STABILITY AND BOUNDS IN AGGREGATE SCHEDULING NETWORKS

THÈSE N^{O} 3998 (2008)

PRÉSENTÉE LE 15 FÉVRIER 2008

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

LABORATOIRE POUR LES COMMUNICATIONS INFORMATIQUES ET LEURS APPLICATIONS 2

PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Gianluca RIZZO

laurea in ingegneria elettronica, Politecnico di Torino, Italie
et de nationalité italienne

acceptée sur proposition du jury:

Prof. C. Petitpierre, président du jury
Prof. J.-Y. Le Boudec, directeur de thèse
Prof. R. Cruz, rapporteur
Prof. D. Starobinski, rapporteur
Prof. P. Thiran, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2008

Abstract

We study networks of FIFO nodes, where flows are constrained by arrival curves. A crucial question with 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 issues: Some examples show 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.

The importance of this issue lies in the necessity of hard bounds on packet delay and queue size, in order to enable QoS guarantees in these networks. For this reason we choose to tackle this problem through a deterministic approach, based on worst-case behavior.

Our first result is the determination of a general method to derive sufficient conditions for the stability of a network: We show how, with a proper choice of the observed variables in the network and with the use of network calculus results, it is possible to derive the expression of an operator whose properties are associated to the stability of the network.

Exploiting this method on a simple example, we first derive a generalization of the RIN result to heterogeneous settings and to leaky bucket constrained flows. Through some realistic examples, we show how this method allows networks to achieve a level of utilization which is more than three times larger than the best existing result.

By applying the general method to three different variable classes, we derive some new sufficient conditions for stability, that perform largely better than all the main existing results, and we show how they can all be derived from the new sufficient conditions.

We generalize the ring result to a whole class of networks, whose cycles present a given structure. Finally, we present a new formula for the computation of end-to-end delay bounds in a network of GR nodes.

Keywords

Stability, Network Calculus, Quality of Service, Differentiated Services, Aggregate Scheduling.

Version abregee

Nous étudions des réseaux de noeuds FIFO, où les flux sont contraints par des courbes d'arrivée. Une question cruciale dans ces réseaux est: pouvons-nous dériver une limite pour le délai maximum qu'un paquet peut subir en traversant le réseau, et à la taille maximale d'une file d'attente à chaque noeud? Pour un réseau générique de noeuds FIFO, ce sont encore des questions ouvertes: quelques exemples montrent que, contrairement au bon sens, il est possible de construire un exemple d'un réseau instable de noeuds FIFO, avec n'importe quelle valeur de l'utilisation maximale des noeuds. L'importance de ce problème se situe dans la nécessité de disposer de limites rigides sur le délai des paquets et sur la taille des files d'attente, afin de pouvoir offrir des garanties de qualité du service dans ces réseaux.
Pour cette raison, nous choisissons d'aborder ce problème par une approche déterministe, basée sur le comportement des pires cas. Notre premier résultat est la détermination d'une méthode générale pour dériver des conditions suffisantes pour la stabilité d'un réseau: nous montrons comment, avec un choix approprié des variables observées dans le réseau et avec l'utilisation des outils de network calculus, il est possible de dériver l'expression d'un opérateur dont les propriétés sont associées à la stabilité du réseau. En appliquant cette méthode sur un exemple simple, nous dérivons d'abord une généralisation du résultat de RIN aux configurations hétérogènes et avec une contrainte de type leaky bucket. À travers de quelques exemples réalistes, nous montrons comment cette méthode permet un niveau d'utilisation des ressources de réseau qui est plus de trois fois supérieur au meilleur résultat existant. En appliquant la méthode générale à trois classes de variables différentes, nous dérivons de nouvelles conditions suffisantes pour la stabilité, qui sont meilleures que les principaux résultats existants, et nous montrons comment ces resultats peuvent être dérivés à partir de nos conditions suffisantes.
Nous généralisons le résultat d'anneau à une classe entière de réseaux, dont les cycles présentent une structure bien précise. En conclusion, nous présentons une nouvelle formule pour le calcul du délai de bout en bout dans un réseau de noeuds Guaranteed Rate.

Mots-clés

Stabilité, Network Calculus, Qualité du Service, Services Différenciés, Ordonnancement Global.

To my parents

Acknowledgements

First of all I want to express my deep gratitude to my supervisor Prof. Jean-Yves Le Boudec, for his guidance, his availability and his infinite patience. Working with him has been a great experience.
I would like to thank all the people in the LCA lab, that make it a very pleasant and stimulating environment. I am in debt with all the past and present colleagues, who all supported me in many ways during this experience. Working with them, sharing ideas or even simply a cup of coffee has been something very precious to me.

I am grateful to our LCA system managers Philippe Chammartin, Jean-Pierre Dupertuis, and Marc-André Lüthi, as well as our secretaries Danielle Alvarez, Holly Cogliati, Angela Devenoge, and Patricia Hjelt for their smiling and efficient support.

A special thanks to Mme Fisch, for making my french grow slowly over these four years.

Finally, my gratitude goes to Laura and Nino, my parents, as well as to Ciccia, for their love, encouragement, and forbearance.

Contents

Chapter 1

Introduction

1.1 Objectives

The main objective of this dissertation is the derivation of sufficient conditions for stability in the networks of FIFO aggregate schedulers. In order to be of practical interest, these sufficient conditions must be:

- applicable to generic network topologies and to the most widespread deterministic characterization of flows: in particular, to leaky bucket constrained flows;
- they must imply a network utilization that is as high as possible; and
- they must give practical indications as to how to route flows and how to allocate resources in order to have a stable network.

1.2 Layout

In Chapter 2 we introduce the problem of stability in aggregate scheduling network, presenting the basic network calculus results and the main existing positive results for these networks.

In Chapter 3 we outline a new approach for the derivation of sufficient conditions for stability, describing its main steps.

A first example of application of this new approach is presented in Chapter 4, together with a first result: Specifically, the generalization of the RIN result to heterogeneous networks.

A more complex application of this approach is described in Chapter 5, together with a new set of sufficient conditions for stability derived from it. An algorithm is described for testing these conditions, and its performance is assessed on some network examples.

We finally present in Chapter 6 a result which is not directly related to the problem of stability. For networks of non-FIFO Guaranteed Rate nodes, we present a refinement of a network calculus result relative to the concatenation of Guaranteed Rate nodes, which makes it possible to compute a tight upper bound to end-to-end delay in networks without cycles. We conclude with Chapter 7, where we present our conclusions and outline possible future directions of research.

Chapter 2

The Problem of Stability

In this chapter, we

- introduce the problem of stability;
- recall the main concepts of deterministic queuing systems analysis; and
- present the main existing results.

2.1 Introduction

Some of the most widespread applications of the Internet are based on packet exchanges between peers that imply a delay which can be kept low, below a given limit. In general, one of the most important problems in a network in which some form of quality of service is to be implemented is the determination of a method to ensure that the maximum packet delay does not overcome a given limit value, usually associated to a specific service guarantee.
We assume the traffic in the considered network is organized in *flows*, following fixed paths. The necessity for the mechanism that ensures quality of service to be *scalable* makes the issue of QoS guarantees a complex problem: If some form of per-flow service guarantee is to be offered in the network, then this can become hard to realize in the core of large networks, where a large number of flows can potentially traverse each node.
One of the solutions proposed for the problem of scalability is *aggregate scheduling*: A node is an aggregate scheduler if it serves packets from all input flows without taking into account to which flow each packet belongs to. This implies that if the node offers some form of service guarantees, they are offered to the whole set of flows traversing the node, rather than to each single flow. As a result, the amount of service received by a single flow depends strongly on the arrival patterns of all the flows at the node.

It is the interdependence among the amounts of service received by each flow at each node in the network that makes the derivation of bounds to delay and backlog in such networks a complex task. Aggregate scheduling is part of the network model on which Differentiated Services are based. Other examples of applications are represented by very high speed switches, by MPLS networks, and by some examples of Network-On-Chip.
Then, a problem arises in such networks: How do we allocate resources to groups of flows in order for the traffic to meet QoS requirements? Even the intuition that a sufficiently high level of over-provisioning might bring low packet latency has revealed itself as wrong: Examples show (see Section 2.1.2) that even with very low levels of utilization of network resources, we can have queue sizes growing in time without bound.

The central problem addressed in our work is therefore the determination of sufficient conditions under which, in a given aggregate scheduling network, the queue size and the packet delay at each node are bounded in time. If finite bounds are available, then the network is said to be *stable*.
There are essentially two main approaches to the problem of stability. In the stochastic approach, some form of stochastic guarantees are derived from the analysis of the statistical behavior of packet arrivals at nodes. In the present work, we chose to adopt a deterministic approach, based on worst case analysis of network behavior, as our interest is to derive hard bounds on delay and backlog. The set of results used for this kind of analysis is named network calculus (Section 2.2).

One of the main factors influencing the stability of a network is the *scheduling algorithm*. Among the many existing scheduling algorithms, some (such as "Longest-in-System", "Shortest-in-System", "Farthest-to-Go") are known to ensure network stability [4]. In the present work, we choose to focus primarily on the First in, First Out (FIFO) scheduling algorithm, which for its simplicity is one of the most commonly used.

2.1.1 Stability and uniform stability

As already stated, we use the term *flow* to refer to a path through the network, together with all the packets that are injected along the path. A flow can actually represent a set of end-user micro flows, that are aggregated at the network edge, and take the same path along the considered network.

We assume each flow is constrained at the source (the network ingress point) by an *arrival curve*, which limits the maximum number of packets that can be emitted from the source in any time interval, in function of the duration of the time interval (Section 2.2.2). We have then the following formal definition of stability:

Definition 2.1.1 (Stability) *Consider a network traversed by F flows, with a given buffer content at each node at $t = 0$. If for any array of input sequences $\mathbf{R}(t) = (R_1(t), ..., R_F(t))$ relative to flows at the input to the network, and compatible with the given arrival curve constraints, $\exists \Gamma > 0$ such that the maximum queue size at each node is upper bounded by Γ, then the network is* stable.

The definition of stability, however, implies that we can have stable networks, in which the maximum queue size and the maximum packet delay do not grow indefinitely over time, but in which the value of these maximal quantities varies in function of the flows input sequences.
In practical applications, in order to offer hard QoS guarantees in the network, we are interested in the possibility to derive bounds to backlog and delay valid for any possible input pattern. Therefore, a more practical definition of stability is the following:

Definition 2.1.2 (Uniform Stability) *Consider a network with a given buffer content at each node at $t = 0$, and traversed by F flows, each constrained at the source by a given arrival curve. If $\exists \Gamma > 0$ such that for any array of input sequences $\mathbf{R}(t) = (R_1(t), ..., R_F(t))$ relative to flows at the input to the network, and compatible with the given arrival curve constraints, the maximum queue size at each node is upper bounded by Γ, then the network is said to be* uniformly stable.

We can see that uniform stability of a network implies its stability, but the converse is not true. In our work we focus on uniform stability: For sake of simplicity, in the rest of this document, we will call simply "stability" what we defined here as uniform stability.

2.1.2 An example of an unstable network

Although the problem of stability is quite old, for the case of more practical interest (FIFO nodes and leaky bucket constrained flows), the only available examples of instability have both been recently described by Andrews [2, 3].

In a first instability example [2] the maximum node utilization in the network is of $1 - 3 \cdot 10^{-9}$ (and therefore very close to critical node load condition).
The most interesting example of instability is in [3], where it is proven that for any value of the maximum node utilization, it can be built a network whose backlog grows in time without bound.
We describe here the main features of the network considered in [3]. The scheme of the network is shown in Fig. 2.1: It has a grid structure, and its size and the path lengths of flows both depend on parameters K and L. There are three classes of nodes: The ones

marked in red have service rate equal to one, and the other two (marked in yellow and in blue, respectively) have a service rate which is inferior to one, and which is function of some parameters, among which ϵ, K and L.
All flows are leaky bucket constrained: Their rates and burstinesses are parameterized by ϵ, K and L, and other parameters. There are only three classes of flows: Flows belonging to the first class (in blue in the figure) enter the network at a red node (one fresh blue flow per red node), and traverse the network with a path whose structure is the one indicated in Fig. 2.1. Note that the paths of all flows keep the structure described by wrapping around (i.e., all flows that "get out" of the network on the right border of the grid, get into the grid again from the left border, and the same holds for flows getting out of the grid from the lower border).
Flows belonging to the remaining two classes traverse each a single node: The ones indicated in green in Fig. 2.1 traverse only blue nodes (one for each blue node), and the ones in red traverse only red nodes (one for each red node).

Theorem 2.1.1 (Instability example [3]) *With reference to Fig. 2.1, $\forall \epsilon > 0$, $\exists$ a set of values for the parameters K and L, and a set of input sequences (one per flow in the network) such that:*

- *The maximum node utilization is inferior to ϵ; and*
- *the maximum node backlog in the network grows in time without bound.*

Some important features of the result are the following:

- It is derived under the assumption of a fluid model for the traffic;
- the example holds independently of buffer content at $t = 0$; and
- the value of the parameters (in particular, of K and L) is function of ϵ: precisely, the size of the network (and the maximum hop count in the network, equal to $3L - 2 + (2K + 1)L$) increase with decreasing ϵ, so that for very small node utilization the network can be very large.

Following a classical procedure in instability examples, the proof proceeds through a sequence of steps, periodic with some period T, and such that at the end of each period the content of the buffers in the network has increased with respect to the end of the previous period.

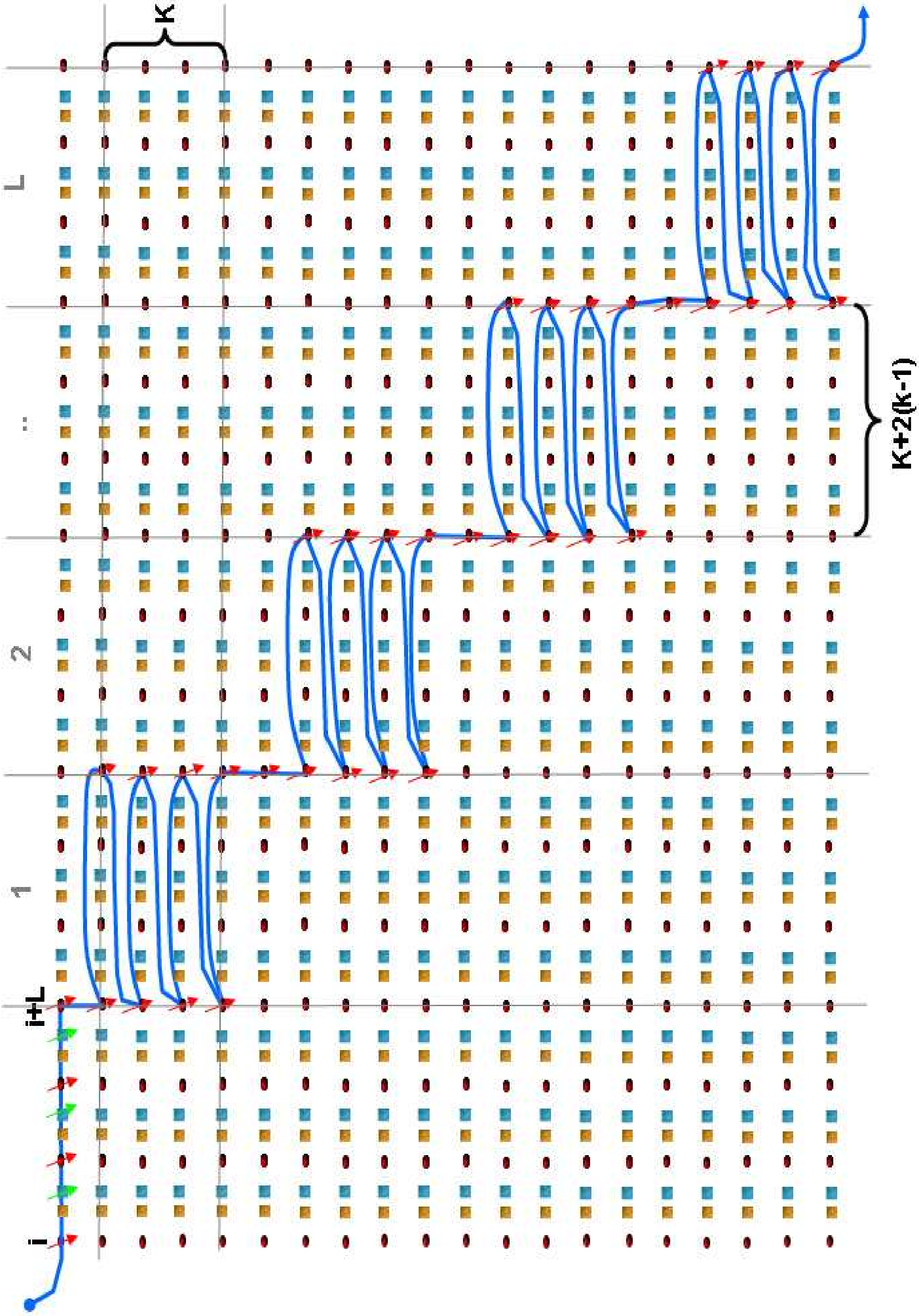

Figure 2.1 – Scheme of the network relative to the instability example in [3], with $K = L = 4$. For sake of clarity, only some of the flows in the network have been drawn.

Table 2.1 – Notation used in Chapter 2.

Symbol	Definition
$\mathcal{V}$	set of wide sense increasing sequences or functions which are zero for $t < 0$, and whose range is $[0, +\infty]$
R	Input function
R^*	Output function
α	Arrival curve
β	Service curve
$\beta_{r,T}$	Service curve of the rate-latency type, with rate r and latency T
For any $x \in \mathbb{R}$, we denote with $[x]^+$ the smallest integer larger than or equal to x.	

2.2 Network calculus: A deterministic approach

In this section we recall for completeness some of the basic concepts and terminology of network calculus, and some results that have been used in the present work.

As we already stated, the approach taken in the present work to the problem of stability is *deterministic*: We do not make any hypothesis on the properties of the input sequence of packets, but only assume they are constrained by *arrival curves*. We then look at upper bounds to delay and backlog at each node, through a worst case analysis.

The set of algebraic results we used in this deterministic approach to queuing systems performance goes under the name of network calculus. It is based on a min-plus algebra, and it provides a set of concepts and of results that allow for a worst-case analysis of queuing systems. A complete reference is provided in [26] and [11], while its origin can be found in the articles by Cruz [15, 16], and by Parekh and Gallager [31].

2.2.1 Basic concepts

Let's denote with $\mathcal{V}$ the set of wide sense increasing sequences or functions which are zero for $t < 0$, and whose range is $[0, +\infty]$. For sequences, we assume they are left-continuous. We introduce the operation of min-plus convolution, as follows:

Definition 2.2.1 (Min-plus Convolution) *Let $f,\ g \in \mathcal{V}$. Then the min-plus convolution of f and g is the function*

$$(f \otimes g)(t) = \begin{cases} \inf_{0 \le s \le t}\{f(t-s) + g(s)\} & t \ge 0 \\ 0 & t < 0 \end{cases} \tag{2.1}$$

Let's consider a generic system $\mathcal{S}$, that can be a buffer, a node in a network or a whole network, traversed by a flow. To that flow we associate the cumulative functions $R(t)$ and $R^*(t)$, that represents the total number of bytes of that flow observed in the time interval $[0, t]$ at the input and at the output of the system $\mathcal{S}$, respectively. We call $R(t)$ the *input function*, and $R^*(t)$ the *output function* associated to the considered flow, and to the system $\mathcal{S}$. $R(t)$ and $R^*(t)$ belong to $\mathcal{V}$, and we assume by convention that $R(0) = R^*(0) = 0$. In what follows we always consider the case in which the system $\mathcal{S}$ is lossless.

Definition 2.2.2 *Let's consider a system* $\mathcal{S}$*:*

- *the* backlog *at time* t *is* $R(t) - R^*(t)$*; and*
- *the* virtual delay *at time* t *is*

$$d(t) = \inf\{\tau \geq 0 : R(t) \leq R^*(t + \tau)\}$$

2.2.2 Arrival curve and service curve

Definition 2.2.3 (Arrival Curve) *Let* $\alpha \in \mathcal{V}$*. Let's consider a flow, and the cumulative function* $R(t)$ *associated to it and relative to a given point in the network (input or output to a node). We say that in that point of the network the flow is constrained by the* arrival curve α *if and only if for all* s *such that* $0 \leq s \leq t$*,*

$$R(t) - R(s) \leq \alpha(t - s)$$

Two important types of arrival curves are the leaky bucket and the staircase arrival curves.

- **Leaky bucket**: they have an expression of the form $\sigma + \rho t$, where σ is called *burstiness*, and ρ *sustainable rate*. It finds its main utilization in the DiffServ and IntServ network models.
- **Staircase**: its expression is of the form $\lceil \frac{t-\tau}{T} \rceil$, with T the *period* (or *interval*) and τ the *tolerance*. Staircase arrival curves are used in ATM.

The concept of service curve is used to model in an abstract way the behavior of packet schedulers. Its function is to define a minimum (and maximum) amount of service that a node can offer.

Definition 2.2.4 (Minimum Service Curve) *We say that the system* $\mathcal{S}$ *offers to the flow a* minimum service curve β *if and only if* β *is wide sense increasing,* $\beta(0) = 0$*, and* $R^* \geq R \otimes \beta$*.*

In what follows, when we talk of a service curve, we always refer to minimum service curves, unless explicitly indicated.

Definition 2.2.5 (Rate-latency functions $\beta_{r,T}$) *We say that a service curve is of the* rate-latency type *if its expression takes the form $\beta(t) = R[t - T]^+$, where R is called* rate*, and T* latency*. We indicate such a service curve with $\beta_{R,T}$.*

Definition 2.2.6 (Strict Service Curve) *We say that the minimum service curve β that the system $\mathcal{S}$ offers to the flow is a* strict *service curve if, during any backlogged period of duration u, the output of the system is at least $\beta(u)$.*

A busy (or backlogged) period at a buffer is a time interval during which the buffer is non empty. Examples of strict service curve elements are given by priority schedulers, and by FIFO constant rate schedulers [26].

Not all systems offering a minimum service curve to a flow offer a strict service curve: An example is represented by a system with a service curve of the form $\beta_{R,T}$, introducing a constant delay T to all packets. Assume that at the input we have a sequence of packets with interarrival time of $T/2$, and of size $RT/4$: In such a system, we can well see that during a backlogged period of duration $3T/2$ the number of packets served is one, while they would be two if the service curve would have been strict.

An upper bound to the maximum service that a system can offer to the whole set of input flows is given through the concept of maximum service curve:

Definition 2.2.7 (Maximum Service Curve) *We say that the system $\mathcal{S}$ offers to the flow a* maximum service curve *γ if and only if $\gamma \in \mathcal{V}$, and $R^* \leq R \otimes \gamma$.*

The maximum service curve is useful to model, for example, an output buffer at a node: The rate of the maximum service curve is in this case the link rate of the link at the output of the buffer, as this is the maximum rate at which packets can be sent on that link.

Definition 2.2.8 (Constant Rate Server) *Consider a node offering a minimum service curve $\beta_{r,0}$, and a maximum service curve equal to its minimum service curve. We say that the node is a* constant rate server*, with rate r.*

Let's consider now a network, where each node n offers a maximum service curve of the form $r_n t$. We assume that each flow f is leaky bucket constrained, with rate ρ_f and burstiness σ_f, and we indicate with N_n the set of flows traversing node n.

Definition 2.2.9 (Node and network utilization) *Given a node n, we define its* utilization *the quantity*

$$\frac{\sum_{f \in \mathcal{N}_n} \rho_f}{r_n}$$

The network utilization *is the maximum of the node utilizations over all nodes in the network.*

Definition 2.2.10 (Node Serviceability Condition) *We say that a node satisfies the* serviceability condition *if its utilization is strictly inferior to one.*

2.2.3 Some basic results

Based on the concepts defined in the previous paragraph, some of the basic network calculus results can be derived [15], [26]. They are related to bounds on delay and backlog.

Theorem 2.2.1 (Backlog Bound) *Consider a flow traversing a system $\mathcal{S}$, which offers a minimum service curve β. Assume the flow at the input of the system is constrained by an arrival curve α. Then for any $t \geq 0$, the backlog $R(t) - R^*(t)$ satisfies*

$$R(t) - R^*(t) \leq \sup_{s \geq 0}\{\alpha(s) - \beta(s)\}$$

In order to introduce the result on delay bound, we have to introduce the concept of horizontal deviation between two functions or sequences belonging to $\mathcal{F}$:

Definition 2.2.11 (Horizontal Deviation) *Let $f,\ g \in \mathcal{F}$ The* horizontal deviation *between f and g, denoted with $h(f, g)$, is defined as*

$$h(f,g) = \sup_{t \geq 0} \{\inf \{d \geq 0 : f(t) \leq g(t+d)\}\}$$

Intuitively, $h(f, g)$ is the maximum horizontal distance between f and g.

Theorem 2.2.2 (Delay Bound) *Consider a flow traversing a system $\mathcal{S}$, offering a minimum service curve β. Assume that the flow is constrained, at the input to the system, by an arrival curve α. Then we have that $\forall t$, $d(t) \leq h(\alpha, \beta)$.*

Another important issue is the derivation of an arrival curve for a flow or a set of flows at the output of a system offering a given minimum service curve:

Theorem 2.2.3 (Output Bound) *Consider a system that offers a minimum service curve β, traversed by a single flow. Assume the flow is constrained, at the input to the system, by an arrival curve α. Then at the output, the flow is constrained by $\alpha \oslash \beta$.*

Theorem 2.2.3 is of limited utility, as it only allows to derive an arrival curve for the whole set of flows traversing the node, and not for a subset of it. In those cases in which we dispose of a delay bound at the node, one of the ways to derive such an arrival curve is to exploit the following theorem:

Theorem 2.2.4 (Output Bound By Delay Bound) *With the same assumptions as in the previous theorem, if we know a bound D to packet delay at the node, then the at the output, the flow is constrained by $\alpha(t + D)$.*

We consider now the case in which an aggregate scheduling system with service curve β, is traversed by two flows (two for simplicity, but the results can be immediately extended to the case of more than two input flows), with generic arrival curves, indicated with α_1 and α_2 respectively, .

Theorem 2.2.5 (Output Bound in FIFO Multiplexing Systems [26]) *With the given assumptions, if the system servers packet in a FIFO order, an output bound for flow 1 is given by $\inf_{\theta \geq 0}(\alpha_1 \oslash \beta^1_\theta(t))$, with*

$$\beta^1_\theta(t) = \begin{cases} [\beta(t) - \alpha_2(t - \theta)]^+ & t > \theta \\ 0 & t \leq \theta \end{cases}$$

With leaky bucket constrained flows, and rate latency service curves, the burstiness bound at the output for flow 1 is given by $\sigma_1 + \rho_1 \left(T + \frac{\sigma_2}{r}\right)$.

2.2.4 The Guaranteed Rate class of scheduling algorithms

An alternative formulation of service guarantees at a node is the one that brings to the definition of the *Guaranteed Rate* class of scheduling algorithms. At its basis there is the concept of *Guaranteed Rate clock value* [19]:

Definition 2.2.12 (GR clock value) *Consider a flow that is served at a node to which it is associated a service rate r (in bit/s). Let p^j denote the j-th packet of the flow, and l^j its length. Let $GRC(p^j)$ and $A(p^j)$ denote respectively the guaranteed rate clock value of packet p^j and its arrival time at the node. The* guaranteed rate clock value *for packet p^j is given by:*

$$GRC(p^j) = \begin{cases} 0 & j = 0 \\ \max\{A(p^j),\ GRC(p^{j-1})\} + \frac{l^j}{r} & j \geq 1 \end{cases}$$

The concept of GR clock value is used to define the *Guaranteed Rate* (GR) node, as follows:

Definition 2.2.13 (GR node [19]) *Consider a node that serves a flow. Packets are numbered in order of arrival. The node is a* Guaranteed Rate node *for the flow, with rate r and latency e, if it guarantees that packet p^j of the flow is transmitted by $GRC(p^j) + e$, where e depends on the scheduling algorithm and the server.*

Let us indicate with L_{max} the maximum packet size for all the flows. We have the following result:

Theorem 2.2.6 *A GR node with rate r and latency e offers a minimum service curve $\beta_{r,e+\frac{L_{max}}{r}}$.*

In order to derive a bound to delay in a GR node, we use the following result:

Theorem 2.2.7 (Delay Bound at a GR Node) *Consider a flow traversing a GR node with rate r and latency e. Assume the flow is constrained by an arrival curve $\alpha(t)$ at the input to the node. Then the delay is upper bounded by*

$$\sup_{t>0}\left\{\frac{\alpha(t)}{r} - t\right\} + e$$

An important property of GR nodes is the following:

Theorem 2.2.8 (Equivalence with service curve [26]) *A GR node with rate r and latency e, with L-packetized input, is the concatenation of a service curve element, with service curve equal to the rate-latency function $\beta_{r,\,e}$, and an L-packetizer. If the GR node is FIFO, then so is the service curve element.*

Corollary 2.2.1 ([26]) *A GR node (with rate r and latency e) offers a minimum service curve $\beta_{r,\,e+\frac{l_{max}}{r}}$.*

A broad class of scheduling algorithms belongs to the class of Guaranteed Rate scheduling algorithms [19, 20]. Among them we can find many practical implementations of the GPS scheduling algorithm, such as Virtual Clock scheduling [22], Packet-by-Packet Generalized Processor Sharing [31], Self-Clocked Fair Queuing [18]. Its relevance is also due to the fact that it finds application in the node model adopted in the Differentiated Services framework, and because it is a class that includes FIFO nodes that offer a minimum service curve of the rate-latency type. We precise that a scheduling algorithm belonging to the GR class is not necessarily FIFO.

2.3 The time stopping method and the Charny-Le Boudec result

One of the main methods for the derivation of sufficient conditions for stability of a network is the *time stopping method*, first derived in [16].
It represents a very general method, at the basis of some of the main existing results. The method can be resumed in two main steps: At first, by assuming that the network is stable, we derive finite upper bounds to quantity of interests (maximum packet delay at a node, or maximum flow burstiness) related to the stability of the network. This derivation is usually conditioned to the satisfaction of some constraints on the network (e.g., on flow rates, in case of leaky bucket flows).

In a second phase, we show that if those conditions are satisfied the network is stable, and those upper bounds hold.
The central steps of the method are:

- The choice of the variables. They can be the flow burstiness (Section 2.3.1), or the maximum packet delay in the network, as in the Charny-Le Boudec result (Section 2.3.2), or some quantity related to node backlog, like in the ring result [35], and in general any other parameter of the network which can be related to the stability of the network; and
- the derivation of upper bounds to the variables.For this we can exploit the network calculus results for output bounds [26], together with the concatenation result [26], and in some special cases, we can also exploit the topology of the network (as in the ring result).

2.3.1 The time stopping method with flow burstiness

We show here an example of application of the time stopping method to a network of leaky bucket constrained flows. We derive burstiness constraints for all flows on any link in the network, in function of the burstiness constraints of the flows at the input of the network itself, and of the rate of all flows, given some conditions on flows rates. The method exploits the result on delay bound presented in Section 2.2.3.

We assume each node has a rate latency service curve, with rate r_n and latency T_n, and it performs aggregate scheduling. We assume each flow f at the input to the network has an arrival curve given by $\sigma_f^0 + \rho_f t$.

Definition 2.3.1 (Flow Burstiness at the input to a node) *For any flow f and any node n, the* burstiness of flow f at the input to node n *and relative to time t (indicated with $\sigma_f^n(t)$) is the maximum backlog that this flow would generate in a constant rate* *server with rate ρ_f, traversed only by flow f, and situated at the input of node n, in the time interval* $[0, t]$.

An analogous definition holds for flow f at the output of node n, and we indicate the corresponding quantities as $\sigma_f^{n*}(t)$.
The time stopping method can be divided in two phases:

First phase: Fix a time $t_0 > 0$. For each flow f in the network and for any traversed node n, we consider the quantities $\sigma_f^n(t_0)$ and $\sigma_f^{n*}(t_0)$. All those quantities are finite for a given t_0, as the number of packets that enter the network in the time interval $[0, t_0]$ is finite.
Now let's consider flow f at the input to node n: The definition of the quantity $\sigma_f^n(t_0)$ implies that flow f is constrained by $\rho_f t + \sigma_f^n(t_0)$ at the input to node n, and by $\rho_f t + \sigma_f^{*n}(t_0)$ at its output, respectively.
Using this observation we can derive an upper bound for the burstiness of every flow at the input to every node in the network in the following way:

- For any flow f, at the input to the first node of its path n_f, we have

$$\sigma_f^{n_f}(t_0) \leq \sigma_f^0$$

- At any node n, we can derive an upper bound to packet delay by using Theorem 2.2.2. Then, by Theorem 2.2.4, we have for any flow f traversing node n,

$$\sigma_f^{n*}(t_0) \leq \sigma_f^n(t_0) + \rho_f \left(\frac{\sum_{f' \in \mathcal{N}^n} \sigma_{f'}^n(t_0)}{r_n} \right)$$

If we define the vector $\boldsymbol{\sigma}(t_0)$ as composed by all the variables defined in the network, for all flows f and all nodes n, we can write the set of inequalities obtained as

$$\boldsymbol{\sigma}(t_0) \leq A\boldsymbol{\sigma}(t_0) + \mathbf{b}$$

A is a nonnegative matrix, and $\mathbf{b}$ a nonnegative vector which does not depend on $\boldsymbol{\sigma}(t_0)$. Now, if the spectral radius of the matrix A is less than 1, then the matrix $(I-A)$ (where I is the identity matrix) is invertible, and a bound to $\boldsymbol{\sigma}$ is given by

$$\boldsymbol{\sigma}(t_0) \leq (I - A)^{-1}\mathbf{b} \tag{2.2}$$

From these bounds we can derive bounds to delay at each node and to backlog.

Second phase: For any time $t_0 > 0$, the sufficient condition to derive a bound to quantities $\boldsymbol{\sigma}(t_0)$ (that is, that the spectral radius of the matrix A is less than 1) as well as the expression of the bounds derived in the previous step do not depend on t_0. Then for $t_0 \to +\infty$, we finally have that if the spectral radius of the matrix A is less than 1, a bound to flow burstiness is given by Equation (2.2).

We can observe that although the time stopping method allows to determine if a given network satisfies some sufficient conditions for stability, in general it does not give any indication as to how to route flows or to how to assign rates in order to have a stable network. Some exceptions to this are represented by the ring [4, 35], and by the Charny - Le Boudec result (Section 2.3.2).

2.3.2 The Charny-Le Boudec result

In what follows we present the Charny-Le Boudec result [12, 21], one of the main existing positive results about stability in generic (non FIFO) networks, and one of the few that can be applied on realistic node and network models. It constitutes an application of the time stopping method on networks of Guaranteed Rate nodes, with leaky bucket constrained flows.

Let's consider a network of Guaranteed Rate nodes. The assumptions are the following:

- At the ingress to the network, every flow f has an arrival curve of the form $\sigma_f + \rho_f t$.
- Nodes are aggregate GR schedulers. Every GR node n has a rate r_n and a latency e_n, and therefore it offers a minimum service curve $\beta_{r_n, e_n + \frac{L_{max}}{r}}$ to the aggregate of all input flows. L_{max} is the maximum packet size for all the flows in the network. Every node n offers a maximum service curve of the form $r_n t$.
- Every node satisfies the node serviceability condition (Section 2.2.2);
- h is the maximum hop count over all flows in the network;
- ν is the network utilization;
- $\forall n,\ \mathcal{N}^n$ is the set of all flows traversing node n; and
- Define $\tau = \max_n \left\{ \frac{\sum_{f \in \mathcal{N}^n} \sigma_f}{r_n} \right\}$, and $e = \max_n e_n$.

Theorem 2.3.1 *If $\nu < \frac{1}{h-1}$, then*

- *a bound to delay at each node is given by*

$$D = \frac{e + \tau}{1 - (h-1)\nu}$$

- *$\forall n$, a backlog bound at node n is given by $r_n D + L_{max}$.*

Slightly improved versions of this result exist, that exploit the knowledge of the total incoming bit rate at the node [26]. An interesting consequence of Theorem 2.3.1 is that if the maximum hop count in the network is equal to 2, the network is stable. But in practical cases h can take quite large values (more than 20 for IP networks [5]). Therefore, in practical situations the sufficient condition for stability in Theorem 2.3.1 leads to very low values of node utilization in realistic scenarios, and is therefore of little practical utility.

An important result is the following [26]:

Theorem 2.3.2 *Given $h_0 \in \mathbb{N}$, $h_0 > 1$. Then for any $D' > 0$ it exists a network with $h = h_0$ and $\nu > \frac{1}{h_0 - 1}$ such that the worst case delay is at least D'.*

This result indicates that, for network utilizations larger than the ones in Theorem 2.3.1, in order to derive better sufficient conditions for stability it is necessary to exploit other characteristics of the network than maximum hop count.

2.4 The RIN result

One of the main existing positive results on stability is the ”RIN result”. It has been first developed in [13, 14], and further developments are in [10, 37, 25, 30]. Its relevance is due to the fact that it does not depend on any assumption on the topology of the network, and it allows to derive good bounds on delay and backlog at each node.

2.4.1 Network model and assumptions

At the basis of the “RIN result” there is a network model close to the one of an ATM network. Consider a network with the following assumptions:

- Time is discrete, and every event in the network (packet arrivals and departures) takes place at integer time units;
- data is organized in flows, following fixed paths. All packets have equal size, equal to one; and
- every node in the network is modeled as a collection of output buffers, one per outgoing link. All links are FIFO. Every output buffer with the associated outgoing link is a FIFO constant rate scheduler, with rate equal to the rate of the link, serving all flows in an aggregate manner. The service time for each packet is equal to one time unit. Any propagation time in the network is constant in time, and takes an integer value.

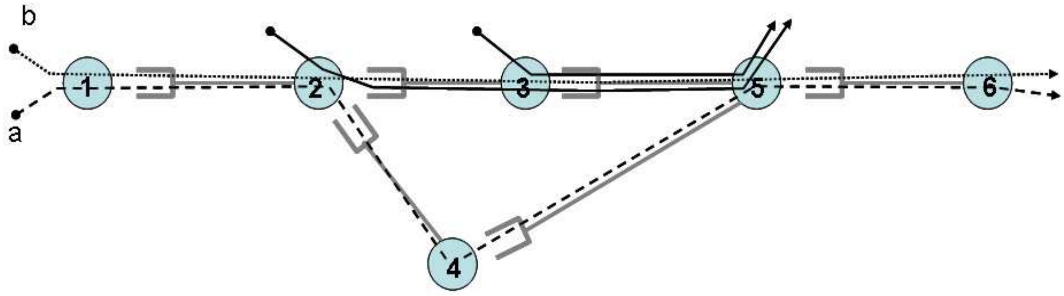

Figure 2.2 – An example of a network analyzed in the RIN result. A node is represented by a circle surrounded by output buffers, one per outgoing link.

The RIN Result is based on the concept of *interference* between flows:

Definition 2.4.1 (Interference between flows) *We say that two flows* interfere *at a node, if they arrive at it from different nodes, and they get out of the node through the same output buffer. The* interference number *of a flow at a node is the total number of flows interfering with the considered flow at the node.*

Definition 2.4.2 (Route Interference Number) *The* Route Interference Number (RIN) *of a flow is the sum of the interference numbers of the flow over all the nodes on its path.*

As an example, in Fig. 2.2 we have a network, with four flows. In this network, the RIN of flow a is 2, and the one of flow b is 4. Note that, at node 5 although four flows are traversing it, flow a interferes only with flow b, as flow b is the only one with which flow a shares the output buffer at that node. Note also how flow a interferes twice with the same flow. In conclusion, the RIN of a flow is a quantity that depends on its path, and therefore both on the topology of the network and on the traffic distribution in it.

2.4.2 The Source Rate Condition

Definition 2.4.3 (Source Rate Condition [14, 26]) *We say that a flow satisfies the* Source Rate Condition *if the minimum spacing in time between packets at the source is at least equal to the the RIN of the flow plus one.*

The Source Rate Condition is equivalent to imposing that, in every time interval of duration equal to the RIN of the flow plus one, the source of that flow can send at most one packet. Equivalently, this can be expressed by saying that the flow is constrained at its ingress to the network by a staircase arrival curve $v_{T+1,0} = \lceil \frac{t}{T+1} \rceil$, with T equal to the RIN of the flow.

Theorem 2.4.1 (RIN Result [14, 26]) *If all flows in the network satisfy the Source Rate condition, then:*

- *The backlog at each output buffer is bounded by the minimum interference number at that node among all the flows traversing the buffer;*
- *the end-to-end delay of a packet is the sum, over each of the traversed nodes, of the minimum interference number at the node among all the flows traversing the same output link as the considered flow;*
- *there is at most one packet per flow in every busy period; and*
- *consider a link, and a subset of m flows of the set composed by all the flows traversing the link. Assume that over all the flows in this set, the minimum residual route interference for the rest of their path, is equal to n. Then over any time interval of duration $n + m$, the maximum number of packets, relative to the considered set of flows, that can arrive over that link is upper bounded by m.*

Some observations:

- The main downside of this result is that it relies on assumptions that restrict its applicability, making it not useful in many practical cases. In particular, it relies on equal sized packets, on nodes that are constant rate servers and all equal between them, and which are synchronized between them.
- Another downside is the poor network utilization due to the source rate condition, especially in those networks with many flows. It implies the allocation of a higher throughput to ”local” flows, with a limited number of hops, at the expense of ”transit” flows.
- The original result from [14] implicitly assumes the network to be empty at $t = 0$, and it does not hold when this condition is not satisfied. This can be easily seen by considering a flow in the network, and by assuming that the initial buffer content at the first node traversed by the flow is at least equal to the RIN of that flow. It would be sufficient to have more than one packet of that flow in the same busy period, and all the other conclusions of Theorem 2.4.1 would not be valid. With the present state of the art, there exists no straightforward extension of the RIN result to non-empty networks.

- The original formulation of the theorem applies to every network topology, including multistage (feed-forward) networks. As a result, in this formulation multistage networks are not stable for any value of the network utilization factor inferior to one.

Among the main positive aspects of this result there is its simplicity, due also to the assumptions on the network on which it is based.

2.5 Other approaches

A number of other results exploit the characteristics of the topology of the network, in order to derive bounds to delay and to backlog.
In a first place, it has been showed that a unidirectional ring of nodes (not necessarily FIFO) is stable for any value of network utilization inferior to one [35]. This result is obtained with the time stopping method, combined with a choice of the observed variables, that allows to take full advantage of the special topology of the network.
Quite a number of works [24, 23, 17, 1] concentrate on feed forward networks of FIFO aggregate schedulers, and try to improve the available network calculus results for the derivation of tighter bounds to packet delay and to flow burstiness at any point in the network. In particular, Fidler [17] extends the concatenation result to a network of aggregate schedulers with interfering traffic at each node, and it derives the best existing formula for an end-to-end service curve in a feed-forward FIFO network.

2.6 Summary

In this chapter we introduced the problem of stability, presenting the main approaches to this problem and the most significative results available. We introduced some basic definitions and concepts from network calculus, on which many of those results, as well as our work, is based.

Chapter 3

A General Method To Prove Stability

In the present chapter we describe in very general terms a new approach, valid for any topology, which allows to derive sufficient conditions for stability of a network.
More precisely:

- We explain how to choose variables and how to derive an operator that upper bounds the values of the variables at any time instant;
- we show how to derive, from the properties of this operator, a sufficient condition for the stability of a network; and
- we present a method to derive practical results, based on linear operators that upper bound the operator previously derived.

In Chapter 4 we present a first example of application of this method. This example will help to make the method clearer and to understand better its potentiality. An important feature of the proposed approach is that it requires very loose assumptions on the network, and in particular, it can be applied to non-FIFO nodes. The content of this chapter is mainly derived from [34].

3.1 Notation and assumptions

We model a network as a directed graph, where each vertex (hereafter called also "node") models a buffer at the input of a physical link. Each node offers to the aggregate of flows a service curve, generally different for each node. We consider a network whose nodes perform aggregate scheduling.

We assume the traffic in the network is organized in flows. To each flow it is associated an ordered sequence of traversed nodes. We assume there is some form of

Table 3.1 – Notation used in Chapter 3.

Symbol	Definition
$p \in \mathbb{N}$	Index of the p-th relevant network event
$\mathcal{X}$	Set of all the arrays of variables x.
$\mathbf{x}$	An element of the set $\mathcal{X}$.
$\mathbf{x}[p]$ $(p = 0, 1, 2, ...)$	A sequence of elements of $\mathcal{X}$, indexed by p.
We use the bold type to denote arrays. For instance, $\mathbf{x}$ is the array of all the variables x.	
We make dependency over the index p of relevant network events visible through square brackets.	

constraint on the amount of packets injected by a source in the network over a given time interval.

We assume that each flow has, in general, packets of different size, and that for each flow there exists a finite set of possible packet sizes.

Definition 3.1.1 (relevant network event) *We define* a relevant network event *an event of dequeuing of a packet at a node.*

In order to study how packet delay and backlog evolve in time, it is sufficient to observe the network behavior at each time a relevant network event takes place.

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. Therefore, we observe the network at time instants $t_p,\ p \in \mathbb{N}$. When two or more network event take place at the same time, we label them in an arbitrary order.

For any time dependent quantity, we make dependency over the index p of relevant network events visible through square brackets. In Table 3.1 we have a list of the notation used in the present chapter.

3.2 Step one: Choosing the right variables

In order to characterize the dynamics of the network over time and more precisely, the evolution over time of maximum packet delay and queue size, a crucial aspect is the choice of the observed quantities.
The requirements of the method on this choice are:

- Variables have to be related to packet delay and to backlog at nodes, because the stability of the network is related to the maximum value that these quantities can assume over time; and

- variables must be indexed by p, and they must represent the maximum value taken by a given parameter in the network (e.g. packet delay at a node) in the time interval $[0\ t_p]$.

A straightforward choice (but not the only possible one) is to take these two quantities (e.g. delay and backlog) as variables. But this choice might not reveal itself as optimal, as for the method to be effective it is also important to be able to derive good bounds for the variable values.

3.3 Step two: Derivation of an upper bounding operator

The heart of the method consists in finding finite bounds to the values that the variables can take at any time, from which we are able to say something on the stability of the network itself. In order to derive these bounds, an intermediate step is to derive a *monotonic* operator that upper bounds the value of the variables at a given time t_p, $\forall p > 0$, in function of the values of the variables at time t_{p-1}.
Therefore, if with $\mathcal{X}$ we denote the set of all the arrays of variables x, indexed by p, and with $\mathbf{x}[p]$ the array of the variables at time t_p, the upper bounding operator should take the following form:

$$\begin{cases} \mathbf{x}[p] \leq \Pi(\mathbf{x}[p-1]) \quad \forall p > 0 \\ \mathbf{x}[0] \leq \mathbf{x}_0 \end{cases} \tag{3.1}$$

where Π is a monotonic operator $\mathcal{X} \to \mathcal{X}$, and $\mathbf{x}_0$ an array that upper bounds $\mathbf{x}[0]$.

In order to derive Π, we can exploit the availability of constraints on packet arrivals at every node, in the form of arrival curves, together with other properties of the flows related to the topology of the network and to the characteristics of their paths. In general, the availability of some form of service guarantees at each node allows to derive some form of service guarantee for a single flow, up to time t_p. From these guarantees we can derive bounds on the variables.

3.4 Step three: Deriving sufficient condition for stability

3.4.1 A first result on stability

The properties of the upper bounding operator Π play a crucial role in determining sufficient conditions for stability of the network. Intuitively, if for $p \to \infty$ (and therefore, for $t_p \to \infty$, as we have a finite number of network events in a finite time interval) the upper bounds to the variable values in Equation (3.1) are finite, then the network is stable, as we have a bound to packet delay and to busy period duration at each node. In this section we show how from the properties of the operator Π in Equation (3.1) we can derive sufficient conditions for the stability of a network. We start by recalling the definition of super-additive closure of an operator:

Definition 3.4.1 (Super-additive Closure [26]) *Let $\mathcal{E}$ be a poset such that, for any enumerable subset of it, the supremum is well defined, and let Φ be a mapping $\mathcal{E} \to \mathcal{E}$. Denote with $\Phi^{(l)}$ the mapping $\mathcal{E} \to \mathcal{E}$ obtained by composing l times the mapping 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*

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

We apply now this definition to the mapping Π defined in Equation (3.1).

Theorem 3.4.1 *Let Π be the monotonic operator defined in Section 3.3, and let $\mathbf{x}^* = \Pi^*(\mathbf{x}_0)$. If $\mathbf{x}^*$ is finite, then we have that:*

- *the network is stable; and*
- *bounds to packet delay and backlog at all nodes are derived by using the values in $\mathbf{x}^*$.*

Proof of Theorem 3.4.1

In order to prove Theorem 3.4.1, we first need the following lemma:

Lemma 3.4.1 *With the given assumptions on Π, we have that $\forall p \in \mathbb{N}\ \mathbf{x}[p] \leq \Pi^*(\mathbf{x}_0)$*

Proof (Lemma 3.4.1): From Equation (3.1) we have that $\forall p > 0$

$$\mathbf{x}[p] \leq \Pi(\mathbf{x}[p-1]) \leq$$

As Π is monotonic,

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

As $\mathbf{x}[0] \leq \mathbf{x}_0$,

$$\leq \Pi^{(p)}(\mathbf{x}_0) \leq \sup_{p' \geq 0} \Pi^{(p')}(\mathbf{x}_0) = \Pi^*(\mathbf{x}_0)$$

□

Proof (Theorem 3.4.1): This theorem derives from Lemma 3.4.1 and from the properties of the variables chosen (Section 3.2): Indeed by the way we chose the variables, if $\mathbf{x}^*$ is finite, then from it we can derive finite bounds to the maximum queue size at each node, and/or to maximum packet delay. □

3.4.2 A linear upper bounding operator

Theorem 3.4.1 defines a sufficient condition for stability of a network. Nonetheless, this result requires the computation of the super-additive closure of Π which can be difficult to derive. Other features of the operator (such as a very high number of variables, or its nonlinearity) could make Theorem 3.4.1 not practically useful.
The approach proposed in order to derive practical results is to use an operator that upper bounds Π. This approach is based on the following result:

Theorem 3.4.2 *Given the operator* Π *defined in Section 3.3, if it exists a monotonic operator* $\Pi' : \mathcal{X} \to \mathcal{X}$ *such that* $\forall\ \mathbf{x} \in \mathcal{X}$*,* $\Pi'(\mathbf{x}) \geq \Pi(\mathbf{x})$*, and such that the fixed point problem* $\Pi'(\mathbf{x}) = \mathbf{x}$ *admits a finite solution* $\mathbf{x}_s \geq \mathbf{x}_0$*, then the network is stable.*

Theorem 3.4.2 outlines a procedure that can be followed in order to derive finite bounds to variables.

- If the operator Π is nonlinear, or if its expression is too complex to handle, a first step consists in deriving the expression of a *linear* operator Π' that upper bounds Π, usually through some approximations procedures over the expression of Π.

- A second step is to impose that the spectral radius of the matrix associated to Π' be smaller than one. If this condition holds, and if the solution of the fixed point problem is larger than or equal to $\mathbf{x}_0$, then we have finite bounds to the variable values, valid for any time.

A way of ensuring that the solution of the fixed point problem be larger than or equal to $\mathbf{x}_0$ can be to choose an operator Π' such that $\forall \mathbf{x} \in \mathcal{E},\ \Pi'(\mathbf{x}) \geq \mathbf{x}_0$.

Instead of computing the spectral radius, we can also choose to exploit one of the many existing upper bounds to the spectral radius. In this way we can derive different sets of sufficient conditions for the stability of the network. Different upper bounding techniques, and different operators Π bring to different sufficient conditions, and each of these choices can potentially give some insight on which features of the network can have an impact on its stability.

Proof of Theorem 3.4.2

The proof derives from the following two results:

Lemma 3.4.2 *If* Φ_1 *and* Φ_2 *are two mappings* $\mathcal{E} \to \mathcal{E}$ *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: 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 $\Phi_1^*(x) \geq \Phi_2^*(x)$. □

Lemma 3.4.2 implies that in Theorem 3.4.1 we can use the super-additive closure of a mapping that upper bounds Π.
The second result needed to derive Theorem 3.4.2 is the following:

Lemma 3.4.3 *If* $\Sigma : \mathcal{E} \to \mathcal{E}$ *is a monotonic operator, and* $x_s \in \mathcal{E}$ *is a solution (finite or not) of the fixed point problem* $x = \Sigma(x)$ *such that* $x_s \geq x_0$ *with* $x_0 \in \mathcal{E}$*, then we have that* $x_s \geq \Sigma^*(x_0)$.

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

3.5 Summary

In this chapter we outlined in very general terms a method for deriving sufficient conditions for the stability of a generic network. We presented the main steps of the method, providing in this way a guide for understanding the derivation of the main practical results. In Chapters 4 and 5 we apply this method to different upper bounding operators, and obtain different new results.

Chapter 4

Generalization of the RIN Result

In the present chapter:

- We describe an example of application of the method presented in the previous chapter;
- we derive a generalized version of the RIN result to heterogeneous networks and leaky bucket constrained flows; and
- we assess the validity of this result on some network examples.

The contents of this chapter are mainly based on the work in [33].

4.1 Assumptions and notation: The heterogeneous setting

We make the following assumptions on the network, in addition to the ones made in Section 3.1:

- nodes are FIFO, store-and-forward schedulers. Each node is indexed by an integer value n;
- each node n offers to the aggregate of flows a service curve of the rate-latency type $\beta_{r_n,T_n}(t) = r_n(t - T_n)^+$, generally different for each node;
- at each node, service curves are *strict* (Section 2.2.2). This is a very general node model, encompassing many scheduling disciplines (e.g. priority schedulers, or FIFO constant rate schedulers);
- no losses are present at buffers in the network (buffers of infinite capacity);

- every flow in the network is indexed by an integer value f; and
- flows are leaky bucket constrained: For any flow f, we denote with ρ_f its rate and with σ_f^0 its burstiness.

Moreover, for any flow f and any strongly connected component e, we denote with $\mathbf{path}(f, e)$ the longest subpath of flow f in the e-th strongly connected component.

With Δ_n we denote the propagation delay of the physical link at the output of the buffer at node n.

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.

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 E *strongly connected components* (see for instance [28]). The graph of the interconnections between the strongly connected components of a network is structured in a succession of stages. We assume then that the labeling is such that if any two strongly connected components are labeled respectively e_1 and e_2, and if the component e_1 belongs to an inferior stage with respect to the component labeled e_2, then it must be $e_1 < e_2$.

Definition 4.1.1 (Inclusion between node sequences) *For any couple of node sequences* $\mathbf{n}$, $\mathbf{n}'$*, we say that* $\mathbf{n}'$ is included in $\mathbf{n}$ *(or that* $\mathbf{n}'$ is a strict subsequence of $\mathbf{n}$*) 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* $\mathbf{n} = (\mathbf{n}_1, \mathbf{n}', \mathbf{n}_2)$.

We observe that this is a partial order relation over the set of all possible ordered sequences. We define a *subpath* of flow f in the e-th strongly connected component any sequence of nodes $\mathbf{n} \subseteq \mathbf{path}(f, e)$.

In Table 4.1 we have a list of the notation used in the present chapter.

As we can see, the set of assumptions made makes our network model more general than the one used in the "RIN result". In particular, assuming packets of different size, lack of synchronization between nodes, assuming nodes have service rates generally different between them. In the rest of the present work we refer to this setting as the *heterogeneous* network setting, to distinguish it from the homogeneous one used in the "RIN result".

4.2 Characterizing chains of busy periods

In this subsection we recall some definitions from the "RIN result" (Section 2.4), on which we base our results in the present chapter.

Table 4.1 – Notation used in Chapter 4.

Symbol	Definition
e	Index of the e-th strongly connected component
$\mathbf{path}(f, e)$	Ordered sequence of nodes, constituted by all the nodes traversed by flow f in the e-th strongly connected component.
$\mathcal{N}^n$, N^n	Set of all flows passing through node n, of cardinality N^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)
$\mathcal{M}$	Set of all the arrays of quantities $m_f^e \in \mathbb{R} \cup \{+\infty\}$, indexed by e and $f \in \mathcal{F}$
$\mathbf{m}$	An element of the set $\mathcal{M}$
$\mathbf{m}[p]$ $(p = 0, 1, 2, ...)$	A sequence of elements of $\mathcal{M}$, indexed by p
$\mathcal{I}(\mathbf{n})$	Set of all the nodes in the ordered sequence of nodes $\mathbf{n}$
$\mathcal{U}_f^n$	Set of those nodes belonging to the path of flow f that precede node n
$\mathcal{Q}_f^n$ (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
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
$prec(n, f)$	Node that precedes node n on the path of flow f
L_f	Maximum packet size for flow f

Definition 4.2.1 (Relation of precedence between packets) *Given two packets c and d and a node n, we say that packet c precedes packet d (and we indicate it with $c \preccurlyeq_n d$) if c and d are in the same busy period at n, and c leaves n before d.*

Definition 4.2.2 (Super chain) *Consider a sequence of packets $\mathbf{c} = (c_0, ..., c_K)$ and a sequence of nodes $\mathbf{n} = (n_1, ..., n_K)$ (all different). We say that $(\mathbf{c}, \mathbf{n})$ is a* super chain *if*

- *nodes $n_1, ..., n_K$ are all on the path of packet c_0;*
- *$c_{j-1} \preccurlyeq_{n_j} c_j,\ j = 1 \div K$; and*
- *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 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 4.2.3 (Packet included in a super chain) *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 $c_{l-1} \preccurlyeq_{n_l} c \preccurlyeq_{n_l} c_l$.*

4.3 Definition of the variables

We have seen (Section 2.4) how at the basis of the derivation of the RIN result there is the concept of *super chain* (Section 4.2). It is used to put a set of packets (all those packets that are included in a same super chain) in a relation of cause and effect with respect to delay. Indeed, all packets that can delay the service of a given packet at any node along its path in the network are included in a same super chain as the considered packet. So not only those packets that precede the considered packet in a given busy period are responsible for this packet's delay, but also all those that, in some way, influenced the delay of the packets that precede the considered one at any of the traversed node. This chain of cause-effect interdependencies is modeled through the concept of super chain.
As we saw in the description of the RIN result, the sufficient condition for stability (the "Source Rate Condition") implies that at the source packets are "sufficiently" spaced in time so that, in any point of the network, any super chain relative to a given flow never contains more than one packet of that flow. All the other bounds are derived from this bound to the maximum number of packets.

In an heterogeneous environment we cannot expect to derive such a simple result: Time is not divided into slots, service times vary at each node, packet lengths can vary a lot.

Our definitions of the observed variables in the network is therefore the following: For any flow f in the network, and for any strongly connected component, we consider the path of f in that subnet. Then for any time instant t_p, we define $m_f^e[p]$ as the maximum number of bytes belonging to f that can be included in any super chain $(\mathbf{c}, \mathbf{n})$ relative to flow f, up to time t_p, and with $\mathbf{n} \subseteq \mathbf{path}(f, e)$.

The justification of this choice lies in the fact that these variables are related to the definition of stability of a network. Indeed the maximum number of packets served in a busy period at a node is upper bounded by the maximum number of packets (from all the flows traversing the node) inserted in a super chain that "includes" that node. In an unstable network these quantities grow over time without bound.

4.4 An upper bound to variables for any time t_p

In this section we derive the expression 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}, $\forall p > 0$.

Theorem 4.4.1 *For any integer $p > 0$, we have:*

$$\begin{cases} \mathbf{m}[p] \leq \Pi(\mathbf{m}[p-1]) \\ \mathbf{m}[0] \leq \mathbf{m}_0 \end{cases} \tag{4.1}$$

where $\Pi: \mathcal{M} \to \mathcal{M}$ *is the operator defined by* $\Pi(\mathbf{x}) = \mathbf{y}$*, with* $\forall f \in \mathcal{F}$*,* $\forall e$*,*

$$y_f^e = \sum_{e' \leq e} \sum_{n \in \mathcal{I}(\mathbf{path}(f,e'))} a_f^n + \rho_f \left(\sum_{e' \leq e} \sum_{f' \in \mathcal{F}} x_{f'}^{e'} \sum_{\mathbf{n}' \in \mathcal{G}(f,f',e')} S(\mathbf{n}') + \right.$$

$$\left. + \sum_{n \in \mathcal{I}(\mathbf{path}(\mathbf{f},\mathbf{e}'))} \left\{ \frac{\max_{f' \in \mathcal{Q}_f^n} L_{f'} + \sum_{f' \in \mathcal{N}^n} a_{f'}^n}{r_n} + 2T_n + \Delta_n \right\} \right) + \sigma_f \tag{4.2}$$

where:

- $\mathbf{m}_0$ *is an element of* $\mathcal{M}$*, such that* $\forall f \in \mathcal{F}$*,* $\forall e$*,* $(m_0)_f^e = L_f$ *with* L_f *the maximum packet size for flow* f*;*
- $\forall f, f' \in \mathcal{F}$ *and* $\forall e$*,* $\mathcal{G}(f, f', e)$ *is the set of maximal common subpaths between flows* f *and* f' *in the* e*-th strongly connected component; and*
- $\forall f, f' \in \mathcal{F}$*,* $\forall e$*,* $\forall \mathbf{n} = (n_1, ..., n_K) \in \mathcal{G}(f, f', e)$*,*

$$S(\mathbf{n}) = \frac{1}{r_{n_1}} + \sum_{j=2}^{K} \left(\frac{1}{r_{n_j}} - \frac{1}{r_{n_{j-1}}} \right)^+$$

The proof of this result is in Section 4.8.

4.5 A RIN Result for heterogeneous networks and leaky bucket flows

We can observe that the operator Π derived in Section 4.4 is linear. Therefore in this case, following the method described in Chapter 3, we do not need to derive an operator that upper bounds Π, but we can consider directly the matrix associated to Π. As a result, we have the following theorem:

Theorem 4.5.1 (Generalized Source Rate Condition) *With the given assumptions on the network, and using* $\forall f$*,* $\forall e$ *the notation* $\mathbf{path}(f, e) = (n_1^e, ..., n_{K^e}^e)$*, if for any flow* f *its rate* ρ_f *satisfies the condition* $\rho_f < \min_e h^e$*, with:*

$$h^e = \left[\frac{N^{n_1^e}}{r_{n_1^e}} + \sum_{j=2}^{K^e} \frac{N^{n_j^e} - Q_f^{n_j^e}}{r_{n_j^e}} + Q_f^{n_j^e} \left(\frac{1}{r_{n_j^e}} - \frac{1}{r_{n_{j-1}^e}} \right)^+ \right]^{-1} \tag{4.3}$$

then the network is stable.

An alternate form of the condition 4.3 is the following:

$$\forall f,\ \rho_f < \min_e \left[\sum_{f' \in \mathcal{F}} \sum_{\mathbf{n}' \in \mathcal{G}(f,f',e)} S(\mathbf{n}') \right]^{-1}$$

Proof: We know from Section 3.4.2 that if the spectral radius of the matrix associated to the operator Π is smaller than one, then the network is stable. From the expression of the operator Π in Theorem 4.4.1 we see that the upper bounds to those variables associated to a given strongly connected component e are function of variables associated to components with $e' \leq e$. As the graph of the strongly connected components of a network has a feed-forward structure, the matrix associated to Π can be put in block triangular form, with one block per strongly connected subnet. Then the spectral radius of the matrix is the maximum of the spectral radius among all the diagonal blocks.
An upper bound to the spectral radius of each block is given by the maximum row sum of the block [27]. By applying this result on the matrix associated to Π, we get the bound in Equation (4.3). □

In what follows, we will address the result in Theorem 4.5.1 as the "generalized RIN" (or GRIN) result: Indeed, one of the main features of this theorem is that it generalizes the "RIN result". It is possible to show that the "RIN result" can be entirely derived as a special case of Theorem 4.5.1:

Corollary 4.5.1 *We consider the homogeneous network model, with time divided into slots, and node synchronized between them. We assume packet arrivals and departures can only take place at integer time units. We assume arrival curves are of the staircase type. If for any flow f in the network the packet inter-arrival time τ_f at the source satisfies the condition $\tau_f \geq RIN_f + 1$, where RIN_f is the* route interference number *of flow f [14] relative to the whole network, then the network is stable.*

Proof: We note that each flow constrained by a staircase arrival curve with period τ_f is also constrained by a leaky bucket arrival curve, with rate $\rho_f = \frac{1}{\tau_f}$ and burstiness equal to one. By applying Theorem 4.5.1 in this setting, we can verify that if $\forall f,\ \rho_f < \frac{1}{RIN_f+1}$ then the network is stable.
In order to pose this condition in terms of packet inter-arrival time at the source, we note that if $\forall f,\ \rho_f = \frac{1}{RIN_f+1}$, then each flow f is constrained by $\left\lfloor 1 + \frac{t}{RIN_f+1} \right\rfloor$, and this implies a minimum inter arrival time of RIN_f time slots for flow f. Therefore a sufficient condition for stability is that $\forall f$, the inter-arrival time be $\geq RIN_f + 1$. □

We can therefore conclude that the sufficient stability conditions in Theorem 4.5.1 extend the "RIN result" to leaky bucket constrained flows and removes all the constraints associated to the *network homogeneity*.

The condition in Theorem 4.5.1 can be used as a rule for choosing flow paths and rates in such a way as to have a stable network: It gives an indication as to what features of flow paths can influence stability.
The quantity at the denominator of the bound in Equation (4.3) can be interpreted as a generalization of the concept of route interference number (RIN) of a flow. In this generalized version, the weight of each interference is related to the characteristics of the common subpath between the two flows that interfere starting at that node. A contribution to this weight is inversely proportional to the service rate of the node at which the interference takes place. Also the differences in service rates at the nodes which lie downwards the node at which the interference takes place have a role in increasing the weight of each interference event.

Some more observations:

- A first difference with respect to the “RIN result” is that while that result is derived by implicitly assuming the network to be empty at $t = 0$, the sufficient conditions derived here are valid independently of the buffer content at nodes at $t = 0$.
- In the homogeneous network settings, the generalized RIN result actually performs better than the “RIN result”, as it considers separately each strongly connected subnet, instead of looking at the whole network at once. As a consequence, from the “RIN result” we cannot derive the stability of feed-forward networks for any value of node utilization inferior to one.
- Finally, we note that in the heterogeneous setting, the new sufficient conditions in Theorem 4.5.1 hold also for staircase arrival curves, as each flow constrained by a staircase arrival curve with period τ is also leaky bucket constrained, with rate $\rho = \frac{1}{\tau}$.

Moreover, in case all the assumptions of the original RIN result hold, but the arrival curves are of the leaky bucket type, from Theorem 4.5.1 we derive the following result:

Corollary 4.5.2 *With the same assumptions as for Corollary 4.5.1, but when flows are leaky bucket constrained, a sufficient condition for stability is given by*

$$\forall f,\ \rho_f < \frac{1}{RIN_f + 1} \tag{4.4}$$

where RIN_f is the route interference number *of flow f.*

This is a sufficient condition for stability similar to the Source Rate Condition (Section 2.4.2), valid for homogeneous networks, but for arrival curves of the leaky bucket type.

4.5.1 Delay bounds

When the Generalized RIN condition is satisfied, there exist a unique solution $\mathbf{m}^*$ of the fixed point problem $\Pi(\mathbf{m}) = \mathbf{m}$. Then the derivation of delay bounds is based on the following result:

Theorem 4.5.2 *If $\mathbf{m}^*$ is finite, then for each node n in the network an upper bound to packet delay is given by*

$$\min_{f \in \mathcal{N}^n} \left\{ \sum_{f' \in \mathcal{N}^n \setminus \mathcal{Q}_f^n} \frac{m_{f'}^*}{r_n} + \sum_{f' \in \mathcal{Q}_f^n} m_{f'}^* \left(\frac{1}{r_n} - \frac{1}{r_{prec(n,f)}} \right)^+ \right\} +$$

$$+ \frac{\sum_{f' \in \mathcal{N}^n} a_{f'}^n + \max_{f' \in \mathcal{N}^n} L_{f'}}{r_n} + T_n + \Delta_n \tag{4.5}$$

*Proof:*It derives immediately from Lemma 4.8.1 in Section 4.8. □

4.5.2 Other possible sufficient conditions

As we have seen, the generalized RIN result derives from imposing that the maximum of the row sums from all the diagonal blocks of the matrix associated to the operator Π be smaller than one, as this constitutes an upper bound to the the spectral radius of that matrix.
Using different upper bounds to the spectral radius, many other sufficient conditions can be derived. These conditions can be derived using the following result:

Theorem 4.5.3 (Upper Bound to Spectral Radius [27]) *For any induced matrix norm $\| \ \|$, and for any matrix $A \in \mathbb{R}^{n \times n}$, if with $\rho(A)$ we indicate its spectral radius, we have that $\rho(A) \leq \|A\|$.*

The Generalized RIN result is derived from applying the norm 1 (row sums), over the matrix associated to the operator Π. By Theorem 4.5.3, using different norms, we can obtain other sufficient conditions.

Let's consider for simplicity the homogeneous settings, and a network composed by a single strongly connected component. For any couples of flows $f, \ f'$, we indicate with $n_{f,f'}$ the total number of times that the two flows interfere in the network. Then if we apply the norm infinite (column sum), we get the following sufficient condition:

$$\forall f,\ C_f = \sum_{f'} n_{f,f'} \rho_{f'} + \rho_f < 1 \tag{4.6}$$

where C_f represents the column sum relative to flow f. A way to look at this condition is to observe that it is equivalent to a node serviceability condition at a node traversed by all those flows that flow f meets along its path (flow f included), and where each flow f' is counted with multiplicity $n_{f,f'}$.

Another sufficient condition can be derived by imposing that maximum of the geometric norm of the i-th row and the i-th column, be smaller than one. If we indicate $\forall f,\ \rho_{f,RIN} = 1/(\sum_{f'} n_{f,f'})$, we have

$$\forall f,\ \sqrt{C_f \frac{\rho_f}{\rho_{f,RIN}}} < 1 \tag{4.7}$$

Similarly, from the algebraic norm, we get

$$\forall f,\ \frac{C_f + \frac{\rho_f}{\rho_{f,RIN}}}{2} < 1 \tag{4.8}$$

which can be also written as

$$\forall f,\ \rho_f + \sum_{f'} n_{f,f'} \left(\frac{\rho_f + \rho_{f'}}{2} \right) < 1 \tag{4.9}$$

4.6 Tightness of the Generalized Source Rate Condition

For a generic network, determining the tightness of the sufficient conditions for stability in Theorem 4.5.1 is an open issue. Indeed, this would imply having a method by which we can determine, for a generic network, what are the settings that can lead to a queue size at nodes (at least to some nodes in the network) which grows in time without bound. To the best of our knowledge, such method is not known up to now for the considered settings (leaky bucket constrained flows, FIFO aggregate scheduling nodes). One of the best known examples of unstable networks in this setting, given by Andrews in [3], has been described in Section 2.1.2. We therefore checked the tightness of the sufficient conditions in Theorem 4.5.1 by applying them over the network described in [3]. We see that, with the flow rate allocation described in this instability example, the maximum node utilization is at least a factor of K^3 larger than the one we obtain when imposing on flow rates the Generalized RIN sufficient conditions, where $K \geq 2$ is a parameter relative to the size of the network and to the path length of a flow.

Another way to get an idea of how the GRIN conditions perform is to apply them to some network examples for which we know their stability behavior.
One example is given by feed-forward networks, known to be always stable, provided that node utilization is inferior to one. For these networks, each node represents a strongly connected component of the network: Sufficient conditions in Theorem 4.5.1, applied to each node in isolation, always imply a maximum node utilization of 1.
Another example of a network that is known to be always stable is the ring [35], provided that node utilization is strictly smaller than one. In these networks, sufficient conditions in Theorem 4.5.1 perform differently according to the number of flows in the ring and to their path, and in general they impose tighter conditions on flow rates than those derived by the natural condition at each node.

4.7 Numerical assessment of the results

The generalized version of the RIN result carries with itself a set of parameters, that depend on various features of the network. In order to assess numerically the quality of the sufficient conditions in Theorem 4.5.1 in function of those parameters, we applied the GRIN conditions over a set of 100 networks.

Generation of the physical topology of the networks. The physical topology of the considered networks has a core-edge structure. The physical network can be represented by an undirected graph, where nodes are routers, and edges are the bidirectional links interconnecting them. We divided nodes in the network in two categories: The ”core” nodes, usually with high degree, that carry mostly long haul traffic, and interconnected between them through high speed links; the ”edge” nodes, lying in ”peripheral” parts of the graph.
In our model, each core node is connected to a small ”regional” network, with nodes interconnected by lower capacity links. As to link capacity distribution, we allocated a capacity of 2488 Mbit/s for all links between core nodes, and we assumed the capacity for all the other links to be uniformly distributed between 100 Mbit/s and 2488 Mbit/s. We assume every edge node to be of degree at least two (we assume they are like this

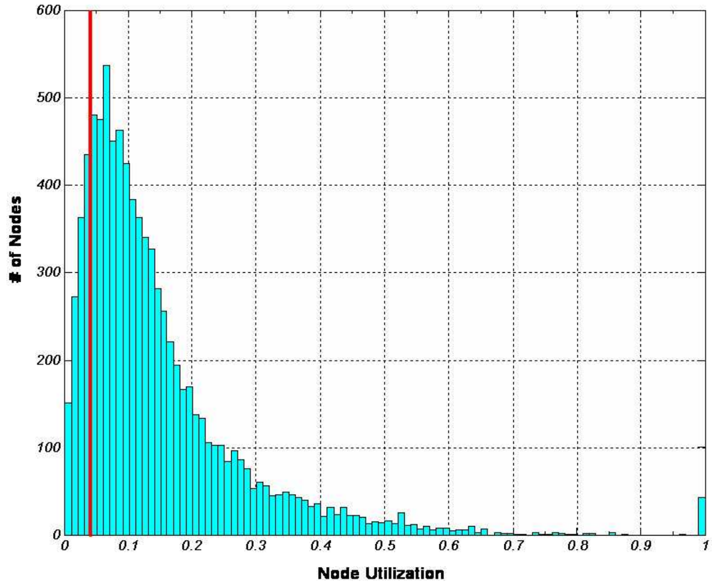

Figure 4.1 – Distribution of node utilization in 100 networks, with all flow rates equal to 99.5% of the bounds in Theorem 4.5.1. The average node utilization is of 0.1460, with a standard deviation of 0.1340. The vertical line represents the maximum node utilization value that can be derived by applying the "Charny-Le Boudec bound" [12], equal to 0.0455 for the considered networks. The new sufficient conditions for stability bring an *average* network utilization which is more than three times larger than the *maximum* node utilization that we can have by applying the "Charny-Le Boudec bound".

for resilience purpose). The physical topology of each network was generated with the Barabasi-Albert method, using the BRITE network generator [29].

Traffic matrix and flow paths distribution. For each possible couple of nodes, we assumed a bidirectional traffic relation. In order to allocate paths to flows, we used the Dijkstra algorithm, setting at each step link weights proportional to the number of flows traversing them, in order to have traffic more uniformly distributed throughout the network. The resulting networks have a number of links between 80 and 100, a total number of flows between 180 and 200 an average hop count of 10.33, and a maximum hop count of 23. The shape of the hop count distribution is close to experimental ones in [5], but the experimental ones have an higher average (around 18).

In Fig. 4.1 we have the empirical distribution of node utilization in the considered networks, where flow rates have been set to 99.5% of the value determined by the bounds in Theorem 4.5.1.

We see that we have an average node utilization of 0.1460, with a standard deviation of 0.1340. The only other available result for this kind of network and flow arrival rates is the “Charny-Le Boudec bound”, that depends on the maximum hop count in the network. Experimentally [5] it has been verified that the maximum hop count in a network can be larger than 23: so the “Charny-Le Boudec bound” would bring to a maximum node utilization in the network of 0.0455. Therefore, in this network the new sufficient conditions for stability bring an average node utilization which is more than three times larger than the (maximum) node utilization that we can have by applying the “Charny-Le Boudec bound”.

Comparison to other generalizations of the RIN Result

In [30] Otel derives a generalization of the “RIN result” to heterogeneous networks. This result is based on a sufficient condition on stability in 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 in the framework of network calculus. That is, this sufficient condition cannot be cast into an arrival curve constraint. This makes the result not very useful in practical cases. Moreover, in the sufficient condition in [30] 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. As an example, in a network where all flows satisfy the sufficient condition in [30] with a maximum packet size of 60 bytes, increasing the maximum packet size to 1500 bytes would decrease the upper bound to maximum network utilization by a factor of 25.

4.8 Proof of Theorem 4.4.1

In order to derive Theorem 4.4.1, we need to introduce some more definitions related to a super chain, and a partial result, relative to an upper bound to packet delay at a node.

Definition 4.8.1 (Super chain time) *The* super chain time *of a super chain is the time at which the last packet in the super chain is served at the last node of the super chain.*

Definition 4.8.2 (Duration of a super chain) *The* duration *of a super chain with time t is the time interval from the emission time of the first packet of the super chain up to time t.*

Also, with reference to Definition 4.2.2 of a super chain, 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 4.8.1 *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*

$$\frac{\Theta}{r_n} + \Gamma\left(\frac{1}{r_n} - \frac{1}{r'}\right)^+ + \frac{A + \max_{f' \in Q_f^n} L_{f'}}{r_n} + T_n + \Delta_n \tag{4.10}$$

where Θ and Γ 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 (Lemma 4.8.1): 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 Equation (4.10) 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. □

Proof (Theorem 4.4.1): We proceed by induction on the index p of relevant network events.

Base case: p=1. In order to upper bound the quantities $m_f^e[1]$, 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 $n \in \mathcal{I}(\mathbf{path}(f, e))$.

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

$$m_f^e[1] \leq \sigma_f^0 + \rho_f t_0 + a_f^n$$

Now we have that

$$t_0 \leq \frac{m_f^e[0] + m_{f'}^e[0] + L_f}{r_n} + \Delta_n + T_n$$

as the first packet of the considered super chain belongs to flow f, so inequality in Equation (4.1) holds.

Induction step. We derive now an upper bound to the variable $m_f^e[p]$. This quantity is upper bounded by the sum of two contributions:

- the first contribution is the total number of bytes from all the flow f's packets, present at $t = 0$ in all the strongly connected components e', with $e' \leq e$; and
- we consider the set of all the super chains relative to flow f, whose path lays inside the e-th strongly connected subnet, and with time $\leq t_{p-1}$. The second contribution is given by the maximum number of bytes that can be emitted by the source of flow f in a time interval equal to the maximum duration among all the super chain in this set. We consider super chains with time $\leq t_{p-1}$, and not time t_p, as in order to be served by time t_p or before at the last node of the super chain, a packet must arrive at that node by time t_{p-1}.

In what follows, we derive an upper bound to this maximal super chain duration.

Let's consider the nodes in the set $\bigcup_{e' \leq e} I(\mathbf{path}(f, e'))$. For each node n'_k in this set, we define the following quantities:

- $\Delta t_{int,k}$ is the delay experienced at the node by the open packet at that node (for nodes not in the super chain we consider the open packet to coincide with the close packet);
- Γ^k is the sum of packet sizes for packets served at the node before the open packet, in the same busy period as the open packet, and arriving at the node from the same origin as the open packet; and
- Θ^k is the sum of packet sizes for packets served at the node before the open packet, in the same busy period as the open packet, and arriving at the node from nodes (or sources) other than the one of the open packet.

Then using Lemma 4.8.1 we have for $k \geq 1$

$$\Delta t_{int,k} \leq \frac{\Theta^k}{r_{n'_k}} + \Gamma^k \left(\frac{1}{r_{n'_k}} - \frac{1}{r_{prec(n'_k, f)}} \right)^+ +$$

$$+ \frac{A'_k + \max_{f' \in Q_f^{n'_k}} L_{f'}}{r_{n'_k}} + T_{n'_k} + \Delta_{n'_k} = B_k + T_{n'_k} + \Delta_{n'_k} \qquad (4.11)$$

where A'_k is relative to packets present at the node at $t = 0$.

If Δt_k is the time interval between the departure time of the open packet and the departure time of the close packet, using the definition of strict service curve, we write

$$\Delta t_{int,k} + \Delta t_k \leq B_k + 2T_{n'_k} + \Delta_{n'_k} + \frac{\gamma^k + \theta^k + A_k}{r_{n'_k}} \tag{4.12}$$

where:

- γ^k represents the sum of packet sizes for packets included in the considered super chain and arrived at node n'_k from node n'_{k-1} (or from the same source as flow f);
- θ^k represents the sum of packet sizes for packets included in the considered super chain and arrived at node n'_k from nodes other than n'_{k-1} (or from fresh flows different than f); and
- 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 the bounds in Equation (4.12) over all the nodes in the path of flow f.

At each node n'_k, each of the terms $\gamma^k,\ \theta^k,\ \Gamma^k,\ \Theta^k,\ A_k,\ A'_k$ 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 $\Theta^k = \sum_{f' \in \mathcal{N}^{n'_k}} \Theta^k_{f'}$. For any flow f' (not necessarily distinct from f), $\forall \mathbf{n}' = (n'_l, ..., n'_{l+K'}) \in \mathcal{G}(f, f', e)$, the contribution to the upper bound to the super chain duration is

$$\frac{\Theta^l_{f'} + \theta^l_{f'}}{r_{n'_l}} + \sum_{k=l}^{l+K'} \left[\Gamma^k_{f'} \left(\frac{1}{r_{n'_k}} - \frac{1}{r_{n'_{k-1}}} \right)^+ + \frac{\gamma^k_{f'} + a^{n'_k}_{f'}}{r_{n'_k}} \right] \tag{4.13}$$

By definition of the variables $m^e_f[p]$, we have that

$$((...(\Theta^l_{f'} + \theta^l_{f'} + \Gamma^l_{f'} + \gamma^l_{f'}) \vee \Gamma^{l+1}_{f'} + \gamma^{l+1}_{f'}) \vee \Gamma^{l+2}_{f'} + \gamma^{l+2}_{f'}) \vee ... \vee \Gamma^{l+K'}_{f'} + \gamma^{l+K'}_{f'} \leq m^e_{f'}[p-1]$$

Therefore, the quantity in Equation (4.13) is upper bounded by

$$m^e_{f'}[p-1] S(\mathbf{n}') + \sum_{n' \in \mathbf{n}'} \frac{a^{n'}_{f'}}{r_{n'}} \tag{4.14}$$

If a given super chain $(\mathbf{c}, \mathbf{n})$ is such that $\mathbf{n} \subseteq \mathbf{path}(f, e)$ but $\mathbf{n} \neq \mathbf{path}(f, e)$, for any node $n'_k \in \mathcal{U}_f^{n_1}$ we have that $\Delta t_k = 0$, and $\Delta t_{int,k}$ is upper bounded as in Equation (4.11). Then for any super chain 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 super chain duration:

$$\sum_{e' \leq e} \sum_{f' \in \mathcal{F}} m_{f'}^{e'}[p-1] \sum_{\mathbf{n}' \in \mathcal{G}(f, f', e')} S(\mathbf{n}') +$$
$$+ \sum_{n \in \mathcal{I}(\mathbf{path}(f,e))} \left(\frac{\max_{f' \in \mathcal{Q}_f^n} L_{f'} + \sum_{f' \in \mathcal{N}^n} a_{f'}^n}{r_n} + 2T_n + \Delta_n \right)$$

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 expression in Equation (4.2). □

4.9 Summary

In the present chapter we showed an example of application of the method described in Chapter 3. As a result, we derived a generalized version of the RIN result to heterogeneous networks and leaky bucket constrained flows. We showed with a numerical example that in heterogeneous network settings the new conditions imply an average network utilization which is much larger than the one obtainable with the main existing result valid in these settings.

Chapter 5

Making the Most of the New Approach

In this chapter,

- we apply the algebraic method described in Chapter 3 in a network of FIFO nodes, on three different classes of variables, associated to maximum node delay, to maximum flow burstiness, and to the maximum number of bytes included in a super chain. We derive the expression of an upper bounding operator, which is complex and nonlinear;
- we derive a new set of sufficient conditions for stability, valid in heterogeneous settings, and we describe a polynomial time algorithm to test these conditions on a generic network;
- we describe an algorithm for delay bound computation; and
- we assess numerically the obtained results on a network example.

The content of this chapter is mainly derived from [34].

5.1 Model and assumptions

The assumptions for the network and the flows are the same as in Section 4.1.

According to the criteria described in Section 3.2, the choice of variables in the network is the following:

- For any flow f in the network, for any strongly connected component e, and for any subpath $\mathbf{n}$ of the path of flow f inside the e-th strongly connected component, for any time instant t_p, we define $m_f^{\mathbf{n}}[p]$ as the maximum number of bytes belonging to f that can be included in any super chain $(\mathbf{c}, \mathbf{n}')$ relative to flow f, up to time t_p, and with $\mathbf{n}' \subseteq \mathbf{n}$.

Table 5.1 – Notation used in Chapter 5.

Symbol	Definition
$\mathcal{P}_f^e$	Set of all the possible subsequences of $\mathbf{path}(f,e)$.
$\mathcal{M}$	Set of all the arrays of quantities $m_f^{\mathbf{n}} \in \mathbb{R} \cup \{+\infty\}$, indexed by $\mathbf{n} \in \mathcal{P}_f$ and $f \in \mathcal{F}$.
$\mathbf{m}$	An element of the set $\mathcal{M}$.
$\mathbf{m}[p]$ $(p = 0, 1, 2, ...)$	A sequence of elements of $\mathcal{M}$, indexed by p.
$\mathcal{D}$	Set of all the arrays of the quantities $d^{\mathbf{n}} \in \mathbb{R} \cup \{+\infty\}$, for all $n \in \mathcal{N}$.
$\mathbf{d}$	An element of the set $\mathcal{D}$, i.e. an array of numbers indexed by n.
$\mathbf{d}[p]$ $(p = 0, 1, 2, ...)$	A sequence of elements of $\mathcal{D}$, indexed by p.
$\mathcal{S}$	Set of all the arrays of the quantities $\sigma_f^n \in \mathbb{R} \cup \{+\infty\}, \forall n \in \mathcal{N}, \forall f \in \mathcal{F}$.
$\boldsymbol{\sigma}$	An element of the set $\mathcal{S}$, i.e. an array of numbers indexed by n and f.
$\boldsymbol{\sigma}[p]$ $(p = 0, 1, 2, ...)$	A sequence of elements of $\mathcal{S}$, indexed by p.
$\mathcal{N}$ ($\mathcal{N}^e$), N (N^e)	Set of all nodes in the network, (resp. in the e-th strongly connected component), of cardinality N (N^e)

- For any node n, for any time instant t_p, we indicate with $d^n[p]$ the maximum packet delay at node n up to time t_p.
- For any flow f, for any node n, and for any time instant t_p, we define the variable $\sigma_f^n[p]$ as the maximum flow burstiness (see Section 2.3.1) of flow f at the input to node n, up to time t_p.

The justification of this choice lies in the fact that all the three classes of variables are related to the definition of stability of a network. Indeed, in an unstable network, maximum packet delay grows over time without bound, as well as the maximum duration of a busy period at a node does, to which the concept of super chain is related, and the maximum flow burstiness. In Table 5.1 we have a description of the notation used in the present chapter.

5.2 An upper bounding operator for three variables classes

In this section we derive the expression of an operator that upper bounds the value of the variables in each of the three classes defined in Section 5.1 at a given time t_p, in function of the values of the variables at time t_{p-1}, $\forall p > 0$.

The derivation of an upper bound to the value of each variable in the three classes at time t_p brings together the approach taken in the derivation of the operator in Chapter 4 with the use of some network calculus results relative to bounds at the output of an aggregate scheduling node (Section 2.2.3). More specifically:

- the bounds to variables associated to super chains are derived in a similar way as for the derivation of the operator Π in Chapter 4;
- the bounds to variables associated to maximum packet delay exploit Theorem 2.2.2; and
- the bounds to variables associated to maximum packet delay are derived by using the output bound for FIFO aggregate schedulers (Theorem 2.2.5).

The detailed expression of the resulting operator is presented in Theorem 5.2.1.

Theorem 5.2.1 *For any integer $p > 0$, we have:*

$$\begin{cases} (\mathbf{m}[p], \mathbf{d}[p], \boldsymbol{\sigma}[p]) \leq \Pi(\mathbf{m}[p-1], \mathbf{d}[p-1], \boldsymbol{\sigma}[p-1]) \\ \mathbf{m}[0] \leq \mathbf{m}_0 \\ \mathbf{d}[0] \leq \mathbf{d}_0 \\ \boldsymbol{\sigma}[0] \leq \boldsymbol{\sigma}_0 \end{cases} \tag{5.1}$$

where $\Pi : \mathcal{M} \times \mathcal{D} \times \mathcal{S} \rightarrow \mathcal{M} \times \mathcal{D} \times \mathcal{S}$ is the operator defined by $\Pi(\mathbf{m}, \mathbf{d}, \boldsymbol{\sigma}) = (\mathbf{m}', \mathbf{d}', \boldsymbol{\sigma}')$, with $\forall f \in \mathcal{F}$, $\forall e$, $\forall \mathbf{n} = (n_1, ..., n_K) \in \mathcal{P}_f^e$,

$$m_f'^{\mathbf{n}} = \sum_{e' \leq e} \sum_{n \in \mathcal{I}(\mathbf{path}(f,e'))} a_f^n + \sigma_f^0 + \rho_f \left(\sum_{n \in \mathcal{U}_f^{n_1}} (b^n \wedge d^n) + \sum_{f'} \sum_{\mathbf{n}' \in \mathcal{G}(f,f',\mathbf{n})} m_{f'}^{\mathbf{n}'} S(\mathbf{n}') + \right.$$

$$\left. + \sum_{n \in \mathcal{I}(\mathbf{n})} \left\{ \frac{\max_{f' \in \mathcal{Q}_f^n} L_{f'} + \sum_{f' \in \mathcal{N}^n} a_{f'}^n}{r_n} + 2T_n + \Delta_n \right\} \right) \tag{5.2}$$

$$d'^n = \sum_{f' \in \mathcal{N}^n} \frac{a_{f'}^n}{r_n} + T_n + \Delta_n + \min_{f \in \mathcal{N}^n} \left\{ \sum_{f' \in \mathcal{N}^n \setminus \mathcal{Q}_f^n} \frac{m_{f'}^n}{r_n} + \frac{\max_{f' \in \mathcal{N}^n} L_{f'}}{r_n} + \right.$$

$$+ \sum_{f' \in \mathcal{Q}_f^n} m_{f'}^n \left(\frac{1}{r_n} - \frac{1}{r_{prec(n,f)}} \right)^+ \Bigg\} \wedge \frac{\sum_{f' \in \mathcal{N}^n} \left(\sigma_{f'}^0 + \rho_{f'} \sum_{n' \in \mathcal{U}_{f'}^{n'}} d^{n'} \right) \wedge \sigma_{f'}^n}{r_n} \tag{5.3}$$

$\forall f \in \mathcal{F}$, for any node n on the path of flow f and different than the first node of its path,

$$\sigma_f'^n = \sigma_f^{prec(n,f)} + \rho_f \left(T_n + \frac{\sum_{f' \in \mathcal{N}^n, f' \neq f} (\sigma_{f'}^{prec(n,f)} + a_{f'}^n) + a_f^n + L_f}{r_n} \wedge d^n \right) \tag{5.4}$$

If n is the first node on the path of flow f, then it simply holds $\sigma_f'^n \leq \sigma_f^0$.
$\forall n$, the quantities b^n in Equation (5.2) are defined as follows:

$$b^n = \min_{f' \in \mathcal{N}^n \setminus \mathcal{H}(f,n,n_1)} \Bigg\{ \sum_{f'' \in \mathcal{N}^n \setminus (\mathcal{Q}_{f'}^n \cup \mathcal{H}(f,n,n_1))} \frac{m_{f''}^n}{r_n} + \frac{\max_{f'' \in \mathcal{N}^n} L_{f'}}{r_n} +$$

$$+ \sum_{f'' \in \mathcal{Q}_{f'}^n} m_{f''}^n \left(\frac{1}{r_n} - \frac{1}{r_{prec(n,f')}} \right)^+ \Bigg\} \wedge$$

$$\wedge \frac{\sum_{f' \in \mathcal{N}^n \setminus \mathcal{H}(f,n,n_1)} (\rho_{f'} (\sum_{n'' \in \mathcal{U}_{f'}^n} d^{n''}) + \sigma_{f'}^0)}{r_n} + \sum_{f' \in \mathcal{N}^n} \frac{a_{f'}^n}{r_n} + T_n + \Delta_n \tag{5.5}$$

where:

- $\mathbf{m}_0$ *is an element of* $\mathcal{M}$*, such that* $\forall f \in \mathcal{F}$, $\forall e$, $\forall \mathbf{n} \in \mathcal{P}_f^e$, $(m_0)_f'^{\mathbf{n}} = L_f$ *with* L_f *the maximum packet size for flow* f*;*
- $\mathbf{d}_0$ *is an element of* $\mathcal{D}$*, such that* $\forall n \in \mathcal{N}$, $d_0^n = \frac{\max_{j \in \mathcal{N}^n} L_j}{r_n} + T_n + \Delta_n$*;*
- $\forall f \in \mathcal{F}$, $\forall \mathbf{n} = (n_1, ..., n_K)$, $S(\mathbf{n}) = \frac{1}{r_{n_1}} + \sum_{j=2}^K \left(\frac{1}{r_{n_j}} - \frac{1}{r_{n_{j-1}}} \right)^+$

- $\mathcal{H}(f,n,n')$ *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;*
- $\forall f, f' \in \mathcal{F}$*,* $\forall e$*,* $\forall \mathbf{n} \in \mathcal{P}_f^e$*, consider the set* $\{\mathbf{n}' = (n'_0, ..., n'_k) : \mathbf{n}' \in \mathcal{P}_f^e \cap \mathcal{P}_{f'}^e,\ n'_k \in \mathcal{I}(\mathbf{n})\}$*. Then* $\mathcal{G}(f,f',\mathbf{n})$ *is the subset of all the maximal elements of that set.*
- $\forall n$*,* $\forall f \in \mathcal{N}^n$*,* $(\boldsymbol{\sigma}_0)_f^n = \sigma_f^0$*.*

The proof of this result is in Section 5.7.1.

When flows are constrained by a generic arrival curve, indicating with α_f an arrival curve for flow f, a similar operator can be derived:

Corollary 5.2.1 *For any integer* $p > 0$*, we have:*

$$\begin{cases} (\mathbf{m}[p], \mathbf{d}[p]) \leq \Pi(\mathbf{m}[p-1], \mathbf{d}[p-1]) \\ \mathbf{m}[0] \leq \mathbf{m}_0 \\ \mathbf{d}[0] \leq \mathbf{d}_0 \end{cases} \tag{5.6}$$

where $\Pi : \mathcal{M} \times \mathcal{D} \to \mathcal{M} \times \mathcal{D}$ *is the operator defined by* $\Pi(\mathbf{m}, \mathbf{d}) = (\mathbf{m}', \mathbf{d}')$*, with* $\forall f \in \mathcal{F}$*,* $\forall e$*,* $\forall \mathbf{n} = (n_1, ..., n_K) \in \mathcal{P}_f^e$*,*

$$m_f'^{\mathbf{n}} = \sum_{e' \leq e} \sum_{n \in \mathcal{I}(\mathbf{path}(f,e'))} a_f^n + \alpha_f \left(\sum_{n \in \mathcal{U}_f^{n1}} (b^n \wedge d^n) + \sum_{f'} \sum_{\mathbf{n}' \in \mathcal{G}(f,f',\mathbf{n})} m_{f'}^{\mathbf{n}'} S(\mathbf{n}') + \right.$$
$$\left. + \sum_{n \in \mathcal{I}(\mathbf{n})} \left\{ \frac{\max_{f' \in \mathcal{Q}_f^n} L_{f'} + \sum_{f' \in \mathcal{N}^n} a_{f'}^n}{r_n} + 2T_n + \Delta_n \right\} \right) \tag{5.7}$$

$$d'^n = \sum_{f' \in \mathcal{N}^n} \frac{a_{f'}^n}{r_n} + T_n + \Delta_n + \min_{f \in \mathcal{N}^n} \left\{ \sum_{f' \in \mathcal{N}^n \setminus \mathcal{Q}_f^n} \frac{m_{f'}^n}{r_n} + \frac{\max_{f' \in \mathcal{N}^n} L_{f'}}{r_n} + \right.$$
$$\left. + \sum_{f' \in \mathcal{Q}_f^n} m_{f'}^n \left(\frac{1}{r_n} - \frac{1}{r_{prec(n,f)}} \right)^+ \right\} \wedge \sup_{\tau \geq 0} \left[\frac{\sum_{f' \in \mathcal{N}^n} \alpha_{f'} \left(\sum_{n'' \in \mathcal{U}_{f'}^n} d^{n''} + \tau \right)}{r_n} - \tau \right] \tag{5.8}$$

with

$$b^n = \min_{f' \in \mathcal{N}^n \setminus \mathcal{H}(f,n,n_1)} \left\{ \sum_{f'' \in \mathcal{N}^n \setminus (\mathcal{Q}^n_{f'} \cup \mathcal{H}(f,n,n_1))} \frac{m^n_{f''}}{r_n} + \frac{\max_{f'' \in \mathcal{N}^n} L_{f'}}{r_n} + \right.$$

$$\left. + \sum_{f'' \in \mathcal{Q}^n_{f'}} m^n_{f''} \left(\frac{1}{r_n} - \frac{1}{r_{prec(n,f')}} \right)^+ \right\} \wedge$$

$$\wedge \sup_{\tau \geq 0} \left[\frac{\sum_{f' \in \mathcal{N}^n \setminus \mathcal{H}(f,n,n_1)} \alpha_{f'} (\sum_{n'' \in \mathcal{U}^n_{f'}} d^{n''} + \tau)}{r_n} - \tau \right] + \sum_{f' \in \mathcal{N}^n} \frac{a^n_{f'}}{r_n} + T_n + \Delta_n \quad (5.9)$$

5.3 A linear upper bounding operator that allows to derive practical results

As we can see from its expression, the operator in Theorem 5.2.1 is nonlinear, and involves a number of variables that grows very rapidly with the size of the network and the number of flows in it ($< FN! + FN$, where F and N are respectively the total number of flows and of nodes in the network).

In this case it is convenient, in order to derive practical results, to use Theorem 3.4.2, by deriving a monotone operator Ψ that upper bounds Π, and with a simpler expression.

Lemma 5.3.1 *For any integer $p > 0$, we have:*

$$\begin{cases} (\mathbf{m}[p], \mathbf{d}[p], \sigma[p]) \leq \Psi(\mathbf{m}[p-1], \mathbf{d}[p-1], \sigma[p-1]) \\ \mathbf{m}[0] \leq \mathbf{m}_0 \\ \mathbf{d}[0] \leq \mathbf{d}_0 \\ \boldsymbol{\sigma}[0] \leq \boldsymbol{\sigma}_0 \end{cases} \quad (5.10)$$

Ψ is the operator $\mathcal{M} \times \mathcal{D} \times \mathcal{S} \to \mathcal{M} \times \mathcal{D} \times \mathcal{S}$ defined by $\Psi(\mathbf{m}, \mathbf{d}, \boldsymbol{\sigma}) = (\mathbf{m}^\star, \mathbf{d}^\star, \boldsymbol{\sigma}^\star)$, where:

- *for any flow f, $\forall e$, and any subpath $\mathbf{n}$ of $\mathbf{path}(f,e)$, for $m^{\star\mathbf{n}}_f$ it holds the upper bound obtained from the one in Equation (5.2), by replacing for any interfering flow f', any variable $m^{\mathbf{n}'}_{f'}$ with the variable $m^{\mathbf{path}(f,e)}_{f'}$, and by replacing at any node n, $b^n \wedge d^n$ with d^n;*

- *for any node n, for $d^{\star n}$ it holds the upper bound obtained from the one in Equation (5.3), by replacing for any flow f',and any strongly connected component e, the variables $m^n_{f'}$ (with $n \in \mathcal{N}^e$) with $m^{\mathbf{path}(f,e)}_{f'}$; and*
- *for any flow f and any node n, for $\sigma^{\star n}_f$ it holds the sum of the upper bound in Equation (5.4), plus the quantity σ^0_f.*

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 $m^{\mathbf{n}}_f[p] \leq m^{\mathbf{n}'}_f[p]$. We then observe that in Equation (5.2) and Equation (5.3) $(\mathbf{m}', \mathbf{d}')$ is a function of variables which are each relative to a single strongly connected subnet. Lemma 5.3.1 is derived from Theorem 5.2.1, by using this property, and taking into account that any arrival curve $\alpha(t)$ is a nondecreasing function of t. □

5.4 Checking stability of a generic FIFO network in polynomial time

In this section we apply on the operator Ψ the method presented in Section 3.4. As a result:

- We derive a new set of sufficient conditions for stability;
- we describe a polynomial time algorithm that allows to decide whether a network, with leaky bucket constrained flows and with FIFO aggregate scheduling nodes, satisfies the new set of sufficient conditions for stability; and
- we describe a polynomial time algorithm that allows to compute delay bounds.

5.4.1 The new sufficient conditions for stability

The new sufficient conditions for stability consist in imposing that, for each strongly connected component in the network, the spectral radius of at least one among three different matrices be smaller than one.

Theorem 5.4.1 *For any strongly connected component e, let us consider the following square nonnegative matrices:*

- *V^e_1, of dimension F^e, whose elements are:*

$$\forall f,\ f' \in \mathcal{F}^e,\ (V^e_1)_{f,f'} = \rho_f \sum_{\mathbf{n} \in \mathcal{G}(f,f',\mathbf{path}(f,e))} S(\mathbf{n}) \qquad (5.11)$$

- *V_2^e, of dimension N^e, whose elements are:*

$$\forall n,\ n' \in \mathcal{N}^e \quad (V_2^e)_{n,n'} = \begin{cases} \sum_{f \in \mathcal{M}(n',n)} \frac{\rho_f}{r_n} & n \neq n' \\ 0 & n = n' \end{cases} \tag{5.12}$$

where $\mathcal{M}(n', n)$ is the set of flows traversing node n' and n (in the order).

- *V_3^e, of dimension $\leq N^e F^e$, whose elements are, $\forall f \in \mathcal{F}^e,\ \forall n \in I(path(f, e))$,*

$$(V_3^e)_{(f,n),(f',n')} = \begin{cases} 1 & f' = f, \quad n' = prec(n, f) \\ \frac{\rho_f}{r_n} & f' \neq f, \quad n' = prec(n, f) \\ 0 & else \end{cases} \tag{5.13}$$

If for all e the spectral radius of at least one of the three matrices is inferior to one, then the network is stable.

The proof of this result is in Section 5.7.2.

We note that neither the burstiness of fresh flows nor the amount of buffer content at nodes at $t = 0$ are involved in the sufficient condition for stability in Theorem 5.4.1.

Based on Theorem 5.4.1, we outline here an algorithm that can be used to test the newly derived sufficient conditions.

Algorithm 1 Stability in FIFO networks with leaky bucket constrained flows

INPUT:

- The graph of the network;
- for any node n: r_n;
- for any flow f, σ_f^0 and ρ_f; and
- for any flow f, $\forall e$, $\mathbf{path}(f, e)$.

```
compute the strongly connected components of the network;
for any strongly connected component e do
    for any flow f in the e-th strongly connected component do
        for any flow f' in the e-th strongly connected component do
            compute G(f, f', path(f, e))
```

```
            compute
                $(V_1^e)_{f,f'} = \rho_f \sum_{\mathbf{n} \in \mathcal{G}(f,f',\mathbf{path}(f,e))} S(\mathbf{n})$
        end for
    end for
    for any node $n$ in the $e$-th strongly connected component do
        for any node $n$' in the $e$-th strongly connected component do
     compute $\mathcal{M}(n', n)$
     compute
            $(V_2^e)_{n,n'} = \sum_{f \in \mathcal{M}(n',n)} \frac{\rho_f}{r_n}$
        end for
    end for
    for any flow $f$ in the $e$-th strongly connected component do
        for any flow $f'$ in the $e$-th strongly connected component do
            for any node $n$ in the $e$-th strongly connected component do
                for any node $n$' in the $e$-th strongly connected component do
                    if $n, n' \in I(path(f,e))$ and $n' = prec(n,f)$ and $f' \in \mathcal{N}^n$ then
                        if $f' = f$ then
                            $(V_3^e)_{(f,n),(f',n')} = 1$
                        else
                            $(V_3^e)_{(f,n),(f',n')} = \frac{\rho_f}{r_n}$
                        end if
                    else
                        $(V_3^e)_{(f,n),(f',n')} = 0$
                    end if
                end for
            end for
        end for
    end for
    compute spectral radius of $V_1^e$, $V_2^e$ and $V_2^e$
    if all of the three spectral radii are $\geq 1$ then
     return network may be unstable
    end if
end for
     return network is stable
```

Let us analyze the worst-case time complexity of the various parts that compose Algorithm 1:

- Computation of the strongly connected components: $O(FN + N^2)$.
- computation of matrices V_1^e, V_2^e and V_2^e, for any strongly connected component e: $O(N(F^2N^2 + FN^2))$; and

- computation of the spectral radius (with a relative error bound of 2^{-b}[36]): $O(N[N^3 + \log(b) N \log^2 N + F^3 + \log(b) F \log^2 F])$.

We have therefore a polynomial time complexity.

5.4.2 Comparison to the "Charny-Le Boudec bound"

We observe how the "Charny-Le Boudec bound" [12] can be derived from Theorem 5.4.1:

Theorem 5.4.2 *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 5.4.1 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. □

5.5 An algorithm for delay bound computation

We observe that when a network is stable according to Algorithm 1, an extension of the same algorithm can be used to derive delay bounds at all nodes. We outline here the algorithm for delay bound computation, whose complexity is polynomial. The algorithm assumes that Algorithm 1 has been run, so that the strongly connected components of the network have been derived, and that for each strongly connected component e, matrices V_1^e and V_2^e have been derived.

Algorithm 2 Delay bounds in FIFO networks with leaky bucket constrained flows

INPUT:

- the graph of the network, subdivided in E strongly connected components;
- for any node n: r_n, T_n, Δ_n;
- for any flow f, $\forall e$: $\mathbf{path}(f, e)$, σ_f^0 and ρ_f, and for any node n, a_f^n; and
- for any strongly connected component e: matrices V_1^e, V_2^e and V_3^e.

for $e = 1 : E$ **do**
  for any flow f in the e-th strongly connected component **do**
    compute σ_f^e (the flow burstiness at the input to the e-th strongly connected component)
  end for

if spectral radius of V_1^e is < 1 **then**
for any flow f in the e-th strongly connected component **do**
compute

$$b_f^e = \sigma_f^e + \sum_{n \in \mathcal{I}(\mathbf{path}(f,e))} a_f^n + \rho_f \sum_{n \in \mathcal{I}(\mathbf{path}(f,e))} \left\{ \frac{\max_{f' \in \mathcal{Q}_f^n} L_{f'} + \sum_{f' \in \mathcal{N}^n} a_{f'}^n}{r_n} + 2T_n + \Delta_n \right\}$$

end for
compute $\mathbf{m}^* = (I - V_1^e)^{-1} \mathbf{b}_f$
for any node n in the e-th strongly connected component **do**
compute

$$d_1^n = \min_{f \in \mathcal{N}^n} \left\{ \sum_{f' \in \mathcal{N}^n \setminus \mathcal{Q}_f^n} \frac{m_{f'}^*}{r_n} + \frac{\max_{f' \in \mathcal{N}^n} L_{f'}}{r_n} + \sum_{f' \in \mathcal{Q}_f^n} m_{f'}^* \left(\frac{1}{r_n} - \frac{1}{r_{prec(n,f)}} \right)^+ \right\}$$

end for
end if
if spectral radius of V_2^e is < 1 **then**
for any node n in the e-th strongly connected component **do**
compute $c_n = \sum_{f \in \mathcal{N}^n} \frac{\sigma_f^e + a_f^n}{r_n} + T_n + \Delta_n$
end for
compute $\mathbf{d}_2 = (I - V_2^e)^{-1} \mathbf{c}$
end if
if spectral radius of V_3^e is < 1 **then**
for any flow f in the e-th strongly connected component **do**
for any node n in the e-th strongly connected component **do**
if $n \in I(path(f, e))$ **then**
if n is the first node of the path of flow f in the e-th strongly connected subnet **then**
compute $g_f^n = \sigma_f^e + \rho_f \left(T_n + \frac{\sum_{f' \in \mathcal{N}^n} a_{f'}^n + L_f}{r_n} \right)$
else

$$g_f^n = \rho_f \left(T_n + \frac{\sum_{f' \in \mathcal{N}^n} a_{f'}^n + L_f}{r_n} \right)$$

end if
else

$$g_f^n = 0$$

end if
end for
end for
compute $\boldsymbol{\sigma}^* = (I - V_3^e)^{-1} \mathbf{g}$
end if
for any node n in the e-th strongly connected component **do**

compute

$$d^n = \min\left\{ d_n^2, d_n^1 + \sum_{f \in \mathcal{N}_n} \frac{a_f^n}{r_n} + T_n + \Delta_n, \sum_{f' \in \mathcal{N}_n} \frac{\left(\sigma_{f'}^{*prec(n,f')} + a_{f'}^n\right)}{r_n} + T_n \right\}$$

end for

end for

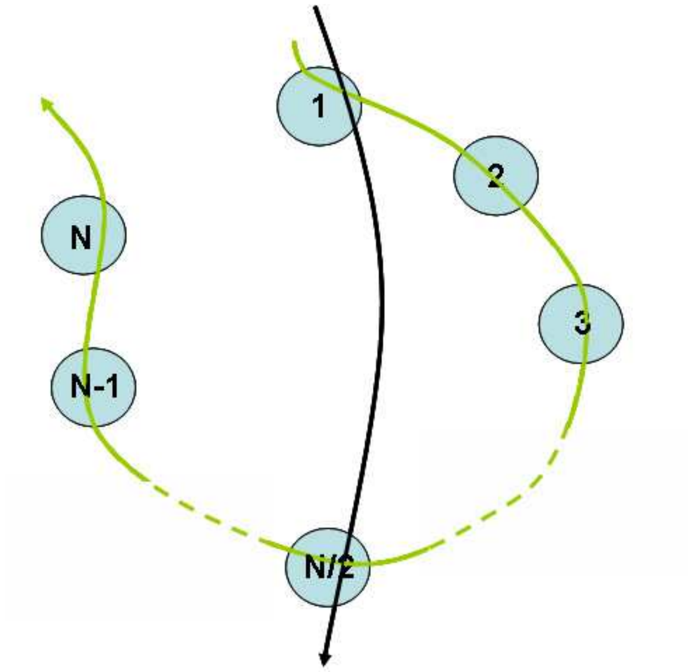

Figure 5.1 – An example of the symmetric network on which the algorithms in Section 5.4 have been run. The nodes are disposed on a ring; at each node i $i = 1 \div N$ there is a "type a" fresh flow, with rate ρ_a, that traverses clockwise all nodes on the ring, and exits the network at node $(i + N - 1)_{mod(N)}$, and a "type b" flow, that that traverses nodes i and $(i + N/2)_{mod(N)}$, and then exits the network. Only the paths of fresh flows at node 1 are shown in the figure. We assume all service rates are equal to 1, all flow burstiness equal to 1, and all buffers in the network are empty at $t = 0$.

5.6 The largest inner bound to the stability region

To assess and analyze the two algorithms derived in the previous section, we applied them over some networks examples. In order to characterize the stability properties of a network, we make use of the concept of *stability region*:

Definition 5.6.1 (Stability Region) *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 stable.*

In Fig. 5.1 we have an example of network on which we evaluated the two algorithms in Section 5.4. The network is composed of N nodes (with N even) in a ring structure, labeled from 1 to N. For simplicity, we assume all flows are of two types, "a" (or "long-haul" flows, traversing all the nodes in the network) and "b" (that represent "local" traffic, traversing only two nodes), and that all flows of the same type

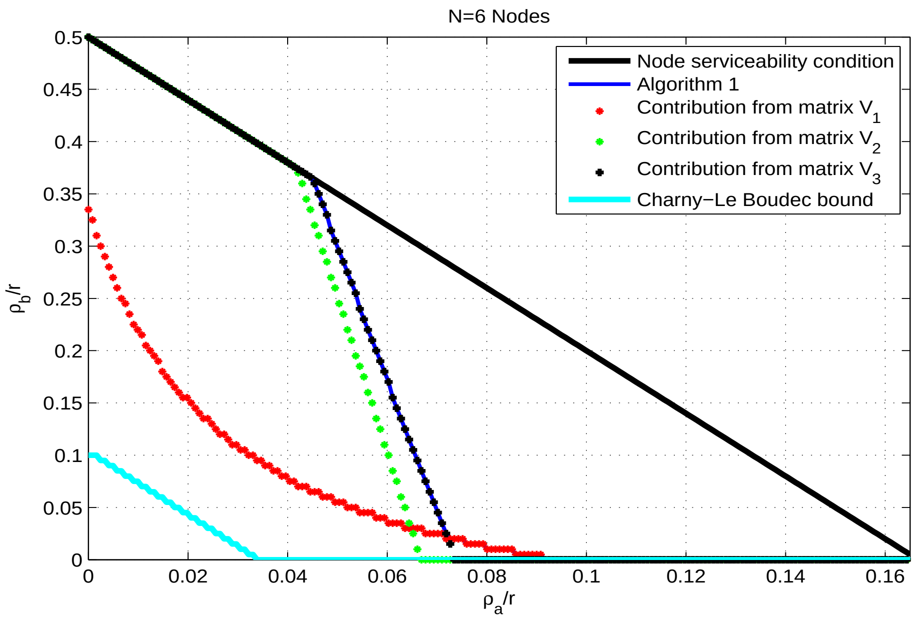

Figure 5.2 – Inner and outer bounds to the stability region of the network in Fig. 5.1, with $N = 6$ nodes. The black line limits the region in which the node serviceability condition is satisfied, and it represents therefore an outer bound to the stability region of the considered network. The other curves represent inner bounds to the stability region of the network: the blue line represents the bound obtained by Algorithm 1; in red, green and black stars we have the bounds derived by imposing that the spectral radius of the matrices V_1, V_2 and V_3, respectively, be smaller than one; finally, the light blue line represents the inner bound derived through the “Charny-Le Boudec bound” [12].

have the same rate (respectively ρ_a and ρ_b) and the same burstiness $\sigma_a = \sigma_b$. At any node $i,\ i = 1 \div N$, a fresh flow of ”type a” enters the network, traverses clockwise all nodes on the ring, and exits the network at node $(i + n - 1)_{mod(N)}$. At any node i we also have a fresh flow of ”type b”, that traverses nodes i and $(i + N/2)_{mod(N)}$ and then exits the network.

By using Algorithm 1, we derived an inner bound to the stability region of the network in Fig. 5.1. In this example, the fact of forcing the rates of all flows in the network to take only two different values (one for each ”type”) has been done in order to make the stability region bidimensional, and to give visual evidence of the performance of the new sufficient conditions, in function of some properties of the network topology and of flow paths. The network is composed by a single strongly connected component, therefore Algorithm 1 evaluates three matrices: V_1 associated to variables in $\mathcal{M}$, V_2 associated to variables in $\mathcal{D}$, and V_3 associated to variables in $\mathcal{S}$.

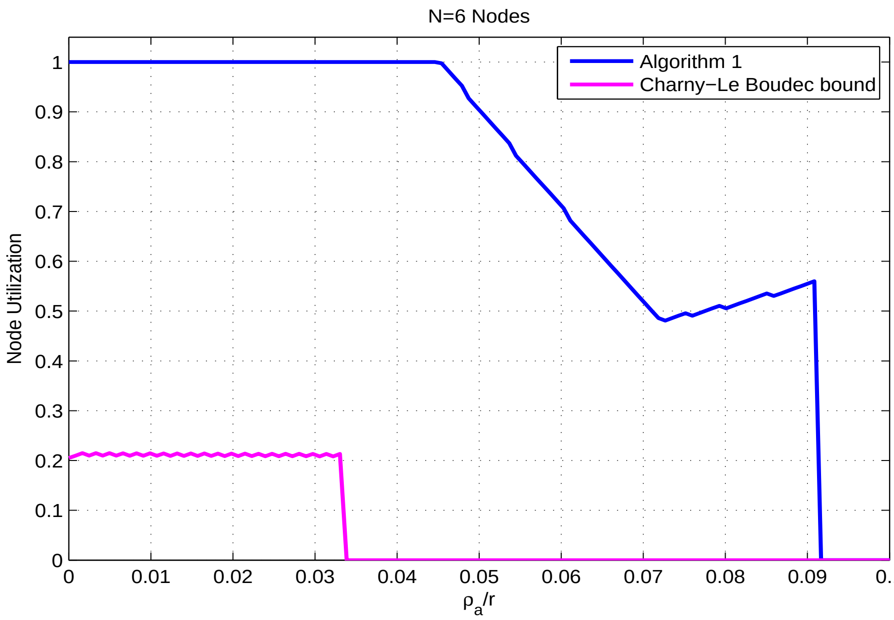

Figure 5.3 – Maximum node utilization for the network in Fig. 5.1, in function of ρ_a, when flow rates satisfy the "Charny-Le Boudec bound" (pink line), and the sufficient conditions tested by Algorithm 1 (blue line), with $N = 6$ nodes.

In Fig. 5.2, relative to a network with $N = 6$, the black line limits the region in which the node serviceability condition is satisfied, and it represents therefore an outer bound to the stability region of the considered network. The other curves represent inner bounds to the stability region of the network: the blue line represents the bound obtained with Algorithm 1 in Section 5.4; in red, green and black stars we have the bounds derived by imposing that the spectral radius of the matrices V_1, V_2 and V_3, respectively, be smaller than one; finally, the light blue line represents the inner bound derived through the the "Charny-Le Boudec bound" [12].

We can observe how with Algorithm 1 we have an inner bound that is much larger than the one derived by applying the "Charny-Le Boudec bound" : with $N = 6$, the inner bound obtained with Algorithm 1 has an area which is 58.8% of the area of the outer bound given by node serviceability condition, while with the "Charny-Le Boudec bound" this ratio is of 4.29%.
We also note that the inner bound derived from the "Charny-Le Boudec bound" is always contained in the one relative to the new sufficient conditions for stability: In Section 5.4.2 we show how the "Charny-Le Boudec bound" can be derived from the new sufficient conditions.

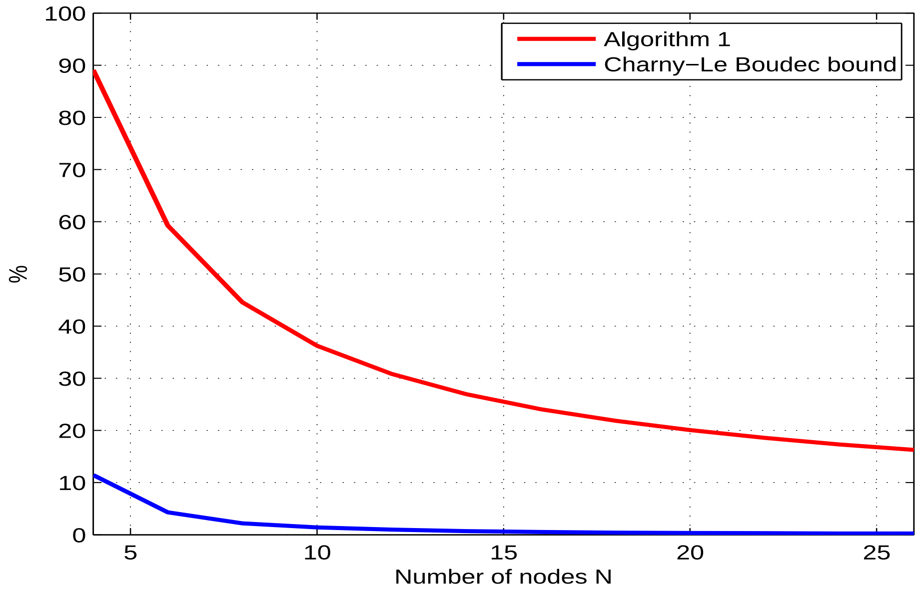

Figure 5.4 – The area of the inner bound to the stability region given by Algorithm 1 (blue line) and by the "Charny-Le Boudec bound" (pink line), in function of N, in % of the outer bound area given by the node serviceability condition. The network is the one in Fig. 5.1.

In Fig. 5.2 we see how for low values of long-haul flow rates the network is stable for any value of "local" flow rates that satisfies the node serviceability condition. In Fig. 5.3 we have the maximum node utilization that we can have for a given value of ρ_a, when the sufficient stability conditions are satisfied, for Algorithm 1 and for the Charny-Le Boudec result, with $N = 6$ nodes. We can see that Algorithm 1 allows for a wider range of values for long-haul flows rates, and for all those values the maximum node utilization obtainable with a given value of ρ_a is far larger than the one obtainable with the "Charny-Le Boudec bound", and in any case larger that 0.45.
In Fig. 5.3 we can also see that the trend of the blue line is close to expectations: for very low values of ρ_a the maximum network utilization is one (indeed, for $\rho_a = 0$ the maximum number of hops in the network becomes 2, and we know from the "Charny-Le Boudec bound" that those networks are always stable). For high values of ρ_a and low values of ρ_b, the network configuration gets closer to a ring network, that we know is always stable. Therefore, we expect for those values of flow rates to have a maximum node utilization that gets close to one. Although in Fig. 5.3 the maximum node utilization keeps growing with increasing values of ρ_a, up to $\rho_a/r = 0.092$, we do not observe the maximum achievable node utilization getting close to one. This shows

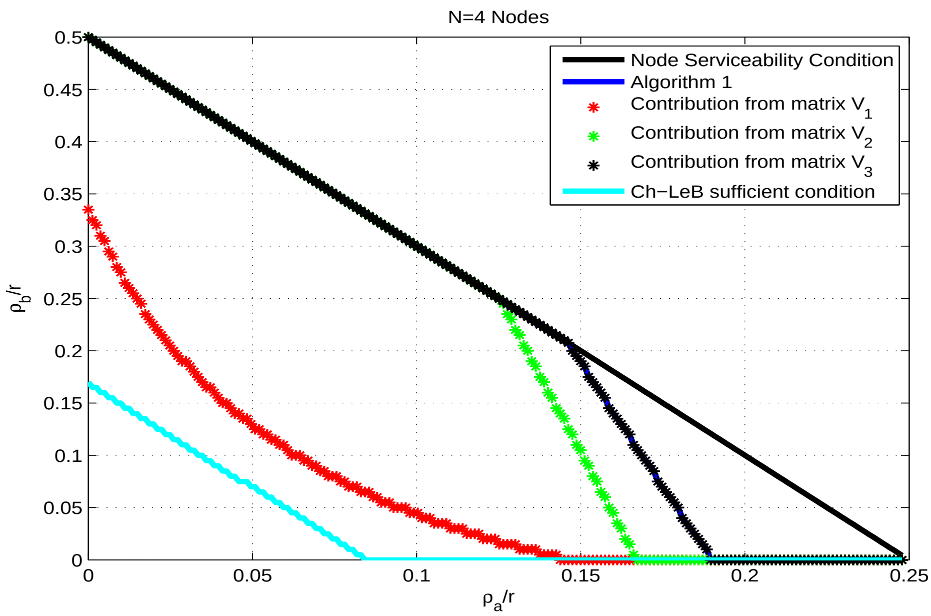

Figure 5.5 – Inner and outer bounds to the stability region of the network in Fig. 5.1, with $N = 4$ nodes.

that the sufficient stability conditions tested by Algorithm 1 do not revert to the ”ring result” ([35]).

An important parameter influencing the performance of the new sufficient conditions for stability, in terms of relative area of the inner bound to the stability region, is the maximum number of hops of a flow in the network. Therefore, we compared the ratio between the area of the inner bounds given by Algorithm 1 and by the “Charny-Le Boudec bound”, and the area of the node serviceability region, taking it as a measure of the ”efficiency” of the algorithm, that is of its ability in guaranteeing stability for the widest range of values of flow rates. It can also be seen as a measure of the tightness of the sufficient conditions tested by Algorithm 1, as the larger the relative area, the smaller the residual area (and therefore the range of values for flow rates) for which an unstable behavior could possibly be present.

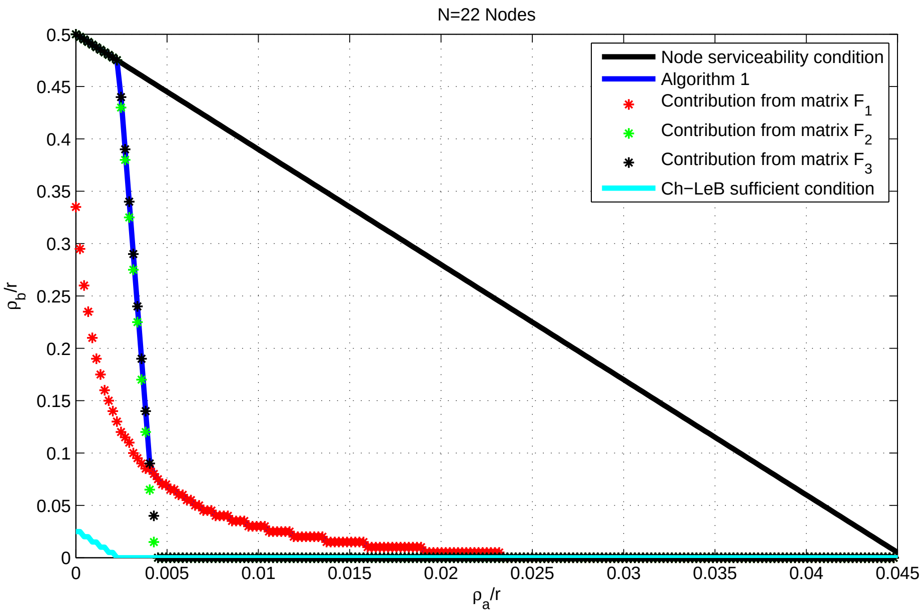

Figure 5.6 – Inner and outer bounds to the stability region of the network in Fig. 5.1, with $N = 22$ nodes.

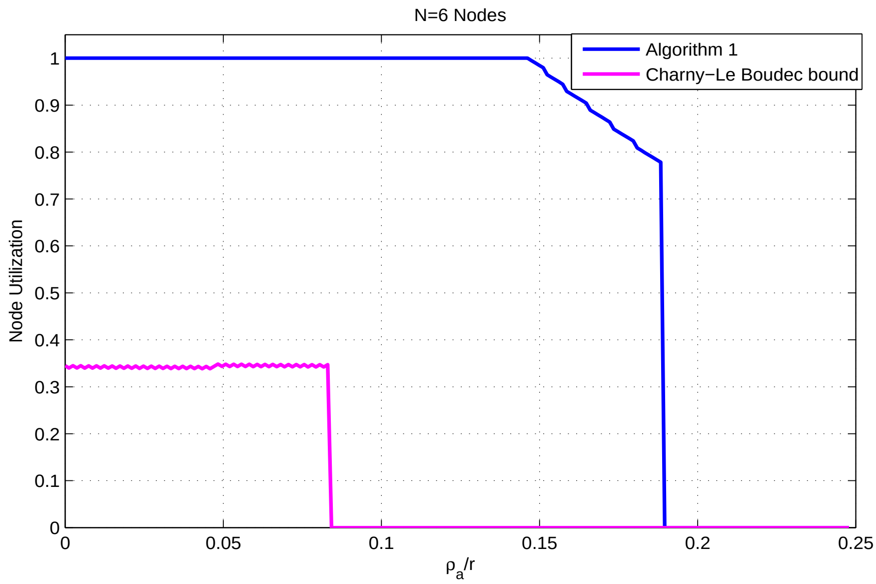

Figure 5.7 – Maximum node utilization for the network in Fig. 5.1, in function of ρ_a, when flow rates satisfy the sufficient condition for stability of the the "Charny-Le Boudec bound" (pink line), and the sufficient conditions tested by Algorithm 1 (blue line), with $N = 4$ nodes.

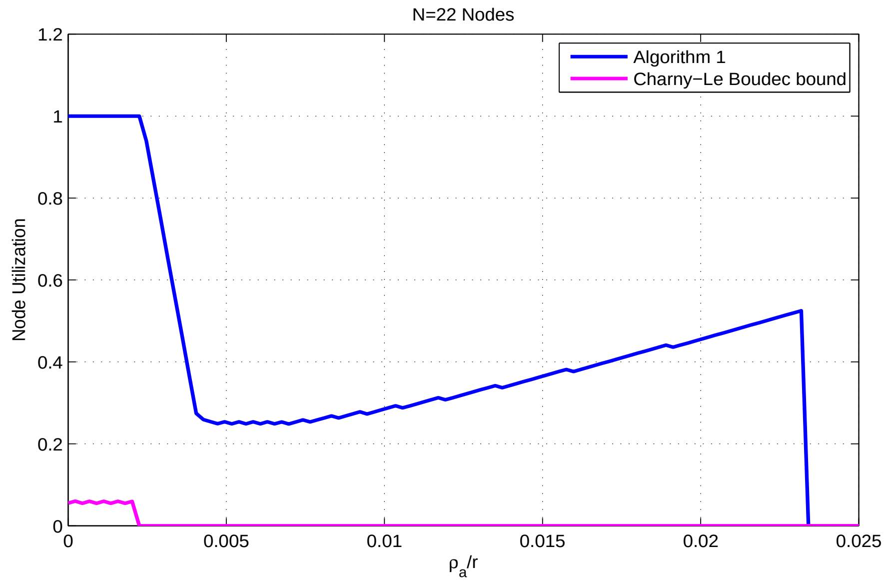

Figure 5.8 – Maximum node utilization for the network in Fig. 5.1, in function of ρ_a, when flow rates satisfy the sufficient condition for stability of the the "Charny-Le Boudec bound" (pink line), and the sufficient conditions tested by Algorithm 1 (blue line), with $N = 22$ nodes.

In Fig. 5.4 we draw the relative areas for both Algorithm 1 and the "Charny-Le Boudec bound", in function of N, which for the considered network represents both the total number of nodes and the maximum hop count in the network. We can see how the relative area of the inner bounds obtained with both methods decreases with increasing N: however, in all cases the area relative to Algorithm 1 is at least an order of magnitude larger than the one from the "Charny-Le Boudec bound".

The behavior of the relative area of the inner bound with increasing values of N can be explained by considering that both the "Charny-Le Boudec bound" and our approach is ultimately based on worst case analysis of the interaction between flows at aggregate scheduling nodes, and of the effects of these interactions on flow burstiness. For each flow, the difference between average burstiness and worst case burstiness amplifies and sums up as the flow traverses the network. As a result, in networks with increasing average number of hops the distance between average behavior and worst case behavior gets more pronounced, and the bounds obtained with this worst case analysis are consequently less tight.

As we can observe from the structure of Algorithm 1, the sufficient conditions is tested, for every strongly connected component, on three different matrices, each as-

sociated to one of the variable classes described in Section 5.1. In Fig. 5.5 and 5.6 we draw the inner bounds to the stability region, for $N = 4$ and 22, respectively. The line with green markers represents the contribution of the matrix V_2 to the inner bound of Algorithm 1, the one with red markers is relative to the contribution of the matrix V_1, and the one with black markers to V_3. We can see how for high values of maximum hop count in the network, the contribution of matrix V_1 (relative to variables associated to super chains) is more important, and accounts for the highest values of long-haul flow rates for which the network is known to be stable. We also note how for the considered network, the contribution from V_3 brings always to a larger inner bound than the one from V_2, but they tend to coincide for increasing values of N.
The use of upper bounding linear operators, introduced through some form of approximation, instead of the direct application of the operator Π described in Theorem 5.2.1 comes naturally at some price. In order to evaluate this, for each point in the region in which the node serviceability condition is satisfied, 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, whose border is indicated in Fig. 5.9 with a black dashed line, gives an idea of what we could gain if we were able to derive $\Pi^*(\mathbf{x}_0)$ (the subadditive closure of the operator Π in $\mathbf{x}_0$). The lower bound to the stability region obtained in this way gives a better idea of what is the region of the graph (and therefore the range of values for flow rates) for which instability could potentially arise. In particular, we can see that the direct use of the operator Π results in an increase of the inner bound to the stability region, and mainly of the maximum values of the long-haul flows rates ρ_a for which the network is stable.

On the same network of Fig. 5.1, with $N = 6$, we also applied Algorithm 2, and we derived the values of the upper bound to node delay, in function of the rates ρ_a and ρ_b. In Fig. 5.10 we have a graph of the value of the delay bounds obtained, where the color is associated to the logarithm of the delay bound value, through the color scale on the side of the graph. We can observe that for values of flow rates which are close to the border of the inner bound to the stability region the delay bound values increase, but this does not happen when this border coincides with the border of the outer bound given by the node serviceability condition.

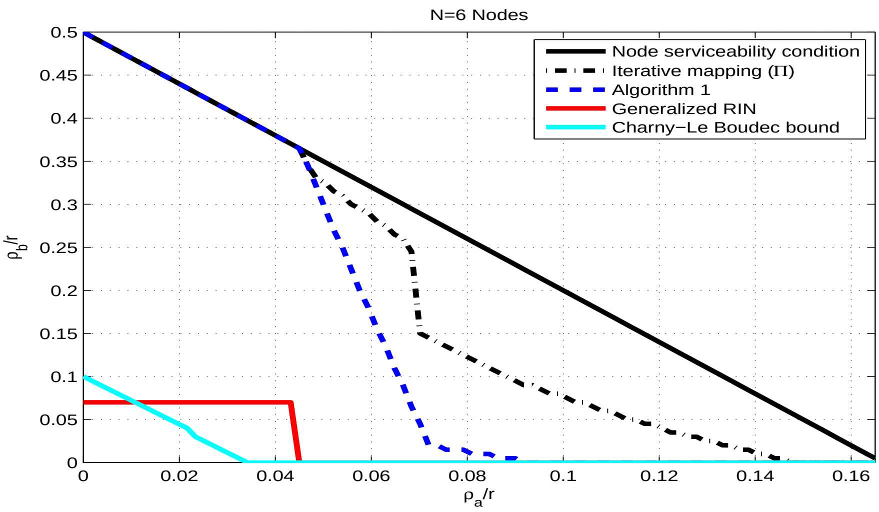

Figure 5.9 – Comparison between the inner and outer bounds to the stability region in Fig. 5.2, and the inner bound (black dashed line) obtained by computing the quantity $\sup_{0\leq l\leq n}\Pi^{(l)}(\mathbf{x}_0)$ for increasing values of n. The region represents those values of flow rates for which the increase in the value of this quantity between two consecutive values of n is inferior to 1%, with $n \leq 100$.

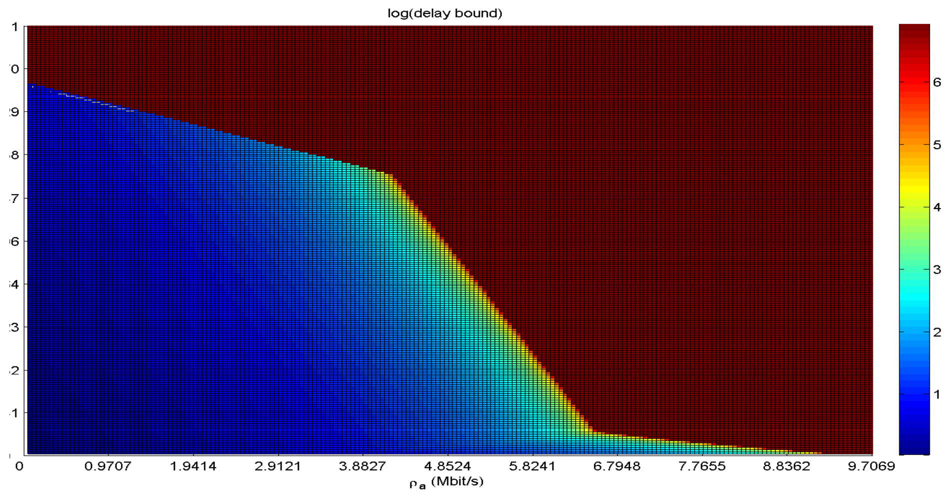

Figure 5.10 – Bounds to node delay for the network in Fig. 5.1, on logarithmic scale, in function of the values of flow rates, inside the inner bound to the stability region obtained with Algorithm 1. Delay bounds are obtained with Algorithm 2, with all service rates equal to 100 Mbit/s, all flow burstiness equal to 1 Mbit, node latency equal to 0.3 ms, propagation delay equal to 0.2 ms, and all buffers in the network empty at $t = 0$. In dark red we have the region in which no delay bound is available.

5.7 Proofs

5.7.1 Proof of Theorem 5.2.1

Proof: We proceed by induction on the index p of relevant network events.
Base case: p=1. In order to upper bound the quantities $m_f^{\mathbf{n}}[1]$, for any flow f, the derivation is exactly as in the proof of Theorem 4.4.1.

For maximum 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 $m_f^{\mathbf{n}'}[1]$ 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.
For any flow f, at the input to any node n along its path, we now derive an upper bound to $\sigma_f^n[1]$. For $p = 1$, only two cases are worth considering: Both are relative to the case in which both the first and the second packet served in the network are served at the same node, n.

- The first and the second packet served both belong to flow f. In this case, we have that the maximum flow burstiness at the output of node n is upper bounded by $\sigma_f^0 + \rho_f \left(\frac{L_f}{r_n} + T_n\right)$.
- Only the second packet served at the node belongs to flow f. Then the bound found in the previous case holds also in this case.

Induction step. For any flow f, at the input to any node along its path, the value of the variable $\sigma_f^n[p]$ would be the same in the case in which, for $t > t_{p-1}$,

- no more packets are injected into the network by any source; and
- no more packet arrive at any node.

Therefore, we compute a bound to $\sigma_f^n[p]$ in this case. This brings to using the output bound results, as the one for FIFO aggregate scheduling nodes in [26, Corollary 6.2.3], where the burstiness of the input flows are given by the values of the variables $\sigma_{f'}^{prec(n,f)}[p-1]$, $\forall f' \in \mathcal{N}^{prec(n,f)}$. To these quantities we must add the buffer content at the considered node at $t = 0$.
$\forall n \in \mathcal{N}$, the upper bound to $d^n[p]$ derives from Lemma 4.8.1, 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. Here too, the use of the variables $\sigma_f^n[p-1]$,

$\forall n \in \mathcal{N}, \forall f \in \mathcal{F}$, relative to time $p-1$ is justified by the same consideration made in the derivation of the bound to these variables.
For any flow f, $\forall e$, for any $\mathbf{n} \in \mathbf{path}(f, e)$, we derive now an upper bound to the variable $m_f^{\mathbf{n}}[p]$. 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. Let's indicate the sequence of nodes in the path of flow f as $n'_k,\ k = 1, ..., K$. We indicate with $\Delta t_{int,k}$ the delay experienced at node n'_k by the open packet at that node (for those nodes which are not in the sequence $\mathbf{n}$ we consider the open packet to coincide with the close packet). We denote with Θ^k and Γ^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 4.8.1 we have for $k \geq 1$

$$\Delta t_{int,k} \leq \frac{\Theta^k}{r_{n'_k}} + \Gamma^k \left(\frac{1}{r_{n'_k}} - \frac{1}{r_{prec(n'_k,f)}} \right)^+ +$$

$$+\frac{A'_k + \max_{f' \in Q_f^{n'_k}} L_{f'}}{r_{n'_k}} + T_{n'_k} + \Delta_{n'_k} = B_k + T_{n'_k} + \Delta_{n'_k} \qquad (5.14)$$

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

$$\Delta t_{int,k} + \Delta t_k \leq B_k + 2T_{n'_k} + \Delta_{n'_k} + \frac{\gamma^k + \theta^k + A_k}{r_{n'_k}} \qquad (5.15)$$

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 Equation (5.15) over all the nodes in the path of flow f. At each node n'_k, each of the terms $\gamma^k,\ \theta^k,\ \Gamma^k,\ \Theta^k,\ A_k,\ A'_k$ 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 $\Theta^k = \sum_{f' \in \mathcal{N}^{n'_k}} \Theta^k_{f'}$. For any flow f' (not necessarily distinct from f), $\forall \mathbf{n}' = (n'_l, ..., n'_{l+K'}) \in \mathcal{G}(f, f', \mathbf{n})$, the contribution to the upper bound to the super chain duration is

$$\frac{\Theta^l_{f'} + \theta^l_{f'}}{r_{n'_l}} + \sum_{k=l}^{l+K'} \left[\Gamma^k_{f'} \left(\frac{1}{r_{n'_k}} - \frac{1}{r_{n'_{k-1}}} \right)^+ + \frac{\gamma^k_{f'} + a^{n'_k}_{f'}}{r_{n'_k}} \right] \tag{5.16}$$

By definition of the variables $m^{\mathbf{n}'}_f[p]$, we have that

$$((...(\Theta^l_{f'} + \theta^l_{f'} + \Gamma^l_{f'} + \gamma^l_{f'} +) \vee \Gamma^{l+1}_{f'} + \gamma^{l+1}_{f'}) \vee \Gamma^{l+2}_{f'} + \gamma^{l+2}_{f'}) \vee ... \vee \Gamma^{l+K'}_{f'} + \gamma^{l+K'}_{f'} \leq m^{\mathbf{n}'}_{f'}[p-1]$$

Therefore, the quantity in Equation (5.16) is upper bounded by

$$m^{\mathbf{n}'}_{f'}[p-1] S(\mathbf{n}') + \sum_{n' \in \mathbf{n}'} \frac{a^{n'}_{f'}}{r_{n'}} \tag{5.17}$$

For any node $n'_k \in \mathcal{U}^{n_1}_f$, $\Delta t_k = 0$, 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 $\mathbf{n}' = (n'_k, ..., n_1, ...)$. For the contributions relative to all the other flows, we note that

$$\Gamma^k_{f'} + \gamma^k_{f'} + \Theta^k_{f'} + \theta^k_{f'} \leq m^{n'_k}_{f'}[p-1]$$

We get in this way the upper bound in Equation (5.5). 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:

$$\sum_{n \in \mathcal{U}^{n_1}_f} (b^n[p-1] \wedge d^n[p-1]) + \sum_{f'} \sum_{\mathbf{n}' \in \mathcal{G}(f, f', \mathbf{n})} m^{\mathbf{n}'}_{f'}[p-1] S(\mathbf{n}') +$$

$$+ \sum_{n \in \mathcal{I}(\mathbf{n})} \left(\frac{\max_{f' \in \mathcal{Q}^n_f} L_{f'} + \sum_{f' \in \mathcal{N}^n} a^n_{f'}}{r_n} + 2T_n + \Delta_n \right)$$

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 Equation (5.2). □

5.7.2 Proof of Theorem 5.4.1

Proof: In order to prove Theorem 5.4.1, we use Theorem 3.4.2. Let's consider first the case in which the network is composed by a single strongly connected component. From Lemma 5.3.1, we note that for any flow the only variable associated to super chains that comes into play is the one 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 those inequalities in Lemma 5.3.1 relative to $\mathbf{m}^{path}$ in the form

$$\begin{array}{ll} \mathbf{m}^{path}[p] \leq \Psi_1(\mathbf{m}^{path}[p-1]) & \forall p \geq 1 \\ \mathbf{m}^{path}[0] \leq \mathbf{m}'_0 & \end{array} \tag{5.18}$$

where $\mathbf{m}'_0$ is such that $\forall f$, $m'_{0,f} = L_f$.

The set of inequalities in Lemma 5.3.1 relative to $\mathbf{d}$ takes the form

$$\begin{array}{ll} \mathbf{d}[p] \leq \Psi_3(\mathbf{m}^{path}[p-1]) \wedge \Psi_2(\mathbf{d}[p-1]) & \forall p \geq 1 \\ \mathbf{d}[0] \leq \mathbf{d}_0 & \end{array} \tag{5.19}$$

and the ones relative to $\boldsymbol{\sigma}$ take the form

$$\begin{array}{ll} \boldsymbol{\sigma}[p] \leq \Psi_4(\boldsymbol{\sigma}[p-1]) \wedge \Psi_5(\mathbf{d}[p-1]) & \forall p \geq 1 \\ \boldsymbol{\sigma}[0] \leq \boldsymbol{\sigma}_0 & \end{array} \tag{5.20}$$

As flows are leaky bucket constrained, all operators Ψ_i, $i = 1$ to 5 are linear.
We observe that, due to the expression of the operator Ψ defined in Lemma 5.3.1, any solution of the fixed point $(\mathbf{m}, \mathbf{d}, \boldsymbol{\sigma}) = \Psi(\mathbf{m}, \mathbf{d}, \boldsymbol{\sigma})$ is larger than $(\mathbf{m}_0, \mathbf{d}_0, \boldsymbol{\sigma}_0)$. Therefore we can apply Theorem 3.4.3. We observe now that in networks with leaky bucket constrained flows, if we have delay bounds at each node, we can derive backlog bounds at each node. Therefore in these networks, if $(\mathbf{m}^*, \mathbf{d}^*, \boldsymbol{\sigma}^*)$ is a solution of the fixed point $(\mathbf{m}, \mathbf{d}, \boldsymbol{\sigma}) = \Psi(\mathbf{m}, \mathbf{d}, \boldsymbol{\sigma})$, by the same definition of stability, a sufficient condition for stability, is that one among the arrays $\mathbf{d}^*$, $\mathbf{m}^{path,*}$ and $\boldsymbol{\sigma}$ be finite (by the structure of the operator Ψ, if $\mathbf{m}^{path,*}$ is finite then $\mathbf{m}^*$ is finite too). This implies that if at least one of the fixed point problems $\mathbf{m}^{path} = \Psi_1(\mathbf{m}^{path})$, $\mathbf{d} = \Psi_2(\mathbf{d})$ and $\boldsymbol{\sigma} = \Psi_4(\boldsymbol{\sigma})$ admits a finite solution, then the network is stable. As these operators are linear, each of the fixed points admits a finite solution if the spectral radius of the associated matrix is inferior to one.
We denote with F_1, F_2 and F_4 the matrix associated to Ψ_1, Ψ_2 and Ψ_4, respectively (or to the array of variables $\mathbf{m}^{path}$, $\mathbf{d}$ and $\boldsymbol{\sigma}$, respectively).

When we have more than one strongly connected component, we derive an operator Ψ' by upper bounding Ψ in Equation (5.10) with the following strategy: For each strongly connected subnet, we keep as independent variables those whose associated matrix has a spectral radius smaller than one, and we make dependent variables all the others. A set of variables is made independent when the corresponding upper bounds in Equation (5.10) are function only of the variables in the same set. A set of variables is made dependent when those upper bounds are made dependent only from the variables in the other sets. As we can see, nonlinearity in the operators Ψ_i derives from a $min()$ operation among linear combinations of variables. When we make each set of variables dependent from only one set of variables, we do it by choosing in the bounds in Ψ only one of the terms of the $min()$ operation. We build in this way an upper bounding operator Ψ' which is linear, and the matrix associated to it can be put in block triangular form, with each block corresponding to a strongly connected component. From the structure of the upper bound, we have that the matrices in those blocks are the ones whose spectral radius is smaller than one by hypothesis. □

5.8 Summary

In this chapter we presented a more complex application of the method described in Chapter 3, and we derived a set of sufficient conditions for the stability of an heterogeneous network of FIFO nodes, with leaky bucket constrained flows. We showed how the obtained conditions perform largely better than any of the existing results.

Chapter 6

”Pay Bursts Only Once” and Non-FIFO Guaranteed Rate Nodes

The focus of the present chapter is on networks of Guaranteed Rate nodes and leaky bucket constrained flows. We consider the case in which the nodes are not FIFO per flow.
Therefore, in the present chapter,

- we show with the help of a counterexample that a known result on end-to-end delay bounds in a succession of Guaranteed Rate nodes [19] is *not* valid when nodes are *not* FIFO per flow;
- we derive bounds on end-to-end delay and to flow burstiness that are valid in the general, possibly non-FIFO case. We show that these bound are tight; and
- we derive a delay bound for the more realistic case in which the latency of a node can be split in a variable and in a fixed part.

The content of the present chapter is largely derived from [32].

6.1 Introduction

In the differentiated services framework [7], in networks without cycles end-to-end delay bounds may be obtained by assuming that sources satisfy leaky bucket [26] traffic specifications, and that routers can be modeled as *Guaranteed Rate* (GR) [19, 20] nodes. One of the main properties of a network of GR nodes is that a tight upper bound on end-to-end delay can be obtained, given the parameters of the leaky buckets at the source (burstiness and sustainable rate), and the parameters of the traversed GR nodes (delay and service rate) [24]. This end-to-end bound, derived in the original paper

[19], is recalled in Section 6.3.1; it has the remarkable property known as "pay bursts only once", i.e. when a bursty flow traverses a number of GR nodes in sequence, the effect of burstiness on the end-to-end delay bound is the same as if the flow traversed only one node. Another way to look at this property is that the end-to-end delay bound is much less than the sum of delay bounds at each node.

The GR node model may be used in the differentiated services framework as follows ([26]). End-user flows (called "microflows") are grouped into "aggregates" at the network edge; inside the network, each aggregate is handled as an individual flow, in other words, the "flow" that a GR node sees inside the network is in fact an aggregate. Now, in practice, although this procedure usually preserves the ordering of packets within each microflow (in order to preserve sequence at the TCP or RTP layer), packet reordering can take place inside an aggregate between packets belonging to different microflows. In routers with multistage fabrics, this reordering is due to the presence of multiple parallel paths between input and output ports [9, 6]. Thus the GR node model of a differentiated services router cannot always be assumed to be FIFO per flow. More generally, the GR class encompasses a great variety of algorithms, which are not necessarily FIFO per flow [8]. In what follows we use the term "FIFO" to indicate a GR node that is FIFO per flow (because the definition of GR node is relative to the treatment it gives to a flow viewed as a single entity).

6.2 Model and assumptions

We consider a flow that traverses a tandem of M nodes. As they traverse this succession of nodes, packets belonging to the flow experience a delay that accumulates along their path, and that can be different in principle at each node for each packet.

To the flow we associate the cumulative function $R(t)$, which counts the number of bits seen on the flow in the time interval $[0, t]$. The flow is leaky bucket constrained, with rate ρ and burstiness σ. We consider that nodes are *Guaranteed Rate* (Section 2.2.4) with service rate r (equal for all nodes) and delay e (in principle different for each node). Nodes are not necessarily FIFO. In Table 6.1 we have a list of the notation used in the present chapter. We assume that each node has also a maximum service curve with latency zero and with rate equal to the GR rate of the node. We assume also that all the nodes in the network satisfy the node serviceability condition (Section 2.2.2).

We assume that the arrival time of a packet at a node is the arrival time of the last bit of the packet, and the departure time of a packet from a node is the departure time of the last bit of the packet. This leads us to observe instantaneous packet arrivals and departures. In what follows we consider that each link between two nodes m and $m+1$ ($m \in [1, M]$) has a constant propagation delay $\tau^{m,m+1}$ (with $\tau^{M,M+1}$ we indicate the

Table 6.1 – Notation used in Chapter 6

$R(t)$	cumulative packet arrival function of the flow
σ	burstiness of the considered flow
ρ	rate of the considered flow
r	service rate for the flow
e^m	delay of the m-th GR node
M	Total number of nodes traversed by the flow
p^j	j-th packet of the flow
l^j	length of packet p^j
$l_{max}(l_{min})$	maximum (minimum) packet length for the flow
$GRC^m(p^j)$	GR clock value at node m for p^j (the j-th packet at the input of the m-th node)
$A^m(p^j)$	arrival time at node m of p^j (the j-th packet at the input of the m-th node)
$d^m(p^j)$	departure time from node m of p^j (the j-th packet at the input of the m-th node)
$\tau^{m,m+1}$	propagation delay between nodes m and $m+1$
α^m	$e^m + \tau^{m,m+1}$
$\beta_{r,e}$	service curve of the form $r(t-e)^+$
$f^m(t)$	arrival curve for the flow at the input to node m

propagation delay of the link between the M-th node and the destination).

6.3 The existing end-to-end delay bounds in GR nodes require FIFO assumption

6.3.1 The existing results

The main result about end-to-end delay bounds in a succession of (not necessarily FIFO) GR servers has been first derived in [19], and extended in [20]. In those papers a method is defined to derive an end-to-end delay bound, based on the following result:

Theorem 6.3.1 ([19]) *Consider a flow that traverses a succession of M nodes in a network. If the scheduling algorithm at each server $m \in (1, M)$ on the path of the flow belongs to GR for the given flow, with service rate r for the flow and delay e^m, then a bound to the end-to-end delay of the j-th packet of the flow, denoted with D^j, is given by*

$$D^j \leq GRC^1(p^j) - A^1(p^j) + (M-1) \max_{n \in 1,\dots,j} \frac{l^n}{r} + \sum_{m=1}^{M} \alpha^m \tag{6.1}$$

where $\alpha^m = e^m + \tau^{m,m+1}$ *and* $\tau^{m,m+1}$ *is the propagation delay between nodes* m *and* $m+1$.

The difference $GRC^1(p^j) - A^1(p^j)$ in Equation (6.1) depends on source traffic specification. For a leaky bucket constrained flow with burstiness σ and rate ρ, Equation (6.1) takes the form [19]

$$D^j \leq \frac{\sigma}{r} + (M-1) \max_{n \in 1,\dots,j} \frac{l^n}{r} + \sum_{m=1}^{M} \alpha^m \tag{6.2}$$

6.3.2 Counterexample

However, from the analysis of even simple examples of non-FIFO behavior in GR nodes, we can verify that in a sequence of non-FIFO GR nodes the end-to-end delay for a packet can actually be higher than the bounds in Equation (6.1) and Equation (6.2).

As an example we consider a sequence of two packets, belonging to a leaky bucket constrained flow with burstiness σ and rate ρ that traverses two GR non-FIFO nodes (see Fig. 6.1). To simplify the example, we took the propagation delay at all links equal to zero, $e = 0$ for both nodes, $\sigma = l$ bits, $\rho = r = 1$. We assume that packet 1 is of length l, and that packet 2 is of length $l - \varepsilon$. At the input to node 1, packet 1 arrives at time $t = 0$, while packet 2 arrives at time $l - \varepsilon$.

As there is no delay at the nodes, the maximum departure time for packet 1 is given by its Guaranteed Clock value at node 1, equal to l time units. Then we can assume that packet 1 leaves the first node at time $t = l$, and that packet 2 gets no delay at the node, so that its departure time equals its arrival time at the node. In this way at the input to node 2 we have that packet 2 precedes packet 1.

At node 2, we assume again that the departure time of packet 1 equals its maximum departure time (its GR clock value at the node) so that, as the GR clock value for packet 2 is $t = 2(l - \varepsilon)$, the GR clock value for packet 1 is $3l - 2\varepsilon$, and this is also the end-to-end delay for this packet.

The end-to-end delay bound from Equation (6.2) in this case would instead be of $2l$ time units, so that if $\varepsilon < \frac{l}{2}$ the delay of packet 1 in the example is larger than the delay bound in Equation (6.2). This simple example shows that the existing end-to-end delay bounds in a succession of GR nodes are not valid for non-FIFO nodes.

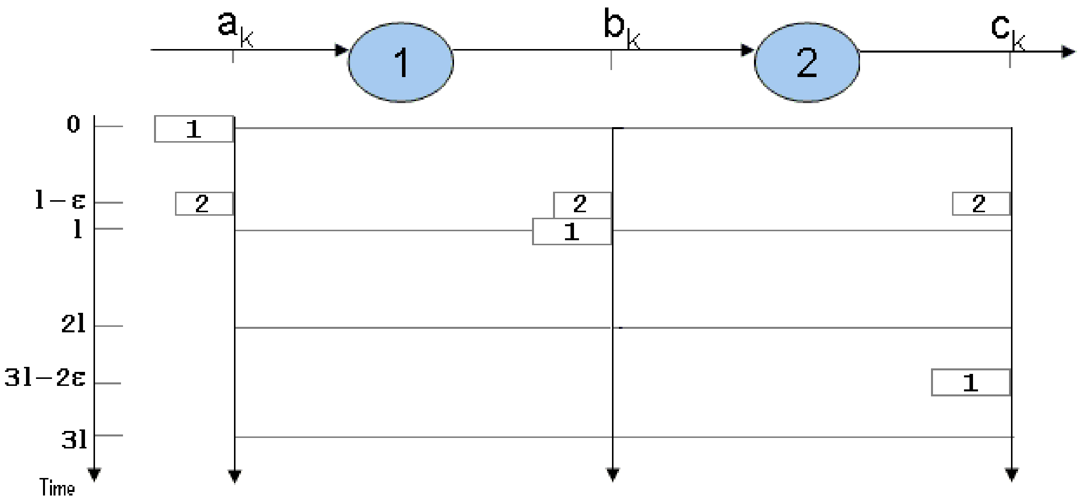

Figure 6.1 – Example of non-FIFO behavior of two GR nodes, traversed by a leaky bucket constrained flow, with $\sigma = l,\ \rho = r = 1,\ \tau^{1,2} = \tau^{2,3} = 0$. a_k and b_k are respectively the arrival times of packets at node 1 and 2, and c_k are the departure times of packets from node 2. At all nodes only packet 1 takes its maximum possible delay at the node, equal to its GR clock value at the node, whereas all other packets get no delay from the nodes. The end-to-end delay of packet 1 is of $3l - 2\varepsilon$ time units and, if $\varepsilon < \frac{l}{2}$, it gets a larger delay than the delay bound in [19], which is of $2l$ time units.

6.3.3 The hidden FIFO assumption in [19]

In this section, we analyze the original derivation of the end-to-end delay bounds of Equation (6.1) and Equation (6.2) from [19], in order to put in evidence that those results are actually valid only for FIFO.

The hidden assumption is in the proof of the following lemma derived in [19], used to derive the bounds in Equation (6.1) and Equation (6.2):

Lemma 6.3.1 ([19]) *If the scheduling algorithm at servers m and $m+1$ along the path of the flow belongs to GR for that flow, then*

$$GRC^{m+1}(p^j) \leq GRC^m(p^j) + \max_{k\in[1,..,j]} \frac{l^k}{r} + \alpha^m, \quad j \geq 1 \tag{6.3}$$

where p^j is the j-th packet of the flow, l^k is the length of the k-th packet of the flow, r is the guaranteed rate for the flow at nodes m and $m+1$.

The hidden FIFO assumption lies in the following inequality, between equations (23) and (24) of the proof of Lemma 6.3.1 in [19]:

$$GRC^m(p^{j+1}) \geq GRC^m(p^j) + \frac{l^{j+1}}{r} \tag{6.4}$$

The hidden assumption is in the use of packet indices at two consecutive nodes m and $m+1$. In the proof, index j refers to the succession of packet arrivals at node $m+1$: Indeed, from Equation (6.4) (that derives from the GR clock definition at node m) we see that the same packet index j is used for the succession of packets at the input to node m. This implies that no packet reordering takes place at node m, and that node m is assumed to be FIFO for the flow.

Indeed, if we look at the example in Fig. 6.1, we can clearly see that, if node 1 is non-FIFO, then Equation (6.4) is false for packet 1.

This can be shown by comparing the GR clock values of packet 1 at nodes 1 and 2. At node 1, the GR clock value for packet 1 is l. At node 2, the GR clock value for packet 2 (which is the first to arrive at the node, at time $l - \varepsilon$) is $2(l - \varepsilon)$, and the GR clock value for packet 1, arrived at the node at time l, is $3l - 2\varepsilon$. Now, for packet 1 Equation (6.4) translates into the following inequality:

$$3l - 2\varepsilon \leq 2l \tag{6.5}$$

As we saw in the example, if $\varepsilon < \frac{l}{2}$ then $3l - 2\varepsilon > 2l$, and Equation (6.3) does not hold in this case.

As in the non-FIFO case the Lemma in [19] does not hold, the whole proofs of the delay bounds in Equation (6.1) and Equation (6.2) in [19], which rely on that lemma, are not valid in the non-FIFO case.

6.4 An end-to-end delay bound valid in the non-FIFO case

6.4.1 The delay bound

As we showed in the previous section, when nodes are not FIFO Theorem 6.3.1 and Equation (6.2) cannot hold. Therefore the question that arises is whether the end-to end delay in a network of generic non-FIFO nodes is bounded, and which is the expression of the bound in this case. To our knowledge the issue is still open, as no result in the present literature addresses it in an exhaustive way. We answer to this with the following theorem:

Theorem 6.4.1 *Consider a leaky bucket constrained flow, with burstiness σ and rate ρ, that traverses a sequence of M GR nodes. If all nodes reserve the same service rate r to the flow, and if all nodes satisfy the node serviceability condition, an end-to-end delay bound for a packet belonging to the flow is given by*

$$d = M\frac{\sigma}{r} + \frac{\rho}{r}\frac{l_{max}}{r}\frac{M(M-1)}{2} + \frac{\rho}{r}\sum_{m=1}^{M}\sum_{i=1}^{m-1} e^i + \sum_{m=1}^{M}\left(e^m + \tau^{m,m+1}\right) \tag{6.6}$$

where l_{max} is the maximum packet length for the flow. Moreover, if we denote with σ_M the burstiness of the arrival curve for the flow at the output of node M, we have that

$$\sigma_M = \sigma + M\rho\frac{l_{max}}{r} + \rho\sum_{m=1}^{M} e^m \tag{6.7}$$

By comparing the bound in Equation (6.6) with the one that can be obtained when we know that nodes are FIFO, we can clearly see how, in the non-FIFO case, the contribution to the end-to-end delay which is due to the burstiness σ of the initial flow is multiplied by a factor M. Hence, in the non-FIFO case, the non-validity of the concatenation result for the computation of an end-to-end delay bound brings to "pay" burst M times, instead of only once [26].

We can also observe that another consequence of the non-FIFO behavior of GR nodes is an increment of the burstiness of the flow at the output of the last node by the quantity $\rho\frac{l_{max}}{r}$, with respect to the FIFO case.

Proof (Theorem 6.4.1): We first observe that the node serviceability condition implies $\rho \leq r$ at all the M nodes. As GR nodes are not necessarily FIFO, for the end-to-end delay computation we can exploit some properties of GR nodes that do not depend on their FIFO behavior. Among the network calculus results still valid in the non-FIFO case we have Theorem 2.2.8 and Corollary 2.2.1 in Section 2.2.4.

As the equivalence between a GR node and a rate-latency service curve element holds also for non-FIFO nodes, a sequence of M GR nodes can still be studied as the concatenation of service curve elements, each one of the form $\beta_{r,\, e+\frac{l_{max}}{r}}$. The link between two nodes on the path of the flow can be modeled as a FIFO constant delay element, with a minimum service curve of the form $\delta_{\tau^{m,m+1}}$, and a maximum service curve with the same expression [26].

The equivalence between a GR node and a service curve element implies that, if we consider a flow constrained by an arrival curve $f(t)$ that gets into a GR node, at the output of the node that flow is constrained by the arrival curve $f\left(t + e + \frac{l_{max}}{r}\right)$.

Although in a FIFO GR node this implies that a bound to the delay at the node is given by $e + \frac{l_{max}}{r}$, this is not true in a non-FIFO node, and the actual maximum delay that a packet can experience can be larger, as packet overtaking can take place inside the node (intuitively, a packet that takes its maximum delay in a FIFO node would be delayed by exactly $e + \frac{l_{max}}{r}$, whereas in a non-FIFO node, if "overtaken" by some other packets that followed it at the input, it can leave the node with a larger delay, as it must first wait for the overtaking packets to leave the node).

In order to derive a delay bound at each of the M non-FIFO GR nodes we can exploit Theorem 2.2.7 on delay bound at a GR node. If we consider a leaky bucket constrained flow that traverses a sequence of M GR nodes, the sequence of GR servers offers to it a minimum service curve given by the min-plus convolution between the service curves of all the GR nodes and links in the sequence, and a maximum service curve given by the min-plus convolution between the maximum service curves of all links.

Proposition 6.4.1 ([26]) *Consider a flow that traverses a sequence of service curve elements in a network. In order to compute an output bound for the flow, fixed delay elements on the path of the flow can be ignored.*

As a consequence, if we use the notation $f^{m+1}(t)$ for an arrival curve of the flow at the input to the $m+1$-th node in the sequence ($m \in [1, M]$), using the properties of the deconvolution operator [26] we have that

$$f^{m+1}(t) = \left[(\sigma + \rho t) \otimes \delta_{\sum_{i=1}^{m} \tau^{i,i+1}}\right] \oslash \beta_{r,\, \sum_{i=1}^{m}\left[\frac{l_{max}}{r} + e^i + \tau^{i,i+1}\right]} =$$

$$= \sigma + \rho \left(t + \sum_{i=1}^{m} \left[\frac{l_{max}}{r} + e^i \right] \right)$$

where $\otimes$ is the convolution operator, $\oslash$ is the deconvolution operator [26], and $\beta_{r,\, \sum_{i=1}^{m}\left[\frac{l_{max}}{r} + e^i + \tau^{i,i+1}\right]}$ is the service curve [26] of the concatenation of nodes $1, ..., m$ and of the links between them.

Using the delay bound for a single node in Theorem 2.2.7, a delay bound at the m-th node along the succession of the M GR nodes is given by

$$d_m = \frac{\sigma}{r} + (m-1)\frac{\rho}{r}\frac{l_{max}}{r} + \frac{\rho}{r}\sum_{i=1}^{m-1} e^i + e^m \tag{6.8}$$

and a delay bound for the concatenation of the m-th node and the link between nodes m and $m+1$ is given by $d_m + \tau^{m,m+1}$.

An end-to-end delay bound for the packets of the flow is obtained by summing the delay bounds in Equation (6.8) at each node along the path of the flow, and taking into account the propagation delays at all links:

$$d = \sum_{m=1}^{M} (d_m + \tau^{m,m+1}) =$$

$$M\frac{\sigma}{r} + \frac{\rho}{r}\frac{l_{max}}{r}\sum_{m=1}^{M}(m-1) + \frac{\rho}{r}\sum_{m=1}^{M}\sum_{i=1}^{m-1} e^i + \sum_{m=1}^{M}\left(e^m + \tau^{m,m+1}\right) =$$

$$= M\frac{\sigma}{r} + \frac{\rho}{r}\frac{l_{max}}{r}\frac{M(M-1)}{2} + \frac{\rho}{r}\sum_{m=1}^{M}\sum_{i=1}^{m-1} e^i + \sum_{m=1}^{M}\left(e^m + \tau^{m,m+1}\right)$$

□

6.4.2 The delay bound in the non-FIFO case is tight

Theorem 6.4.2 *With the same assumptions as in Theorem 6.4.1, the bounds in Equation (6.6) and Equation (6.7) are tight. More precisely, we can always define a succession of packets and a series of scheduling behaviors of the chain of GR nodes such that the burstiness of the flow at the output of the M-th node achieves the bound in Equation (6.7), and that at least one packet from the given flow experiences an end-to-end delay equal to the bound in Equation (6.6).*

Proof: The proof of Theorem 6.4.2 is by example. Let's take a leaky bucket constrained flow, with burstiness σ and rate ρ, that traverses a sequence of M non-FIFO GR nodes, all with the same delay e and the same service rate r for the flow.

We assume for simplicity that $e = k$ time units, with $k \in \mathbb{N}$, that $\forall m,\ \tau^{m,m+1} = \tau = \frac{l}{r}$ and we take $\sigma = nl > (k+1)l$. In order to simplify the notation, we assume that $\rho = r$, and that all packets are of the same length l.

The example can be built as follows:

The sequence of packets: we consider the following sequence of packets, at the input to the first of the M nodes:

- at $t = 0$, we have the arrival of a burst of dimension $\sigma = nl$;
- then, with a period $P(i) = \frac{\sigma}{r} + (i-1)\frac{l}{r} + ik\frac{l}{r},\ i \geq 1$, we have the arrival of a burst of dimension $\sigma = nl$. The arrival time of the i-th burst at the first of the M nodes is given by

$$t_i = \sum_{j=1}^{i} P(j) = i\frac{\sigma}{r} + \frac{l}{r}\sum_{j=1}^{i}(j-1) + k\frac{l}{r}\sum_{j=1}^{i} j$$

- For $i \geq 1$ we define the time instants $t_i^* = t_i - (i-1)\frac{l}{r} - ik\frac{l}{r}$. Then we assume that in the time intervals $[t_i^*,\ t_i),\ i \geq 1$ we have a packet arrival at time t_i^* and then the arrival of a packet each $\frac{l}{r}$ time units, so that a total of $i - 1 + ik$ packets arrive in each interval $[t_i^*,\ t_i)$.

We can verify that such a succession of packets is leaky bucket constrained, with burstiness σ and rate ρ. On Fig. 6.2 we have an example of a succession of packets with these characteristics, with $\sigma = 4l$, $e = 2\frac{l}{r}$ and $\frac{l}{r} = 1$ time unit.

The scheduling behavior: given the initial burst of the sequence, of dimension $\sigma = nl$, which arrives at the first of the M nodes at time 0, we consider one of the packets that compose it, and we indicate it with p_n (in order to distinguish it from p^n, the n-th packet to get into a given node).
We assume that, at the input to the m-th node along the path of the flow:

- all packets that precede packet p_n (p_n included) at the input of the m-th node get the *maximum* delay at the node;
- if p_n is part of a burst of packets, arrived at a node in the same time instant as p_n, it is always the last to be served;
- all packets p^j ([1]) that get into the m-th node after packet p_n, and in time intervals $[t_i^* + (m-1)\tau,\ t_i + (m-1)\tau),\ i \geq 1$, get a delay equal to $e + \frac{l}{r}$, but do not get out of the node after time $t_i + (m-1)\tau$. That is, their departure time is

$$d^m(p^j) = \min\left\{A(p^j) + e + \frac{l}{r},\ t_i + (m-1)\tau\right\}$$

- all packets that get into the m-th node after packet p_n, and at time instants $t_i + (m-1)\tau,\ i \geq 1$, get the *minimum* delay for that node.

At each GR node, the maximum departure time for each packet is equal to the sum of its Guaranteed Rate clock value and of the delay of the node, whereas its minimum departure time is equal to its arrival time at the node. As a consequence, given the

[1] We underline here that packet indices refer to the succession of packets at the input to a specific node, so that in general different packet indices are to be used for packets at the input to each node.

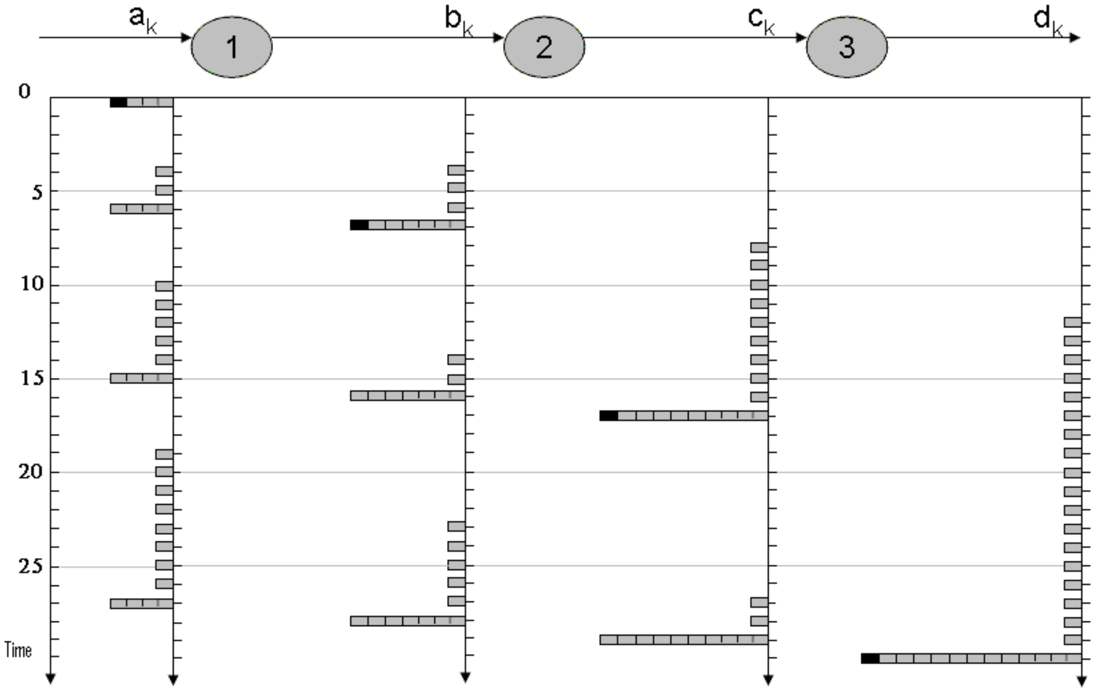

Figure 6.2 – Evolution of a sequence of packets, at the input to each of three GR nodes on its path, and at the output of node 3. At the input to node 1, there is a sequence with the characteristics described in the proof of Theorem 6.4.2, and which is constrained by an arrival curve of the leaky bucket type, with burstiness σ and rate ρ, with $\frac{l}{r} = 1$ time unit, $\sigma = 4l$ and $e = 2\frac{l}{r}$ time units. As the propagation delay at all links is of 1 time unit, the delay experimented by the packet marked in black at the input to nodes 2, 3, and at the output to node 3 (taking into account propagation delay of the link at the output of node 3) is respectively of 7, 10 and 13 time units, and the burstiness of the output flow is respectively of $7l$, $10l$ and $13l$, as predicted by Th. 6.4.2.

structure of the sequence of packets, for all packets that precede or arrive at the same time as packet p_n at the input of a node (packet p_n included), we observe at the output, starting from the first packet, one packet departure each $\frac{l}{r}$ time units.
In order to demonstrate the tightness of the bounds in Equation (6.6) and Equation (6.7), we use induction on the index m of the succession of the M nodes on the path of the flow.
Base case: $m = 1$, the first node of the path. The departure time of the first packet to be served at node 1 is $d^1(p^1) = A^1(p^1) + \frac{l}{r} + e = (k+1)\frac{l}{r}$, since we assumed that the arrival time at node 1 is at $t = 0$.
As all the n packets that arrived at $t = 0$ leave the node with the maximum delay, starting from time $t = d^1(p^1)$ we observe at the output of node 1 one packet leaving

the node each $\frac{l}{r}$ time units, up to packet p_n, which then leaves the node at time

$$d^1(p_n) = A^1(p^1) + n\frac{l}{r} + e = \frac{\sigma}{r} + e$$

Taking into account propagation delay on the link between nodes 1 and 2, the delay of packet p_n at the input to node 2 is equal to $d^1(p_n) + \tau$, and in this case the delay bound in Equation (6.6) is verified.
In the time interval $[\frac{\sigma}{r},\ \frac{\sigma}{r} + k\frac{l}{r})$ at the input to node 1, we have one packet arrival each $\frac{l}{r}$ time units and a total of k packet arrivals in the whole interval. The first of this packets arrives at node 1 at time $t = \frac{\sigma}{r}$, and it leaves the node at time $t' = \frac{\sigma}{r} + k\frac{l}{r}$. This implies that all packets arrived in that time interval leave the node at time t'. Then, at time t' at the input of the node we have the arrival of a burst of dimension $\sigma = nl$, that is not delayed by node 1. As $d^1(p_n) = t'$, we have that at time t' at the output of the node the flow has a burst of dimension $(n + k + 1)l$, achieving the burstiness bound in Equation (6.7).
Iterative step: The inductive hypothesis is that Equation (6.6) and Equation (6.7) hold for the sequence of nodes from 1 to m. We want to demonstrate that they hold also for the sequence of nodes from 1 to $m + 1$ ($m + 1 \leq M$).
By the inductive hypothesis, the time at which packet p_n arrives at node m can be obtained by Equation (6.6):

$$A^m(p_n) = (m-1)\frac{\sigma}{r} + \frac{l}{r}\sum_{j=1}^{m-1}(j-1) + k\frac{l}{r}\sum_{j=1}^{m-1} j + (m-1)\tau$$

and the time at which it leaves node m is given by

$$d^{m+1}(p_n) = m\frac{\sigma}{r} + \frac{l}{r}\sum_{j=1}^{m}(j-1) + k\frac{l}{r}\sum_{j=1}^{m} j + (m-1)\tau$$

Due to the structure of the sequence of packets and the scheduling behavior of nodes, after the arrival of p_n at node m we have, in the time interval $[t^*_m + (m-1)\tau,\ t_m + (m-1)\tau]$, the arrival of $n + mk + m - 1$ packets, with

$$t_m = m\frac{\sigma}{r} + \frac{l}{r}\sum_{j=1}^{m}(j-1) + k\frac{l}{r}\sum_{j=1}^{m} j$$

Then, for the scheduling behavior of the sequence of nodes, the packet that arrives at node 1 at time t^*_m (and so, the first packet to arrive at node m in the time interval $[t^*_m + (m-1)\tau,\ t_m + (m-1)\tau]$) takes by each node in the succession $1, ..., m$ a delay equal to $(k+1)\frac{l}{r}$, and by each link a delay of τ. Therefore, it leaves node m at a time

$t' + (m-1)\tau$, where $t' = \min\left\{t_m^* + m(k+1)\frac{l}{r},\ t_m\right\}$.
As $t_m - t_m^* = mk\frac{l}{r} + (m-1)\frac{l}{r}$, then we have that $t' = t_m$, and all the $n+mk+m-1$ packets leave node m at time $t_m + (m-1)\tau$.
As $A^{m+1}(p_n) = t_m + m\tau$, at time $t_m + m\tau$ at the input to node $m+1$ we have the arrival of $n + mk + m$ packets, and a burst of dimension $n + m(k+1)$.
In general, at the output of each node x, $x \in 1, ..., m$, at time $t_x + (x-1)\tau$ we have the departure of packet p_n and of $n + xk + x - 1$ other packets. As at all nodes p_n is always the last packet to be served among those that arrived at the same time as p_n, and as all packets served before p_n get the maximum delay, we have that

Remark 6.4.1 *At the input of nodes* $2, ..., m+1$*, the arrival at time* t *of packet* p_n *is always preceded by the arrival, at time* $t - \frac{l}{r}$*, of another packet.*

Another result that is important for the rest of the proof, is the following:

Lemma 6.4.1 *At nodes* $1, ..., m+1$*, for all packets* p *that arrive at the node before packet* p_n*, we have that*

$$GRC(p) = A(p) + \frac{l}{r} \tag{6.9}$$

Proof (Lemma 6.4.1): At the node, the GRC of the first packet that arrives is given by $GRC(p^1) = A(p^1) + \frac{l}{r}$, as no packet precedes it. The GRC of the second packet is $GRC(p^2) = \max\{A(p^2), GRC(p^1)\} + \frac{l}{r} = \max\left\{A(p^2), A(p^1) + \frac{l}{r}\right\} + \frac{l}{r}$.
As all packets that precede p_n at the node get their maximum delay, packet interarrival times are at least of $\frac{l}{r}$ time units. So we have that $A(p^2) \geq A(p^1) + \frac{l}{r}$, and $GRC(p^2) = A(p^2) + \frac{l}{r}$.
For the same reason, in general (for all packets p^j that get at a node before p_n) we have that $A(p^j) \geq A(p^{j-1}) + \frac{l}{r}$, and Equation (6.9) holds. □

At node $m+1$, the GR clock value of the first packet to be served among those arrived at time $t_m + m\tau$ (that we denote with p^j) is given by:

$$GRC^{m+1}(p^j) = \max\left(t_m + m\tau,\ GRC^{m+1}(p^{j-1}) + \frac{l}{r}\right) + \frac{l}{r}$$

where p^{j-1} is the packet that precedes packet p^j at the input to node $m+1$.
Using Lemma 6.4.1 and Remark 6.4.1, we have that $t_m + m\tau = GRC^{m+1}(p^{j-1}) + \frac{l}{r}$.
So we have $GRC^{m+1}(p^j) = t_m + m\tau + \frac{l}{r}$.
Then packet p_n, that is the last packet to be served among the $n + mk + m$ packets arrived at time $t_m + m\tau$, will have a GR clock value at node $m+1$ given by

$GRC^{m+1}(p_n) = t_m + m\tau + (n + mk + m)\frac{l}{r}$, and the departure time from node $m+1$ for packet p_n is

$$\begin{aligned} d^{m+1}(p_n) &= GRC^{m+1}(p_n) + k\frac{l}{r} = t_m + m\tau + (n + (m+1)k + m)\,\frac{l}{r} = \\ &= (m+1)\frac{\sigma}{r} + \frac{l}{r}\sum_{j=1}^{m+1}(j-1) + k\frac{l}{r}\sum_{j=1}^{m+1} j + m\tau \end{aligned}$$

and taking into account the propagation delay of the link at the output of node $m+1$, we have that the end-to end delay for packet p_n for the succession of nodes $1, ..., m+1$ is given by $d^{m+1}(p_n) + \tau$, and it equals the end-to-end delay bound in Equation (6.6).
Also, with a similar procedure to the one followed at node m, we have that, at time $t_{m+1} + m\tau$ $(= d^{m+1}(p_n))$ at the output of node $m+1$ we have a burst of $n + (m+1)(k+1)$ packets, so that the flow at the output of node $m+1$ achieves the burstiness bound in Equation (6.7). □

6.4.3 A refined result

We now introduce a new node model, more realistic than the simple GR node model in what packet delay has a minimal value greater than zero. Specifically, this new model is composed by a FIFO GR node, with rate r and zero delay, followed by a FIFO constant delay element with delay e_a, and by a non-FIFO variable delay element, with maximum delay e_b.

Although the GR model of a scheduler allows for a packet to have zero delay at the node, real schedulers do not have a minimal delay equal to zero: They usually always introduce some amount of delay to packets. That is, the departure time of a packet p (arrived at the node at time $A(p)$) is

$$A(p) + e_a \leq d'(p) \leq GRC(p) + e_a + e_b \tag{6.10}$$

As an example, this happens in input buffer switches, in which the minimum delay for a packet in the node is due to the minimum time necessary for a packet to traverse the fabric. In the presented model, the fabric is modeled by the succession of the constant delay element and of the variable delay element. In this sense, the model presented captures more closely and realistically the characteristics of network nodes. In this case, we have the following proposition:

Proposition 6.4.2 *It is given a GR node, with rate r and delay $e_a + e_b$, at which the departure time for a generic packet falls inside the interval $[A(p) + e_a,\ GRC(p) + e_a + e_b]$. Such a node is equivalent to the succession of a FIFO GR server, with rate r and zero delay, followed by a FIFO constant delay element with delay e_a, and by a non-FIFO variable delay element, with maximum delay e_b.*

Proof: In order to show this equivalence, let us analyze the delay of a packet at the output of such a succession of elements. By definition, the departure time of a packet p at the FIFO GR server is upper bounded by the GR clock $GRC(p)$ for that packet at the GR node.

Now, for any $r' \geq 0$, a variable delay element is a GR node, with rate r' and delay $[e_b - \frac{l_{min}}{r'}]^+$ [2], where l_{min} is the minimum packet size for the flow [26]. If we indicate with $GRC'(p)$ the GR clock value of packet p at the variable delay element, and the arrival time of packet p at the variable delay element as $A'(p)$, the total delay $d'(p)$ at this element is given by

$$A'(p) \leq d'(p) \leq GRC'(p) + \left[e_b - \frac{l_{min}}{r'}\right]^+$$

Now, letting $r' \to \infty$, we have $GRC'(p) = A'(p)$, and the total delay $d'(p)$ at the variable delay element falls in the interval $A'(p) \leq d'(p) \leq A'(p) + e_b$. Then the total delay of the succession of the FIFO constant delay element and the variable delay element falls in the interval $[e_a,\ e_a{+}e_b]$. Taking into account also the delay of the FIFO GR element, we have that the departure time $d(p)$ of packet p from the succession of the three elements is

$$A(p) + e_a \leq d(p) \leq GRC(p) + e_a + e_b$$

□

We than have another version of Theorem 6.4.1:

Theorem 6.4.3 *Consider a flow constrained by an arrival curve of the leaky bucket type, with burstiness σ and rate ρ, that gets into a sequence of M nodes. We assume that each node m, $m \in\ 1, ..., M$ is a GR node, with rate r and delay $e_a^m + e_b^m$, at which the departure time for a generic packet falls inside the interval $[A^m(p) + e_a^m,\ GRC^m(p) + e_a^m + e_b^m]$. If the node serviceability condition is satisfied at each node, an end-to-end delay bound for a packet belonging to the flow is given by*

[2] The notation $[x]^+$ stands for $\max(x,\ 0)$.

$$d = M\frac{\sigma}{r} + \frac{\rho}{r}\frac{l_{max}}{r}\frac{M(M-1)}{2} + \frac{\rho}{r}\sum_{m=1}^{M}\sum_{j=1}^{m-1} e_b^j + \sum_{m=1}^{M}\left(e_a^m + e_b^m + \tau^{m,m+1}\right) \quad (6.11)$$

Moreover, the burstiness of the arrival curve for the flow at the output of node M, σ_M, is given by

$$\sigma_M = \sigma + M\rho\frac{l_{max}}{r} + \rho\sum_{m=1}^{M} e_b^m \quad (6.12)$$

Proof: For each GR node we make use of the equivalence in Proposition 6.4.2. First of all, we can consider the FIFO constant delay element at the m-th GR node ($m \in 1, ..., M$) and the link between this node and node $m+1$ as a single FIFO delay element, which introduces a constant delay $\alpha^m = e_a^m + \tau^{m,\, m+1}$.

At the m-th GR node, the service curve of the succession of the FIFO GR element and of the non-FIFO variable delay element is given by the min-plus convolution between:

- the service curve of the FIFO GR element, equal to $\beta_{r,\,0}$; and
- the service curve of the non-FIFO variable delay element that, by the equivalence with a GR node [26], is equal to $\beta_{r',\,[e_b^m - \frac{l_{min}}{r'}]^+}$, for any $r' \geq 0$. Letting $r' \to \infty$, it becomes equal to $\delta_{e_b^m}$.

The resulting service curve is then given by $\beta_{r,\, e_b^m}$.

The proof then proceeds similarly to the one of Theorem 6.4.1, with e_b^m instead of e^m at each node, and substituting $\tau^{m,\, m+1}$ with α^m. □

In order to have an idea of the difference between the values assumed by delay bounds in Equation (6.2) and those given by Equation (6.11) of Theorem 6.4.3, in Table 6.2 we reported the values assumed by the two bounds, for different values of the burstiness σ for the considered flow. We can observe that the actual delay bound in the non-FIFO case can be many times larger than the one holding for FIFO nodes.

6.4.4 The FIFO case

When GR nodes are FIFO per flow, the result in Theorem 1 in [19] is valid. Another way to derive it in the FIFO case is to exploit the network calculus result for the concatenation of FIFO GR nodes [26]:

σ	Delay bound (ms) (non-FIFO case, Equation (6.11))	Delay bound (ms) (FIFO case, Equation (6.2)[19])
512 bytes	117.49	31.47
1 kB	146.16	35.57
1.5 kB	174.83	39.66
2 kB	203.50	43.76

Table 6.2 – Delay bounds comparison between the FIFO and the non-FIFO case. Values assumed by the delay bounds in Equation (6.2) from [19] and in Equation (6.11). We made the following assumptions: $M = 7$ nodes; all nodes have the same delay e (and the same fixed part of delay e_a); all links introduce the same delay $\tau = 400\mu s$; all nodes guarantee a service rate $r = 1$ Mbit/s to the flow; $\rho = r$, $e_a = 100ns$, $e_b = 10ns$; all packets have the same length $l = 512$ bytes.

Theorem 6.4.4 (Concatenation FIFO GR nodes[26]) *The concatenation of M GR nodes (that are FIFO per flow) with rates r_m and latencies e^m is GR with rate $r = \min_m(r_m)$ and latency $e = \sum_{m=1}^{M} e^m + \sum_{m=1}^{M-1} \frac{l_{max}}{r_m}$.*

Using Theorem 2.2.7, a delay bound for the concatenation of the M FIFO GR nodes, when the flow that traverses them is leaky bucket constrained, and when all nodes reserve the same service rate r for the flow, is given by Equation (6.2), so that we find the result of Theorem 6.3.1.

6.5 Summary

In this chapter we considered end-to-end delay bounds in networks of Guaranteed Rate nodes. We showed that the validity of the available methods to derive end-to-end delay bounds in such networks is restricted to the case in which nodes are globally FIFO (that is, they are FIFO per flow and per microflow). We proved with a counterexample that those delay bounds are not valid in the non-FIFO case. We put in evidence the implicit FIFO assumption in the original derivation of the bounds, and we determined new bounds that are valid in the non-FIFO case. We showed the tightness of the bounds derived in the non-FIFO case. We also gave evidence of how, in a realistic scenario, the new bounds can be sensibly higher than those valid in the FIFO case.
The results obtained in the present chapter are important when applying the method described in Chapter 3 to a generic network of GR nodes.

Chapter 7

Conclusion

7.1 Achievements

- A first result of the present research is the elaboration of a general method that allows to derive sufficient conditions for stability in a generic network of aggregate schedulers. The method is very general, and it applies to a very broad range of assumptions on the network.

- Exploiting this general method, we derived a generalized version of the “RIN result”, which extends the classical result to leaky bucket constrained flows and to heterogeneous settings. We verified on some realistic network examples how the generalized version of the “RIN result” allows an average node utilization that is far larger than the maximum node utilization allowed by previous results in the heterogeneous settings.

- We derived a set of sufficient conditions for stability in a generic network of FIFO aggregate schedulers, which holds for any topology and in an heterogeneous setting, and is therefore applicable in realistic scenarios. The method outperforms the two main existing results (the “RIN result” and the “Charny-Le Boudec bound”) and we proved that they can be derived from it.

- Finally we determined, for a network of generic (not necessarily FIFO) Guaranteed Rate nodes, a correct formula for the computation of an end-to-end delay bound, correcting a previously existing result that implicitly assumed FIFO behavior at each node.

7.2 Future work

- *Extension to the ring case*: the only existing result to which the sufficient conditions we derived do not revert to is the one of the ring [35]. Then a way to make more powerful the set of sufficient conditions derived in the present work is to define a proper set of variables and to build and a set of upper bounding relations that can allow to derive the ring result within the approach described here.

- *Extension to the Stochastic Domain*: it could be of interest to apply the general method defined here in a network in which stochastic network calculus is used to define service guarantees at nodes and flows constraints. In particular, it could be interesting to see which inner bounds to the stability region of a network can be derived in the stochastic domain.

- *Improvement of the node utilization level* Derivation of sufficient conditions for stability that imply a higher node utilization level than the ones derived, especially in large networks with a high maximum hop count.

List of Figures

List of Tables

Bibliography

[1] S. Lagrange A. Bouillard, B. Gaujal and E. Thierry. Optimal routing for end-to-end guarantees: the price of multiplexing. 2007.

[2] M. Andrews. Instability of fifo in session-oriented networks. In *Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2000)*, pages 440–447, January 2000.

[3] 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.

[4] 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.

[5] F. Begtasevi and C. Van Mieghem. Measurements of the hopcount in internet. In *Proceedings of Passive and Active Measurement (PAM2001)*, Amsterdam, April 2001.

[6] Partridge C. Shectman N. Bennet, J.C.R. Packet reordering is not pathological network behavior. *IEEE/ACM Trans. on Networking*, 7(6):789–798, December 1999.

[7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services, December 1998. RFC 2475, IETF.

[8] J. M. Blanquer and B. Ozden. Fair queuing for aggregated multiple links. In *Proc. Sigcomm 2001*, pages 189–197, September 2001.

[9] J-Y Le Boudec and A. Charny. Packet scale rate guarantee for non-fifo nodes. *IEEE/ACM Transactions on Networking*, 11(5):810–820, October 2003.

[10] Jean-Yves Le Boudec and Gérard Hébuterne. Comments on "a deterministic approach to the end-to-end analysis of packet flows in connection oriented networks". *IEEE/ACM Transactions on Networking*, 8(1):121–124, 2000.

[11] C.S. Chang. *Performance Guarantees in Communication Networks*. Springer-Verlag, New York, 2000.

[12] Anna Charny and Jean-Yves Le Boudec. Delay bounds in a network with aggregate scheduling. In *QofIS*, pages 1–13, September 2000.

[13] I. Chlamtac. Bounded delay routing for real-time communication in tactical radio networks. *Proc. of IEEE Military Communications Conference (MILCOM'89), Boston, MA,*, pages 37–41, October 1989.

[14] 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*, (6)4:422–431, August 1998.

[15] R.L. Cruz. A calculus for network delay, part i: Network elements in isolation. *IEEE Trans. Inform. Theory, vol 37-1*, pages 114–131, January 1991.

[16] R.L. Cruz. A calculus for network delay, part ii: Network analysis. *IEEE Trans. Inform. Theory, vol 37-1*, pages 132–141, January 1991.

[17] Markus Fidler. Extending the network calculus pay bursts only once principle to aggregate scheduling. pages 19–34.

[18] S. J. Golestani. A self clocked fair queuing scheme for high speed applications. In *Proceedings of Infocom 1994*, 1994.

[19] P. Goyal, S. S. Lam, and H. Vin. Determining end-to-end delay bounds in heterogeneous networks. In *5th Int Workshop on Network and Op. Sys support for Digital Audio and Video, Durham NH*, April 1995.

[20] P. Goyal and H. Vin. Generalized guaranteed rate scheduling algorithms: a framework. *IEEE/ACM Trans. Networking, vol 5-4*, pages 561–571, August 1997.

[21] Y. Jiang. Delay bounds for a network of guaranteed rate servers with fifo aggregation. *Comput. Networks*, 40(6):683–694, 2002.

[22] Zhang L. Virtual clock: A new traffic control algorithm for packet switching networks. In *Proceedings of ACM SIGCOMM '90*, pages 19–29, August 1990.

[23] E. Mingozzi G. Stea L. Lenzini, L. Martorini. A methodology for deriving per-flow end-to-end delay bounds in sink-tree diffserv domains with fifo multiplexing. In *Proceedings of ISCIS 2004*.

[24] G. Stea L. Lenzini, E. Mingozzi. Delay bounds for fifo aggregates: A case study. In *Lecture Notes in Computer Science, Springer*, volume 2811, pages 31–40, 2003.

[25] J.-Y. Le Boudec and G. Hebuterne. Comment on a deterministic approach to the end-to-end analysis of packet flows in connection oriented network. *IEEE/ACM Transactions on Networking*, 8:121–124, February 2000.

[26] 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.

[27] M. Marcus and H. Minc. *A survey of matrix theory and matrix inequalities*. Dover Publications Inc., New York, 1992.

[28] M.Blesa. Deciding stability in packet-switched fifo networks under the adversarial queuing model in polynomial time. *Proc. of the 19th International Symposium on Distributed Computing, LNCS Vol. 3724*, pages 429–441, 2005.

[29] Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers. BRITE: Universal topology generation from a user's perspective. Technical Report 2001-003, 1 2001.

[30] Florian-Daniel Otel and Jean-Yves Le Boudec. Deterministic end-to-end delay guarantees in a heterogeneous route interference environment. In *QofIS*, pages 21–30, October 2003.

[31] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single node case. *IEEE/ACM Trans. Networking, vol 1-3*, pages 344–357, June 1993.

[32] G. Rizzo and J.Y. Le Boudec. ´ pay bursts only once´ does not hold for non-fifo guaranteed rate nodes. *Performance 2005*, 62(1-4), Oct 2005.

[33] G. Rizzo and J.Y. Le Boudec. Generalization of the rin result to heterogeneous networks and to leaky bucket constrained flows. *ICON 2007, Adelaide*, to appear, Nov 2007.

[34] G. Rizzo and J.Y. Le Boudec. Stability and delay bounds in heterogeneous networks of aggregate schedulers. *Infocom 2008, Phoenix*, to appear, Apr 2008.

[35] L. Tassiulas and L. Georgiadis. Any work-conserving policy stabilizes the ring with spatial re-use. *IEEE/ACM Transactions on Networking*, 4(2):205–208, August 1996.

[36] Z.Q. Chen V.Y. Pan and A. Zheng. The complexity of the algebraic eigenproblem. December 1998.

[37] Hongbiao Zhang. A note on deterministic end-to-end delay analysis in connection oriented networks. In *Proc of IEEE ICC'99, Vancouver, pp 1223–1227*, 1999.

Curriculum Vitae

Personal Data

16.02.1975	:	Born in Galatina (LE), Italy

Education

09/1981 – 06/1986	:	Elementary School
09/1986 – 06/1989	:	Gymnasium
09/1989 – 07/1994	:	Secondary Education
09/1994 – 02/2000	:	Studies in Electronic Engineering, Politecnico di Torino
03/2001	:	Diploma in Electronic Engineering
01/2004 – 11/2007	:	PhD student at LCA, EPFL Lausanne

Civil Service

03/2000 – 02/2001	:	UILDM Torino

Employment

09/2001 – 12/2003	:	Telecom Italia Labs (CSELT), Torino
Since 01/2004	:	EPFL Lausanne