
LECTURE NOTES IN CONTROL AND INFORMATION SCIENCES

308

Sophie Tarbouriech
Chaouki T. Abdallah
John Chiasson (Eds.)

# Advances in Communication Control Networks

Lecture Notes
in Control and Information Sciences 308

Editors: M. Thoma · M. Morari

S. Tarbouriech · C.T. Abdallah · J. Chiasson (Eds.)

# Advances in Communication Control Networks

With 99 Figures

Springer

## Series Advisory Board

A. Bensoussan · P. Fleming · M.J. Grimble · P. Kokotovic · A.B. Kurzhanski · H. Kwakernaak · J.N. Tsitsiklis

## Editors

Dr. Sophie Tarbouriech
LAAS-CNRS
7, Avenue du Colonel Roche
31077 Toulouse cedex 4
France

Prof. Chaouki T. Abdallah
The University of New Mexico
Department of Electrical and Computer Engineering
EECE Bldg. MSC01 1100
Albuquerque, NM 87131-0001
USA

Prof. John Chiasson
University of Tennessee
ECE Department
1508 Middle Drive
Knoxville, TN 37996-2100
USA

ISSN 0170-8643

ISBN 3-540-22819-5 **Springer Berlin Heidelberg New York**

Library of Congress Control Number: 2004111522



**Springer is a part of Springer Science+Business Media**

springeronline.com


Printed in Germany



Typesetting: Data conversion by the authors.
Final processing by PTP-Berlin Protago-TeX-Production GmbH, Germany
Cover-Design: design & production GmbH, Heidelberg
Printed on acid-free paper 62/3020Yu - 5 4 3 2 1 0

# Preface

The area of communication and computer networks has become a very active area of research by the Control Systems community in the last few years. The recent special issues in journals (e.g., *IEEE Transactions on Automatic Control*, Vol. 47, No. 6, June 2002, *Automatica*, Vol. 35, December, 1999, and *Control Engineering Practice*, Vol. 11, 2003 ) and the many special sessions in control conferences certify to this strong interest (see, for example, the plenary sessions of T. Basar and of F. Kelly in the IEEE 2003 Conference on Decision and Control, and the 2003 European Control Conferences, respectively, as well as the more than 15 sessions in the IEEE 2003 Conference on Decision and Control). As described in a special issue of IEEE Control Systems Magazine (vol.21, no.1, February 2001), the area of communication networks is of great interest to control researchers because its challenging problems fall within the scope and background of systems and control engineering. Furthermore, the increasing need to control dynamical systems via communication networks provides fertile grounds for control theorists as well as practitioners.

As a result, both analysis and synthesis approaches by the control community have appeared in the context of network control. Tools from convex optimization and control theory are playing increasing roles in efficient network utilization, fair resource allocation, and communication delay accommodation. On the other hand, as feedback control systems become increasingly distributed and dependent on shared networks for their efficient operation, the field of Networked Control Systems (NCS) is fast becoming a mainstay of control systems research and applications. In what follows, the term "communication control networks" will refer to both networks under control *(control in networks)* as well as networked control systems *(control over networks)*.

The complexity in the design and operation of communication/control networks, along with their real-time requirements, severely limit the ability to obtain accurate mathematical models. Such models are usually hybrid (containing both time-driven and event-driven dynamics), uncertain due to intentional simplification (fluid model approximations), or to parametric inaccuracies (uncertain delays), or still from the presence of additive disturbances (disruptive loads and congestion). Furthermore, in all practical situations, the network devices and systems are limited in their range

of operation (limited queue lengths and rates) and therefore are subject to amplitude and rate saturation. Hence, large amplitude disturbances or uncertainties may drive the states and/or the controls into saturation where the system may operate in a mode from which it may be difficult to preserve performance or even stability requirements. Finally, the inherent presence of packets loss, delays, and the limited-capacity communication medium are important features that need to be taken into account to guarantee the performance and stability of such networks.

Our objective in editing this book with solicited contributions from experts in the various areas of communication/control networks, is to present interesting and complementary techniques that treat problems arising in the control of networks as well as in control across networks. We also see this book as a modest attempt to reverse the trend of fragmentation and specialization of the related fields of control, communication, and computing. The book had its genesis in a multidisciplinary collaborative effort between France and the USA, under the auspices of CNRS and NSF, concerning problems in stability, stabilization, and optimization for time-delay systems and related applications including the control of communication networks. Hence, the developments proposed in the sixteen chapters are *interdisciplinary* as they cover various research fields including *Controls*, *Communications*, *Applied Mathematics*, and *Computer Science*.

The book is organized as follows.

- Part 1 is devoted to Fluid/Flow Models and consists of chapters 1 through 5.
- Part 2 is devoted to Congestion and Nonlinear Systems and consists of chapters 6 through 10.
- Part 3 is devoted to Load Balancing and Teleoperation and consists of chapters 11 through 13.
- Part 4 is devoted to "Emerging Control Theory and Information Complexity" and consists of chapters 14 to 16.

Note that this partition is somewhat arbitrary as most of the chapters are interconnected, and mainly reflects the editors' biases and interests.

We hope that this volume will help in claiming many of the communications and networking problems for controls researchers, and to alert graduate students to the many interesting ideas at the boundary between communications, computing, and controls.

## Acknowledgements

The idea of this edited book was formed through a series of e-mail exchanges and face-to-face discussions when the three of us met at conferences as part of the project "Time-delay systems: analysis, computer aided design, and applications," started in 2002 and jointly funded by CNRS (France) and NSF (USA). Indeed, many contributors ar e amongst the participants of that project, and most of the French participants al so belong to *CNRS GDR Automatique: Delay systems*, a research network

in Autom atic Control in existence since 1995. However, we also invited researchers outside these research teams in order to provide a deeper and more balanced representation of the areas.

First and foremost, we would like to thank all the contributors of the book. Wit hout their encouragement, enthusiasm, and patience, this book would have not been possible. A list of contributors is provided at the end of the book. Professors S.I. Niculescu and K. Gu in particular have contributed more than their share to the birthing of this volume. Next, we would like to thank CNRS and NSF, and more specifically our program man agers Claire Giraud, Jean-Luc Clément (CNRS), Rose Gombay and Kishan Baheti (NSF) for funding the joint research which made this book possible. We would also like to thank Luis Farinas Del Cerro (DRI-CNRS). Thanks also go to *GDR Automatique* (France), *MENRT* (France) and the laboratory *LAAS-CNRS*. Finally, thank also Isabelle Queinnec (LAAS-CNRS) for her help regarding latex problems.

We also wish to thank Springer for agreeing to publish this book. We wish to express our gratitude to the Editor-in-Chief Dr. Manfred Thoma, to Dr. Thomas Ditzinger (Engineering Editor), and Ms. Heather King (International Engineering Editorial) for their careful consideration and helpful suggestions regarding the format and organization of the book.

Toulouse, France, May 2004, *Sophie Tarbouriech*
Albuquerque, USA, May 2004, *Chaouki T. Abdallah*
Knoxville, USA, May 2004, *John N. Chiasson*

# Contents

# Control of Communication Networks Using Infinitesimal Perturbation Analysis of Stochastic Fluid Models

Christos Panayiotou[1], Christos G. Cassandras[2], Gang Sun[3], and Yorai Wardi[4]

[1] Dept. of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus, christosp@ucy.ac.cy
[2] Dept. of Manufacturing Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446, cgc@bu.edu
[3] Dept. of Manufacturing Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446, gsun@bu.edu
[4] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, wardi@ee.gatech.edu

## 1 Introduction

Managing and operating large scale communication networks is a challenging task and it is only expected to get worse as networks grow larger. The difficulties associated with network management stem from the fact that modeling and analysis of large scale communication networks is an excessively difficult task. On one hand, the enormous traffic volume in today's Internet makes packet-by-packet analysis infeasible. On the other hand, queueing systems (the natural modeling framework for packet-based communication networks) are largely based on Poisson processes and does not capture the bursty nature of realistic traffic. Moreover, the discovery of self-similar patterns in the Internet traffic distribution [1] and the resulting inadequacies of Poisson traffic models [2] further complicate queueing analysis. At the same time we need to account for the fact that the stochastic processes involved are time-varying, i.e., no stationarity assumptions hold. In addition, we need to explicitly model buffer overflow phenomena which typically defy tractable analytical derivations. Consequently, performance analysis techniques that do not depend on detailed traffic distributional information are highly desirable.

An alternative modeling paradigm based on fluid models has become increasingly attractive. The argument leading to the popularity of fluid models is that random phenomena may play different roles at different time scales. When the variations on the faster time scale have less impact than those on the slower time scale, the use of fluid models is justified. The efficiency of a fluid model rests on its ability to aggregate multiple events. By ignoring the micro-dynamics of each discrete entity and focusing on the change of the aggregated flow rate instead, a fluid model allows

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 1–26, 2004.

the aggregation of events associated with the movement of multiple packets within a time period of a constant flow rate into a single rate change event. Introduced in [3] and later proposed in [4] for the analysis of multiplexed data streams and network performance [5], fluid models have been shown to be especially useful for simulating various kinds of high speed networks [6, 7, 8, 9]. A *Stochastic* Flow Model (SFM) has the extra feature that the flow rates are treated as general stochastic processes, which distinguishes itself from the approach adopted in [10, 11, 12] that deal with deterministic or Markov modulated fluid rates.

On the other hand, the fluid modeling paradigm forgoes the identity and dynamics of individual packets and focuses instead on the aggregate flow rate. As a result, this paradigm is more suitable for network-related measures, such as buffer levels and packet loss volumes, rather than packet-related measures such as sojourn times (although it is still possible to define fluid-based sojourn times [13]). A Quality of Service (QoS) metric that depends on the identity of certain packets, for example, cannot be obviously captured by a fluid model. Furthermore, for the purpose of performance analysis of networks with QoS requirements, the accuracy of SFMs depends on traffic conditions, the structure of the underlying system, and the nature of the performance metrics of interest. Moreover, some metrics may depend on higher-order statistics of the distributions of the underlying random variables involved, which a fluid model may not be able to accurately capture.

In this chapter, our goal is to explore the use of SFMs for the purpose of *control and optimization* rather than *performance analysis*. In this case, it is not unreasonable to expect that one can identify the solution of an optimization problem based on a model which captures only those features of the underlying "real" system that are needed to lead to the right solution, without the need to estimate the corresponding optimal performance with accuracy. Even if the exact solution cannot be obtained by such "lower-resolution" models, one can still obtain near-optimal points that exhibit robustness with respect to certain aspects of the model they are based on. Such observations have been made in several contexts (e.g., [14]), including recent results related to SFMs reported in [15] where a connection between the SFM and queueing-system-based solution is established for various optimization problems in queueing systems.

Using the SFM modeling framework, a new approach for network management is being developed which is based on Infinitesimal Perturbation Analysis (IPA) [16, 17, 18, 19] (IPA is covered in detail in [20, 21]). In this approach, we estimate the gradient of the performance measure of interest (e.g., packet loss rate) with respect to the control parameters of interest (e.g., buffer thresholds) and use them in standard stochastic approximation algorithms to determine the optimal parameter setting. This approach has some very important advantages.

- The gradient estimation is done *on-line* thus the approach can be implemented on the real system: as the operating conditions change, it will aim at *continuously* seeking to optimize a generally time-varying performance metric.
- The gradient estimation process does not require any knowledge of the system's underlying stochastic processes; in other words, it is model free.

- The estimators are shown to be unbiased when evaluated based on SFM sample paths[5].
- It turns out that the estimators consist only of accumulators and timers and are generally easy to implement.

It is also worth pointing out that, even though the estimators are derived based on a SFM, their simplicity allows us to evaluate them based on the sample paths of *discrete-event systems* . Furthermore, simulation results indicate that such an approach works nicely, although the SFM-based estimators evaluated based on discrete event sample paths may no longer be unbiased. On-line management is appealing in today's computer networks and will become even more important as high speed network technologies become popular. In such cases, huge amounts of resources may suddenly become available or unavailable. Since manually managing network resources has become unrealistic, it is critical for network components, i.e., routers and end hosts, to automatically adapt to rapidly changing conditions.

This chapter consists of a tutorial on some of the main results that have appeared in the literature of IPA of SFMs. Section 2 presents an overview of the general methodology employed when analyzing such systems. The subsequent sections present some specific results on different important models. Section 3 analyzes a single node with a single class of fluid. Section 4 extends the analysis to multiple classes of fluid, while Section 5 presents a series of nodes but with a single class of fluid. Subsequently, Section 6 presents some simulations examples and finally Section 7 concludes and presents plans for future directions.

## 2 General Methodology

The basic SFM, used in this chapter follows the ones described in [13, 16, 17, 19, 23] where the system is characterized by a number of stochastic processes, all defined on a common probability space $(\Omega, \mathcal{F}, P)$. In general, all stochastic processes are classified as *defining* or *derived*; *Defining processes* are all *external* inflow processes (typically denoted by $\{\alpha(t;\theta)\}$) and all service processes (denoted by $\{\beta(t;\theta)\}$) where $\theta$ is some controllable parameter. *Derived processes* are the ones that result from the defining processes, the system dynamics and the controllable parameters ($\theta$); examples of such processes are the buffer outflow (denoted by $\{\delta(t;\theta)\}$), buffer occupancy ($\{x(t;\theta)\}$) and fluid overflow ($\{\gamma(t;\theta)\}$). Examples of these processes are shown in Fig. 1 for the single node system. How the derived processes are derived from the defining processes is considered in detail in the following sections.

The main motivation behind this research is the optimization of some cost functions of the form

$$J(\theta;\mathbf{x}(0),T) = \mathbb{E}\left[\mathcal{L}(\theta;\mathbf{x}(0),T)\right]$$

where, $\theta$ constitutes the controllable parameter (possibly a vector of parameters) and $\mathcal{L}(\theta;\mathbf{x}(0),T)$ is some sample function of interest evaluated in the interval $[0,T]$ when

[5] This is a desirable property that allows us to reliably use them with stochastic optimization algorithms e.g., [22].

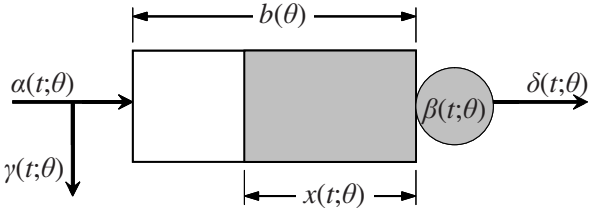



**Fig. 1.** Basic Stochastic Fluid Model (SFM)

the initial conditions are $\mathbf{x}(0)$. Loss volume, loss probability, average workload, and throughput are some of the cost functions that can be addressed in this approach. However, to limit the length of the chapter, in the sequel we only address the loss volume ($L(\theta;\mathbf{x}(0),T)$) and average workload ($Q(\theta;\mathbf{x}(0),T)$) which will be explicitly defined in the following sections. Note that from the workload metric it is possible to obtain a delay metric using appropriate forms of Little's law (e.g., see [13]). The general solution approach to the optimization problem above adopted in this chapter consists of three main steps which are briefly described in the following subsections.

### 2.1 Stochastic Approximation Algorithm

It is generally difficult (if at all possible) to obtain closed form expressions for $J(\theta;\mathbf{x}(0),T)$. Therefore, one needs to resort to iterative methods such as stochastic approximation algorithms (e.g., [22]) which are driven by estimates of the gradient of a cost function with respect to the parameters of interest. In the case of the cost minimization problem above, we are interested in estimating $dJ/d\theta$ based on directly observed (or simulated) data. We can then seek to obtain $\theta^*$ such that it minimizes $J(\theta;\mathbf{x}(0),T)$ through an iterative scheme of the form

$$\theta_{n+1} = \theta_n - \sigma_n H_n(\theta_n;\mathbf{x}(0),T,\omega_n^{SFM}), \quad n = 0,1,\cdots \tag{1}$$

where $H_n(\theta_n;\mathbf{x}(0),T,\omega_n^{SFM})$ is an estimate of $dJ/d\theta$ evaluated at $\theta = \theta_n$ and based on information obtained from a sample path of the SFM denoted by $\omega_n^{SFM}$. Furthermore, $\{\sigma_n\}$ is an appropriate sequence of step sizes. To simplify the notation, in the sequel we assume that $\mathbf{x}(0) = \mathbf{0}$ and we will omit the initial condition, the observation interval $T$ and the sample point $\omega \in \Omega$ unless it is necessary to stress the dependence. However, we emphasize that *all performance measures of interest are evaluated over a finite interval* $[0,T]$. Infinitesimal Perturbation Analysis (IPA) is used to obtain sample derivatives $d\mathcal{L}/d\theta$. These derivatives can be used in (1) if they are *unbiased* estimators of $dJ/d\theta$. The derivation of such estimators and their unbiasedness properties are addressed next.

### 2.2 IPA Derivative Estimates

In this section we show the general process for employing Infinitesimal Perturbation Analysis (IPA) [20, 21] to derive the sample derivatives $\mathcal{L}'(\theta) = d\mathcal{L}(\theta)/d\theta$. Before

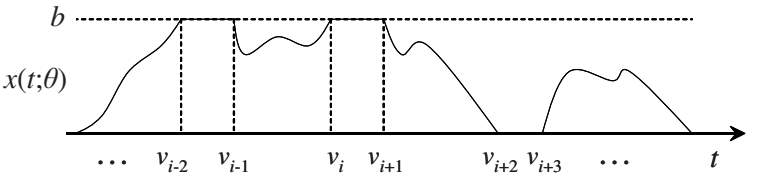



**Fig. 2.** Typical sample path of a single SFM node

we present the general derivation approach, we define the notion of *events* along a sample path which indicate changes in either the defining or the derived processes. Of particular interest are what we refer to as *exogenous* and *endogenous* events (exogenous events refer to changes in the input (defining) processes and endogenous events refer to changes in the output (derived) processes). The precise definition of these events depends on the nature of the model under investigation. In general, however, for a given fixed $\theta \in \Theta$, an *exogenous event* coincides with a point where the *net inflow* (inflow minus outflow) to certain buffers changes sign either in a continuous fashion or due to a discontinuity of the defining processes ($\alpha(t;\theta)$ and $\beta(t;\theta)$). *Endogenous events* correspond to points where some buffer becomes either full or empty or points where the buffer content crosses certain thresholds. Fig. 2 shows a typical sample path of the buffer content of a single SFM node with buffer capacity $b$. The sequence $\{v_i : i = 1,2,\cdots\}$ indicates examples of such events; $v_{i-2}$ and $v_{i+2}$ indicate the events *buffer becomes full* and *buffer becomes empty* respectively while, $v_{i+1}$ and $v_{i+3}$ indicate the events *buffer ceases to be full* and *buffer ceases to be empty* respectively. The latter two are exogenous events since they occur due to changes in the input processes, while the first two are endogenous events since their occurrence depends on the system dynamics (the detailed dynamics will be given in (2)).

**Corollary 1.** *Exogenous events are independent of the control parameters* $\theta$*, in other words, the event time derivatives of exogenous events are equal to zero.*

The performance measures of interest are generally functions of the derived processes. Thus, obtaining sample derivatives involves the differentiation of these processes with respect to the control parameter $\theta$ which can be done if the sample paths are segmented into smaller intervals. It turns out that a "convenient" segmentation is to divide the sample path at points where some of the exogenous or endogenous events occur. For example, in Fig. 2 the segments $s_i = [v_i, v_{i+1})$, $i = 0,\cdots,N_T$ constitute such a sample path segmentation where in the observation interval $[0,T]$ there are $N_T < \infty$ (w.p. 1) such segments. Such segmentation makes the sample function differentiation easier, but the result is generally an iterative algorithm that determines the required sample derivatives during segment $s_i$ given the sample derivatives of segment $s_{i-1}$. These iterative algorithms are usually sufficient to numerically evaluate the derivatives of the sample functions of interest. However, we point out that for certain systems it is possible to derive closed-form expressions for these derivatives (examples of such derivations will be given in the following sections where specific models will be investigated).

Once the sample function derivatives $d\mathcal{L}/d\theta$ are obtained we need to investigate whether they are unbiased estimates of the required $dJ/d\theta$ so they can be used in (1). This is done in the next section.

### 2.3 Unbiasedness

In this section we address the *unbiasedness* properties of the IPA estimators obtained above. As IPA estimator is *unbiased* if the following holds

$$\frac{dJ(\theta)}{d\theta} = \frac{d\mathbb{E}[\mathcal{L}(\theta)]}{d\theta} = \mathbb{E}\left[\frac{d\mathcal{L}(\theta)}{d\theta}\right] = \mathbb{E}\left[\mathcal{L}'(\theta)\right].$$

As mentioned earlier, this result allows us to use the derived IPA estimates in the stochastic approximation algorithm (1). In general, the unbiasedness of an IPA derivative $\mathcal{L}'(\theta)$ has been shown to be ensured by the following two conditions (see [24], Lemma A2, p.70):

**Condition 1.**

*a*. For every $\theta \in \Theta$ (where $\Theta$ is a closed bounded set), the sample derivative $\mathcal{L}'(\theta)$ exists w.p.1.

*b*. W.p.1, the random function $\mathcal{L}(\theta)$ is Lipschitz continuous throughout $\Theta$, and the (generally random) Lipschitz constant has a finite first moment.

The existence of the sample derivatives studied in this chapter is guaranteed by Assumption 1 shown below.

**Assumption 1.**

*a*. W.p.1, all *defining processes* (e.g., arrival and service rate functions $\alpha(t) \geq 0$ and $\beta(t) \geq 0$) are piecewise analytic in the interval $[0,T]$.

*b*. For every $\theta \in \Theta$, w.p. 1, two events cannot occur at exactly the same time. An exception is allowed for pairs of events such that the occurrence of one forces the immediate occurrence of the other.

*c*. W.p.1, no two processes $\{\alpha(t)\}$ or $\{\beta(t)\}$, have identical values during any open subinterval of $[0,T]$.

All three parts of **Assumption 1** are mild technical conditions which hold for all problems considered in the sequel. Regarding parts *b* and *c*, we point out that even if they do not hold, it is possible to use one-sided derivatives and still carry out similar analysis. However, in order to keep the analysis and notation manageable we impose these conditions.

Consequently, establishing the unbiasedness of $\mathcal{L}'(\theta)$, reduces to verifying the Lipschitz continuity of the sample function $\mathcal{L}(\theta)$ with appropriate Lipschitz constants. For all estimators presented in the sequel this has been established. The unbiasedness proofs are rather tedious and the interested reader is referred to the appropriate reference for the details. Next, we apply the three steps briefly described above for controlling system parameters such as the buffer thresholds and the server processing rates for different systems.

## 2.4 More Notation and Preliminaries

In this section we define some of the concepts and quantities that we will use in the sequel.

### Boundary and Non-Boundary Periods ($B_k$, $\bar{B}_k$)

These define a possible partition of a sample path. Boundary periods are maximal intervals where the buffer level $x(t;\theta)$ is constant and equal to some boundary or some threshold (i.e., $dx(t;\theta)/dt = 0$). Equivalently, non-boundary periods are intervals such that $x(t;\theta)$ is not on a boundary. In Fig. 2, $[v_i, v_{i+1})$ and $[v_{i+2}, v_{i+3})$ are boundary intervals while $[v_{i-1}, v_i)$ and $[v_{i+1}, v_{i+2})$ are non-boundary intervals. Also, with $N_B$ and $N_{\bar{B}}$ we denote the random number of boundary and non-boundary periods respectively observed in the interval $[0, T]$.

### Resetting Cycle ($\mathcal{C}_k$)

A non-boundary period followed by a boundary period forms a *resetting cycle* where the evaluation of event time derivatives is independent of the past history. However, one should not confuse the concept of *resetting cycle* with that of *regenerating cycle* because the evolution of the stochastic process itself might not always be independent of its past history. Furthermore, the $k$th resetting cycle $\mathcal{C}_k$ includes $R_k + 1$ events, that is $\mathcal{C}_k = [v_{k,0}, v_{k,R_k})$, where $v_{k,j}$ correspond to $v_i$ defined earlier but re-indexed based on the resetting cycle they belong to. In Fig. 2 the intervals $[v_{i-1}, v_{i+1})$ and $[v_{i+1}, v_{i+3})$ correspond to resetting cycles each of which includes 3 events. In addition, with $N_{\mathcal{C}}$ we denote the random number of resetting cycles in the interval $[0, T]$.

### Empty and Non-Empty Periods ($\mathcal{E}_k$, $\bar{\mathcal{E}}_k$)

Another sample path partitioning due to queueing theory is into *busy* and *idle* periods. Busy periods are periods where a buffer is not empty and idle otherwise. When using SFMs however, it is possible to have an empty buffer, while the server is not idle (e.g., when the inflow is less than the maximum outflow), so, we prefer to use the more appropriate terms of *empty* and *non-empty* periods. In Fig. 2 the interval $[v_{i-3}, v_{i+2})$ corresponds to a non-empty period while the interval $[v_{i+2}, v_{i+3})$ corresponds to an empty period. The $k$th non-empty period includes $S_k + 1$ events, thus $\mathcal{E}_k = [v_{k,0}, v_{k,S_k})$, where again $v_{k,j}$ are re-indexed based on the non-empty period they belong to. Also, with $N_{\mathcal{E}}$ and $N_{\bar{\mathcal{E}}}$ we denote the random number of empty and non-empty periods respectively observed in the interval $[0, T]$.

### The Prime Notation ($\cdot'$)

In the sequel we use the prime notation to indicate the derivative with respect to the control parameter of interest (typically either $\theta$ or $\rho$). For example, $x'(t;\theta)$ indicates the derivative of $x(t;\theta)$ with respect to $\theta$ and $v_i'$ indicates the derivative of the event time $v_i$ again with respect to $\theta$.

## 3 Single-Class Single-Node System

In this section we investigate the single-class single-node system shown in Fig 1. We assume that the system inflow is $\alpha(t)$, the maximum outflow is $\rho\beta(t)$ and the buffer size is $\theta$ where, $\rho$ and $\theta$ are the controllable parameters of interest. The parameter $\rho \in [0,1]$ denotes the proportion of the server capacity (i.e., $\beta(t)$) allocated to the specific queue by the resource scheduler. The processes $\alpha(t)$ and $\beta(t)$ are independent of both parameters $\theta$ and $\rho$. The system dynamics are given by

$$\frac{dx(t;\theta,\rho)}{dt^+} = \begin{cases} 0 & \text{if } x(t;\theta,\rho) = 0 \text{ and } \alpha(t) - \rho\beta(t) < 0 \\ 0 & \text{if } x(t;\theta,\rho) = \theta \text{ and } \alpha(t) - \rho\beta(t) > 0 \\ \alpha(t) - \rho\beta(t) & \text{otherwise} \end{cases} \tag{2}$$

The performance measures of interest are the average workload $Q(\theta,\rho)$ and the loss volume $L(\theta,\rho)$ defined in (3) and (4) respectively.

$$Q(\theta,\rho) = \int_0^T x(t;\theta,\rho)dt \tag{3}$$

$$L(\theta,\rho) = \int_0^T \gamma(t;\theta,\rho)dt \tag{4}$$

Next we derive the sample function derivatives with respect to the parameters $\theta$ and $\rho$. The IPA derivation can be done using either the resetting cycles or empty and non-empty periods. In [16, 19] the derivation is done using empty and non-empty periods, so here we present an alternative analysis based on the resetting cycles. For this system, any resetting cycle (say the $k$th one) consists of two periods $[v_{k,0}, v_{k,1})$ where the system will be in a non-boundary period and $[v_{k,1}, v_{k,2})$ where the system will be in a boundary period (i.e., $R_k = 2$ for all $k$). Using this sample path partitioning, we can rewrite the objectives as:

$$Q(\theta,\rho) = \sum_{k=1}^{N_C} q_k(\theta,\rho) = \sum_{k=1}^{N_C} \int_{v_{k,0}}^{v_{k,2}} x(t;\theta,\rho)dt \tag{5}$$

$$L(\theta,\rho) = \sum_{k=1}^{N_C} \int_{v_{k,1}}^{v_{k,2}} \gamma(t;\theta,\rho)dt \tag{6}$$

where, as mentioned earlier, $N_C$ is the random number of such cycles that appear in the interval $[0,T]$ and $q_k(\cdot) = \int_{v_{k,0}}^{v_{k,2}} x(\cdot)dt$, $k = 0,1,\cdots$.

### 3.1 IPA Derivative with respect to $\theta$

In this section we assume $\rho = 1$ constant (so it is omitted from all expressions), and derive $Q'(\theta) = dQ(\theta)/d\theta$ and $L'(\theta) = dL(\theta)/d\theta$. Differentiating (5) with respect to $\theta$ we obtain

$$Q'(\theta) = \sum_{k=1}^{N_C} q_k'(\theta) = \sum_{k=1}^{N_C} \int_{v_{k,0}}^{v_{k,2}} x'(t;\theta)dt \tag{7}$$

where we used the fact that cycle beginning and ending points $v_0$ and $v_2$ respectively are independent of $\theta$ since they correspond to exogenous events and thus $v_0' = v_2' = 0$ (Corollary 1). The loss volume derivative is given by

$$L'(\theta) = \sum_{k=1}^{N_C} \left[ -\gamma(v_{k,1};\theta) v_{k,1}' + \int_{v_{k,1}}^{v_{k,2}} \gamma'(t;\theta) dt \right].$$

Furthermore, the loss rate during any boundary period is

$$\gamma(t;\theta) = \begin{cases} 0 & \text{if } x(t;\theta) = 0 \\ \alpha(t) - \beta(t) & \text{if } x(t;\theta) = \theta \end{cases}, \; t \in [v_{k,1}, v_{k,2}), \; k = 1, \cdots, N_C.$$

Both cases are independent of $\theta$, therefore, $\gamma'(t;\theta) = 0$ and thus the above derivative simplifies to

$$L'(\theta) = -\sum_{k=1}^{N_C} \gamma(v_{k,1};\theta) v_{k,1}' \tag{8}$$

Next, we derive $q_k'(\theta)$ and $v_{1,k}'$ by analyzing each cycle independently. Before we proceed with the four possible cases, we recognize that during a non-boundary period the buffer content goes from one boundary to another. Therefore,

$$\theta \mathbf{1}[x(v_{k,0}) = \theta] + \int_{v_{k,0}}^{v_{k,1}} (\alpha(t) - \beta(t)) dt = \theta \mathbf{1}[x(v_{k,1}) = \theta]$$

Differentiating both sides we get

$$(\alpha(v_{k,1}) - \beta(v_{k,1})) v_{k,1}' = \mathbf{1}[x(v_{k,1}) = \theta] - \mathbf{1}[x(v_{k,0}) = \theta] \tag{9}$$

Now, depending on the boundary where the cycle starts and ends (empty (E) or full (F)), we identify four possible cases.

Case EE: *The cycle starts and ends with an empty period*. In this case,

$$x(t;\theta) = \begin{cases} \int_{v_{k,0}}^{t} (\alpha(t) - \beta(t)) dt & \text{for } t \in [v_{k,0}, v_{k,1}) \\ 0 & \text{for } t \in [v_{k,1}, v_{k,2}) \end{cases}$$

and $x'(t;\theta) = 0$ for all $t \in \mathcal{C}$. Also, $\gamma(t;\theta) = 0$ for all $t \in \mathcal{C}$ (no loss during the cycle), thus $\gamma(v_{k,1};\theta) = 0$. As a result

$$q_k'(\theta) = 0 \;\text{ and }\; \gamma(v_{k,1};\theta) v_{k,1}' = 0 \tag{10}$$

Case EF: *The cycle starts with an empty and ends with a full period*. Thus,

$$x(t;\theta) = \begin{cases} \int_{v_{k,0}}^{t} (\alpha(t) - \beta(t)) dt & \text{for } t \in [v_{k,0}, v_{k,1}) \\ \theta & \text{for } t \in [v_{k,1}, v_{k,2}) \end{cases}.$$

Differentiating with respect to $\theta$ we get,

$$x'(t;\theta) = \begin{cases} 0, & \text{for } t \in [v_{k,0}, v_{k,1}) \\ 1, & \text{for } t \in [v_{k,1}, v_{k,2}) \end{cases}.$$

Substituting into (7) we get, $q'_k(\theta) = v_{k,2} - v_{k,1}$. For the loss volume, $\gamma(v_{k,1},\theta) = \alpha(v_{k,1}) - \beta(v_{k,1})$ and, from (9), we get that $(\alpha(v_{k,1}) - \beta(v_{k,1}))v'_{k,1} = 1$, therefore

$$q'_k(\theta) = v_{k,2} - v_{k,1} \text{ and } \gamma(v_{k,1};\theta)v'_{k,1} = 1. \tag{11}$$

Case FF: *The cycle starts and ends with a full period*. In this case,

$$x(t;\theta) = \begin{cases} \theta + \int_{v_{k,0}}^{t} (\alpha(t) - \beta(t))\,dt & \text{for } t \in [v_{k,0}, v_{k,1}) \\ \theta & \text{for } t \in [v_{k,1}, v_{k,2}) \end{cases}$$

therefore, $x'(t;\theta) = 1$ for the entire cycle. As a result, from (7), $q'_k(\theta) = v_{k,2} - v_{k,0}$. In addition, from (9) we get that $v'_{k,1} = 0$ and therefore

$$q'_k(\theta) = v_{k,2} - v_{k,0} \text{ and } \gamma(v_{k,1};\theta)v'_{k,1} = 0. \tag{12}$$

Case FE: *The cycle starts with a full and ends with an empty period*. In this case,

$$x(t;\theta) = \begin{cases} \theta + \int_{v_{k,0}}^{t} (\alpha(t) - \beta(t))\,dt & \text{for } t \in [v_{k,0}, v_{k,1}) \\ 0 & \text{for } t \in [v_{k,1}, v_{k,2}) \end{cases}$$

and therefore $x'(t;\theta) = 1$ for $t \in [v_{k,0}, v_{k,1})$ and 0 for $t \in [v_{k,1}, v_{k,2})$. As a result, from (7) $q'_k(\theta) = v_{k,1} - v_{k,0}$. For the loss volume, $x(v_{k,1};\theta) = 0$ therefore $\gamma(v_{k,1};\theta) = 0$. Hence,

$$q'_k(\theta) = v_{k,1} - v_{k,0} \text{ and } \gamma(v_{k,1};\theta)v'_{k,1} = 0. \tag{13}$$

**Theorem 1.** *The sample derivatives $Q'(\theta)$ and $L'(\theta)$ with respect to $\theta$ are*

$$Q'(\theta) = \sum_{k=1}^{N_{\mathcal{C}}} [(v_{k,2} - v_{k,1})\mathbf{1}_{EF} + (v_{k,2} - v_{k,0})\mathbf{1}_{FF} + (v_{k,1} - v_{k,0})\mathbf{1}_{FE}]$$
$$L'(\theta) = -N_{\mathcal{C}_{EF}}$$

*where $N_{\mathcal{C}_{EF}}$ is the number of EF cycles that were observed during $[0,T]$ and $\mathbf{1}_{yy}$ is the usual indicator function that takes the value of 1 if the cycle is of the yy type and 0 otherwise.*

**Proof:** Follows immediately from (10)-(13). ∎

**Corollary 2.** *The estimators of Theorem 1 are precisely the estimators obtained in [16], in other words,*

$$Q'(\theta) = \sum_{j=1}^{N_{\bar{\mathcal{E}}}} (v_{j,S_j} - v_{j,1}) \quad \text{and} \quad L'(\theta) = -N_{\mathcal{C}_{EF}}$$

*where $j$ counts the number of non-empty periods (not cycles), $v_{j,1}$ indicates the first overflow point of the $j$th non-empty period and $v_{j,S_j}$ indicates the end of the $j$th non-empty period. If no overflow occurs $v_{j,1} = v_{j,S_j}$.*

**Proof:** Follows by recognizing that the union of a non-empty period with the following empty period consists either of a single $EE$ cycle or it starts with an $EF$ cycle, followed by $m$ $FF$ cycles $m = 0, 1, 2, \cdots$, and ends with an $FE$ cycle. Furthermore, note that the number of non-empty periods with at least some loss is equal to the number of $EF$ cycles $N_{\mathcal{C}_{EF}}$. ∎

In [16] it is also shown that the above estimators are *unbiased*. Furthermore, we point out that the implementation of the above estimators is extremely simple; they simply accumulate the time between certain events, or they count the number of $EF$ cycles.

### 3.2 IPA Derivative with respect to ρ

In this section, we assume $\theta = b$ constant (so it is omitted from all expressions), and derive $Q'(\rho) = dQ(\rho)/d\rho$ and $L'(\rho) = dL(\rho)/d\rho$. Differentiating (5) with respect to ρ we obtain

$$Q'(\rho) = \sum_{k=1}^{N_{\mathcal{C}}} q'_k(\rho) = \sum_{k=1}^{N_{\mathcal{C}}} \left[ x(v_{k,2};\rho)v'_{k,2} - x(v_{k,0};\rho)v'_{k,0} + \int_{v_{k,0}}^{v_{k,2}} x'(t;\rho)dt \right] \tag{14}$$

In this case, the cycle beginning and ending points $v_{k,0}$ and $v_{k,2}$ respectively are not always due to exogenous events. The events *buffer ceases to be empty or full* occur at points where a sign change of $\alpha(t) - \rho\beta(t)$ occurs. The sign change can be due to a jump (discontinuity) in either $\alpha(t)$ or $\beta(t)$ which corresponds to exogenous events, or it can occur in a continuous fashion, thus the switching time depends on ρ. Nevertheless, the following result holds.

**Lemma 1.** *At the cycle beginning and ending points $v_{k,0}$ and $v_{k,2}$ respectively,*

$$(\alpha(v_{k,0}) - \rho\beta(v_{k,0}))\, v'_{k,0} = (\alpha(v_{k,2}) - \rho\beta(v_{k,2}))\, v'_{k,2} = 0$$

**Proof:** We make the argument only for the beginning point of $\mathcal{C}_k$; the argument for the ending point is the same since it coincides with the beginning point of $\mathcal{C}_{k+1}$. As mentioned above, $v_{k,0}$ is due to a sign change of $\alpha(t) - \rho\beta(t)$ from positive to negative or vise versa. This can happen either due to a discontinuity in the processes $\alpha(t)$ and $\beta(t)$ or it can happen in a continuous fashion. If this happens due to a discontinuity of either $\alpha(t)$ or $\beta(t)$, then it corresponds to an exogenous event, thus $v'_{k,0} = 0$. If on the other hand, the sign change occurs in a continuous fashion, $\alpha(v_{k,0}) - \rho\beta(v_{k,0}) = 0$. Hence, the lemma is proved. ∎

Next, note that the buffer content is given by

$$x(t;\rho) = \begin{cases} b\mathbf{1}[x(v_{k,0};\rho) = b] + \int_{v_{k,0}}^{t} (\alpha(t) - \rho\beta(t))\, dt & \text{for } t \in [v_{k,0}, v_{k,1}) \\ b\mathbf{1}[x(v_{k,1};\rho) = b] & \text{for } t \in [v_{k,1}, v_{k,2}) \end{cases} \tag{15}$$

therefore, using Lemma 1, the derivative with respect to ρ is given by

$$x'(t;\rho) = \begin{cases} -\bar{\beta}(t, v_{k,0}) & \text{for } t \in [v_{k,0}, v_{k,1}) \\ 0 & \text{for } t \in [v_{k,1}, v_{k,2}) \end{cases}. \tag{16}$$

where

$$\bar{\beta}(t_2,t_1) = \int_{t_1}^{t_2} \beta(t)dt.$$

For the general case, the above integral can be evaluated numerically, however, to simplify the analysis, in the sequel we assume the $\beta(t) = \bar{\beta}$ constant, and thus $\bar{\beta}(t_2,t_1) = \bar{\beta}(t_2 - t_1)$. Therefore, the last term of (14) simplifies to

$$\begin{aligned}\int_{v_{k,0}}^{v_{k,2}} x'(t;\rho)dt &= -\int_{v_{k,0}}^{v_{k,1}} \bar{\beta}(t - v_{k,0})dt \\ &= -\frac{\bar{\beta}}{2}\left(v_{k,1}^2 - 2v_{k,1}v_{k,0} - v_{k,0}^2 + 2v_{k,0}^2\right) \\ &= -\frac{\bar{\beta}}{2}\left(v_{k,1} - v_{k,0}\right)^2.\end{aligned}$$

Substituting in (14), each term of the workload derivative simplifies to

$$q'_k(\rho) = x(v_{k,2};\rho)v'_{k,2} - x(v_{k,0};\rho)v'_{k,0} - \frac{\bar{\beta}}{2}\left(v_{k,1} - v_{k,0}\right)^2 \tag{17}$$

Similarly, the loss volume derivative is given by

$$L'(\rho) = \sum_{k=1}^{N_{\mathcal{C}}} \left[\gamma(v_{k,2};\rho)v'_{k,2} - \gamma(v_{k,1};\rho)v'_{k,1} + \int_{v_{k,1}}^{v_{k,2}} \gamma'(t;\rho)dt\right].$$

The loss rate during any boundary period is

$$\gamma(t;\rho) = \begin{cases} 0 & \text{if } x(t;\rho) = 0 \\ \alpha(t) - \rho\beta(t) & \text{if } x(t;\rho) = b \end{cases}, \; t \in [v_{k,1}, v_{k,2}), \; k = 1,\cdots,N_{\mathcal{C}}. \tag{18}$$

Therefore, $\gamma'(t;\rho) = -\beta(t)$ if $x(v_{k,1};\rho) = b$ and 0 otherwise. Also, from Lemma 1, the term $\gamma(v_{k,2};\rho)v'_{k,2} = (\alpha(v_{k,2}) - \rho\beta(v_{k,2}))v'_{k,2} = 0$. As a result, the above derivative simplifies to

$$L'(\rho) = \sum_{k=1}^{N_{\mathcal{C}}} \lambda'_k(\rho) = -\sum_{k=1}^{N_{\mathcal{C}}} \left[\gamma(v_{k,1};\rho)v'_{k,1} + \bar{\beta}(v_{k,2} - v_{k,1})\mathbf{1}[x(v_{k,1};\rho) = b]\right] \tag{19}$$

Next, we focus on a single cycle and derive $q'_k(\rho)$ and $\lambda'_k(\rho)$, but first we recognize that during any non-boundary period $x(t;\rho)$ goes from one boundary to another, thus

$$b\mathbf{1}[x(v_{k,0};\rho) = b] + \int_{v_{k,0}}^{v_{k,1}} \left(\alpha(t) - \rho\bar{\beta}\right)dt = b\mathbf{1}[x(v_{k,1};\rho) = b].$$

Differentiating both sides with respect to $\rho$, and using Lemma 1, we get

$$\left(\alpha(v_{k,1}) - \rho\bar{\beta}\right)v'_{k,1} - \bar{\beta}\left(v_{k,1} - v_{k,0}\right) = 0 \tag{20}$$

Depending on the boundary where the cycle starts and ends (empty (E) or full (F)), we identify four possible cases.

Case EE: *The cycle starts and ends with an empty period*. In this case, $x(v_{k,0};\rho) = x(v_{k,2};\rho) = 0$, and, from (17), $q'_k(\rho) = -\bar{\beta}\,(v_{k,1} - v_{k,0})^2/2$. Also, $\gamma(t;\rho) = 0$ for all $t \in \mathcal{C}_k$ (no loss during the cycle), thus, from (19) $\lambda'_k(\rho) = 0$. Summarizing,

$$q'_k(\rho) = -\frac{\bar{\beta}}{2}\,(v_{k,1} - v_{k,0})^2 \;\text{ and }\; \lambda'_k(\rho) = 0 \tag{21}$$

Case EF: *The cycle starts with an empty and ends with a full period*. In this case, $x(v_{k,0},\rho) = 0$ and $x(v_{k,2},\rho) = b$ so, in (17) the term $x(v_{k,0},\rho)v'_{k,0} = 0$. Next, from (20) we get,

$$\gamma(v_{k,1};\rho)v'_{k,1} = \left(\alpha(v_{k,1}) - \rho\bar{\beta}\right)v'_{k,1} = \bar{\beta}\,(v_{k,1} - v_{k,0})\,.$$

Substituting in (17) and (19), we get

$$q'_k(\rho) = bv'_{k,2} - \frac{\bar{\beta}}{2}\,(v_{k,1} - v_{k,0})^2 \;\text{ and }\; \lambda'_k(\rho) = -\bar{\beta}(v_{k,2} - v_{k,0}) \tag{22}$$

Case FF: *The cycle starts and ends with a full period*. In this case, $x(v_{k,0},\rho) = b$ and $x(v_{k,2},\rho) = b$. Using (20) we get

$$\gamma(v_{k,1};\rho)v'_{k,1} = \left(\alpha(v_{k,1}) - \rho\bar{\beta}\right)v'_{k,1} = \bar{\beta}\,(v_{k,1} - v_{k,0})\,,$$

and again, substituting in (17) and (19), we get

$$q'_k(\rho) = bv'_{k,2} - bv'_{k,0} - \frac{\bar{\beta}}{2}\,(v_{k,1} - v_{k,0})^2 \;\text{ and }\; \lambda'_k(\rho) = -\bar{\beta}(v_{k,2} - v_{k,0}) \tag{23}$$

Case FE: *The cycle starts with a full and ends with an empty period*. In this case, $x(v_{k,0},\rho) = b$ and $x(v_{k,1},\rho) = x(v_{k,2},\rho) = 0$, so that in (17) the term $x(v_{k,2},\rho)v'_{k,2} = 0$. Also, from (20) we arrive at

$$\gamma(v_{k,1};\rho)v'_{k,1} = \left(\alpha(v_{k,1}) - \rho\bar{\beta}\right)v'_{k,1} = \bar{\beta}\,(v_{k,1} - v_{k,0})\,,$$

and again, substituting in (17) and (19), we get

$$q'_k(\rho) = -bv'_{k,0} - \frac{\bar{\beta}}{2}\,(v_{k,1} - v_{k,0})^2 \;\text{ and }\; \lambda'_k(\rho) = -\bar{\beta}(v_{k,1} - v_{k,0}) \tag{24}$$

**Theorem 2.** *The sample derivatives of $Q(\rho)$ and $L(\rho)$ are given by*

$$Q'(\rho) = -\frac{\bar{\beta}}{2}\sum_{k=1}^{N_{\mathcal{C}}}(v_{k,1} - v_{k,0})^2$$

$$L'(\rho) = -\sum_{k=1}^{N_{\mathcal{C}}}\left[(v_{k,2} - v_{k,0})(\mathbf{1}_{EF} + \mathbf{1}_{FF}) + (v_{k,1} - v_{k,0})\mathbf{1}_{FE}\right]$$

**Proof:** The theorem follows by combining (21)-(24). For $Q'(\rho)$ note also that $v_{k,2} = v_{k+1,0}$ and that every $EF$ cycle is followed by either an $FF$ or an $FE$ cycle and every $FF$ is followed by another $FF$ or $FE$ cycle. As a result all terms of the form $bv'_{k,2}$ are cancelled by $bv'_{k+1,0}$. ∎

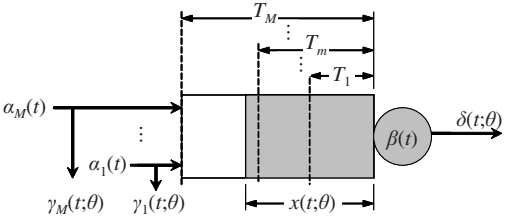



**Fig. 3.** M-Class Stochastic Fluid Model (SFM)

**Corollary 3.** *Theorem 2 gives precisely the estimators obtained in [19].*

We do not present the estimators explicitly since they require the introduction of some new notation. In any case, the proof of this corollary follows along the lines of the proof of Corollary 2. In [19] it is also shown that the above estimators are *unbiased*. Furthermore, we point out that the implementation of the above estimators is extremely simple; they simply accumulate the time between certain events. Finally, before we leave the single-node single-class flow model we mention that [19] also derives IPA estimators with respect to parameters of the arrival process $\alpha(t;\theta)$.

## 4 Multiple Classes

In this section, we enrich the modeling framework introduced earlier by introducing multiple classes of flows which are merged into a First Come First Serve (FIFO) buffer. The various classes of flows are differentiated according to a *Threshold Policy* [25] which works as follows: When fluid from class $m$ arrives, it is accepted in the buffer if the state $x(t;\theta) < T_m$, otherwise the fluid is rejected. Threshold $T_m$, $m$, $m = 1,...,M$, is associated with the class $m$ flow and we assume $0 = T_0 < T_1 < \cdots < T_m < \cdots < T_M$. In this way, class $M$ has the highest priority and class 1 the lowest priority. This policy has been shown to provide good protection to the higher priority classes [25] and was proposed for the Differentiated Services (DS) architecture [26]. The inflow rate of class $m$ at time $t$ is denoted by $\alpha_m(t)$ and the corresponding loss rate by $\gamma_m(t;\theta)$. As in the previous section, the service rate is denoted by $\beta(t)$, and $x(t;\theta)$ is the buffer level at time $t$. For the purpose of our analysis, we choose any one of the thresholds, say, $T_{\overline{m}}$, as the one with respect to which we wish to carry out sensitivity analysis and denote this parameter by $\theta$. We also assume that the processes $\{\alpha_m(t)\}$, $m = 1,...,M$, and $\{\beta(t)\}$ are independent of $\theta$ and **Assumption 1** holds. For notational economy we define:

$$A_m(t) \equiv \sum_{n=m}^{M} \alpha_n(t) - \beta(t), \quad m = 1,\cdots,M \tag{25}$$

and observe that $A_m(t) \geq A_{m+1}(t)$, $m = 1,...,M-1$.

Fig. 3 depicts the SFM described above. Furthermore, we assume that the parameter $\theta$ is confined to a bounded (compact) interval $\Theta = (T_{\overline{m}-1}, T_{\overline{m}+1})$. In what follows,

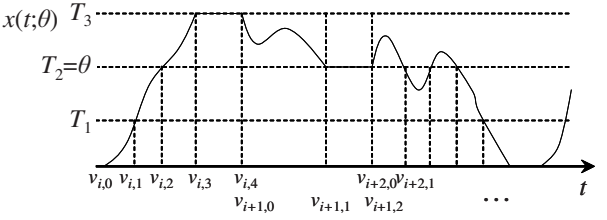



**Fig. 4.** Typical sample path segment ($M = 3$)

again we consider two performance metrics, the *Cumulative Workload*(or just *Work*) $Q(\theta)$ and the $m$th class *Loss Volume* $L_m(\theta)$, $m = 1,...,M$ defined as follows:

$$Q(\theta) = \int_0^T x(t;\theta)dt, \quad L_m(\theta) = \int_0^T \gamma_m(t;\theta)dt. \tag{26}$$

For this system, we identify the following two event types. Event type $e_1$ is an event where the buffer content *leaves* the value $x(t;\theta) = T_m$, for some $m = 0,\cdots,M$, after it has maintained it for some finite length of time. This is an exogenous event since it is caused by a sign change of $A_m(t)$ in (25) for some $m = 1,...,M$. Event type $e_2$ is defined to occur whenever the buffer content *reaches* or *crosses* the value $x(t;\theta) = T_m$, for any $m = 0,\cdots,M$. The interval between two consecutive events of type $e_1$ define a cycle (say $\mathcal{C}_k$). These events are assumed to occur at time instants $v_{k,0}$, $k = 1,\cdots,N_{\mathcal{C}}$ where $N_{\mathcal{C}}$ is the random number of such cycles in the interval $[0,T]$. During the $k$th cycle $\mathcal{C}_k$, we observe $R_k - 1$ endogenous ($e_2$) events, at time instants $v_{k,i}$ $i = 1,\cdots,R_k - 1$. (see Fig. 4 for examples of such events). Thus, the $\mathcal{C}_k$ cycle is divided into $R_k$ periods

$$p_{k,i} \equiv [v_{k,i}, v_{k,i+1}), \;\; i = 0,\cdots,R_{k-1}.$$

The corresponding open interval $(v_{k,i}, v_{k,i+1})$ is denoted by $p^o_{k,i}$. For the purpose of our analysis, we view each threshold as a *boundary*, thus any period $p_{k,i}$ where $x(t;\theta) = T_m$, $t \in p_{k,i}$, $m = 0,\cdots,M$, is considered as a *boundary period*. Otherwise, $p_{k,i}$ is a *non-boundary* period. We emphasize that each cycle ends with a boundary period $p_{k,R_k-1}$ and thus the cycle definition is consistent with the definition of Section 2. A typical sample path for the case of $M = 3$ with three cycles $\mathcal{C}_i$, $\mathcal{C}_{i+1}$ and $\mathcal{C}_{i+2}$ is shown in Fig. 4. In the figure, note also that $v_{i,0} = v_{i-1,R_i}$, $i = 1,\cdots,N_{\mathcal{C}}$.

During a boundary period $p_{k,R_k-1}$, the buffer content dynamics are

$$\frac{dx(t;\theta)}{dt^+} = 0. \tag{27}$$

On the other hand, during a non-boundary period $p_{k,i}$, $i = 0,...,R_k - 2$, if $T_{m-1} < x(t;\theta) < T_m$ for $t \in p^o_{k,i}$, the buffer content dynamics are

$$\frac{dx(t)}{dt^+} = A_m(t). \tag{28}$$

### 4.1 IPA with Respect to Thresholds

Our objective here is to estimate the sample derivatives $Q' = dQ(\theta)/d\theta$ and $L'_m = dL_m(\theta)/d\theta, m = 1,...,M$. We proceed by first evaluating the sample derivatives $Q'(\theta)$ and $L'_m(\theta)$ in terms of event time derivatives $v'_{k,i} = dv_{k,i}/d\theta$, and then provide an algorithm for evaluating these event time derivatives based on observable quantities along a given sample path. Again for notational convenience, similar to our definition of $A_m(t)$ in (25), let us define

$$A_{m,k,i} \equiv \sum_{n=m}^{M} \alpha_n(v_{k,i}) - \beta(v_{k,i}) \tag{29}$$

**Work IPA Derivative**

Using the sample path partition into cycles, we write (3) as

$$Q(\theta) = \sum_{k=1}^{N_C} q_k(\theta) = \sum_{k=1}^{N_C} \int_{v_{k,0}}^{v_{k,R_k}} x(t;\theta)dt. \tag{30}$$

where, as earlier, $q_k(\theta) = \int_{v_{k,0}}^{v_{k,R_k}} x(t;\theta)dt$. Thus,

$$Q'(\theta) = \sum_{k=1}^{N_C} q'_k(\theta) = \sum_{k=1}^{N_C} \frac{d}{d\theta} \int_{v_{k,0}}^{v_{k,R_k}} x(t;\theta)dt = \sum_{k=1}^{N_C} \int_{v_{k,0}}^{v_{k,R_k}} x'(t;\theta)dt \tag{31}$$

where we use the fact that $v_{k,0}$ and $v_{k,R_k}$ are independent of $\theta$ since they correspond to exogenous events.

**Theorem 3.** *The sample derivative of $Q(\theta)$ with respect to $\theta$ is given by*

$$Q'(\theta) = \sum_{k=1}^{N_C} \sum_{j=0}^{R_k-1} x'_{k,j} \left(v_{k,j+1} - v_{k,j}\right)$$

*where* $x'_{k,R_k-1} = \mathbf{1}\left[x(v_{k,R_k-1};\theta) = \theta\right]$ *and for* $j = 0,\cdots,R_k-2$

$$x'_{k,j} = \begin{cases} \mathbf{1}\left[x(v_{k,j};\theta) = \theta\right] - A_{m+1,k,j}v'_{k,j} & \text{if } \forall t \in p^o_{k,j}, T_m < x(t;\theta) < T_{m+1} \\ \mathbf{1}\left[x(v_{k,j};\theta) = \theta\right] - A_{m,k,j}v'_{k,j} & \text{if } \forall t \in p^o_{k,j}, T_{m-1} < x(t;\theta) < T_m \end{cases},$$

The proof follows easily from (31) by writing the expression of $x(t;\theta)$ in any interval $p_{k,j}$ and subsequently differentiating with respect to $\theta$ (see [23] for details). In [23] it is also shown that the estimator is unbiased. In order to evaluate $Q'(\theta)$ one simply needs to observe the net inflow rates $A_{m,k,i}$ at specific points in time and measure the intervals $(v_{k,j+1} - v_{k,j})$. In addition, one also needs the event time derivatives $v'_{k,j}$ which are determined in a subsequent subsection.

**Class $m$ Loss IPA Derivatives**

Again, using the sample path partition into cycles, we may write $L_m(\theta)$ from (26) as follows:

$$L_m(\theta) = \sum_{k=1}^{N_C} \lambda_{m,k}(\theta) = \sum_{k=1}^{N_C} \int_{v_{k,0}}^{v_{k,R_k}} \gamma_m(t;\theta)dt \tag{32}$$

For the purpose of our analysis, a useful way of grouping periods $p_{k,i}$, $k = 1,\ldots,N_C$ within a typical cycle is by defining sets associated with each class $m = 1,\cdots,M$ as follows:

Partial Loss Period Set $U_m$. For any $p_{k,i} \in U_m$, the buffer content is $x(t;\theta) = T_m$ for all $t \in p_{k,i}$, and class $m$ traffic experiences partial loss. In particular, the traffic flows satisfy

$$A_m(t) > 0 \text{ and } A_{m+1}(t) < 0 \tag{33}$$

so that the processing capacity $\beta(t)$ can accommodate the cumulative incoming flow $\sum_{n=m+1}^{M} \alpha_n(t)$ due to classes $m+1,\cdots,M$, but not the flow $\sum_{n=m}^{M} \alpha_n(t)$ that includes the next lower priority class $m$. In this case, the system accepts only the portion of the class $m$ traffic that can be accommodated and incurs a "partial" loss with rate $\gamma_m(t;\theta) = A_m(t)$. Formally, we define $U_m$ as follows:

$$U_m := \{p_{k,i} : x(t;\theta) = T_m,\ \ t \in p_{k,i}\}. \tag{34}$$

Note that the starting point $v_{k,i}$ of such a period corresponds to an endogenous event $e_2$, whereas the ending point $v_{k,i+1}$ corresponds to an exogenous event $e_1$ and is, therefore, locally independent of $\theta$. Also note that the elements of $U_m$ set are always the last interval of a cycle. In addition, the last interval of a cycle, $p_{k,R_k-1}$, if during it $x(t;\theta) \neq 0$, it must be an element of $U_m$ for a some $m \in \{1,\cdots,M\}$.

Full Loss Period Set $V_m$. During such periods, the buffer content is $x(t;\theta) > T_m$ (excluding the starting point $v_{k,i}$) and *all* class $m$ traffic is lost, i.e., $\gamma_m(t;\theta) = \alpha_m(t)$. Formally, we define $V_m$ as follows:

$$V_m := \{p_{k,i} : x(t;\theta) > T_m,\ \ t \in p^o_{k,i}\} \tag{35}$$

No Loss Period Set $W_m$. During such periods the buffer content is $x(t;\theta) < T_m$ (excluding the starting point $v_{k,i}$) and no class $m$ loss occurs, i.e., $\gamma_m(t;\theta) = 0$. Formally, we define $W_m$ as follows:

$$W_m := \{p_{k,i} : x(t) < T_m,\ \ t \in p^o_{k,i}\} \tag{36}$$

Note that each of the sets above is locally independent of $\theta$, and that for any $m$, $U_m \cup V_m \cup W_m = [0,T)$ with all sets being mutually exclusive.

**Theorem 4.** *The sample derivatives $L'_m(\theta)$, $m = 1,\cdots,M$ are given by*

$$L'_m(\theta) = \sum_{k=1}^{N_C} \lambda'_{m,k}(\theta)$$

*where*

$$\lambda'_{k,m}(\theta) = \sum_{j=0}^{R_k-1} \Big[\mathbf{1}\big[p_{k,j} \in V_m\big]\left(\alpha_m(v_{k,j+1})v'_{k,j+1} - \alpha_m(v_{k,j})v'_{k,j}\right) - \mathbf{1}\big[p_{k,R_k-1} \in U_m\big] \cdot A_{m,k,R_k-1} v'_{k,R_k-1}\Big].$$

Again the proof follows easily by differentiating (32) and writing the loss volume during each interval $p_{k,j}$. For details see [23] where it is also shown that the above estimators are unbiased. We also point out that as with $Q'(\theta)$, evaluating this estimator we only need some rates at specific points in time and the event time derivatives $v'_{k,j}$ which are determined next.

**Event Time Derivatives**

Theorems 3 and 4 provide estimators of the sample derivatives $Q'(\theta)$ and $L'(\theta)$ respectively. Both estimators require the event time derivatives $v'_{k,j}$, $j = 1, \cdots, R_k - 1$ and $k = 1, \cdots, N_{\mathcal{C}}$. As already discussed, the beginning and end points of $\mathcal{C}_k$ are independent of $\theta$ because they correspond to exogenous events ($e_1$), so $v'_{k,0} = v'_{k,R_k} = 0$. The following theorem provides an iterative algorithm for determining the event time derivative $v'_{k,i+1}$ given $v'_{k,i}$. To simplify the notation we also use

$$x_{k,i} = x(v_{i,k}; \theta)$$

**Theorem 5.** *The event time derivative of any endogenous event occurring at time $v_{i+1}$, $i = 0, \cdots, R_k - 1$, is given by*

$$v'_{k,i+1} = F_{k,i} \cdot v'_{k,i} + G_{k,i},$$

*where $F_{k,i}$ and $G_{k,i}$ and are given below*

$$F_{k,i} = \begin{cases} S_{k,i}\frac{A_{m+1,k,i}}{A_{m+1,k,i+1}} & \text{if } x_{k,i} = T_m \text{ and } x_{k,i+1} = T_{m+1} \\ S_{k,i}\frac{A_{m,k,i}}{A_{m,k,i+1}} & \text{if } x_{k,i} = T_m \text{ and } x_{k,i+1} = T_{m-1} \\ \frac{A_{m+1,k,i}}{A_{m+1,k,i+1}} & \text{if } x_{k,i} = x_{k,i+1} = T_m \text{ and } x(t;\theta) > T_m, t \in p^o_{k,i} \\ \frac{A_{m,k,i}}{A_{m,k,i+1}} & \text{if } x_{k,i} = x_{k,i+1} = T_m \text{ and } x(t;\theta) < T_m, t \in p^o_{k,i} \end{cases}$$

$$G_{k,i} = \begin{cases} -\frac{1}{A_{m+1,k,i+1}} & \text{if } x_{k,i} = T_m = \theta \text{ and } x_{k,i+1} = T_{m+1} \\ -\frac{1}{A_{m,k,i+1}} & \text{if } x_{k,i} = T_m = \theta \text{ and } x_{k,i+1} = T_{m-1} \\ \frac{1}{A_{m,k,i+1}} & \text{if } x_{k,i} = T_{m-1} \text{ and } x_{k,i+1} = T_m = \theta \\ \frac{1}{A_{m+1,k,i+1}} & \text{if } x_{k,i} = T_{m+1} \text{ and } x_{k,i+1} = T_m = \theta \\ 0 & \text{otherwise} \end{cases}$$

*where $S_{k,i} = 1$ if $x_{k,i+1} \neq \theta$ and $S_{k,i} = -1$ if $x_{k,i+1} = \theta$.*

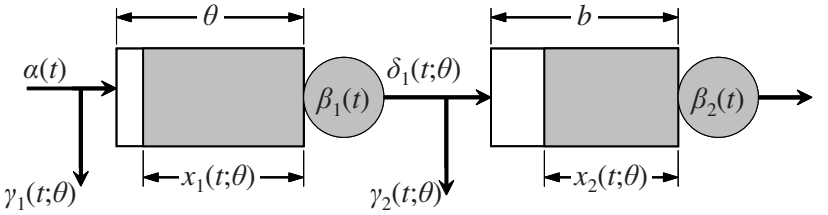



**Fig. 5.** Tandem two-node network

Recall that $A_{m,k,i}$ is given by (29). Also note that $G_{k,i} \neq 0$ only for the intervals $i$ that either start or end at threshold θ. For the proof the interested reader is referred to [23]. We point out that unlike the estimators of the previous section, where we were able to determine a closed-form expression, these estimators are evaluated using the iterative algorithm of Theorem 5. An exception is the two-class case, where again it is possible to obtain a closed-form expression (see [17] and [23]).

## 5 Tandem Networks

In this section, we describe how perturbations in a buffer threshold propagate in a network. For simplicity, we limit ourselves to the two-node network of Fig. 5; for the general $m$-node network the reader is referred to [27]. The control parameter of interest is θ, the buffer size of node 1, and we are interested in the sample derivatives of the workload and loss volume of node 2; recall that the corresponding sample derivatives for node 1 were obtained in Section 3.

For notational convenience, let as focus on node 2 and use $x(t;\theta) = x_2(t;\theta)$ and $y(t;\theta) = x_1(t;\theta)$. Also, let $\gamma(t;\theta) = \gamma_2(t;\theta)$ and define the net inflow rate to node 2

$$A(t;\theta) = \delta_1(t;\theta) - \beta_2(t) \tag{37}$$

where, $\delta_1(t;\theta)$ is the outflow from node 1 and is equal to $\alpha(t)$ if $y(t;\theta) = 0$ or $\beta_1(t)$ if $y(t;\theta) > 0$. Using this notation, the objective functions of interest are

$$Q(\theta) = \int_0^T x(t;\theta)dt, \quad \text{and} \quad L(\theta) = \int_0^T \gamma(t;\theta)dt \tag{38}$$

A typical sample path of this system is shown in Fig. 6. Following the practice of the previous sections, we again partition the sample path of $x(t;\theta)$ into boundary and non-boundary periods and form cycles consisting of a non-boundary period and the following boundary period. The $k$th non-boundary period starts at $v_{k,0}$ with either one of the events $x(t;\theta)$ *ceases to be empty* or $x(t;\theta)$ *ceases to be full*, and ends at $v_{k,r_k}$ with the events $x(t;\theta)$ *becomes either empty or full*. During this interval we also observe $r_k - 1$ buffer $y(t;\theta)$ *becomes empty* events. Similarly, the $k$th boundary period starts at $v_{k,r_k}$ and ends at $v_{k,R_k} = v_{k+1,0}$ (the beginning of the next cycle). During the boundary period we observe $R_k - r_k - 1$ buffer $y(t;\theta)$ *becomes empty* events which

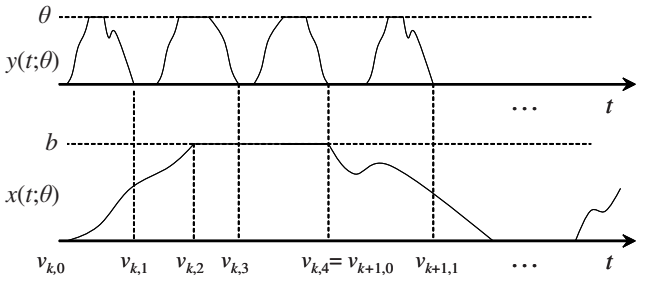



**Fig. 6.** A typical sample path for the tandem two-node network

in [27] are referred to as *active switchover points*; these are important because they are the *only* points that propagate the effect of perturbing θ downstream.

**Theorem 6.** *The workload and loss volume sample derivatives are given by*

$$Q'(\theta) = -\sum_{k=1}^{N_{\mathcal{C}}} \sum_{j=1}^{r_k-1} \left(v_{k,r_k} - v_{k,j}\right) \psi_{k,j} - \sum_{k=1}^{N_{\mathcal{C}}} \left(v_{k,r_k} - v_{k,0}\right) \phi_k$$

$$L'(\theta) = -\sum_{k=1}^{N_{\mathcal{C}}} \sum_{\substack{j=1 \\ j \neq r_k}}^{R_k} \mathbf{1}[x_k]\, \psi_{k,j} + \sum_{k=1}^{N_{\mathcal{C}}} \phi_k$$

*where* $\mathbf{1}[x_k] = \mathbf{1}\left[x(v_{k,r_k};\theta) = b\right]$ *and*

$$\psi_{k,j} = \left(A(v_{k,j}^+;\theta) - A(v_{k,j}^-;\theta)\right) v'_{k,j} = \left(\alpha(v_{k,j}) - \beta_1(v_{k,j})\right) v'_{k,j}$$

$$\phi_k = A(v_{k,0}^+) v'_{k,0}$$

At this point, it is worth pointing out that $v'_{k,R_k} = v'_{k+1,0} = 0$ if the end (or beginning) of a cycle does not coincide with a buffer *y becomes empty* event; note that in this case, the end of a boundary period of $x$ corresponds to an exogenous event. For the general case, when there are $m$ nodes in series, refer to [27] where it is also shown that these estimators are unbiased (under Assumption 1). It is also worth pointing out that in order to evaluate the above estimators, one simply needs a number of timers that measure the intervals $(v_{k,r_k} - v_{k,j})$ for $j = 1, \cdots, r_k - 1$ and $k = 1, \cdots, N_{\mathcal{C}}$. In addition, one needs some rate information at specific points in time and such information is generally easy to get. Finally, the only remaining information is the event time derivatives $v'_{k,j}$. But these are precisely the event time derivatives evaluated in (9).

## 6 Simulation Results

For illustration purposes, in this section, we present some simulation results (also reported in [28]), where we use the estimators obtained in the earlier sections to control

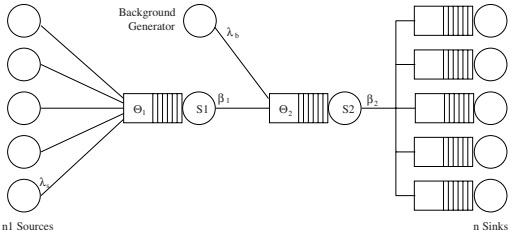



**Fig. 7.** System topology for the two-stage network

the buffer thresholds of a two-node network. The main objective is to optimize a cost function that consists of the weighted sum of loss and workload in the two-queue tandem system shown in Fig. 7. We emphasize that the setting of Fig. 7 may correspond to a "real system", so that it is best captured by a discrete-event model (not the fluid models that were considered so far). However, the simple form of the estimators obtained allows us to evaluate them using information readily available from the sample paths of discrete-event systems. Recall that all estimators consist only of some timers and counters. They simply count the number of cycles with overflow, or measure the interval between the occurrence of certain events (typically events that make a buffer become full or empty and events that make the buffer cease to be full or empty). Even though there is no guarantee that the estimators are still unbiased when evaluated using information from the discrete-event sample path, our simulation results indicate that such estimators can be used to solve practical optimization problems.

In the system of Fig. 7, intended to represent the operation of a communication network, the inflow process at the first queue consists of $n_1$ multiplexed *on–off* data sources generating bursty traffic. When in the *on* state, each source generates a continuous data stream at the rate of $\alpha$ bits per second. These data streams are used to construct UDP packets which are forwarded to the buffer at the first queue and thence across the rest of the network. Each UDP packet consists of a 42-byte header (including UDP, IP, and IEEE 802.3 headers) and a 512 information (data) field, for a total of 554 bytes. The sources provide the content of the information field, and the header is prepended whenever that field becomes full. If the information field is not full at the time the state of a source changes from *on* to *off*, then the incomplete packet waits until the source changes back to the *on* state and completes the information field. In other words, all packets have 554 bytes. The *on* times and *off* times are i.i.d. random variables sampled from the exponential distribution with mean 0.1 seconds. The channel transporting packets from the first queue to the second queue has a capacity of $\beta_1$ bps. The inflow process to the second queue consists of the outflow process from the first queue and of traffic from the background generator. The background traffic consists of $n_2$ independent sources. Each one of these sources has the same statistical characteristics as the sources to the first queue. The outgoing channel from the second queue has a capacity of $\beta_2$ bps.

Note that the average bit rate from either one of the independent sources is $\alpha/2$ bps, since the expected durations of the off periods and the on periods are identical. Therefore, the expected bit rate of the aggregate flow to the first queue is $(n_1\alpha/2)\times(554/512)$, where the latter term accounts for the insertion of the headers. Consequently, the traffic intensity at the first queue, denoted by $\rho_1$, is given by

$$\rho_1 \;=\; n_1 \times \frac{\alpha}{2} \times \frac{554}{512} \times \frac{1}{\beta_1}. \tag{39}$$

Similarly, the traffic intensity of the second queue is denoted by $\rho_2$. All of the experiments were performed using the *Georgia Tech Network Simulator* (*GTNetS*) [29], modified to include the requisite IPA derivative calculations. In our simulation experiments we set $n_1 = n_2 = 100$, $\beta_1 = 10\ Mbps$, and $\beta_2 = 20\ Mbps$. For the simulation results, we set $\rho_1 = 0.95$ and calculated $\alpha$ according to (39).

Let $\theta = [\theta_1, \theta_2]$ denote the two-dimensional parameter vector consisting of the buffer limits at the first and second queue respectively. The loss volumes and workloads at the two queues are denoted by $L_j(\theta)$ and $Q_j(\theta)$, $j = 1,2$ (see (3), (4) and (38)). Let us define the cost function $J(\theta)$ as the weighted sum of the average loss rate and workload rate.

$$J(\theta) = \frac{1}{T}\left[L_1(\theta) + 10Q_1(\theta) + L_2(\theta) + 20Q_2(\theta)\right].$$

Recall that $T$ is the observation interval over which the objective function is defined and it is set to $T = 1$ second. We seek to minimize $\mathbb{E}[J(\theta]$ using a standard stochastic approximation technique (1) which defines a sequence of points $\theta_n = [\theta_1^n, \theta_2^n]$. However, we substitute the gradient of $J(\theta)$, $\mathbf{H}_n(\theta_n; \mathbf{x}(0); \omega_n^{SFM})$ with $\mathbf{H}_n(\theta_n; \mathbf{x}(0); \omega_n^{DES})$ to indicate that the gradient evaluation is done based on data observed from a discrete-event sample path. The required gradient is evaluated through the IPA algorithms described in the previous sections (specifically Theorems 1 and 6). In addition, although all our analysis is based on the assumption that all observed sample paths start with all queues at the empty state, we have nonetheless applied the IPA estimates at the $n$th iteration of the optimization algorithm using the ending state of the $(n-1)$th iteration. Furthermore, we adopt the step size sequence $\sigma_n = 10/n^{0.6}$. Finally, we used a simple heuristic to bound the displacement $\theta_{n+1} - \theta_n$ along each coordinate by modifying the vector $\mathbf{H}_n = [h_1^n, h_2^n]$ as follows. We first computed the partial derivatives $\frac{\partial J(\theta)}{\partial \theta_i}$, $i = 1,2$. If $|\sigma_n \frac{\partial J(\theta)}{\partial \theta_i}| \leq 5$ then we set $h_i^n = \frac{\partial J(\theta)}{\partial \theta_i}$, and if $|\sigma_n \frac{\partial J(\theta)}{\partial \theta_i}| > 5$, then we set $h_i^n = 5\mathrm{sgn}(\frac{\partial J(\theta)}{\partial \theta_i})/\sigma_n$. The parameters $\theta_i^n$ $(i = 1,2)$ were considered as real numbers, but the simulation runs were performed at the respective integer values closest to them.

We ran the optimization algorithm twice, with two different initial parameters: first with $\theta_1 = [5,5]$, and then with $\theta_n = [40,40]$. In either case we ran the algorithm for 100 iterations (i.e., 100 seconds). For each experiment, we plotted the evolution of $\theta_1^n$ and $\theta_2^n$ as a function of iteration $n$, and show the results in Fig. 6 respectively. Each of the figures shows one trajectory for the $\theta_n = [5,5]$ initial condition, and a second one for the $\theta_n = [40,40]$ initial condition. The results indicate asymptotic convergence to approximately ${\theta^*}_n = [15,14]$ within approximately 20 seconds.

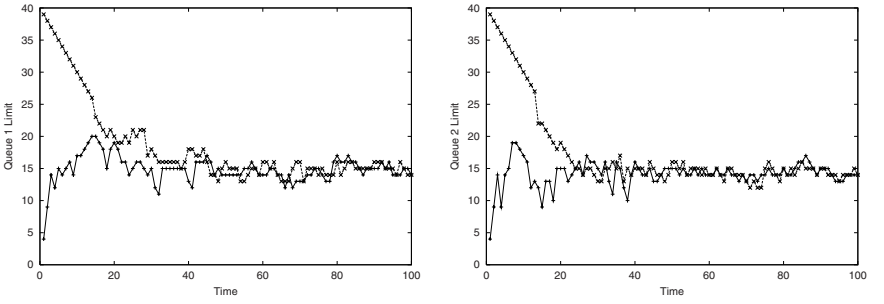



**Fig. 8.** Evolution of $\theta_1^n$ and $\theta_2^n$

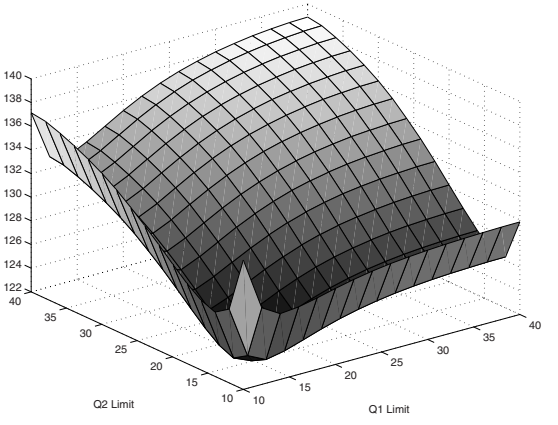



**Fig. 9.** Cost function $J(\theta_1, \theta_2)$

Finally, to add validity to these results, we plotted the graph of $J(\theta_1, \theta_2)$ as shown in Fig. 9. Each point on the plot is the average of 10 separate simulation experiments with $T = 100$ seconds, each with a different seed for the random number generators. However, each set of the 10 simulation experiments uses the same set of 10 random seeds as all other sets of experiments. This graph clearly corroborates the results obtained by the optimization runs, i.e., it shows that $\theta^*{}_n = [15, 14]$ is indeed optimal.

## 7 Conclusions and Future Work

In this chapter we used the stochastic fluid modeling paradigm and derived IPA sensitivity estimates of some performance measures of interest with respect to the control parameters $\theta$ (or $\rho$). Subsequently, these estimators were evaluated from data observed in a discrete-event sample path and they were used in stochastic optimization schemes to solve non-linear stochastic optimization problems. For all problems considered in this chapter, there was no feedback involved, in other words, the inflow

processes where independent of the control parameters. Such models are appropriate for User Datagram Protocol (UDP) traffic, however, they do not capture the Transport Control Protocol (TCP) traffic. The difficulty of TCP stems from the delayed feedback mechanisms that are embedded in this scheme. As a result, models with feedback are still being developed (see for example [30]). Closing, another open question is whether the IPA evaluation process using data observed from a discrete-event sample path still produces unbiased estimators. In general, this process does introduce some bias; however, our experience so far indicates that the estimators obtained are adequate for practical network management and optimization problems.

## Acknowledgements

The work of the second and third authors is supported in part by the National Science Foundation under grants EEC-0088073 and DMI-0330171, by AFOSR under grant F49620-01-0056, and by ARO under grant DAAD19-01-0610.

## References

[1] W. E. Leland, M. S. Taqq, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic," in *ACM SIGCOMM*, pp. 183–193, 1993.
[2] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226–244, June 1995.
[3] D. Anick, D. Mitra, and M. Sondhi, "Stochastic theory of a data-handling system with multiple sources," *The Bell System Technical Journal*, vol. 61, pp. 1871–1894, 1982.
[4] H. Kobayashi and Q. Ren, "A mathematical theory for transient analysis of communications networks," *IEICE Transactions on Communications*, vol. E75-B, pp. 1266–1276, 1992.
[5] R. Cruz, "A calculus for network delay, Part I: Network elements in isolation," *IEEE Transactions on Information Theory*, 1991.
[6] G. Kesidis, A. Singh, D. Cheung, and W. Kwok, "Feasibility of fluid-driven simulation for ATM network," in *Proc. IEEE Globecom*, vol. 3, pp. 2013–2017, 1996.
[7] K. Kumaran and D. Mitra, "Performance and fluid simulations of a novel shared buffer management system," in *Proceedings of IEEE INFOCOM*, March 1998.
[8] B. Liu, Y. Guo, J. Kurose, D. Towsley, and W. Gong, "Fluid simulation of large scale networks: Issues and tradeoffs," in *Proceedings of the Intl. Conf. on Parallel and Distributed Processing Techniques and Applications*, June 1999. Las Vegas, Nevada.
[9] A. Yan and W. Gong, "Fluid simulation for high-speed networks with flow-based routing," *IEEE Transactions on Information Theory*, vol. 45, pp. 1588–1599, 1999.

[10] R. Akella and P. Kumar, "Optimal control of production rate in a failure prone manufacturing system," *IEEE Transactions on Automatic Control*, vol. 31, pp. 116–126, Feb 1986.

[11] J. Perkins and R. Srikant, "The role of queue length information in congestion control and resource pricing," in *Proceedings IEEE Conference on Decision and Control*, pp. 2703–2708, Dec 1999.

[12] J. Perkins and R. Srikant, "Failure-prone production systems with uncertain demand," *IEEE Transactions on Automatic Control*, vol. 46, pp. 441–449, 2001.

[13] Y. Wardi and B. Melamed, "Variational bounds and sensitivity analysis of traffic processes in continuous flow models," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 11, pp. 249–282, 2001.

[14] B. Mohanty and C. Cassandras, "The effect of model uncertainty on some optimal routing problems," *Journal of Optimization Theory and Applications*, vol. 77, pp. 257–290, 1993.

[15] S. Meyn, "Sequencing and routing in multiclass networks. Part I: Feedback regulation," in *Proceedings of the IEEE International Symposium on Information Theory*, pp. 4440–4445, 2000. To appear in SIAM J. Control and Optimization.

[16] C. G. Cassandras, Y. Wardi, B. Melamed, G. Sun, and C. G. Panayiotou, "Perturbation analysis for on-line control and optimization of stochastic fluid models," *IEEE Transactions on Automatic Control*, vol. AC-47, no. 8, pp. 1234–1248, 2002.

[17] C. Cassandras, G. Sun, C. Panayiotou, and Y. Wardi, "Perturbation analysis and control of two-class stochastic fluid models for communication networks," *IEEE Transactions on Automatic Control*, vol. 48, pp. 770–782, May 2003.

[18] G. Sun, C. G. Cassandras, and C. G. Panayiotou, "Perturbation analysis of a multiclass stochastic fluid model with finite buffer capacity," in *Proceedings of 41st IEEE Conf. On Decision and Control*, pp. 2171–2176, 2002.

[19] Y. Wardi, B. Melamed, C. Cassandras, and C. Panayiotou, "IPA gradient estimators in single-node stochastic fluid models," *Journal of Optimization Theory and Applications*, vol. 115, no. 2, pp. 369–406, 2002.

[20] Y. C. Ho and X. Cao, *Perturbation Analysis of Discrete Event Dynamic Systems*. Dordrecht, Holland: Kluwer Academic Publishers, 1991.

[21] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.

[22] H. J. Kushner and D. S. Clark, *Stochastic Approximation for Constrained and Unconstrained Systems*. Berlin, Germany: Springer-Verlag, 1978.

[23] G. Sun, C. Cassandras, and C. Panayiotou, "Perturbation analysis of multiclass stochastic fluid models," *Journal of Discrete Event Dynamic Systems*, 2004. To Appear.

[24] R. Y. Rubinstein and A. Shapiro, *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. New York, New York: John Wiley and Sons, 1993.

[25] I. Cidon, R. Guérin, and A. Khamisy, "On protective buffer policies," *IEEE/ACM Transactions on Networking*, vol. 2, pp. 240–246, Jun 1994.

[26] C. Panayiotou and C. Cassandras, "On-line predictive techniques for "differentiated services" networks," in *Proceedings IEEE Conference on Decision and Control*, pp. 4529–4534, Dec 2001.

[27] G. Sun, C. Cassandras, Y. Wardi, and C. Panayiotou, "Perturbation analysis of stochastic flow networks," in *Proceedings of IEEE Conference on Decision and Control*, pp. 4831–4838, Dec 2003.

[28] G. Sun, C. Cassandras, Y. Wardi, C. Panayiotou, and G. Riley, "Perturbation analysis and optimization of stochastic flow networks," 2003. Submitted.

[29] G. F. Riley, "The Georgia Tech Network Simulator," in *Proceedings of Workshop on Models, Methods, and Tools for Reproducible Network Research (MoMeTools)*, Aug. 2003.

[30] H. Yu and C. G. Cassandras, "Perturbation analysis of feedback-controlled stochastic flow systems," in *Proceedings of the IEEE Conference On Decision and Control*, pp. 6277–6282, 2003.

# Stabilized Vegas *

Hyojeong Choe[1] and Steven H. Low[2]

[1] EEE Postech, Pohang, Korea and Caltech, Pasadena, USA hjchoe@caltech.edu
[2] CS and EE Caltech, Pasadena USA slow@caltech.edu http://netlab.caltech.edu

## 1 Introduction

TCP Vegas was introduced in 1994 [5] as an alternative to TCP Reno. Unlike Reno (or its variants such as NewReno and SACK), that uses packet loss as a measure of congestion, Vegas uses queueing delay as a measure of congestion [15, 18]. Vegas introduces a new congestion avoidance mechanism that corrects the oscillatory behavior of AIMD (Additive Increase Multiplicative Decrease). While the AIMD algorithm induces loss to learn the available network capacity, a Vegas source adjusts its sending rate to keep a small number of packets buffered in the routers along the path. Provided there is enough buffering, a network of Vegas sources will stabilize around a proportionally fair equilibrium and packet loss will be eliminated; see [15] for details. In this paper, we study the stability of this equilibrium in the presence of network delay, motivated by two lines of recent research.

First, extensive experimental results have been conducted to compare the performance of Vegas and Reno, e.g., [1, 5, 7]. Its dynamic and fairness properties have also been studied in [3, 4, 17], but these papers consider only a single bottleneck link and network delay is not accounted for in the study of its dynamics. Optimization based models are used in [15, 18] to analyze a general network of Vegas. In particular, it is shown in [13, 15] that any TCP/AQM (active queue management) protocol can be interpreted as carrying out a distributed primal-dual algorithm over the Internet to maximize aggregate utility, and a user's utility is (often implicitly) defined by its TCP algorithm; see also, e.g., [9, 11, 14, 16, 18]. These models mostly focus on the equilibrium structure, and do not adequately deal with network delay. To complement this series of work, we use here a multi-link multi-source model, described in Section 2, that explicitly includes heterogeneous forward and backward delays to analyze the linear stability of Vegas around an equilibrium. In comparison with previous analytical work, we sacrifice global (nonlinear) dynamics in order to understand the effect of delay on stability of Vegas, and how to stabilize it.

* Preliminary results have appeared in the Proceedings of IEEE Infocom, San Francisco, April 2003.

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 27–47, 2004.

Second, this paper is motivated by the stability theory for linear distributed and delayed system recently developed in [10, 19, 20, 21, 22]. In particular, a TCP/AQM algorithm is designed in [19] that maintains linear stability for *arbitrary* network delays and capacities. It is in the class of "dual" algorithms of [14] that use static source algorithms, and it employs a sophisticated scaling with respect to network delays and capacities to achieve high utilization and fast response without compromising stability. This form of arbitrarily scalable stability, however, dictates a specific source utility function and hence fairness in rate allocation. By introducing a slower timescale dynamics into the source algorithm, the TCP/AQM of [19] is extended in [20] to track *any* utility function, or fairness, on a slow timescale, provided there is a known bound on network delays.

The main insight from this series of work is to scale down source responses with their own round trip times and scale down link responses with their own capacities, in order to keep the gain over the feedback loop under control. It turns out that the (implicit) link algorithm of Vegas has exactly the right scaling with respect to capacity as used in [19]; see [15]. This built-in scaling with capacity makes Vegas potentially scalable to high bandwidth, in stark contrast to Reno and its variants. The source algorithm of Vegas, however, has a different scaling from those in [19] with respect to delay.

We prove in Section 3 a sufficient stability condition that suggests that Vegas can become unstable at large delay. In Section 4 we propose a small modification to stabilize it. We describe an incremental deployment strategy in Section 5 that would allow Vegas sources to work with a mix of routers, some implementing a queue-clearing AQM and some not. We present simulation results in Section 6 to validate our theoretical results. Finally, we conclude with limitations of this work.

In contrast to Reno and its variants, Vegas seems particularly well-suited for high speed networks. Reno and its variants, with RED, become unstable as network capacity increases [5, 12]. It also must maintain an exceedingly small loss probability in equilibrium that is difficult to reliably use for control. Vegas, on the other hand, scales correctly with capacity. Moreover, while the equilibrium queueing delay can be excessive at low capacity, it is reduced as capacity increases. Other problems, such as error in propagation delay estimation due to queues and rerouting [15, 17], may be less severe at high capacity, as buffers clear more frequently.

## 2 Network model

A network is modeled as a set of $L$ links (scarce resources) with finite capacities $c = (c_l, l \in L)$. They are shared by a set of $N$ sources indexed by $r$. Each source $r$ uses a set of links defined by the $L \times N$ routing matrix

$$R_{lr} = \begin{cases} 1 & \text{if source } r \text{ uses link } l \\ 0 & \text{otherwise} \end{cases}$$

Associated with each link $l$ is a congestion measure $p_l(t)$ we will call 'price'; as we will see below, $p_l(t)$ is the scaled queueing delay at link $l$. Each source $r$

maintains a rate $x_r(t)$ in packets/sec. In this paper, we are mainly concerned with linearized model around an equilibrium, so we denote the equilibrium forward delay from source $r$ to link $l$ by $\tau_{lr}^{\rightarrow}$ and the equilibrium backward delay from link $l$ to source $r$ by $\tau_{lr}^{\leftarrow}$. At time $t$, we assume source $r$ observes the aggregate price in its path

$$q_r(t) := \sum_l R_{lr} p_l(t - \tau_{lr}^{\leftarrow}) \tag{1}$$

and link $l$ observes the aggregate source rate

$$y_l(t) := \sum_r R_{lr} x_r(t - \tau_{lr}^{\rightarrow}) \tag{2}$$

Let $T_r$ denote the equilibrium round trip time. We assume that

$$\tau_{lr}^{\rightarrow} + \tau_{lr}^{\leftarrow} = T_r, \quad \forall l \in L \tag{3}$$

Then [15] models TCP Vegas, with its associated queue management, as the following dynamical system:[3]

$$\dot{p}_l(t) = \begin{cases} \frac{1}{c_l}(y_l(t) - c_l) & \text{if } p_l(t) > 0 \\ \frac{1}{c_l}(y_l(t) - c_l)^+ & \text{if } p_l(t) = 0 \end{cases} \tag{4}$$

$$\dot{x}_r(t) = \frac{1}{T_r^2(t)} \operatorname{sgn}\left(1 - \frac{x_r(t) q_r(t)}{\alpha_r d_r}\right) \tag{5}$$

where $(z)^+ = \max\{0, z\}$, $\operatorname{sgn}(z) = 1$ if $z > 0$, $-1$ if $z < 0$, and 0 if $z = 0$. Here, $\alpha_r$ is a Vegas protocol parameter, and $d_r$ is the round trip propagation delay of source $r$. Price $p_l(t)$ is the queueing delay at link $l$ and $q_r(t)$ is the end-to-end queueing delay of source $r$ (see [15]). Round trip time of source $r$ is defined as

$$T_r(t) := d_r + q_r(t)$$

with the equilibrium value $T_r$ defined in (3).

An interpretation of Vegas algorithm is that each source $r$ adjusts its rate (or window) to maintain $\alpha_r d_r$ number of its own packets buffered in the queues in its path. The link algorithm (4) is automatically carried out by the buffer process. The source algorithm (5) increments or decrements the *window* by 1 packet per round trip time, according as the number $x_r(t)q_r(t)$ of packets buffered in the links is smaller or bigger than $\alpha_r d_r$. In equilibrium, $x_r^* q_r^* = \alpha_r d_r$, and the unique equilibrium rates $x^* := (x_r^*, r = 1, \ldots, N)$ maximize aggregate utility $\sum_r U_r(x_r)$ subject to link capacity constraints, with the utility functions (see [15] for details)

$$U_r(x_r) = \alpha_r d_r \log x_r.$$

Hence Vegas achieves weighted proportional fairness [9].

[3] The model in [15] is discrete-time and ignores feedback delay in the interconnection defined by (1) and (2).

The link algorithm of [19] is the same as (4) of Vegas, except that, there, $c$ is a virtual capacity that is strictly less than real link capacity in order to clear the queue in equilibrium. There, $p_l(t)$ can be interpreted as "virtual" queueing delay at a link fed by the same input but drained at the virtual capacity. As shown in [19], scaling down $\dot{p}_l$ by $1/c_l$ is what gives delay, real or virtual, the right scaling with respect to network capacity. In Section 5, we will explain how stabilized Vegas can be incrementally deployed in a network with a mix of links, generating both real and virtual queueing delays.

## 3 Stability of Vegas

The Vegas source algorithm (5) is discontinuous. This can cause oscillation around the equilibrium. The original Vegas algorithm prevents oscillation by enlarging the equilibrium point to a set: source rate $x_r(t)$ (or window) is unchanged as soon as the number $x_r(t)q_r(t)$ of packets buffered in the links enters the range $[\alpha_r d_r, \beta_r d_r]$ with $\alpha_r < \beta_r$. It is however hard to control fairness with $\alpha_r < \beta_r$ [4]. Here, as in [15], we take $\alpha_r = \beta_r$.

In this section, we present a continuous approximation of the Vegas algorithm (5), derive a sufficient condition for linear stability based on this approximation. The condition suggests that a network of Vegas can become unstable when delay increases. In the next section, we will indeed propose to use (a stabilized version of) this continuous function to replace (5) in order to prevent oscillation due to discontinuity.

### 3.1 Approximate model

Note that

$$\mathrm{sgn}(z) \simeq \frac{2}{\pi}\tan^{-1}(\eta z)$$

The approximation becomes exact in the limit as $\eta \to \infty$. Hence, consider the following approximation of (5):

$$\begin{aligned}\dot{x}_r(t) &:= f_r(x_r(t), q_r(t)) \\ &= \frac{2}{\pi}\frac{1}{T_r^2(t)}\ \tan^{-1}\eta\left(1 - \frac{x_r(t)q_r(t)}{\alpha_r d_r}\right) \end{aligned} \tag{6}$$

where again $T_r(t) = d_r + q_r(t)$.

Consider the equilibrium point $(x^*, p^*)$. The rates $x^*$ are unique since the log utility function is strictly concave. Suppose the routing matrix $R$ has full rank, so that the equilibrium prices $p^* = (p_l^*, l = 1, \ldots, L)$ are also unique. Moreover, assume that only bottleneck links are included in the model so that $p_l^* > 0$ for all $l$. In equilibrium, the source rate $x_r^*$ and aggregate price $q_r^*$ satisfy

$$x_r^* q_r^* = \alpha_r d_r \tag{7}$$

Linearize around the equilibrium point,

$$\begin{cases} x_r(t) = x_r^* + \delta x_r(t) \\ q_r(t) = q_r^* + \delta q_r(t) \end{cases}$$

Then, to first order,

$$\dot{x}_r \;=\; \delta\dot{x}_r = \left.\frac{\partial f_r}{\partial x_r}\right|_* \delta x_r(t) + \left.\frac{\partial f_r}{\partial q_r}\right|_* \delta q_r(t)$$

where

$$\begin{aligned} \left.\frac{\partial f_r}{\partial x_r}\right|_* &= -\frac{2\eta}{\pi}\frac{1}{x_r^* T_r^2} \\ \left.\frac{\partial f_r}{\partial q_r}\right|_* &= -\frac{2\eta}{\pi}\frac{1}{q_r^* T_r^2} \end{aligned}$$

where we have used $T_r = d_r + q_r^*$. Hence, in Laplace domain, we have

$$\delta x_r(s) = -\frac{x_r^*}{q_r^*}\frac{a_r}{sT_r + a_r}\,\delta q_r(s) \tag{8}$$

where

$$a_r = \frac{2\eta}{\pi}\frac{1}{x_r^* T_r}. \tag{9}$$

At the links, the equilibrium points $(y_l^*, p_l^*)$ satisfy $y_l^* = c_l$. Linearizing the link algorithm (4) around the equilibrium

$$\begin{cases} y_l(t) = y_l^* + \delta y_l(t) \\ p_l(t) = p_l^* + \delta p_l(t) \end{cases}$$

we have, to first order,

$$\begin{aligned} \delta\dot{p}_l &= \left.\frac{\partial g_l}{\partial p_l}\right|_* \delta p_l(t) + \left.\frac{\partial g_l}{\partial y_l}\right|_* \delta y_l(t) \\ &= \frac{1}{c_l}\delta y_l(t) \end{aligned}$$

and its Laplace transform

$$\delta p_l(s) = \frac{1}{c_l s}\delta y_l(s). \tag{10}$$

In summary, the linearized model of Vegas is described by (8), (9) and (10). To simplify notation, we assume without loss of generality for the rest of the paper that all sources have the same target queue length, $\alpha_r d_r = \alpha$ for all $r$ (otherwise, take $\alpha$ to be the minimum $\alpha_r d_r$ in the stability results that follow).

### 3.2 Stability

Following [19], we can express the error equations of (1)–(2) in matrix form, in Laplace, as

$$\delta y(s) = R(s)\delta x(s) \tag{11}$$
$$\delta q(s) = \mathrm{diag}\{e^{-sT_r}\}R^T(-s)\delta p(s) \tag{12}$$

where

$$R_{lr}(s) = \begin{cases} e^{-s\tau^{\rightarrow}_{ls}} & \text{if } R_{lr} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The routing matrix $R(0) = R$ determines static relationship between equilibrium values, i.e.,

$$y^* = R(0)x^*, \quad q^* = R^T(0)p^* \tag{13}$$

Given any finite positive $a$, let $\theta(a)$ be the unique solution in $(0, \pi/2)$ of

$$\theta \tan\theta = a \tag{14}$$

as a (strictly increasing) function of $a$.

We will characterize the stability of Vegas first in terms of maximum window size and then in terms of minimum queueing delay. The results below say that Vegas is linearly stable if the equilibrium window size is sufficiently small (Theorem 1), or equivalently, if the equilibrium queueing delay is sufficiently large (Corollary 1).

**Theorem 1.** *Suppose for all r, $k_0 T_r \geq \max_r T_r$ for some $k_0$. Let $M$ be an upper bound on the number of links in the path of any source, $M \geq \max_r \sum_l R_{lr}$. The Vegas model described by (4) and (6) is locally asymptotically stable around the equilibrium point $(x_r^*, y_l^*, p_l^*, q_r^*)$ if*

$$\max_r \; x_r^* T_r \; sinc\,\theta\left(\frac{\hat{\eta}}{x_r^* T_r}\right) \; < \; \frac{\alpha}{M k_0^2} \tag{15}$$

*where $\hat{\eta} := 2\eta/\pi$ and $sinc\,\theta = \sin\theta/\theta$.*

*Proof.* See Section 3.3 below.

Note that since $\theta(\cdot)$ is strictly increasing and $\mathrm{sinc}(\cdot)$ is strictly decreasing, the left-hand side of the stability condition in Theorem 1 is strictly increasing in the window size $x_r^* T_r$. Hence the stability condition imposes a limit on the maximum window size.

Since $x_r^* q_r^* = \alpha$, this condition directly translates into one on queueing delay. The left-hand side of the following corollary is strictly increasing in $q_r^*/T_r$, implying a lower bound on queueing delay.

**Corollary 1.** *Suppose for all r, $k_0 T_r \geq \max_r T_r$ for some $k_0$. Let $M$ be an upper bound on the number of links in the path of any source, $M \geq \max_r \sum_l R_{lr}$. The Vegas model described by (4) and (6) is locally asymptotically stable around the equilibrium point $(x_r^*, y_l^*, p_l^*, q_r^*)$ if*

$$\min_r \frac{q_r^*/T_r}{sinc\,\theta\left(\frac{\hat{\eta}}{\alpha}\cdot\frac{q_r^*}{T_r}\right)} > Mk_0^2$$

*where* $\hat{\eta} := 2\eta/\pi$ *and* $sinc\,\theta = \sin\theta/\theta$.

The next result shows that the stability condition cannot be satisfied when there are more than one (bottleneck) link in a source's path.

**Corollary 2.** *The stability condition cannot be satisfied if a source has more than one link, i.e., if there is an r with* $R_{lr} = 1$ *for more than one l.*

*Proof.* The conditions in Theorem 1 and Corollary 1 are the same, so we will work with Corollary 1. Since $\theta(\cdot) < \pi/2$, $\operatorname{sinc}\theta(\cdot) > 2/\pi$. By definition $k_0 \geq 1$. Hence the stability condition in Corollary 1 implies

$$\min_r \frac{q_r^*}{T_r} > \frac{2M}{\pi}.$$

If $M \geq 2$, then the right-hand side is bigger than 1. Yet, the left-hand side cannot exceed 1 since $T_r = d_r + q_r^*$.

We emphasize that the stability condition is only sufficient in the multiple link case. It is however both necessary and sufficient in the single-link-homogeneous-source case. We now illustrate, for this case, the stability region and the effect of protocol parameter $\alpha = \alpha_r d_r$.

**Example 1: Single link with homogeneous sources** $(c,d,N)$
Consider a single link of capacity $c$ shared by $N$ homogeneous sources with round trip propagation delay $d$. For this case, $a_r = a_0$, $T_r = T_0$, and $\omega_r = \omega_0$ for all sources $r$ in the proof of Theorem 1; see Section 3.3. This implies that the stability condition ($M = 1$ and $k_0 = 1$)

$$\frac{q_r^*/T_r}{\operatorname{sinc}\theta\left(\frac{\hat{\eta}}{\alpha}\cdot\frac{q_r^*}{T_r}\right)} > 1 \quad \text{for all } r \tag{16}$$

is both necessary and sufficient. Note that the equilibrium quantities $q_r^*$ and $T_r$ depend on the target queue length $\alpha$. To get insight on the effect of protocol parameter $\alpha$ on stability, we look at a simpler condition.

As noted above, since $\operatorname{sinc}\theta(\cdot) > \frac{2}{\pi}$, a *necessary* condition is

$$\frac{q_r^*}{T_r} > \frac{2}{\pi} \quad \text{for all } r.$$

Since $T_r = d + q_r^*$ and $q_r^* = \alpha/x_r^* = \alpha N/c$ by symmetry, this condition is equivalent to

$$cd < \left(\frac{\pi}{2} - 1\right)\alpha N. \tag{17}$$

Hence, a necessary condition for Vegas stability is that the bandwidth delay product be small. Moreover the stability region is larger with larger target queue length $\alpha$ or number $N$ of sources. □

### 3.3 Proof of Theorem 1

The proof proceeds in three steps. First, we follow the argument of [20, 21] to show that the Nyquist trajectories of the loop gain matrix is contained in the convex hull of $N$ complex functions of $j\omega$. Second, we show that at large enough $\omega$ when at least one of these functions has a phase lag of $-\pi$, all of them are contained in the unit circle, under an appropriate condition, and hence cannot encircle $-1$ in the complex plane. Third, we show that this condition is the one in the theorem.

**Step 1:** Using the linearized equations (8), (10), (11), (12), the return ratio seen at the source is described as :

$$\mathrm{diag}\left(a_r\frac{x_r^*}{q_r^*}\right)\mathrm{diag}\left(\frac{e^{-sT_r}}{sT_r+a_r}\right)R^T(-s)\mathrm{diag}\left(\frac{1}{c_l s}\right)R(s).$$

For stability, it suffices to show that the eigenvalues of this function does not encircle $-1$ in the complex plain for $s=j\omega$, $\omega\geq 0$. The set of eigenvalues is identical to that of

$$L(j\omega) \;=\; \mathrm{diag}\left(\frac{ke^{-j\omega T_r}}{j\omega T_r\,(j\omega T_r+a_r)}\right)\hat{R}^T(-j\omega)\hat{R}(j\omega) \tag{18}$$

where

$$k \;=\; \frac{Ma_rT_r}{q_r^*} \;=\; \frac{2\eta M}{\pi\alpha} \tag{19}$$

and

$$\hat{R}(j\omega) \;:=\; \mathrm{diag}\left(\sqrt{\frac{1}{c_l}}\right)R(j\omega)\mathrm{diag}\left(\sqrt{\frac{x_r^*}{M}}\right)$$

From the lemma of [21], the spectrum of $L(j\omega)$ satisfies

$$\begin{aligned}\sigma(L(j\omega)) &= \sigma\left(\mathrm{diag}\left(\frac{ke^{-j\omega T_r}}{j\omega T_r\,(j\omega T_r+a_r)}\right)\hat{R}^T(-j\omega)\hat{R}(j\omega)\right)\\ &\subseteq \rho\left(\hat{R}^T(-j\omega)\hat{R}(j\omega)\right)\cdot\\ &\quad \mathrm{co}\left(0\bigcup\left\{\frac{ke^{-j\omega T_r}}{j\omega T_r\,(j\omega T_r+a_r)},\; r=1,\ldots,N\right\}\right)\end{aligned}$$

where $\mathrm{co}(\cdot)$ above denotes the convex hull of the $N$ eigentrajectories and the origin.

The spectral radius of $\hat{R}(j\omega)$ satisfies

$$\begin{aligned}&\rho\left(\hat{R}^T(-j\omega)\hat{R}(j\omega)\right)\\ &\qquad\leq \left\|\mathrm{diag}\left(\frac{1}{M}\right)R^T(-j\omega)\,\mathrm{diag}\left(\frac{1}{c_l}\right)R(j\omega)\,\mathrm{diag}(x_r^*)\right\|_\infty\\ &\qquad\leq \left\|\mathrm{diag}\left(\frac{1}{M}\right)R^T(-j\omega)\right\|_\infty\cdot\\ &\qquad\quad \left\|\mathrm{diag}\left(\frac{1}{y_l^*}\right)R(j\omega)\,\mathrm{diag}\,(x_r^*)\right\|_\infty\\ &\qquad= 1.\end{aligned}$$

since, by (13), all the absolute row sums are equal to 1. Hence,

$$\sigma(L(j\omega)) \subseteq \operatorname{co}\left(0 \bigcup \left\{ \frac{ke^{-j\omega T_r}}{(j\omega T_r + a_r)},\ r = 1,\ldots,N \right\}\right).$$

Let

$$\Lambda_r(j\omega) := \frac{ke^{-j\omega T_r}}{j\omega T_r\,(j\omega T_r + a_r)}.$$

We now show that under the condition of the theorem, at no $\omega$ will the convex combination of $\Lambda_r(j\omega)$ encircle the critical point $-1$. **Step 2:** Define $a_0 = \min_r a_r$ and $T_0 = \max_r T_r$. Let $\omega_r$, $r = 0,1,\ldots,N$, be the value in $(0,\pi/2)$ that satisfies

$$\omega_r T_r \tan \omega_r T_r = a_r, \quad r \geq 0. \tag{20}$$

Clearly $\omega_0 \leq \omega_r$ for all $r$. Here $\omega_r$, $r \geq 1$, is the critical frequency when the eigenvalue $\Lambda_r(j\omega)$ has a phase lag of $-\pi$. Hence, for $\omega < \omega_0 \leq \omega_r$, the convex combination of $\Lambda_r(j\omega)$ cannot encircle $-1$ because $\text{phase}(\Lambda_r(j\omega)) > -\pi$ for all $r$. We now show that, for $\omega \geq \omega_0$, all $\Lambda_r(j\omega)$ are in the unit circle and hence their convex combination cannot encircle $-1$ either. For $\omega \geq \omega_0$, since $k_0 T_r \geq T_0$, we have

$$\begin{aligned}
|\Lambda_r(j\omega)| &= \frac{k}{\omega T_r \sqrt{\omega^2 T_r^2 + a_r^2}} \\
&\leq \frac{kk_0^2}{\omega_0 T_0 \sqrt{\omega_0^2 T_0^2 + a_0^2}} \\
&= \frac{kk_0^2}{a_0} \cdot \frac{1}{\omega_0 T_0} \cdot \frac{a_0}{\sqrt{\omega_0^2 T_0^2 + a_0^2}} \\
&= \frac{kk_0^2}{a_0} \cdot \operatorname{sinc}\theta(a_0)
\end{aligned}$$

where the last equality follows from (20) and the definition of $\theta(\cdot)$ in (14). Hence a sufficient stability condition is $|\Lambda_r(j\omega)| < 1$ for all $r$, $\omega \geq \omega_0$, or:

$$\frac{\operatorname{sinc}\theta(a_0)}{a_0} < \frac{1}{kk_0^2} \tag{21}$$

**Step 3:** Substituting $k$, from (19), and $a_0$:

$$k = \frac{\hat{\eta} M}{\alpha} \text{ and } a_0 = \min_r a_r = \frac{\hat{\eta}}{\max_r x_r^* T_r}$$

into the above condition, we have

$$\max_r x_r^* T_r \operatorname{sinc}\theta\left(\frac{\hat{\eta}}{\max_r x_r^* T_r}\right) < \frac{\alpha}{Mk_0^2}.$$

Since $\operatorname{sinc}\theta\left(\hat{\eta}(x_r^*T_r)^{-1}\right)$ is strictly increasing in $x_r^*T_r$, we have

$$\left(\max_r x_r^*T_r\right)\ \operatorname{sinc}\theta\left(\frac{\hat{\eta}}{\max_r x_r^*T_r}\right) = \max_r\left\{x_r^*T_r\ \operatorname{sinc}\theta\left(\frac{\hat{\eta}}{x_r^*T_r}\right)\right\}$$

hence the stability condition in the theorem. □

## 4 Stabilized Vegas

In this section, we propose a PD (proportional differential) controller at each source to stabilize a network of Vegas sources. We modify the (approximate) Vegas algorithm (6) into

$$\dot{x}_r\ =\ fracwT_r^2(t)\cdot\tan^{-1}\left(\eta_r(t)\Delta_r(t)\right) \tag{22}$$

or

$$\dot{x}_r\ =\ \begin{cases}\min\left[\frac{w}{T_r^2(t)}\left(e^{\eta_r(t)\Delta_r(t)}-1\right),\frac{\ln 2}{T_r(t)}x(t)\right], & \text{if } \Delta_r(t)>0\\ \max\left[\frac{w}{T_r^2(t)}\left(1-e^{-\eta_r(t)\Delta_r(t)}\right),-\frac{\ln 2}{T_r(t)}x(t)\right], & \text{otherwise}\end{cases} \tag{23}$$

where $T_r(t)=d_r+q_r(t)$, $\Delta_r(t)=1-\frac{x_r(t)q_r(t)}{\alpha_r d_r}-\kappa_r(t)\dot{q}_r(t)$ and

$$\kappa_r(t)\ =\ \frac{1}{a}\cdot\frac{T_r(t)}{q_r(t)} \tag{24}$$

$$\eta_r(t)\ =\ \frac{\mu a}{w}\cdot x_r(t)T_r(t) \tag{25}$$

Here, the parameter $w$ determines the maximum change in window size per round trip time (for the original Vegas, the maximum change is 1 packet per round trip time). The parameters $a>0$ and $\mu\in(0,1)$ are to be chosen to ensure stability (see below). The overall gain parameter $\eta_r(t)$ is proportional to the current window size: the larger the window, the more aggressive the response. The gain $\kappa_r(t)$ on the differential term is proportional to the ratio of round trip time to end-to-end queueing delay of source $r$, and serves as a normalization to $\dot{q}_r(t)$. The additional differential term $\kappa_r(t)\dot{q}_r(t)$ anticipates the future of $q_r(t)$. Without this term, source rate $x_r(t)$ will be increased if the number $x_r(t)q_r(t)$ of packets buffered in the links is small compared with $\alpha_r d_r$. With this term, even when $x_r(t)q_r(t)$ is small, the source may decrease its rate if prices are rapidly growing, i.e. if $\dot{q}(t)$ is large.

Both (22) and (23) have the same equilibrium point as the original Vegas, and both linearize to the same first-order equations

$$\delta\dot{x}_r\ =\ \left.\frac{\partial f_r}{\partial x_r}\right|_*\delta x_r(t)+\left.\frac{\partial f_r}{\partial q_r}\right|_*\delta q_r(t)+\left.\frac{\partial f_r}{\partial\dot{q}_r}\right|_*\delta\dot{q}_r(t)$$

where

$$\left.\frac{\partial f_r}{\partial x_r}\right|_* = -\frac{\mu a}{T_r}$$
$$\left.\frac{\partial f_r}{\partial q_r}\right|_* = -\frac{\mu a x_r^*}{T_r q_r^*}$$
$$\left.\frac{\partial f_r}{\partial \dot{q}_r}\right|_* = -\frac{\mu x_r^*}{q_r^*}.$$

Its Laplace transform is

$$\delta x_r(s) = -\frac{\mu x_r^*}{q_r^*}\left(\frac{sT_r + a}{sT_r + \mu a}\right)\delta q_r(s). \tag{26}$$

We have chosen $\eta_r$ and $\kappa_r$ so that the lead-lag compensator in (26) for all sources have a common zero $a$ and pole $\mu a$. In contrast, the algorithm of [20] allows $\mu_r$ to depend on $r$, corresponding to unrestricted choice of utility functions. Hence, we need a different stability proof from [20].

**Theorem 2.** *Suppose for all r, $k_0 T_r \geq \max_r T_r$ for some $k_0$. Let M be an upper bound on the number of links in the path of any source, $M \geq \max_r \sum_l R_{lr}$. For any given $a > 0$ and $\mu \in (0,1)$, the modified Vegas model described by (4) and (22)–(25) is locally asymptotically stable around the equilibrium point $(x^*, y^*, p^*, q^*)$ if*

$$\max_r \; x_r T_r \; < \; \frac{\alpha\phi}{\mu k_0 M}\sqrt{\frac{\phi^2 + \mu^2 (k_0 a)^2}{\phi^2 + (k_0 a)^2}} \tag{27}$$

*or equivalently, if*

$$\min_r \frac{q_r^*}{T_r} \; > \; \frac{\mu k_0 M}{\phi}\sqrt{\frac{\phi^2 + (k_0 a)^2}{\phi^2 + \mu^2 (k_0 a)^2}} \tag{28}$$

*where*

$$\phi \; = \; \tan^{-1}\frac{2\sqrt{\mu}}{1-\mu}$$

*and $\alpha = \alpha_r d_r$ is the common target queue length.*

*Proof.* The proof proceeds in two steps. First, we follow the argument of [20, 21] to show that the Nyquist trajectories of the loop gain matrix is contained in the convex hull of $N$ complex functions of $j\omega$. Second, we show that at large enough $\omega$ when at least one of these functions has a phase lag of $-\pi$, all of them are contained in the unit circle, under the conditions in the theorem, and hence cannot encircle $-1$ in the complex plane.

**Step 1:** Using the linearized equations (26), (10), (11) and (12), the return ratio seen at the sources can be written as:

$$\text{diag}\left(\frac{\mu x_r^*}{q_r^*}\right)\text{diag}\left(\frac{sT_r + a}{sT_r + \mu a}e^{-sT_r}\right)R^T(-s)\,\text{diag}\left(\frac{1}{c_l s}\right)R(s).$$

At $s = j\omega$, the set of eigenvalues is identical to that of

$$L(j\omega) = \text{diag}\left(\frac{\mu M T_r}{q_r^*}\right) \cdot \text{diag}\left(\frac{e^{-j\omega T_r}}{j\omega T_r} \cdot \frac{j\omega T_r + a}{j\omega T_r + \mu a}\right) \hat{R}^T(-j\omega)\hat{R}(j\omega)$$

where

$$\hat{R}(j\omega) \;=\; \text{diag}\left(\sqrt{\frac{1}{c_l}}\right) R(j\omega) \text{diag}\left(\sqrt{\frac{x_r^*}{M}}\right). \tag{29}$$

Using (29) and (13), the usual argument gives

$$\rho\left(\hat{R}^T(-j\omega)\hat{R}(j\omega)\right) \le 1.$$

The lemma from [21] then implies that all eigenvalues of $L(j\omega)$ are in the convex hull:

$$\text{co}\left\{0 \bigcup \left\{\frac{M T_r}{q_r^*} \cdot \Lambda(j\omega T_r)\right\},\;\; r = 1, \ldots, N,\;\right\} \tag{30}$$

where

$$\Lambda(j\omega T_r) \;:=\; \mu \cdot \frac{e^{-j\omega T_r}}{j\omega T_r} \cdot \frac{j\omega T_r + a}{j\omega T_r + \mu a}.$$

Note that $\Lambda(\cdot)$ is independent of $r$. By the generalized Nyquist stability criterion [6], the system is stable if the set in (30) does not encircle $-1$.

**Step 2:** Let $\omega_r$ be the critical frequency at which the phase $\angle\Lambda(j\omega_r T_r)$ is $-\pi$ for sources $r$:

$$\omega_r T_r - \angle\frac{j\omega T_r + a}{j\omega T_r + \mu a} \;=\; \frac{\pi}{2}$$

Without loss of generality, we can assume $T_1 \ge T_r$ for all $r$. Then $\omega_1 \le \omega_r$ for all $r$ since $\omega_r T_r = \omega_1 T_1$ for all $r$. Thus, at $\omega \le \omega_1$, the convex hull of (30) cannot encircle $-1$. At $\omega \ge \omega_1$, the set in (30) does not encircle $-1$ if, for all $r$,

$$\frac{M T_r}{q_r^*} \cdot |\Lambda(j\omega T_r)| \;<\; 1. \tag{31}$$

We now show that this is implied by the conditions in the theorem.

For $\omega \ge \omega_1$, we have $\omega T_r \ge \omega_1 T_r \ge \omega_1 T_1 / k_0$. Notice that the magnitude

$$|\Lambda(j\omega T_r)| \;=\; \frac{\mu}{\omega T_r}\sqrt{\frac{(\omega T_r)^2 + a^2}{(\omega T_r)^2 + \mu^2 a^2}}$$

is a strictly decreasing function of $\omega T_r$. Hence, for all $r$,

$$
\begin{aligned}
|\Lambda(j\omega T_r)| &\le \left|\Lambda\left(j\omega_1 \frac{T_1}{k_0}\right)\right| \\
&= \frac{\mu k_0}{\omega_1 T_1}\sqrt{\frac{(\omega_1 T_1)^2+(ak_0)^2}{(\omega_1 T_1)^2+\mu^2(ak_0)^2}} \\
&\le \frac{\mu k_0}{\phi}\sqrt{\frac{\phi^2+(ak_0)^2}{\phi^2+\mu^2(ak_0)^2}}.
\end{aligned}
$$

The last inequality follows from Lemma 1 below which implies that, for all $r$,

$$
\omega_r T_r \;\ge\; \frac{\pi}{2} - \tan^{-1}\frac{1-\mu}{2\sqrt{\mu}} \;=\; \phi
$$

where $\phi$ is defined in the theorem.

The condition (28) in the theorem then guarantees that (31) holds. Since $x_r^* q_r^* = \alpha$, the conditions (27) and (28) are equivalent. Hence the proof is complete with the following lemma.

**Lemma 1.** *Let*

$$
\Lambda(j\omega T_r) \;=\; \mu\cdot\frac{e^{-j\omega T_r}}{j\omega T_r}\cdot\frac{j\omega T_r + a}{j\omega T_r + \mu a}
$$

*where* $0 < \mu < 1$ *and* $a > 0$. *Then, for all r,* $\omega \ge 0$,

$$
\angle\Lambda(j\omega T_r) \;\ge\; -\omega T_r - \frac{\pi}{2} - \tan^{-1}\frac{1-\mu}{2\sqrt{\mu}}.
$$

*Proof.* Let

$$
\begin{aligned}
h_r(\omega) \;&:=\; \angle\frac{j\omega T_r + a}{j\omega T_r + \mu a} \\
&=\; \tan^{-1}\left(\frac{\omega T_r}{a}\right) - \tan^{-1}\left(\frac{\omega T_r}{\mu a}\right).
\end{aligned}
$$

Then

$$
h_r'(\omega) \;=\; \frac{(1-\mu)((\omega T_r)^2 - a^2\mu)a}{((\omega T_r)^2 + a^2)((\omega T_r)^2 + \mu^2 a^2)}.
$$

Since $\mu \in (0,1)$, it can be checked that the solution, $\omega_r^* = \frac{a\sqrt{\mu}}{T_r}$, of $h'(\omega) = 0$ minimizes the phase $h_r(\omega)$. Hence

$$
\angle h_r(\omega) \;\ge\; \tan^{-1}\sqrt{\mu} - \tan^{-1}\frac{1}{\sqrt{\mu}} \;=:\; \psi
$$

which is independent of $r$. Moreover

$$
\tan\psi \;=\; \frac{\sqrt{\mu} - \frac{1}{\sqrt{\mu}}}{2} \;=\; -\frac{1-\mu}{2\sqrt{\mu}}.
$$

Therefore

$$\begin{aligned}\angle\Lambda(j\omega T_r) &= -\omega T_r - \frac{\pi}{2} + \angle h_r(\omega) \\ &\geq -\omega_r T_r - \frac{\pi}{2} - \tan^{-1}\frac{1-\mu}{2\sqrt{\mu}}\end{aligned}$$

and hence the lemma follows.

We remark on the implications of Theorem 2. In the case of homogeneous round trip time, i.e. $k_0 = 1$, the stability (28) becomes

$$M \max_r \frac{T_r}{q_r^*} < \frac{\theta}{\mu}\sqrt{\frac{\theta^2+\mu^2 a^2}{\theta^2+a^2}}. \tag{32}$$

$M$ is a bound on the number of *bottleneck* links in the path of a source. It is typically much less than 10. $\frac{q_r^*}{T_r}$ is the ratio of round trip queueing delay and entire round trip time; both quantities are available at a Vegas source. For the current network, this ratio seems to be small (less than 5) for long delay routes. Hence, a choice of design parameters $a$ and $\mu$ that guarantees that the right-hand side of the stability condition exceeds 100 seems safe. From Fig. 1,

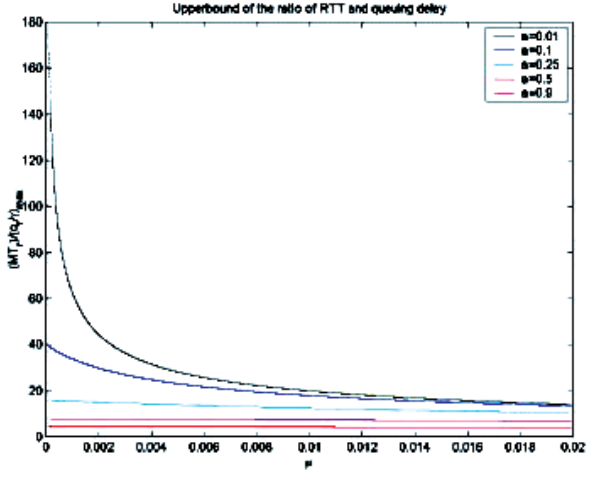



**Fig. 1.** The upper limit of $\frac{MT_r}{q_r^*}$.

this requires small $a$ and $\mu$ (e.g., $a = 0.01$ and $\mu = 0.001$). Recall the definition of (24) and (25): $\kappa_r(t) = (T_r(t)/q_r(t))/a$ and $\eta_r(t) = \mu a(x_r(t)T_r(t)/w)$. A small $a$ implies a large $\kappa_r(t)$, which means that the stabilized Vegas reacts more aggressively to price change $\dot{q}_r(t)$. A small $\mu a$ implies a small $\eta$, which means that the slope of (22)

around equilibrium is small, yielding a smoother overall gain. For the heterogeneous round-trip time case, i.e. $k_0 > 1$, a smaller $a$ than in the homogeneous is required to guarantee stability.

**Example 2: Single link with homogeneous sources** $(c,d,N)$
For a direct comparison with the original Vegas, we consider the same setup as in Example 1: a single link of capacity $c$ shared by $N$ homogeneous sources with round trip propagation delay $d$. For this case, the sufficient condition of Theorem 2 with ($M = 1$ and $k_0 = 1$) is simplified to

$$q_r^*/T_r \;>\; \frac{\mu}{\phi}\sqrt{\frac{\phi^2+a^2}{\phi^2+\mu^2a^2}} \qquad \text{for all } r.$$

Since $T_r = d + q_r^*$ and $q_r^* = \alpha/x_r^* = \alpha N/c$ by symmetry, this condition is equivalent to

$$cd \;<\; \left(\frac{\phi}{\mu}\sqrt{\frac{\phi^2+\mu^2a^2}{\phi^2+a^2}} - 1\right)\alpha N. \tag{33}$$

Hence, like the original Vegas, it also has a larger stability region with larger queue length $\alpha$ or number $N$ of sources. Furthermore, given $\alpha$ and $N$, stabilized Vegas can choose a small ($a > 0, \mu \in (0,1)$) such that the right-hand side of (33) can be larger than that of (17) for stability of the original Vegas. This is illustrated in Fig. 2 where the stability regions in (17) and (33), for Vegas and stabilized Vegas respectively, are plotted, with $(a,\mu) = (0.5, 0.015)$ and $\alpha = 20$ packets, $N = 100$ sources. □

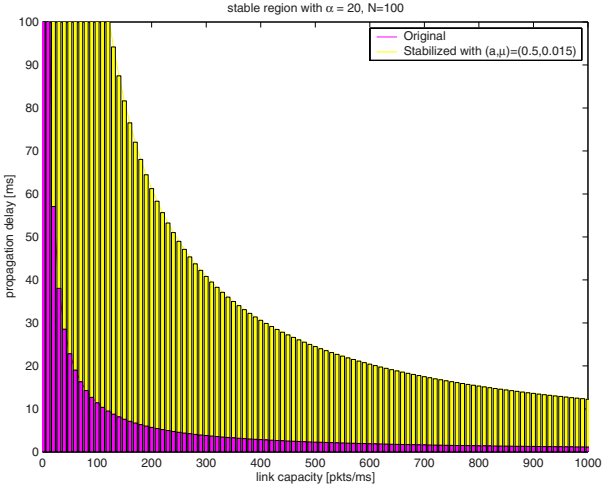



**Fig. 2.** Stability regions of Examples 1 and 2: single-link shared by homogeneous sources.

The stability condition is only sufficient. Indeed, less conservative values can be used for $a$ and $\mu$. For instance, the Nyquist plots of $(MT_r/q_r^*)\cdot\Lambda(j\omega T_r)$ for $MT_r/q_r^* =$

100 are shown in Fig. 3, for the scenario in Example 2 with $a = 0.1$ and $\mu \in [0.001, 0.015]$.

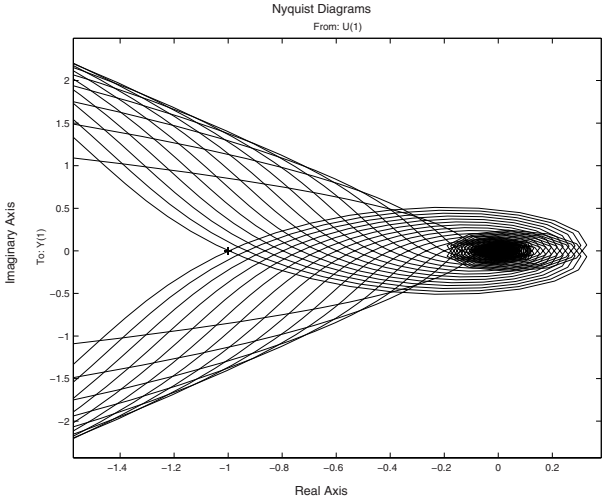



**Fig. 3.** Nyquist Stability of $\Lambda(j\omega T_r)$ for $\frac{MT_r}{q_r^*} = 100$ with $a = 0.1, \mu = [0.001 : 0.015]$.

Even though these $a$ and $\mu$ values do not satisfy the stability condition of Theorem 2, the network is indeed stable, as shown by the Nyquist plots.

## 5 Discussion: implementation and deployment

The most attractive feature of TCP Vegas is its suitability for high speed large delay networks. In this regime, window size is large and TCP Reno or its variants must maintain an extremely small loss probability (e.g., on the order $10^{-10}$) in equilibrium. Using such a small probability reliably is a great challenge.

Vegas on the other hand has two advantages, both stem from the use of delay as a measure of congestion. First, its implicit link algorithm has a built-in scaling with respect to network capacity, which together with stabilized source algorithm, can potentially scale to much larger bandwidth delay product. Second, each measurement of delay by a source provides a much finer-grained estimate of congestion than the binary-valued loss or marking does. As capacity scales up, it is easier to scale at the source to maintain the strength of the congestion signal (delay) by scaling up the $\alpha$ parameter.

The problem that delay may be excessive in the low bandwidth regime in order for Vegas to reach equilibrium is much less severe in the high bandwidth regime. Moreover, problem with error in propagation delay estimation and persistent congestion [15, 17] is also eased with high capacity, as buffers empty more frequently.

Though there are other issues with using delay for congestion control, it seems that unless ECN is widely deployed, these problems are less fundamental than the intrinsic difficulty of reliably using an extremely small loss probability for control. Further study is required to resolve these issues.

We now describe a viable strategy for stabilized Vegas to work with incremental deployment of new AQM and ECN. The link algorithm in Vegas computes the queueing delay as follows:

$$\dot{p}_l(t) = \frac{1}{c_l}(y_l(t) - c_l) \tag{34}$$

The division by $c_l$ is what gives Vegas the built-in scalability with network capacity (see proofs of Theorems 1 and 2). As discussed in [15], Vegas exploits the buffer process to automatically compute congestion prices, at the expense of having to maintain a non-zero queueing delay (these are Lagrange multipliers for the utility maximization problem Vegas is implicitly solving).

The link algorithm in the scalable scheme of [19] uses the same expression as in (34) except that instead of real link capacity $c_l$, a virtual link capacity that is slightly smaller (say, 95% of $c_l$) is used to explicitly compute the price $p_l(t)$. The advantage of using a virtual queue is that while the prices still converge to their non-zero values, the real queue will be cleared in equilibrium. Since queues are now empty, queueing delay can no longer serve as a feedback signal. ECN marking must be used to explicitly feedback the prices, e.g., using REM [2, 19].

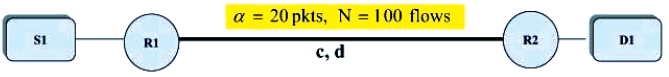
Imagine now a network with both types of links, one does not use ECN nor perform AQM to clear their queues and one does. The first type maintains a queue but does not mark, while the second type has no queueing delay but sends a stream of marks to the sources. A source observes two types of feedback signals: aggregate queueing delay from type 1 links and aggregate prices from type 2 links (after REM estimation). Not only do these two signals not interfere with each other, their sum yields precisely the total price in the path of the source! Hence, by observing both signals and summing them, the source automatically obtains the necessary information for its control, without having to know the type or number of links in its path. As more and more links convert to AQM with ECN, the source algorithm needs no upgrade. The only effect is that queueing delay steadily decreases.

## 6 Simulation results

In this section, we present a simulation experiment to validate the stability results established in Sections 3 and 4.

The experiments all use the topology shown in Figure 4, with a single bottleneck link shared by $N$ TCP sources.

The access links between sources (or destinations) and their router are non-bottlenecks with zero latency, and the link between the two routers is the only bottleneck link with capacity $c$ units and round-trip propagation delay $d$ units. The queueing discipline is FIFO with Droptail and queue capacity is set to $40,000$ packets so




| | capacity c [Gbps] (pkts/ms) | propagation delay d [ms] | expected queue length αN [pkts] | expected window size $\alpha + \frac{cd}{N}$ [pkts] |
|---|---|---|---|---|
| (a) | 0.8 (100) | 10 | 2000 | 30 |
| (b) | 8.0 (1000) | 10 | 2000 | 120 |
| (c) | 0.8 (100) | 100 | 2000 | 120 |

**Fig. 4.** Single bottleneck topology.

the probability of packet loss is negligible. All (data) packets have a size of 1,000 bytes.

We simulate the scenarios of Examples 1 and 2. We fix $\alpha = 20$ packets and $N = 100$ flows, and vary $c$ and $d$, as shown in Table 1. Simulation (a) on Table 1

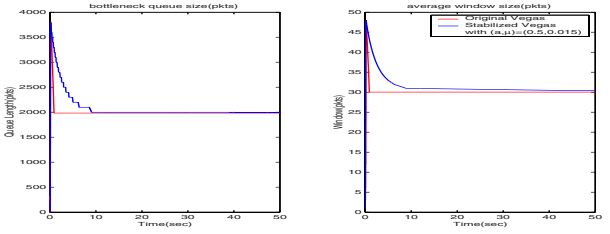
**Table 1.** Simulation: $(c,d)$ combinations in different stability regions; $\alpha = 20$ packets, $N = 100$ flows.

| | $c$ Gbps (pkts/ms) | $d$ ms | queue $\alpha N$ pkts | window $\alpha + cd/N$ pkts |
|---|---|---|---|---|
| (a) | 0.8 (100) | 10 | 2,000 | 30 |
| (b) | 8.0 (1,000) | 10 | 2,000 | 120 |
| (c) | 0.8 (100) | 100 | 2,000 | 120 |

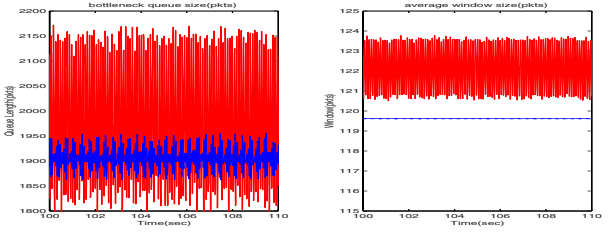
is for small capacity and delay in the stability regions of both Vegas and stabilized Vegas. Simulation (b) scales up the capacity by 10 times, and simulation (c) scales up the delay used in (a) by 10 times. Both (b) and (c) are outside the stability region of the original Vegas, but still in the stability region of stabilized Vegas. The last two columns of Table 1 show the equilibrium queue length and the equilibrium window size calculated from [15].

The simulation results are shown in Figure 5. The first plot of each case shows the total queue length buffered in the bottleneck link. The second plots are the average window size averaged over the $N$ sources.

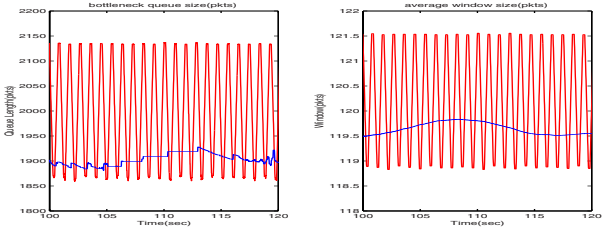
As expected, the original Vegas exhibits instability in cases Fig 5(b) and Fig 5(c), where the stabilized Vegas remains stable.




*(a)* $c = 800$ Mbps (100 pkts/ms) and $d = 10$ms




*(b)* $c = 8$ Gbps (1000 pkts/ms) and $d = 10$ ms




*(c)* $c = 800$ Mbps (100 pkts/ms) and $d = 100$ ms

**Fig. 5.** Simulation: queue length and average window size near equilibrium ($\alpha = 20$ pkts, $N = 100$). For stabilized Vegas, $(a, \mu) = (0.5, 0.015)$.

## 7 Conclusion

In this paper, we have presented a detailed analysis of Vegas stability in a general multi-link multi-source setup with heterogeneous forward and backward delays. We have derived a stability condition that suggests that Vegas can be unstable in the presence of delay. We have proposed a small modification that stabilizes it in the presence of large network delays.

Vegas is particularly attractive for high speed network because of its built-in scalability with network capacity. In the high bandwidth regime, the potential problem

with persistent congestion of Vegas is alleviated. Moreover, it avoids the intrinsic difficulty of having to control based on extremely small loss probability, as Reno must. Despite these advantages, there are challenges associated with delay-based congestion control that must be resolved, especially issues in incremental deployment. We have described one aspect of this: how Vegas source can work gracefully as the network migrates to an ECN-based AQM. Other important aspects have not be studied, such as the interaction of stabilized Vegas with TCP Reno (and its variants such as NewReno and SACK).

**Acknowledgments:** We are grateful to John Doyle, Fernando Paganini, Jiantao Wang, Zhikui Wang, for helpful discussions. We acknowledge the support of Korean Government through BK21 Project, NSF through grant ANI-0113425.

## References

[1] J. S. Ahn, P. B. Danzig, Z. Liu, and L. Yan. Evaluation of TCP Vegas: emulation and experiment. In *Proceedings of SIGCOMM'95*, 1995.

[2] Sanjeewa Athuraliya, Victor H. Li, Steven H. Low, and Qinghe Yin. REM: active queue management. *IEEE Network*, 15(3):48–53, May/June 2001. Extended version in *Proceedings of ITC17*, Salvador, Brazil, September 2001. `http://netlab.caltech.edu`.

[3] Thomas Bonald. Comparison of TCP Reno and TCP Vegas via fluid approximation. In *Workshop on the Modeling of TCP*, December 1998. `http://www.dmi.ens.fr/\%7Emistral/tcpworkshop.html`.

[4] C. Boutremans and J. Y. Le Boudec. A note on the fairness of TCP Vegas. In *Proceedings of International Zurich Seminar on Broadband Communications*, pages 163–170, February 2000.

[5] Lawrence S. Brakmo and Larry L. Peterson. TCP Vegas: end-to-end congestion avoidance on a global Internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–80, October 1995. `http://cs.princeton.edu/nsg/papers/jsac-vegas.ps`.

[6] C. A. Desoer and Y. T. Yang. On the generalized Nyquist stability criterion. *IEEE Trans. on Automatic Control*, 25:187–196, 1980.

[7] U. Hengartner, J. Bolliger, and T. Gross. TCP Vegas revisited. In *Proceedings of IEEE Infocom*, March 2000.

[5] Chris Hollot, Vishal Misra, Don Towsley, and Wei-Bo Gong. A control theoretic analysis of RED. In *Proceedings of IEEE Infocom*, April 2001. `http://www-net.cs.umass.edu/papers/papers.html`.

[9] Frank P. Kelly, Aman Maulloo, and David Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of Operations Research Society*, 49(3):237–252, March 1998.

[10] S. Kunniyur and R. Srikant. A time-scale decomposition approach to adaptive ECN marking. *IEEE Transactions on Automatic Control*, June 2002.

[11] Srisankar Kunniyur and R. Srikant. End–to–end congestion control schemes: utility functions, random losses and ECN marks. In *Proceedings of IEEE Infocom*, March 2000. http://www.ieee-infocom.org/2000/papers/401.ps.

[12] S. H. Low, F. Paganini, J. Wang, and J. C. Doyle. Linear stability of TCP/RED and a scalable control. *Computer Networks Journal*, 43(5):633–647, 2003. http://netlab.caltech.edu.

[13] Steven H. Low. A duality model of TCP and queue management algorithms. *IEEE/ACM Trans. on Networking*, 11(4):525–536, August 2003. http://netlab.caltech.edu.

[14] Steven H. Low and David E. Lapsley. Optimization flow control, I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, December 1999. http://netlab.caltech.edu.

[15] Steven H. Low, Larry Peterson, and Limin Wang. Understanding Vegas: a duality model. *J. of ACM*, 49(2):207–235, March 2002. http://netlab.caltech.edu.

[16] L. Massoulie and J. Roberts. Bandwidth sharing: objectives and algorithms. In *Infocom'99*, March 1999. http://www.dmi.ens.fr/\%7Emistral/tcpworkshop.html.

[17] J. Mo, R. La, V. Anantharam, and J. Walrand. Analysis and comparison of TCP Reno and Vegas. In *Proceedings of IEEE Infocom*, March 1999.

[18] Jeonghoon Mo and Jean Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, October 2000.

[19] Fernando Paganini, John C. Doyle, and Steven H. Low. Scalable laws for stable network congestion control. In *Proceedings of Conference on Decision and Control*, December 2001. http://www.ee.ucla.edu/~paganini.

[20] Fernando Paganini, Zhikui Wang, John C. Doyle, and Steven H. Low. Congestion control for high performance, stability and fairness in general networks. *IEEE/ACM Transactions on Networking, to appear*, 2004.

[21] Glenn Vinnicombe. On the stability of end-to-end congestion control for the Internet. Technical report, Cambridge University, CUED/F-INFENG/TR.398, December 2000.

[22] Glenn Vinnicombe. Robust congestion control for the Internet. submitted for publication, 2002.

# Robust Controller Design for AQM and $\mathcal{H}^\infty$-Performance Analysis

Peng Yan[1] and Hitay Özbay[2]

[1] Department of Electrical & Computer Engineering, The Ohio State University
2015 Neil Ave. Columbus, OH, 43210 yanp@ee.eng.ohio-state.edu
[2] Department of Electrical & Electronics Engineering, Bilkent University
Bilkent, Ankara, Turkey TR-06800
on leave from The Ohio State University ozbay@ee.eng.ohio-state.edu

## 1 Introduction

Active Queue Management (AQM) has recently been proposed in [1] to support the end-to-end congestion control for TCP traffic regulation on the Internet. For the purpose of alleviating congestion for IP networks and providing some notion of quality of service (QoS), the AQM schemes are designed to improve the Internet applications. Earliest efforts on AQM (e.g. RED in [2]) are essentially heuristic without systematic analysis. The dynamic models of TCP ([9, 12]) make it possible to design AQM using feedback control theory. We refer to [11] for a general review of Internet congestion control.

In [12], a TCP/AQM model was derived using delay differential equations. The authors further provided a control theoretic analysis for RED where the parameters of RED can be tuned as an AQM controller [4]. In [5], a Proportional-Integral controller was developed based on the linearized model of [12]. Their controller could ensure robust stability of the closed loop system in the sense of good gain and phase margin of the PI AQM [5, 6]. A challenging nature in the design of AQM is the presence of a time delay which is called *RTT* (round trip time), and the time delays are usually time varying and uncertain. In [14], $\mathcal{H}^\infty$ optimization method was proposed for AQM controller design, which allows for parameter uncertainties of *RTT*, the number of TCP connections and available link capacity. In a similar fashion, we develop in this paper robust AQM controllers based on the $\mathcal{H}^\infty$ control techniques for SISO infinite dimensional systems [3, 16]. However, the model we considered here is a LPV system with *RTT* being the scheduling parameter. We also analyze the $\mathcal{H}^\infty$ performance for the robust controllers with respect to the uncertainty bound of the scheduling parameter *RTT*. Our results show that a smaller operating range of *RTT* results in better $\mathcal{H}^\infty$ performance of the AQM controller, which indicates that switching control among a set of robust controllers designed at selected smaller operating ranges can have better performance than a single $\mathcal{H}^\infty$ controller for the whole range. MATLAB simulations are also given to validate our design and analysis.

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 49–63, 2004.

## 2 Mathematical Model of TCP/AQM

In [12], a nonlinear dynamic model for TCP congestion control was derived, where the network topology was assumed to be a single bottleneck with $N$ homogeneous TCP flows sharing the link. The congestion avoidance phase of TCP can be modeled as AIMD (additive-increase and multiplicative-decrease), where each positive ACK increases the TCP window size $W(t)$ by one per $RTT$ and a congestion indication reduces $W(t)$ by half. Aggregating $N$ TCP flows through one congested router results in the following TCP dynamics [6, 12]:

$$\begin{aligned}\dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)}{2}\frac{W(t-R(t))}{R(t-R(t))}p(t-R(t)) \\ \dot{q}(t) &= [\frac{N(t)}{R(t)}W(t) - C(t)]^+ \end{aligned} \tag{1}$$

where $R(t)$ is the $RTT$, $0 \le p(t) \le 1$ is the marking probability, $q(t)$ is the queue length at the router, and $C$ is the link capacity. Note

$$R(t) = T_p + \frac{q(t)}{C}$$

where $T_p$ is the propagation delay and $q(t)/C$ is the queuing delay.

Assume $N(t) = N$ and $C(t) = C$, the operating point of (1) is defined by $\dot{W} = 0$

$$R_0 = T_p + \frac{q_0}{C} \tag{2}$$

$$W_0 = \frac{R_0 C}{N} \tag{3}$$

$$p_0 = \frac{2}{W_0^2}. \tag{4}$$

Let $\delta q := q - q_0$ and $\delta p := p - p_0$, the linearization of (1) results in the following LPV time delay system, [6],

$$\frac{\delta q(s)}{\delta p(s)} := P_\theta(s) = \frac{K(\theta)e^{-h(\theta)s}}{(T_1(\theta)s+1)(T_2(\theta)s+1)} \tag{5}$$

where

$$K(\theta) = \frac{C^3\theta^3}{4N^2} \tag{6}$$

$$T_1(\theta) = \theta \tag{7}$$

$$T_2(\theta) = \frac{C\theta^2}{2N} \tag{8}$$

$$h(\theta) = \theta \tag{9}$$

and $\theta = R(t) \in [T_p,\ T_p + q_{max}/C]$ is the scheduling parameter of (5) where $q_{max}$ is the buffer size. Note that we employ $\mathcal{L}\{f(t,\theta)|_{\theta=\theta_0}\} = f_{\theta_0}(s)$ to describe the LPV dynamic equations in Laplace domain at fixed parameter values.

## 3 $\mathcal{H}^\infty$ Controller Design for AQM

Consider the nominal system

$$P_0(s) := P_\theta(s)|_{\theta=\theta_0} = \frac{K(\theta_0)e^{-h(\theta_0)s}}{(T_1(\theta_0)s+1)(T_2(\theta_0)s+1)} \tag{10}$$

where $\theta_0 = R_0$ is the nominal *RTT*. We would like to design a robust AQM controller $C_0(s)$ for the nominal plant (10) so that

(i) $C_0(s)$ robustly stabilizes $P_\theta(s)$ for $\forall\theta \in \Theta := [\theta_0 - \Delta\theta,\ \theta_0 + \Delta\theta]$;
(ii) The closed loop nominal system has good tracking of the desired queue length $q_0$ which is a step-like signal.

Notice that the plant (5) can be written as

$$P_\theta(s) = P_0(s)(1 + \Delta P_\theta(s)) \tag{11}$$

where $\Delta P_\theta(s)$ is the multiplicative plant uncertainty.

It can be shown that an uncertainty bound $W_2^{(\theta_0,\Delta\theta)}$ satisfying

$$|\Delta P_\theta(s)|_{s=j\omega} \le |W_2^{(\theta_0,\Delta\theta)}(s)|_{s=j\omega} \quad \forall\omega \in \mathbb{R}^+ \tag{12}$$

is (see details of the derivation in Sect. 4)

$$W_2^{(\theta_0,\Delta\theta)}(s) = a + bs + cs^2 \tag{13}$$

where $a$, $b$ and $c$ are defined in (29). Note that once $\theta_0$ and $\Delta\theta$ are fixed, these coefficients are fixed.

Combining the robust stability and the nominal tracking performance condition, we come up with a two block infinite dimensional $\mathcal{H}^\infty$ optimization problem as follows:

Minimize $\gamma$, such that robust controller $C_0(s)$ is stabilizing $P_0(s)$ and

$$\left\| \begin{bmatrix} W_1(s)S_0(s) \\ W_2^{(\theta_0,\Delta\theta)}(s)T_0(s) \end{bmatrix} \right\|_\infty \le \gamma \tag{14}$$

where

$$\begin{aligned} S_0(s) &= (1 + P_0(s)C_0(s))^{-1} \\ T_0(s) &= 1 - S_0(s) = P_0(s)C_0(s)(1 + P_0(s)C_0(s))^{-1}, \end{aligned}$$

and $W_1(s) = 1/s$ is for good tracking of step-like reference inputs.

By applying the formulae given in [16] and [3], the optimal solution to (14) can be determined as follows [14]:

$$C_0(s) = \frac{\gamma(T_1(\theta_0)s+1)(T_2(\theta_0)s+1)}{cK(\theta_0)s^2} \frac{1}{1 + A(s) + F(s)} \tag{15}$$

where

$$A(s) = \frac{\beta\xi\gamma^2}{s} \tag{16}$$

and $F(s)$ is a finite impulse response (FIR) filter with time domain response

$$f(t) = \begin{cases} (\alpha+\xi-\beta\xi\gamma^2)\cos(\frac{t}{\gamma}) \\ +(\alpha\xi\gamma+\beta\gamma-\frac{1}{\gamma})\sin(\frac{t}{\gamma}) & \text{for } t < h(\theta_0) \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

where

$$\begin{aligned} \beta &= \sqrt{x} \\ \xi &= \frac{1}{c\gamma}\sqrt{\frac{\gamma^2-a^2}{x}} \\ \alpha &= \sqrt{\frac{(b^2-2ac)\gamma^2-c^2}{c^2\gamma^2}+2\sqrt{x}-\frac{\gamma^2-a^2}{c^2\gamma^2 x}} \end{aligned} \tag{18}$$

with $x$ the unique positive root of

$$x^3+\frac{b^2-2ac-a^2\gamma^2}{c^2\gamma^2}x^2-(\gamma^2-a^2)\frac{(2ac-b^2)\gamma^2+c^2}{c^4\gamma^4}x-\frac{(\gamma^2-a^2)^2}{c^4\gamma^4}=0 \tag{19}$$

The optimal $\mathcal{H}^\infty$ performance cost $\gamma$ is determined as the largest root of

$$1-\frac{\gamma}{c}e^{-h(\theta_0)s}\frac{s}{(s+\xi)(s^2+\alpha s+\beta)}\bigg|_{s=\frac{i}{\gamma}}=0 \tag{20}$$

Note that an internally robust digital implementation of the $\mathcal{H}^\infty$ AQM controller (15) includes a second-order term which is cascaded with a feedback block containing an FIR filter $F(s)$. The length of the FIR filter is $h(\theta_0)/T_s$, where $T_s$ is the sampling period.

## 4 Multiplicative Uncertainty Bound

In this section we derive an upper bound for the plant uncertainty. A similar analysis for a different version of the TCP/AQM linear dynamics was done in [14].

Recall (5) and (10), we have

$$
\begin{aligned}
&|P_\theta(s)-P_0(s)|_{s=j\omega}\\
&=\left|\frac{K(\theta)\mathrm{e}^{-h(\theta)s}}{(T_1(\theta)s+1)(T_2(\theta)s+1)}-\frac{K(\theta_0)\mathrm{e}^{-h(\theta_0)s}}{(T_1(\theta_0)s+1)(T_2(\theta_0)s+1)}\right|_{s=j\omega}\\
&=\left|\frac{K(\theta)\mathrm{e}^{-\Delta hs}}{(T_1(\theta)s+1)(T_2(\theta)s+1)}-\frac{K(\theta_0)}{(T_1(\theta_0)s+1)(T_2(\theta_0)s+1)}\right|_{s=j\omega}\\
&\le\left|\frac{|K(\theta)\mathrm{e}^{-\Delta hs}-K(\theta)|+|K(\theta)-K(\theta_0)|}{|(T_1(\theta)s+1)(T_2(\theta)s+1)|}\right|_{s=j\omega}\\
&\quad+\left|\frac{K(\theta_0)-\frac{K(\theta_0)(T_1(\theta)s+1)(T_2(\theta)s+1)}{(T_1(\theta_0)s+1)(T_2(\theta_0)s+1)}}{(T_1(\theta)s+1)(T_2(\theta)s+1)}\right|_{s=j\omega}\\
&\le K(\theta)\left|\frac{\frac{\mathrm{e}^{\Delta hs}-1}{s}}{\frac{(T_1(\theta)s+1)(T_2(\theta)s+1)}{s}}\right|_{s=j\omega}+\left|\frac{\Delta K}{(T_1(\theta)s+1)(T_2(\theta)s+1)}\right|_{s=j\omega}\\
&\quad+K(\theta_0)\left|\frac{(T_1(\theta)T_2(\theta)-T_1(\theta_0)T_2(\theta_0))s^2+(\Delta T_1+\Delta T_2)s}{T(s)}\right|_{s=j\omega}
\end{aligned}
\tag{21}
$$

where

$$T(s)=(T_1(\theta)s+1)(T_2(\theta)s+1)(T_1(\theta_0)s+1)(T_2(\theta_0)s+1),$$

and

$$
\begin{aligned}
\Delta h&=h(\theta)-h(\theta_0),\qquad \Delta K=K(\theta)-K(\theta_0)\\
\Delta T_1&=T_1(\theta)-T_1(\theta_0),\quad \Delta T_2=T_2(\theta)-T_2(\theta_0)
\end{aligned}
$$

Note that

$$\left|\frac{\mathrm{e}^{-\Delta hs}-1}{s}\right|_{s=j\omega}\le|\Delta h|$$

and

$$\left|\frac{(T_1(\theta)s+1)(T_2(\theta)s+1)}{s}\right|_{s=j\omega}\ge\max(T_1^-,T_2^-)$$

where

$$
\begin{aligned}
T_1^-&:=\min\{T_1(\theta),\ \theta\in[T_p,\ T_p+q_{max}/C]\}=T_p,\\
T_2^-&:=\min\{T_2(\theta),\ \theta\in[T_p,\ T_p+q_{max}/C]\}=\frac{CT_p^2}{2N}
\end{aligned}
$$

which are straightforward from (7) and (8). Thus

$$\left|\frac{\frac{\mathrm{e}^{\Delta hs}-1}{s}}{\frac{(T_1(\theta)s+1)(T_2(\theta)s+1)}{s}}\right|_{s=j\omega}\le\frac{|\Delta h|}{\max(T_1^-,T_2^-)}.\tag{22}$$

Recall

$$\begin{aligned}\Delta T_{12} &:= T_1(\theta)T_2(\theta) - T_1(\theta_0)T_2(\theta_0)\\ &= (T_1(\theta_0)+\Delta T_1)(T_2(\theta_0)+\Delta T_2) - T_1(\theta_0)T_2(\theta_0)\\ &= \Delta T_1\Delta T_2 + T_1(\theta_0)\Delta T_2 + T_2(\theta_0)\Delta T_1. \end{aligned} \tag{23}$$

We have

$$\begin{aligned}&\left|\frac{\Delta T_{12}s^2+(\Delta T_1+\Delta T_2)s}{T(s)}\right|_{s=j\omega}\\ \le &\left|\frac{|\Delta T_{12}s^2|+|(\Delta T_1+\Delta T_2)s|}{|T(s)|}\right|_{s=j\omega}\\ \le &\left|\frac{|\Delta T_1\Delta T_2|+|T_1(\theta_0)\Delta T_2|+|T_2(\theta_0)\Delta T_1|}{|\frac{(T_1(\theta)s+1)(T_2(\theta)s+1)(T_1(\theta_0)s+1)(T_2(\theta_0)s+1)}{s^2}|}\right|_{s=j\omega} + \frac{|\Delta T_1+\Delta T_2|}{\max(T_1(\theta_0),T_2(\theta_0))}\\ \le &\frac{|\Delta T_1\Delta T_2|+|T_1(\theta_0)\Delta T_2|+|T_2(\theta_0)\Delta T_1|}{T_1(\theta_0)T_2(\theta_0)} + \frac{|\Delta T_1|+|\Delta T_2|}{\max(T_1(\theta_0),T_2(\theta_0))}\\ \le &\frac{|\Delta T_1|}{T_1(\theta_0)} + \frac{|\Delta T_2|}{T_2(\theta_0)} + \frac{|\Delta T_1\Delta T_2|}{T_1(\theta_0)T_2(\theta_0)} + \frac{|\Delta T_1|+|\Delta T_2|}{\max(T_1(\theta_0),T_2(\theta_0))}\end{aligned}$$

Invoking (21) and (22), we have

$$\begin{aligned}&|P_\theta(s)-P_0(s)|_{s=j\omega}\\ \le\ &K(\theta)\frac{|\Delta h|}{\max(T_1^-,T_2^-)} + |\Delta K| + K(\theta_0)\Big(\frac{|\Delta T_1|}{T_1(\theta_0)} + \frac{|\Delta T_2|}{T_2(\theta_0)}\\ &+\frac{|\Delta T_1\Delta T_2|}{T_1(\theta_0)T_2(\theta_0)} + \frac{|\Delta T_1|+|\Delta T_2|}{\max(T_1(\theta_0),T_2(\theta_0))}\Big)\end{aligned} \tag{24}$$

Defining

$$K^+ := \max\{K(\theta),\ \theta\in[T_p,\ T_p+q_{max}/C]\} = \frac{(CT_p+q_{max})^3}{4N^2},$$

and assuming

$$\begin{aligned}\Big|\frac{dh(\theta)}{d\theta}\Big| \le \beta_h \quad \Big|\frac{dT_1(\theta)}{d\theta}\Big| \le \beta_{T_1}\\ \Big|\frac{dT_2(\theta)}{d\theta}\Big| \le \beta_{T_2} \quad \Big|\frac{dK(\theta)}{d\theta}\Big| \le \beta_K,\end{aligned} \tag{25}$$

the additive uncertainty (24) can be rewritten as

$$\begin{aligned}&|P_\theta(s)-P_0(s)|_{s=j\omega} \le \Delta_{(\theta_0,\Delta\theta)}\\ :=\ &\frac{K(\theta_0)\beta_{T_1}\beta_{T_2}}{T_1(\theta_0)T_2(\theta_0)}(\Delta\theta)^2 + \Big(\frac{K^+\beta_h}{\max(T_1^-,T_2^-)} + \beta_K\\ &+\ \frac{K(\theta_0)\beta_{T_1}}{T_1(\theta_0)} + \frac{K(\theta_0)\beta_{T_2}}{T_2(\theta_0)} + \frac{K(\theta_0)(\beta_{T_1}+\beta_{T_2})}{\max(T_1(\theta_0),T_2(\theta_0))}\Big)\Delta\theta.\end{aligned} \tag{26}$$

With (11) and (26), the multiplicative uncertainty $\Delta P_\theta(s)$ can be bounded by

$$|\Delta P_\theta(s)|_{s=j\omega} \le \Delta_{(\theta_0,\Delta\theta)} |P_0(s)^{-1}|_{s=j\omega} = |W_2^{(\theta_0,\Delta\theta)}(s)|_{s=j\omega} \tag{27}$$

where

$$W_2^{(\theta_0,\Delta\theta)}(s) = a + bs + cs^2 \tag{28}$$

with

$$\begin{aligned} a &= \frac{\Delta_{(\theta_0,\Delta\theta)}}{K(\theta_0)} \\ b &= \frac{\Delta_{(\theta_0,\Delta\theta)}(T_1(\theta_0) + T_2(\theta_0))}{K(\theta_0)} \\ c &= \frac{\Delta_{(\theta_0,\Delta\theta)} T_1(\theta_0) T_2(\theta_0)}{K(\theta_0)}. \end{aligned} \tag{29}$$

## 5 $\mathcal{H}^\infty$-Performance Analysis

As shown in Sect.3, the $\mathcal{H}^\infty$ AQM controller (15) is designed for $P_\theta(s)|_{\theta=\theta_0}$ and allows for $\theta \in \Theta = [\theta - \Delta\theta,\ \theta + \Delta\theta]$. In this section, we would like to investigate the $\mathcal{H}^\infty$-performance for the corresponding closed loop system, which indicates the system robustness and system response.

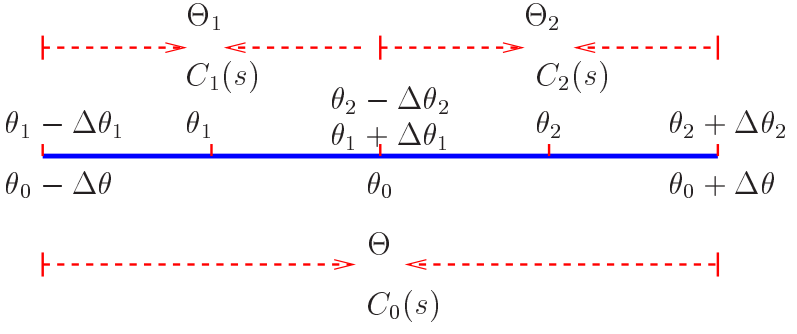



**Fig. 1.** Partition of $\Theta$ by $\Theta_1$ and $\Theta_2$

Define the $\mathcal{H}^\infty$-performance of controller $C_0(s)$ with respect to $P_\theta(s)$ as follows:

$$\gamma_{C_0}(\theta) = \left\| \begin{bmatrix} W_1(s)S(s) \\ W_2^{(\theta_0,\Delta\theta)}(s)P_0(s)C_0(s)S(s) \end{bmatrix} \right\|_\infty \tag{30}$$

for any $\theta \in \Theta = [\theta_0 - \Delta\theta,\ \theta_0 + \Delta\theta]$, where

$$S(s) = (1 + P_\theta(s)C_0(s))^{-1}, \tag{31}$$

here the term $|W_2^{(\theta_0,\Delta\theta)}(j\omega)P_0(j\omega)|$ can be seen as a bound on the additive plant uncertainty.

Furthermore, we define

$$\gamma_{C_0}^{\Delta\theta} := \sup_{\theta\in\Theta}\{\gamma_{C_0}(\theta)\} \tag{32}$$

which corresponds to the worst system response of controller $C_0(s)$ for plant $P_\theta(s)$ with $\forall\theta \in [\theta_0-\Delta\theta,\ \theta_0+\Delta\theta]$. Notice that a smaller $\gamma_{C_0}^{\Delta\theta}$ means better performance of the robust controller within the operating range $\Theta$.

Particularly, we are interested in the scenario depicted in Fig. 1, where $\Theta$ is equally partitioned by $\Theta_1 = [\theta_1-\Delta\theta_1,\ \theta_1+\Delta\theta_1]$ and $\Theta_2 = [\theta_2-\Delta\theta_2,\ \theta_2+\Delta\theta_2]$, with $\Delta\theta_1 = \Delta\theta_2 = \frac{\Delta\theta}{2}$. For $\theta\in\Theta_i$, $i=1,2$, we design $\mathcal{H}^\infty$ controller $C_i(s)$ obeying (15) with the nominal plant $P_i(s) := P_\theta(s)|_{\theta=\theta_i}$. Similar to (30) and (32), we have

$$\gamma_{C_i}(\theta) = \left\| \begin{bmatrix} W_1(s)S_i(s) \\ W_2^{(\theta_i,\Delta\theta_i)}(s)P_i(s)C_i(s)S_i(s) \end{bmatrix} \right\|_\infty \tag{33}$$

for any $\theta\in\Theta_i\ \ i=1,2$, and

$$\gamma_{C_i}^{\Delta\theta_i} := \sup_{\theta\in\Theta_i}\{\gamma_{C_i}(\theta)\}\quad i=1,2 \tag{34}$$

where $S_i(s) = (1+P_\theta(s)C_i(s))^{-1}$ is defined similarly to (31).

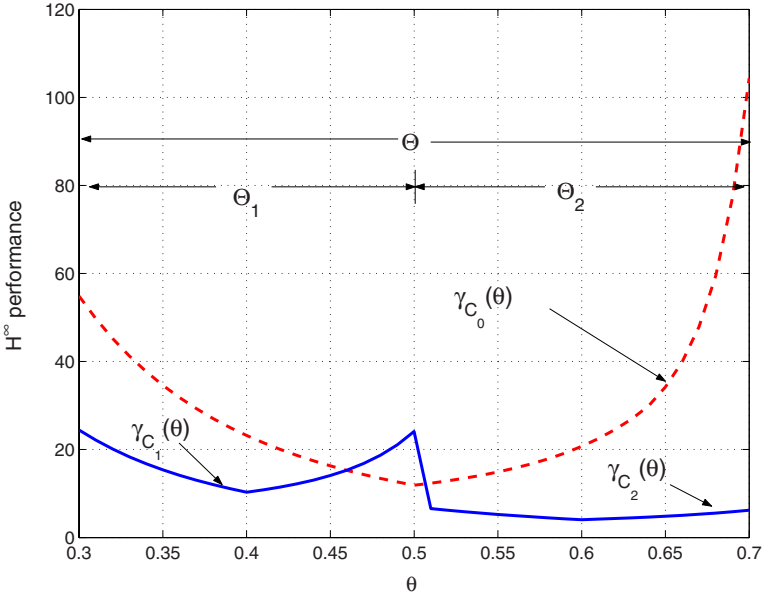



**Fig. 2.** $\mathcal{H}^\infty$ performance with respect to $\theta$

In what follows, we provide numerical analysis of the $\mathcal{H}^\infty$-performance with respect to the operating ranges and corresponding controllers shown in Fig. 1. Assume $N = 150$, $C = 500$, $\Delta\theta = 0.2$, and $\theta_0 = 0.5$, the $\mathcal{H}^\infty$ performance $\gamma_{C_0}(\theta)$ and $\gamma_{C_i}(\theta)$, $i = 1,2$ can be numerically obtained from (30) and (33). As depicted in Fig. 2, it is straightforward to have

$$\max(\gamma_{C_1}^{\Delta\theta_1}, \gamma_{C_2}^{\Delta\theta_2}) = 24.4 < \gamma_{C_0}^{\Delta\theta} = 104.4$$

which means that the partition of Fig. 1 can improve system performance in the sense of smaller $\mathcal{H}^\infty$-performance cost. In fact, it is a general trend that

$$\max(\gamma_{C_1}^{\Delta\theta_1}, \gamma_{C_2}^{\Delta\theta_2}) < \gamma_{C_0}^{\Delta\theta}, \tag{35}$$

which can be further verified by Fig. 3, Fig. 4, and Fig. 5, where $N$ is chosen from 100 to 200, $C$ from 400 to 600.

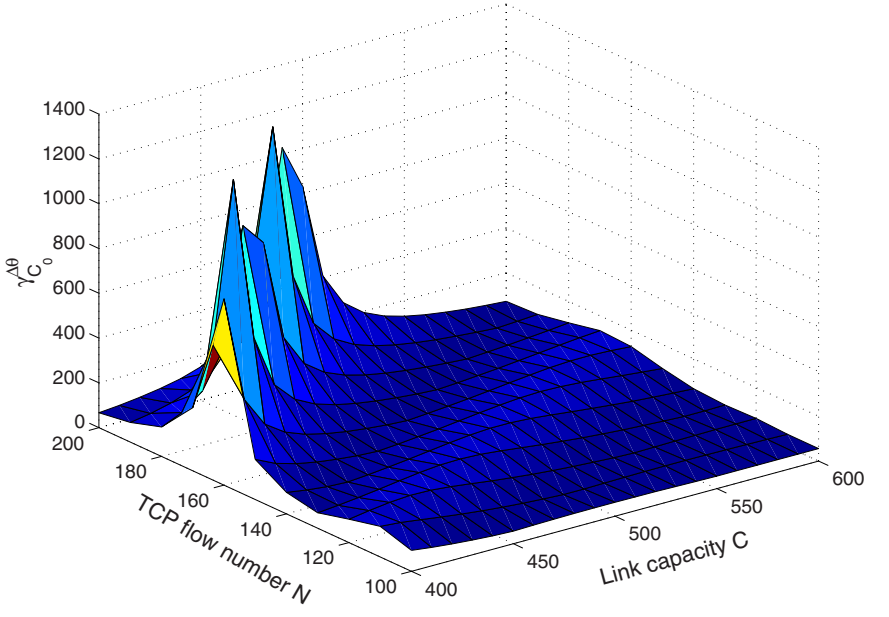



**Fig. 3.** Performance cost $\gamma_{C_0}^{\Delta\theta}$ w.r.t. $N$ and $C$

Based on the observation of better performance obtained by the partition shown in Fig. 1, it is natural to consider switching robust control among a set of $\mathcal{H}^\infty$ controllers, each of which is designed for a smaller operating range. We provide in Sect. 6 the simulation results of switching control between two robust controllers.

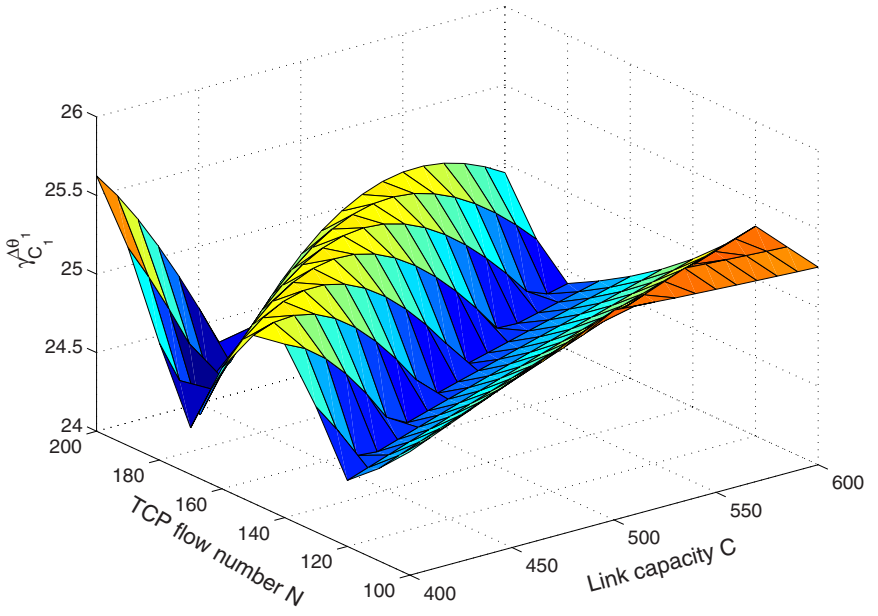



**Fig. 4.** Performance cost $\gamma_{C_1}^{\Delta\theta_1}$ w.r.t. $N$ and $C$

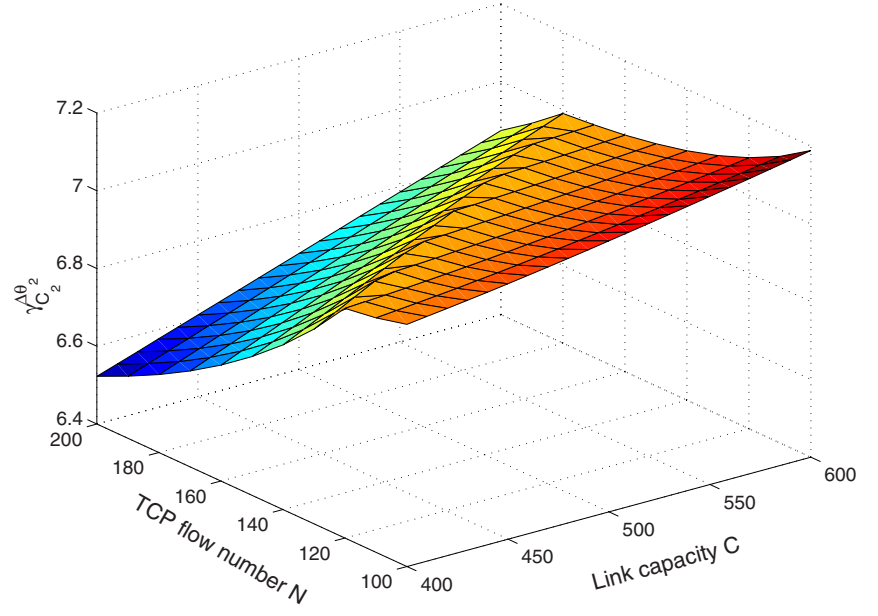



**Fig. 5.** Performance cost $\gamma_{C_2}^{\Delta\theta_2}$ w.r.t. $N$ and $C$

## 6 Simulations

The closed loop system with the determined controllers is implemented in MATLAB to validate the controller design as well as the $\mathcal{H}^\infty$ performance analyzed in previous sections. We assume the TCP flow number $N = 150$, the link capacity $C = 500$ packets/sec. The propagation delay $T_p$ is set to be 0.3 sec and the desired queue size is $q_0 = 100$ packets. Therefore, the nominal $RTT$ is 0.5 sec ($\theta_0 = 0.5$), which is straightforward from (2).

### 6.1 The Case of a Single Controller

We use $\Delta\theta = 0.2$ in the design of $C_0(s)$ and $\Delta\theta_1 = \Delta\theta_2 = 0.1$ in $C_1(s)$ and $C_2(s)$. The following three scenarios are considered:

- Assuming the plant is the nominal one, i.e. $P_\theta(s) = P_0(s)$, we implement controller $C_0(s)$ as well as $C_1(s)$ and $C_2(s)$. It is shown in Fig. 6 that the three controllers can stabilize the queue length because the nominal value $\theta_0$ is within the operating range of $\Theta$, $\Theta_1$, and $\Theta_2$. Note that the system response of $C_0(s)$ is better than the other two due to the fact that it achieves the optimal $\mathcal{H}^\infty$-performance at $\theta_0$.

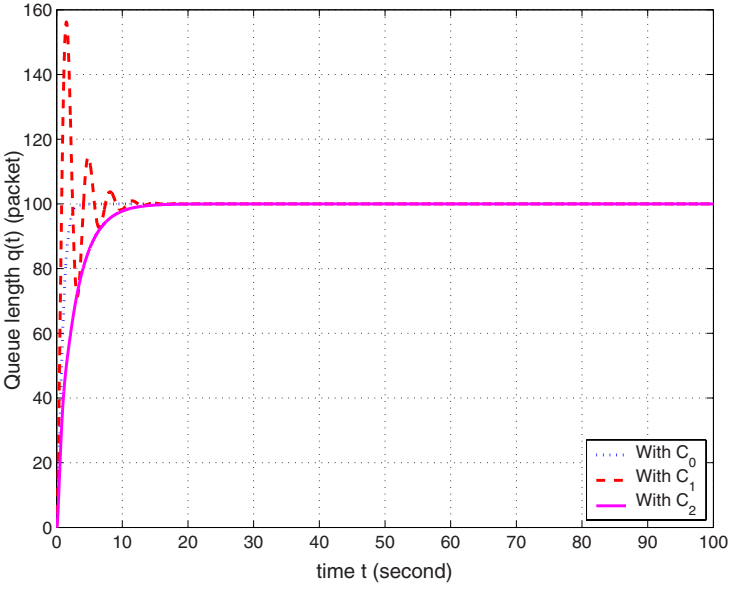



**Fig. 6.** System responses of $C_0$, $C_1$ and $C_2$ at $\theta = \theta_0 = 0.5$

- Assuming $\theta = \theta_0 - \Delta\theta = 0.3$, we implement controller $C_0$ and $C_1$ ($C_2$ is not eligible in this scenario). As depicted in Fig. 7, $C_0$ and $C_1$ can robustly stabilize

the queue length. Observe that the system response of $C_1$ is better because it has much smaller $\mathcal{H}^\infty$ performance cost, which has been shown in Sect. 5.

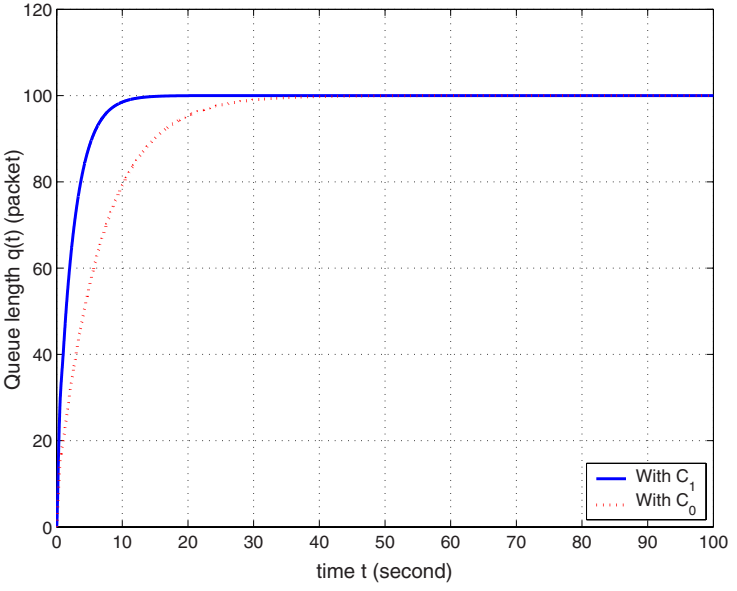



**Fig. 7.** System responses of $C_0$ and $C_1$ at $\theta = \theta_0 - \Delta\theta = 0.3$

- Similarly, we choose $\theta = \theta_0 + \Delta\theta = 0.7$ and repeat the simulation for controller $C_0$ and $C_2$ ($C_1$ is not eligible). As depicted in Fig. 8, the two controllers can robustly stabilize the queue length and their system responses coincide with the $\mathcal{H}^\infty$-performance analysis given previously.

The above simulations show that the proposed robust AQM controllers have good performance and robustness in the presence of parameter uncertainties. Meanwhile, the system responses also affirm a good coincidence with the $\mathcal{H}^\infty$ performance analysis in Sect. 5.

### 6.2 The Case of Switching Control

Motivated by the analysis in Sect. 5, we perform control switching in this experiment. We assume the same simulation configuration as Sect. 6.1 and investigate the closed loop system performance in the presence of switching between $\mathcal{H}^\infty$ controller $C_1$ and $C_2$ for a slow time varying signal $\theta(t) \in \Theta$. For the purpose of comparison, we also provide the system response with a single $\mathcal{H}^\infty$ controller $C_0$. As depicted in Fig. 9 and Fig. 10, the switching control method has better transient behavior in terms of smaller overshoot, faster convergence and less oscillations. Note that the large oscillations around 90 *sec* on both plots are due to the fact that $\theta$ is not assumed to

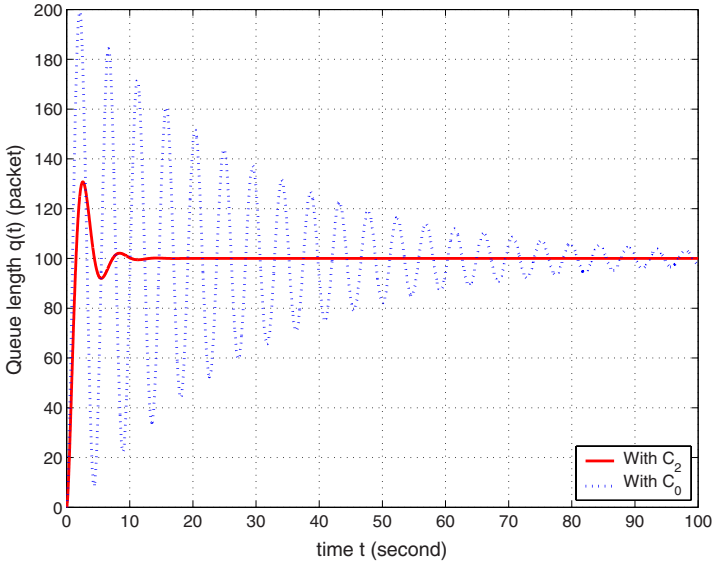



**Fig. 8.** System responses of $C_0$ and $C_2$ at $\theta = \theta_0 + \Delta\theta = 0.7$

be time varying in the proposed design. Instead, we assume it is piece-wise constant but uncertain in the derivation of the system uncertainty bound (see Section 3 and 4 for details).

## 7 Conclusions

We provided in this paper the guidelines of designing robust controllers for AQM, where the $\mathcal{H}^\infty$ techniques for infinite dimensional systems were implemented. The $\mathcal{H}^\infty$-performance was numerically analyzed with respect to the bound of the scheduling parameter $\theta$. It was shown that smaller uncertainty bound could result in better $\mathcal{H}^\infty$-performance of the corresponding closed loop systems. Accordingly, we proposed switching control between two robust controllers which outperforms a single controller. Simulations were conducted to validate the design and analysis.

## References

[1] Braden B, Clark D, Crowcroft J, Davie B, Deering S, Estrin D, Floyd S, Jacobson V, Minshall G, Partridge C, Peterson L, Ramakrishnan K, Shenker S, Wroclawski J, Zhang L (1998) Recommendations on queue management and congestion avoidance in the internet. RFC 2309

[2] Floyd S, Jacobson V (1993) Random Early Detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, vol. 1, No. 4, pp. 397-413

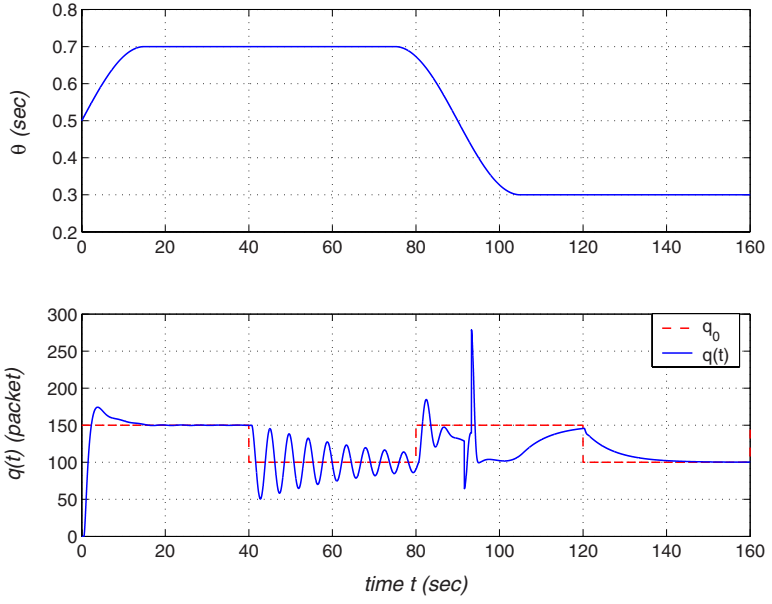



**Fig. 9.** A single controller $C_0$

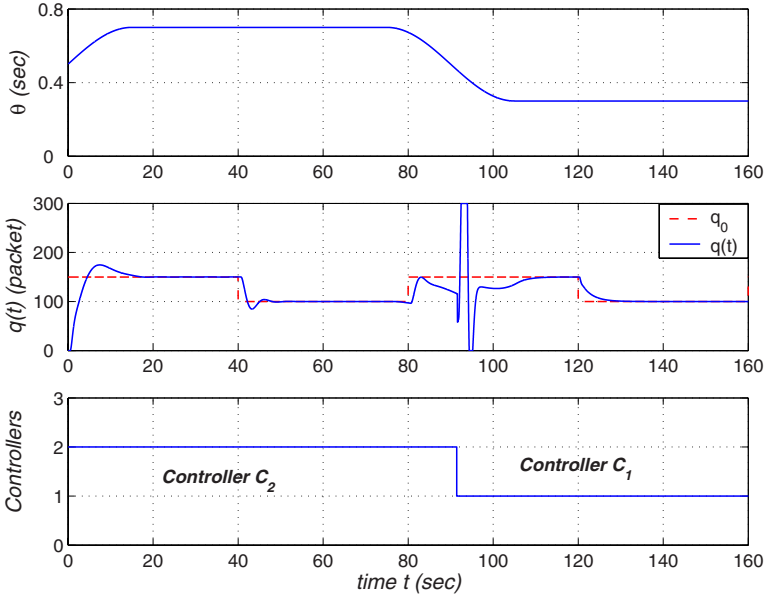



**Fig. 10.** Switching control between $C_1$ and $C_2$

[3] Foias C, Özbay H, Tannenbaum A (1996) Robust Control of Infinite Dimensional Systems: Frequency Domain Methods. Lecture Notes in Control and Information Sciences, No. 209, Springer-Verlag, London
[4] Hollot C, Misra V, Towsley D, Gong W (2001) A control theoretic analysis of RED. *Proc. of IEEE INFOCOM'01*, Alaska, USA
[5] Hollot C, Misra V, Towsley D, Gong W (2001) On designing improved controllers for AQM routers supporting TCP flows. *Proc. of IEEE INFOCOM'01*, Alaska, USA
[6] Hollot C, Misra V, Towsley D, Gong W (2002) Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Trans. on Automatic Control*, vol. 47, pp. 945-959
[7] Jacobson V, Karels M (1998) Congestion avoidance and control. *Proc. of ACM SIGCOMM'88*, CA, USA
[8] Johari R, Tan D (2001) End-to-end congestion control for the internet: delays and stability. *IEEE/ACM Trans. on Networking*, vol. 9, pp. 818-832
[9] Kelly F (2001) Mathematical modeling of the Internet. In: Engquist B, Schmid W (eds) Mathematics Unlimited-2001 and Beyond. Springer-Verlag, Berlin
[10] Lee S.-H, Lim J.T (2000) Switching control of $\mathcal{H}^\infty$ gain scheduled controllers in uncertain nonlinear systems. *Automatica*, vol. 36, pp. 1067-1074
[11] Low S, Paganini F, Doyle J (2002) Internet congestion control. *IEEE Control System Magazine*, vol. 22, pp. 28-43
[12] Misra V, Gong W, Towsley D (2000) Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. *Proc. of ACM SIGCOMM'00*, Stockholm, Sweden
[13] Quet P.-F, Ataşlar B, İftar A, Özbay H, Kang T, Kalyanaraman S (2002) Rate-based flow controllers for communication networks in the presence of uncertain time-varying multiple time-delays. *Automatica*, vol. 38, pp. 917-928
[14] Quet P.-F, Özbay H (2003) On the robust controller design for Active Queue Management scheme supporting TCP flows. *Proceedings of the 42nd IEEE Conference on Decision and Control*, Hawaii, USA, pp. 4220-4224. Also to appear in *IEEE Trans. on Automatic Control*, 2004
[15] Rugh W, Shamma J (2000) Research on gain scheduling. *Automatica*, vol. 36, pp. 1401-1425
[16] Toker O, Özbay H (1995) $\mathcal{H}^\infty$ optimal and suboptimal controllers for infinite dimensional SISO plants. *IEEE Trans. on Automatic Control*, vol. 40, pp. 751-755

# Models and Methods for Analyzing Internet Congestion Control Algorithms

Rayadurgam Srikant

Department of Electrical and Computer Engineering and
Coordinated Science Lab
University of Illinois at Urbana-Champaign
Urbana, IL 61801
rsrikant@uiuc.edu

## 1 Introduction

Congestion control in the Internet was introduced in the late 1980s by Van Jacobson [9]. Jacobson's algorithm which is implemented in the transport layer protocol called TCP (Transmission Control Protocol) has served the Internet well during a time of unprecedented growth. However, the algorithm was designed during a time when the Internet was a relatively small network compared to its size today. Therefore, there has been much interest in reexamining the role of congestion control in the Internet with the goal of enhancing TCP to make it scalable to large networks.

We begin this survey with a simple model of a particular version of Jacobson's algorithm called TCP-Reno. We show that the algorithm may not be stable when the network size becomes large, i.e., if the feedback delays are large or the capacity of the network is large. We then present Kelly's model where congestion control is viewed as a distributed control algorithm for achieving fair resource allocation in a network with many users. The remarkable feature of the algorithm is that the complex interactions between the users in the network can be captured using quantities that can be measured easily by each individual user. Next, we study the stability of this congestion control mechanism in the presence of feedback delay. We primarily concentrate on a new technique using Razumikhin's theorem that proves global stability of the congestion controllers under some assumptions on the control parameters. The interested reader may refer to the results in [10, 17, 22, 24, 28, 29] where local stability results are obtained without restrictions on the network parameters. However, the region of attraction for these local stability results is yet to be established for a general topology network. Finally, we turn our attention to connection-level models of congestion control. In traditional congestion control models, the number of users on the various routes in the network is assumed to be fixed. In the connection-level model, the number of files in the network is time-varying and is described by a Markov chain. Between the time instants when there is a file arrival or departure, it is assumed that congestion control occurs instantaneously. Thus, the connection-level

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 65–86, 2004.

model operates at a slower time-scale compared to the time required for congestion controllers to converge. Under this assumption of time-scale separation, we show that the Kelly model of resource allocation is efficient, i.e., if the total load on each link in the network is less than its capacity, then the network is stable.

The rest of this paper is organized in follows. In Section 2, we present a simple model for Jacobson's algorithm that was introduced in [27] and analyze its local stability in the presence of feedback delay. We summarize the results in [25, 27] which show that TCP-Reno is not stable when the capacity per user is large or if the feedback delay is large. In Section 3, we introduce the Kelly resource allocation model [11] and the associated congestion control algorithms [13, 20]. A stability analysis of the congestion control algorithm in the presence of feedback delays is presented in Section 4. This stability result was obtained in [32] using Razumikhin theorem. A related result has also been obtained in [18] using different techniques. In Section 5, we present the connection-level model introduced in [26] and present a stability result using a drift analysis. This result which was first established in [4] using fluid limits and a special case of this result was presented earlier in [6]. Finally, concluding remarks are presented in Section 6. The survey presented here covers only a small portion of the considerable research on Internet Congestion Control that has taken place over the last few years. For a more comprehensive survey, we refer the reader to [27].

## 2 A Simple Model for TCP-Reno

Congestion control is implemented in the Internet using a *window flow control* algorithm. A source's *window* is the maximum number of unacknowledged packets that the source can inject into the network at any time. For example, if the window size is 1, then the source maintains a counter which has a maximum value of 1. The counter indicates the number of packets that it can send into the network. The counter's value is initially equal to the window size. When the source sends one packet into the network, the counter is reduced by 1. Thus, the counter in this example would become zero after each packet transmission and the source cannot send any more packets into the network till the counter hits 1 again. To increment the counter, the source waits for the destination to acknowledge that it has received the packet. This is accomplished by sending a small packet called the *ack* packet, from the destination back to the source. Upon receiving the ack, the counter is incremented by 1 and thus, the source can again send one more packet. We use the term *round-trip time (RTT)* to refer to the amount of time that elapses between the instant that the source transmits a packet and the instant at which it receives the acknowledgment for the packet. The RTT consists of three components: the propagation delay of the packet through the transmission medium (which is determined by the distance between the source and destination), the queueing delay at the routers in the network and the time taken to process a packet at the routers in the network. Typically, the processing time is negligible compared to the other two components. With a window size of 1, since one

packet is transmitted during every RTT, the source's data transmission rate is $1/RTT$ packets/sec.

If the window is 2, the counter's value is initially set to 2. Thus, the source can send two back-to-back packets into the network. For each transmitted packet, the counter is decremented by 1. Thus, after the first two packet transmissions, the counter is decremented to zero. When one of the packets is acknowledged and the ack reaches the source, then the source increments the counter by 1 and can send one more packet into the network. Once the new packet is transmitted, the counter is again decremented back to zero. Thus, after each ack, one packet is sent, and then, the source has to wait for the next ack before it can send another packet. If one assumes that the processing speed of the link is very fast and that the processing times at the source and destination are negligible, then the source can transmit two packets during every RTT. Thus, the source's transmission rate is $2/RTT$ packets/sec. From the above argument, it should be clear that, if the window size is $W$, then the transmission rate can be approximated by $W/RTT$ packets/sec.

If the link capacity is $c$ and the source's window size $W$ is such that $W/RTT < c$, then the system will be stable. In other words, all transmitted packets will be eventually processed by the link and reach the intended destination. However, in a general network, the available capacity cannot be easily determined by a source. The network is shared by many sources which share the capacities at the various links in the network. Thus, each source has to adaptively estimate the value of the window size that can be supported by the network. The most widely-used algorithm for this purpose in the Internet today is called TCP-Reno.

The TCP-Reno algorithm is quite complicated and therefore, for our modelling purposes, we consider the following simplified version of the algorithm. Assume that there is a mechanism for the receiver to indicate to the source that a packet has been lost in the network. Then, the essential features of the TCP-Reno algorithm can be summarized as below:

- Upon receipt of a ack, the source increases its current window size, denoted by $cwnd$, as follows:
$$cwnd \leftarrow cwnd + 1/cwnd.$$
- Upon being informed of a loss, the source decreases its window size by a factor of two:
$$cwnd \leftarrow cwnd/2.$$

The key feature of TCP-Reno is that it increases its window size when it does not detect congestion which is indicated by the reception of an ack, and it decreases its window size upon detecting congestion, which is indicated by the detection of a lost packet.

We now present a differential equation model that describes the TCP-Reno congestion control algorithm. Consider $N$ TCP-Reno sources, all with the same RTT, accessing a single link. Let $W_r(t)$ denote the window size of flow $r$, $T$ be its RTT, and $q(t)$ be the fraction of packets lost at the link at time $t$. Then, the congestion avoidance phase of TCP-Reno can be modelled as

$$\dot{W}_r = \frac{x_r(t-T)(1-q(t-T))}{W_r} - \beta x_r(t-T)q(t-T)W_r(t), \tag{1}$$

where $x_r(t) = W_r(t)/T$ is the transmission rate. The parameter $\beta$ is the decrease factor and is taken to be $1/2$ although studies show that a more precise value of $\beta$ when making a continuous-time approximation of TCP's behavior is closer to $2/3$. Substituting for $W_r(t)$ in terms of $x_r(t)$ gives

$$\dot{x}_r = \frac{x_r(t-T)(1-q(t))}{T^2 x_r} - \beta x_r(t-T)q(t)x_r(t). \tag{2}$$

The loss probability $q(t)$ is a function of the arrival rate at the link. Thus, let

$$q(t) = f(y(t-T)),$$

where $f(\cdot)$ is an increasing function and $y(t)$ is the total arrival rate at the link and is given by

$$y(t) = \sum_{r=1}^{N} x_r(t).$$

The equilibrium value of $x_r$ is easily seen to be

$$\hat{x}_r = \sqrt{\frac{1-\hat{q}}{\beta\hat{q}}}\frac{1}{T}, \tag{3}$$

where $\hat{q}$ is the equilibrium loss probability. We note that throughout this paper, we use $\hat{}$ to denote equilibrium values. The functional form of $f(y)$ could be quite complicated in general. Among other things, it will depends on the assumptions on the stochastic behavior of the packet arrival process at the router. To simplify the analysis, we will assume that $f(y)$ is of the form suggested in [16]:

$$f(y) = \left(\frac{y-c}{y}\right)^+.$$

Thus, this form of $f(y)$ can be interpreted as a fluid approximation to the loss probability: it is equal to zero if the arrival rate is less than the capacity of the link and is otherwise equal to the fraction by which the arrival rate exceeds the link capacity. Recall that the RTT $T$ consists of two components, namely the propagation delay $T_p$ and the queueing delay at the router. Just like the loss probability, it is difficult to precisely capture the queueing delay using a simple analytical formula. To obtain a tractable expression for the queueing delay, we recall that the TCP-Reno protocol attempts to fill up the buffer at the router and uses the resulting packet loss to obtain congestion information. Therefore, it seems reasonable to assume that the queue is full most of time. Under this assumption, our approximation to the queueing delay takes the form $B/c$, where $B$ is the buffer size at the router. Thus, for all users, the RTT is given by

$$T = T_q + \frac{B}{c}.$$

To study the stability of the congestion controller given in (2), we first linearize the system around its equilibrium point. Defining $\delta x_r = x_r - \hat{x}_r$, and $\delta q = q - \hat{q}$, the linearized form of the congestion control algorithm is given by

$$\dot{\delta x_r} = \hat{x}_r \left( \frac{1-\hat{q}}{T^2 \hat{x}_r^2} \delta x_r + \frac{1}{T^2 \hat{x}_r^2} \delta q + \beta \hat{q} \delta x_r + \beta \hat{x}_r \delta q \right),$$

and

$$\delta q = \frac{c}{\hat{y}^2} \sum_r \delta x_r (t - T).$$

Defining $\delta y = y - \hat{y}$, and using the equilibrium relationship (3) yields

$$\dot{\delta x_r} + \alpha_1 \delta x_r + \alpha_2 \delta x_r (t - T) = 0, \tag{4}$$

where

$$\alpha_1 = 2\beta \hat{q} \hat{x}_r, \qquad \alpha_2 = \beta \hat{x}_r.$$

From Hayes' lemma [7], the linearized delay-differential equation describing TCP-Reno's dynamics is stable if one of the following conditions is satisfied:

- $\alpha_1 \geq \alpha_2$,
- $\alpha_1 < \alpha_2$ and

$$\alpha_2 T \sqrt{1 - \frac{\alpha_1^2}{\alpha_2^2}} < \arccos\left( -\frac{\alpha_1}{\alpha_2} \right).$$

For the first condition to be satisfied, we require $\hat{q} \geq 1/2$. This is not a practical scenario since it requires at least half the packets to be dropped at the router. The second condition can be written as

$$\frac{c}{N} T < \frac{1}{\beta} \frac{(1-\hat{q}) \arccos(-2\hat{q}_r)}{\sqrt{1 - 4\hat{q}^2}}. \tag{5}$$

Note that the equilibrium relationship (3) can be rewritten as

$$\frac{(1-\hat{q})^3}{\hat{q}} = \left( \frac{c}{N} T \right)^2.$$

If we let $c/N$ (which is simply the capacity per user) be a constant and increase the RTT, then it is clear from the previous equation that $\hat{q}$ must decrease. Thus, for large $T$, the right-hand side of the stability condition can be approximated by letting $\hat{q} = 0$ which gives the following condition for stability

$$\frac{c}{N} T < \frac{\pi}{2\beta}.$$

Clearly, this condition will be violated as $T$ increases or $c/N$ increases. From the above analysis, we can conclude that TCP-Reno is not a scalable protocol, i.e., its stability is compromised if either the RTT of the users is large or if the available capacity per user at the router is large. Related linear analysis of TCP dynamics can be found in [8, 23]. In the following sections, we will examine alternatives to TCP-Reno as a congestion control mechanism.

## 3 Resource Allocation and Congestion Control

Consider a network consisting of a set of links denoted by $\mathcal{L}$ and a set of users denoted by $\mathcal{R}$. User $r$ chooses a collection of links called a route to transmit its packets from a source to a destination. We will associate a unique route with a user and therefore, an index $r$ can be used to represent both a user and the route used by the user. The data transmission rate of user $r$ is denoted by $x_r$. The capacity of each link $l$ is denoted by $c_l$. Since the total arrival rate on each link should be less than or equal to the capacity of the link, we have following constraints:

$$\sum_{r:l\in r} x_r \le c_l, \qquad \forall l, \tag{6}$$

where $l \in r$ is a notation for "link $l$ belongs to route $r$." The resource allocation problem in the Internet is to find a set of rates $\{x_r\}$ which satisfy the constraints (6). However, without additional constraints, typically there would be many solutions to this problem and it would be difficult to distinguish between them. For example, consider a single link of unit capacity shared by two users. Then, any set of non-negative values $x_1, x_2$ satisfying $x_1 + x_2 \le 1$ would be feasible. Even if we impose the natural condition that the link should be fully utilized, we still have an infinite number of solutions to $x_1 + x_2 = 1$. Among the many solutions, the particular solutions $x_1 = 0, x_2 = 1$ or $x_1 = 1, x_2 = 0$ are especially unappealing since they are severely unfair to one of the users. Thus, a reasonable goal to aim for in resource allocation is *fairness*.

To define a fair resource allocation problem, associate with each $r$ a utility function $U_r(x_r)$ which indicates the utility to user $r$ when it is allowed to transmit at rate $r$. Then, the fair resource allocation problem is to maximize

$$\max_{\{x_r\}} \sum_r U_r(x_r). \tag{7}$$

Thus, in this problem, a resource allocation is said to be fair if it maximizes the sum of the utilities of all the users subject to the link capacity constraints (6) and the obvious non-negativity constraints

$$x_r \ge 0, \qquad \forall r. \tag{8}$$

In the rest of this section, we will interpret congestion control as a decentralized solution to the resource allocation problem (7).

Instead of solving the resource allocation problem (7) exactly, let us consider the following objective:

$$V(\mathbf{x}) = \sum_r U_r(x_r) - \sum_l \int_0^{\sum_{s:l\in s} x_s} f_l(y)dy, \tag{9}$$

where we interpret $f_l(\cdot)$ as the *penalty function or the barrier function* [3] as the case may be, or simply as the *price* for sending traffic at rate $\sum_{s:l\in s} x_s$ on link $l$. We assume that the price functions $f_l(\cdot)$ are increasing functions. Under this assumption,

it is easy to see that $V(\mathbf{x})$ is a strictly concave function. Now, consider the following problem:

$$\max_{\{x_r \geq 0\}} V(\mathbf{x}). \tag{10}$$

The optimal source rates satisfy

$$\frac{\partial V}{\partial x_r} = 0, \qquad \forall r.$$

This gives the following set of equations:

$$U_r'(x_r) - \sum_{l:l\in r} f_l \left( \sum_{s:l\in s} x_s \right) = 0, \qquad \forall r. \tag{11}$$

Clearly, the above set of equations is difficult to solve in a centralized manner. It requires knowledge of all the sources' utility functions, their routes and the link penalty functions $f_l$'s. Next, we will see how congestion controllers compute the solution to (10) in a decentralized manner.

Let us suppose that each router (node) in the network has the ability to monitor the traffic on each of its links and compute $y_l$ for each link $l$ connected to it. Let $p_l(t)$ denote the price for using link $l$ at time $t$. In other words,

$$p_l(t) = f_l(y_l(t)).$$

Define the price of a route to be the sum of the prices of the links on the route. Now, suppose that the sources and the network have a protocol that allows each source to obtain the sum of the link prices along its route. One way to convey the route prices would be to have a field in each packet's header which can be changed by the routers to convey the price information. For instance, the source may set the price field equal to zero when it transmits a packet. Each router on the packet's route could then add the link prices to this field so that, when the packet reaches the destination, the field would contain the route price. The destination could then simply transmit this information back to the source in the acknowledgment (ack) packet.

Let $q_r$ denote the route price on route $r$, i.e.,

$$q_r = \sum_{l:l\in r} p_l.$$

Using this notation, the condition (11) can be rewritten as

$$U_r'(x_r) - q_r = 0, \qquad \forall r. \tag{12}$$

Consider the following congestion control algorithm for solving the set of equations (12) [13]: for each user $r$,

$$\dot{x}_r = k_r(x_r)\left(U_r'(x_r) - q_r(t)\right), \tag{13}$$

where $k_r(x)$ is an appropriately chosen scaling function. Note that the right-hand side of the differential equation (13) describing the congestion control algorithm is simply

the partial derivative of the objective function (9) with respect to $x_r$, multiplied by $k_r(x_r)$. Thus, the congestion controller is the well-known gradient algorithm that is widely used in optimization theory [3].

The following theorem shows that the congestion control algorithm is globally asymptotically stable, i.e., starting from any initial condition, in the limit as $t \to \infty$, the set of sources rates $x_r(t)$ will converge to the set of non-zero rates $\hat{x}_r$ that maximize the objective $V(\mathbf{x})$ in (9).

**Theorem 1.** *Assume that the utility functions $U_r(x_r)$, the price functions $f_l(y_l)$ and the scaling functions $k_r(x_r)$ are such that*

- *the problem (10) has a unique solution $\hat{x}_r$ such that $\hat{x}_r \neq 0\ \forall r$,*
- *$\hat{x}_r = 0\ \forall r$ is not an equilibrium point of (13) and*
- *$V(\mathbf{x}) \to -\infty$ as $\|\mathbf{x}\| \to \infty$.*

*Then, starting from any initial condition $\{x_r(0) \geq 0\}$, the congestion control algorithm (13) will converge to the unique solution of (10), i.e., $\mathbf{x}(t) \to \hat{\mathbf{x}}$ as $t \to \infty$.*

*Proof.* Note that

$$\dot{V} = \sum_r \frac{\partial V}{\partial x_r}\dot{x}_r = \sum_r k_r(x_r)\left(U_r'(x_r) - q_r\right)^2 \geq 0.$$

Further, note that $\dot{V} > 0$ for $\mathbf{x} \neq \hat{\mathbf{x}}$ and is equal to zero for $\mathbf{x} = \hat{\mathbf{x}}$. Thus, $V(\mathbf{x})$ is a Lyapunov function for the system and by the assumptions of the theorem, the system of congestion controllers is globally, asymptotically stable and converges to the unique solution of (10).

□

In the penalty function method, the link prices $p_l$ are computed based on price functions $f_l(x_l)$, and thus, while it solves the problem (10), it does not solve the original resource allocation problem (7) formulated in the previous section exactly. At equilibrium, the algorithm (13) can be interpreted as computing an approximation to the Lagrange multipliers of the original resource allocation problem (7) and (6). Clearly, the congestion controller will converge to the optimal solution of (7) if the equilibrium prices $\hat{p}_l$ are indeed the Lagrange multipliers. To this end, instead of computing $p_l$ via the static relationship $p_l = f_l(y_l)$, consider the following dynamic equation to update $p_l$ [1, 12, 20, 30]:

$$\dot{p}_l = g_l(p_l)(y_l - c_l)^+, \tag{14}$$

where it is assumed that $h(p_l) > 0$. The congestion controller (13) can be interpreted as computing the primal variables which are the source rates and the dynamic price update (14) can be interpreted as computing the dual variables or the Lagrange multipliers. Therefore, (13) and (14) together is called the primal-dual algorithm. The widely-studied dual algorithm can be viewed as a special case of the above algorithm where $k_r(x_r) \to \infty$ [21, 31]. The exact penalty function approach (also known

as the adaptive virtual queue or AVQ algorithm [15, 16]) cannot be directly interpreted in terms of the primal-dual algorithm presented here, but it is an alternative method to compute the Lagrange multipliers precisely.

Define

$$H_r(x_r) := \int_{\hat{x}_r}^{x_r} \frac{u_r - \hat{x}_r}{k_r(u_r)}\, du,$$

$$J_l(p_l) := \int_{\hat{p}_l}^{p_L} \frac{v_l - \hat{p}_l}{g_l(v_l)}\, dv,$$

and let

$$V(x,p) = \sum_r H_r(x_r) + \sum_l J_l(p_l).$$

Then, it can be verified that $V(x,p)$ is a Lyapunov function for the primal-dual algorithm defined by (13)-(14) and that the system is globally, asymptotically stable [30].

## 4 Stability in the Presence of Delays

In this section, we study the global stability of the primal congestion controller introduced in the previous section in the presence of delays. We first introduce some notation to describe the network with delays. Let $d_f(i,l)$ be the forward delay from source $i$ to link $l$, and $d_b(i,l)$ be the backward delay from link $l$ to source $i$. Denote by $T_i = d_f(i,l) + d_b(i,l)$ the round trip time (RTT) for source $i$. We assume that the RTT is a constant and not time-varying. This is a reasonable assumption if the price functions $f_l(y_l)$ are designed such that the users get early feedback about congestion so that they react before queue buildup occurs at the routers. In that case, the RTT is simply equal to the propagation delay which is a constant.

We consider the following congestion control algorithm:

$$\dot{x}_i(t) = k_i x_i(t - T_i)\left(\frac{a}{x_i^n(t)} - b x_i^m(t) q_i(t)\right), \tag{15}$$

where

$$q_i(t) = \sum_{l \in i} p_l(t - d_b(i,l)),$$

$$p_l(t) = f_l(y_l(t)),$$

$$f_l(y_l(t)) = \left(\frac{y_l(t)}{c_l}\right)^h \tag{16}$$

and

$$y_l(t) = \sum_{k:l \in k} x_k(t - d_f(i,l)).$$

Here $a$, $b$, and $h$ are positive real numbers, and $m$, $n$ are real numbers that satisfy $m + n > 0$. In the above set of equations, $x_i$ is the rate at which source $i$ transmits

data, $y_l$ is the arrival rate at link $l$, $p_l$ is price of link $l$, $q_i$ is the price of source $i$'s route and $f_l(y)$ is the function of the link arrival rate which is used to compute $p_l$. The price of a route is simply the sum of the prices of the links along its path. Also define the quantity $d := \max_j T_j$ which will be useful later. Note that when $d = 0$, the algorithm (15) is a special case of (13). Specifically, here we have chosen

$$U_i'(x_i) = \frac{a}{bx_i^{m+n}}$$

and the scaling function to be $k_i x_i^{m+1}$ for some constant $k_i > 0$.

It has been shown in [32] that $\mathbf{x}(t) > 0$ for all $t$ whenever the initial conditions are non-zero. So we can define the functions

$$W_j(t) = \frac{1}{2}\left(\log x_j(t) - \log \hat{x}_j\right)^2 \tag{17}$$

which are well-defined provided that the network model has a nonzero initial condition. Also define the function

$$W(t) = \max_j W_j(t). \tag{18}$$

In what follows, we will show that there exists an $\alpha > 1$ such that $W(t)$ decreases whenever

$$\alpha^2 W(t) > \max_{t-d \le r \le t} W(r). \tag{19}$$

It then follows from Razumikhin's theorem that the set of delay-differential equations describing the primal controller is globally asymptotically stable. The next lemma shows that, if the Lyapunov function $W$ defined by (17-18) satisfies the condition in (19) at some time instant $t$, then this naturally imposes upper and lower bounds on the functions $x_j$ over the interval $\in [t-d, t]$.

**Lemma 1.** *Suppose the network model has a nonzero initial condition, that at time t the Lyapunov function W satisfies the condition in (19), and that index i is such that that*

$$W(t) = W_i(t).$$

*Then*

$$B_i^{-\beta} \hat{x}_j < x_j(r) < B_i^{\beta} \hat{x}_j, \tag{20}$$

*for all* $\tau \in [t-d, t]$, *where* $B_i = \frac{x_i(t)}{\hat{x}_i}$ *and* $\beta = \mathrm{sgn}(\log B_i)\alpha$.

*Proof.* First note that since $W_i(t) \neq 0$, we have $B_i \neq 1$. By the definition on $W_i(t)$ we have for each index $j$ and $\tau \in [t-d, t]$ that

$$\alpha^2 \left(\log B_i\right)^2 = \alpha^2 W(t) > W_j(t) = \left(\log \frac{x_j(t)}{\hat{x}_j}\right)^2.$$

Note that above inequality can be rewritten as

$$(\log B_i^{\alpha})^2 > \left(\log \frac{x_j(t)}{\hat{x}_j}\right)^2$$

which implies that

$$-\mathrm{sgn}(\log B_i)\log B_i^{\alpha} < \log \frac{x_j(t)}{\hat{x}_j} < \mathrm{sgn}(\log B_i)\log B_i^{\alpha}.$$

From this, we get the inequalities

$$\hat{x}_j B_i^{-\beta} < x_j(r) < \hat{x}_j B_i^{\beta}.$$

□

The next lemma then shows that the route prices are also upper and lower bounded as a consequence of the previous lemma.

**Lemma 2.** *Under the conditions of Lemma 1, the price of each route i is bounded as given by the following expression:*

$$\frac{a}{b} B_i^{-h\beta} \hat{x}_i^{-m-n} < q_i(t) < \frac{a}{b} B_i^{h\beta} \hat{x}_i^{-m-n}.$$

*Proof.* The bounds on the source rates in the previous lemma immediately give a bound on the link arrival rates $y_l$, which in turn give bounds on the link prices $p_l$. It is then easy to see from the definition of the route price $q(t)$ that

$$\hat{q}_i B_i^{-\beta h} < q_i(t) < \hat{q}_i B_i^{\beta h}.$$

Furthermore since $\hat{q}_i = \frac{a}{b}\hat{x}_i^{-m-n}$, we can conclude that

$$\frac{a}{b} B_i^{-h\beta} \hat{x}_i^{-m-n} < q_i(t) < \frac{a}{b} B_i^{h\beta} \hat{x}_i^{-m-n}.$$

**Corollary 1** *If $m+n>h$, the supposition of Lemma 1 holds and $\alpha = (m+n+h)/2h$, then $\dot{W}_i(t) < 0$.*

*Proof.* The derivative $\dot{W}_i(t)$ is given by

$$\dot{W}_i(t) = \frac{\dot{x}_i(t)}{x_i(t)} \log\left(\frac{x_i(t)}{\hat{x}_i}\right) = \frac{\dot{x}_i(t)}{x_i(t)} \log B_i,$$

and thus to prove the result we need to show that $\dot{x}_i(t)\log B_i < 0$. From (15) we see that

$$\dot{x}_i(t) = k_i \frac{x_i(t-T_i)}{x_i^n(t)} \left(a - b q_i(t) x^{m+n}(t)\right).$$

Since $x_i(t-T_i)$ and $x_i(t)$ are both positive it is sufficient to show that

$$\left(a - b q_i(t) x^{m+n}(t)\right) \log B_i < 0. \tag{21}$$

We can show this using Lemma 2 by considering the following two cases. Suppose first that $B_i > 1$, then we have to show that

$$q_i(t)x_i^{m+n}(t) > \frac{a}{b}.$$

From the lower bound in Lemma 2, we have

$$q_i(t)x_i^{m+n}(t) > \frac{a}{b}B_i^{-h\alpha}\hat{x}_i^{-m-n}x_i^{m+n}(t) = \frac{a}{b}B_i^{-h\alpha+m+n}$$

which is greater than $a/b$ since $\alpha h < m+n$. Next, suppose that $B_i < 1$. Then, we have to show that

$$q_i(t)x_i^{m+n}(t) < \frac{a}{b}.$$

From the upper bound in Lemma 2,

$$q_i(t)x_i^{m+n}(t) < \frac{a}{b}B_i^{-h\alpha}\hat{x}_i^{-m-n}x_i^{m+n}(t) = \frac{a}{b}B_i^{-h\alpha+m+n}$$

which is now less than $a/b$ since $\alpha h < m+n$.

Now we state our result on the stability of congestion controllers with delay.

**Theorem 1** *If $m+n > h > 0$, then the network model in (15) is globally asymptotically stable.*

*Proof.* With $W(t)$ and $W_j(t)$ as defined above, it is easy to show that, for every $t \geq 0$,

$$\text{limsup}_{a\to 0^+}\frac{1}{a}\{W(t+a) - W(t)\} < 0 \tag{22}$$

whenever $\alpha^2 W(t) > \max_{t-d\leq r\leq t} W(r)$. Thus, the result follows from Razumikhin's theorem [7]. □

## 5 Stability with File Arrivals and Departures

In this section, we study simple models of the dynamics of a network at the connection level, i.e., at the level of file arrivals and departures. In all the previous sections, we assumed that the number of controlled flows in the network is a constant and studied the congestion control and resource allocation problem for a fixed number of flows. In reality, connections or files or flows arrive and depart from the network. We assume that the amount of time that it takes for the congestion control algorithms to drive the source rates close to their equilibrium is much smaller than the inter-arrival and inter-departure times of files. In fact, we assume that compared to the time-scale of connection dynamics, the congestion control algorithm operates instantaneously, providing a resource allocation dictated by the utility functions of the users of the network.

Consider a network in which files are arriving to route $r$ at rate $\lambda_r$ files-per-second according to a Poisson process. The file sizes are independent and exponentially distributed, with the mean file size being $1/\mu_r$ on route $r$. Let $n_r(t)$ be the number of files on route $r$ at time $t$. It is assumed that each file on route $r$ is allocated $\hat{x}_r(t)$ bits-per-second at time $t$, where the rates $\{\hat{x}_r\}$ are chosen as the optimal solution to the following resource allocation problem:

$$\max_{\{x_r\}} \sum_r w_r n_r \frac{x_r^{1-\alpha}}{1-\alpha} \tag{23}$$

subject to

$$\sum_{r:l\in r} n_r x_r \le c_l, \qquad \forall l,$$
$$x_r \ge 0, \qquad \forall r.$$

In this optimization problem, it is assumed that all files using the same route have the same utility function $w_r x_r^{1-\alpha}/(1-\alpha)$ for some $\alpha > 0$, $\alpha \neq 1$. When $\alpha = 1$, we assume that the utility function is $U_r(x_r) = w_r \log x_r$. This connection-level model assumes that, compared to the inter-arrival and departure times of files, congestion control operates at a very fast time scale so that, at the connection time-scale, it appears as though the congestion controller converges instantaneously to solve the resource allocation problem instantaneously. Thus, a time-scale separation is assumed between congestion control (the fast time-scale) and file arrivals and departures (the slow time-scale).

Note that $\hat{x}_r$ is not uniquely determined when $n_r = 0$. For this reason, it is more convenient to describe the optimization problem in terms of new variables $\{\hat{\chi}_r\}$, where $\hat{\chi}_r$ is the total rate allocated to users on route $r$, i.e., $\hat{\chi}_r = n_r x_r$. These rates are obtained as solutions to the following optimization problem:

$$\max_{\{\chi_r\}} \sum_r w_r n_r^{\alpha} \frac{\chi_r^{1-\alpha}}{1-\alpha} \tag{24}$$

subject to

$$\sum_{r:l\in r} \chi_r \le c_l, \qquad \forall l,$$
$$\chi_r \ge 0, \qquad \forall r.$$

Again $\hat{\chi}_r$ is not uniquely determined when $n_r = 0$. However, $\hat{\chi}_r$ is always less than or equal to $\zeta_r := \max_{l:l\in r} c_l$.

Consider the load on any link $l$ due to the file arrival process. If a route $r$ uses link $l$, then on average, it requires $\lambda_r/\mu_r$ bits-per-second from link $l$. Thus, it is rather obvious that the network can support the traffic load only if

$$\sum_{r:l\in r} \frac{\lambda_r}{\mu_r} < c_l, \qquad \forall l. \tag{25}$$

What is less obvious is that this is also the sufficient condition for stability. In the rest of this section, we will prove that the condition (25) is sufficient for stochastic stability.

From our assumptions on the arrival process and file-size distribution, the evolution of $\mathbf{n}$(t) can be described by a continuous-time Markov chain. It is more convenient to work with a discrete-time Markov chain and therefore, using uniformization [14], we consider the system only at certain event times, where the events correspond to arrival, departure and fictitious departures. Suppose that there are $n_r$ files using route $r$ immediately after an event. Then,

- with probability $\lambda_r / \sum_s (\lambda_s + \mu_s \zeta_s)$, the next event is an arrival to route $r$;
- with probability $\mu_r n_r x_r / \sum_s (\lambda_s + \mu_s \zeta_s)$, the next event is a departure from route $r$; and
- with probability $(\mu_r \zeta_r - \mu_r x_r n_r) / \sum_s (\lambda_s + \mu_s \zeta_s)$, the next event is a fictitious departure from route $r$.

Let $\mathbf{n}(k)$ denote the state of the Markov chain after the $k^{\text{th}}$ event. Then $\mathbf{n}(k)$ is a discrete-time Markov chain. We are interested in showing that this Markov chain is positive chain. For this purpose, we will use the following result known as the Foster-Lyapunov criterion [2].

**Theorem 2.** *Suppose that a Markov chain* $\{\mathbf{n}(k)\}$ *is irreducible and let* $E_0$ *be a finite subset of the state space* $E$. *Then the chain is positive recurrent if, for some* $V : E \longrightarrow \Re$ *and some* $\delta > 0$, *we have* $\inf_{\mathbf{n} \in E} V(\mathbf{n}) > -\infty$ *and*

$$\begin{aligned} E(V(\mathbf{n}(k+1)) | \mathbf{n}(k) = \mathbf{n}) &< \infty, \qquad \mathbf{n} \in E_0, \\ E(V(\mathbf{n}(k+1)) | \mathbf{n}(k) = \mathbf{n}) &\leq V(\mathbf{n}) - \delta, \ \mathbf{n} \notin E_0. \end{aligned} \tag{26}$$

□

It is easy to verify that the Markov chain under consideration is irreducible. Given any state, with non-zero probability, it is possible to reach the zero state where $n_r = 0$ for all routes $r$. This will happen if there are no arrivals for all the routes for a sufficiently long period of time. From the zero state, it is possible to reach any state by considering a sequence of arrival events and no departure events, which again can happen with non-zero probability. Thus, one can reach any state from any other state in finite time with non-zero probability which, by definition, means that the Markov chain is irreducible. Now, to verify the key drift condition of the Foster-Lyapunov criterion, consider the candidate Lyapunov function

$$V(k) = \sum_r \kappa_r n_r^{\alpha+1}(k),$$

where the $\kappa_r$'s are non-negative constants, to be chosen appropriately later. For notational convenience, and without loss of generality, we assume that

$$\sum_s (\lambda_s + \mu_s \zeta_s) = 1.$$

We note that

$$E(V(k+1)-V(k)|\mathbf{n}(k)) = \sum_r \lambda_r \kappa_r \left((n_r+1)^{\alpha+1} - n_r^{\alpha+1}\right) \tag{27}$$

$$+ \sum_r I_{n_r>0} \mu_r \hat{\chi}_r \kappa_r \left((n_r-1)^{\alpha+1} - n_r^{\alpha+1}\right) \tag{28}$$

$$= \sum_r \lambda_r \kappa_r \left((n_r+1)^{\alpha+1} - n_r^{\alpha+1}\right) I_{n_r>0} + \sum_r \lambda_r \kappa_r$$

$$+ \sum_r I_{n_r>0} \mu_r \hat{\chi}_r \kappa_r \left((n_r-1)^{\alpha+1} - n_r^{\alpha+1}\right)$$

Consider the expression

$$(n_r+1)^{\alpha+1} - n_r^{\alpha+1}$$

and assume $n_r > 0$. By Taylor's theorem,

$$(n_r+1)^{\alpha+1} - n_r^{\alpha+1} = (\alpha+1)n_r^{\alpha} + \frac{\alpha(\alpha+1)}{2}\tilde{n}_r^{\alpha-1}$$

for some $\tilde{n}_r$ that satisfies

$$n_r \le \tilde{n}_r \le n_r + 1.$$

Equivalently,

$$1 \le \frac{\tilde{n}_r}{n_r} \le 1 + \frac{1}{n_r} \le 2.$$

Thus, for $\alpha \ge 1$,

$$\left(\frac{\tilde{n}_r}{n_r}\right)^{\alpha-1} \le 2^{\alpha-1}$$

and, for $\alpha < 1$,

$$\left(\frac{\tilde{n}_r}{n_r}\right)^{\alpha-1} \le 1.$$

Letting

$$C_1 = \frac{\alpha(\alpha+1)}{2}\max\{2^{\alpha-1}, 1\},$$

we have

$$(n_r+1)^{\alpha+1} - n_r^{\alpha+1} \le (\alpha+1)n_r^{\alpha} + C_1 n_r^{\alpha-1} \tag{29}$$

if $n_r > 0$.

Next, again by Taylor's theorem, we have

$$(n_r-1)^{\alpha+1} - n_r^{\alpha+1} \le -(\alpha+1)n_r^{\alpha} + \frac{\alpha(\alpha+1)}{2}m_r^{\alpha-1},$$

where $m_r$ satisfies

$$n_r - 1 \le m_r \le n_r.$$

Assuming, $n_r \ne 0$, we have

$$1 - \frac{1}{n_r} \leq \frac{m_r}{n_r} \leq 1.$$

If $\alpha > 1$, then

$$\left(\frac{m_r}{n_r}\right)^{\alpha-1} \leq 1.$$

If $\alpha < 1$ and $n_r > 1$, then

$$\left(\frac{m_r}{n_r}\right)^{\alpha-1} \leq 2^{\alpha-1}$$

since $1 - 1/n_r \geq 1/2$. If $n_r = 1$, denote the corresponding value of $m_r$ as $\tilde{m}$. Defining $C_2 = \frac{\alpha(\alpha+1)}{2} \max\{1, 2^{\alpha-1}, \tilde{m}^{\alpha-1}\}$, we have

$$(n_r - 1)^{\alpha+1} - n_r^{\alpha+1} \leq -(\alpha+1)n_r^{\alpha} + C_2 n_r^{\alpha-1}, \tag{30}$$

when $n_r > 0$. Substituting (29) and (30) in (29), we get

$$\begin{aligned} E(V(k+1) - V(k)|\mathbf{n}(k)) \leq & \sum_r \kappa_r(\alpha+1)n_r^{\alpha}(\lambda_r - \mu_r\hat{\chi}_r) \\ & + C\sum_{n_r} I_{n_r>0} n_r^{\alpha-1} + \sum_r \lambda_r \kappa_r, \end{aligned}$$

where $C = C_1 + C_2$.

We now make use of the following well-known result from concave optimization: suppose $\hat{\mathbf{y}}$ is the optimal solution to $\max_{\mathbf{y}} f(\mathbf{y})$ subject to $\mathbf{y} \in \mathcal{S}$, where $f(\cdot)$ is a concave function and $\mathcal{S}$ is a convex set. Then, for any $\mathbf{y} \in \mathcal{S}$, the following is true:

$$\sum_i \frac{\partial f}{\partial y_i}(\mathbf{y})(y_i - \hat{y}_i) \leq 0.$$

To make use of this result, let us rewrite the drift as

$$\begin{aligned} E(V(k+1) - V(k)|\mathbf{n}(k)) \leq & \sum_r \kappa_r(\alpha+1)n_r^{\alpha}\mu(\frac{\lambda_r}{\mu_r} - \hat{\chi}_r) \\ & + C\sum_{n_r} I_{n_r>0} n_r^{\alpha-1} + \sum_r \lambda_r \kappa_r, \end{aligned}$$

Recall that $\{\hat{\chi}_r\}$ is the optimal solution to (24). Further, if we choose $\chi_r = \frac{\lambda_r}{\mu_r}$, then this set of rates satisfy the link capacity constraints. In fact, from condition (25), there exists an $\varepsilon > 0$ such that

$$(1+\varepsilon)\sum_{r:l\in r} \frac{\lambda_r}{\mu_r} \leq c_l, \qquad \forall l.$$

Thus,

$$\chi_r = (1+\varepsilon)\frac{\lambda_r}{\mu_r}$$

is also a rate allocation that satisfies the link capacity constraints. Thus, from the property of concave optimization discussed earlier, we have

$$\sum_r \frac{n_r^\alpha w_r}{\frac{\lambda_r(1+\varepsilon)}{\mu_r}} \left( \frac{\lambda_r(1+\varepsilon)}{\mu_r} - \hat{\chi}_r \right) \le 0. \tag{31}$$

This can be equivalently written as

$$\sum_r w_r n_r^\alpha \left( \frac{\mu_r}{\lambda_r} \right)^\alpha \left( \frac{\lambda_r}{\mu_r} - \hat{\chi}_r \right) \le -\varepsilon \sum_r w_r n_r^\alpha \left( \frac{\mu_r}{\lambda_r} \right)^{\alpha-1}.$$

Choosing

$$\kappa_r = \frac{w_r}{(\alpha+1)\mu_r} \left( \frac{\mu_r}{\lambda} \right)^\alpha,$$

the drift can then be upper-bounded as

$$E(V(k+1) - V(k)|\mathbf{n}(k)) \le -\varepsilon \sum_r w_r n_r^\alpha \left( \frac{\mu_r}{\lambda_r} \right)^{\alpha-1} + C \sum_{n_r} I_{n_r>0} n_r^{\alpha-1} + \sum_r \lambda_r \kappa_r.$$

Now it is easy to see that, given $\delta > 0$, there exists a $B > 0$ such that, for all $\mathbf{n}$ satisfying $\|\mathbf{n}\| > B$,

$$E(V(k+1) - V(k)|\mathbf{n}(k)) \le -\delta.$$

Thus, by the Foster-Lyapunov criterion, the Markov chain describing the number of files waiting on the various routes in the network is positive recurrent.

In addition to showing positive recurrence, it is also possible to obtain an upper bound on the steady-state value of $E(\|\mathbf{n}\|^\alpha)$. To obtain such a bound, first note that

$$n_r \le \|\mathbf{n}\|, \qquad \forall r,$$

and there exists $j$ such that

$$n_j \ge \frac{\|\mathbf{n}\|}{R},$$

where $R$ is the dimension of $\mathbf{n}$, i.e., $R$ is the total number of routes in the network. Thus, for appropriately chosen positive constants $K_1$ and $K$, the drift can be bounded as

$$E(V(k+1) - V(k)|\mathbf{n}(k)) \le -\varepsilon K_1 \|\mathbf{n}(k)\|^\alpha + C\|\mathbf{n}(k)\|^{\alpha-1} \sum_r I_{n_r(k)>0} + K.$$

Since

$$\frac{1}{\|\mathbf{n}(k)\|} \sum_r I_{n_r(k)>0} \le \frac{R}{\|\mathbf{n}\|},$$

we have

$$\begin{aligned} E(V(k+1)-V(k)|\mathbf{n}(k)) &\leq -\varepsilon K_1\|\mathbf{n}(k)\|^{\alpha}+RC\|\mathbf{n}(k)\|^{\alpha-1}I_{\|\mathbf{n}\|\geq B} \\ &\quad +C\|\mathbf{n}(k)\|^{\alpha-1}I_{\{1\leq\|\mathbf{n}\|<B\}}+K \\ &\leq -\varepsilon K_1\|\mathbf{n}(k)\|^{\alpha}+\frac{RC}{B}\|\mathbf{n}(k)\|^{\alpha} \\ &\quad +RC\max\{1,B^{\alpha-1}\}+K, \end{aligned}$$

where $B$ is an arbitrary positive constant that is chosen, depending on $\varepsilon$, to ensure that

$$\frac{RC}{B}-\varepsilon K_1<0.$$

Thus,

$$(\varepsilon K_1-K_2)E\left(\|\mathbf{n}(k)\|^{\alpha}\right)\leq E(V(k)-V(k+1))+K_3,$$

where

$$K_2=\frac{RC}{B},\qquad K_3=K+RC\max\{1,B^{\alpha-1}\}.$$

Summing over $k$, we get

$$\begin{aligned} \sum_{k=1}^{M}(\varepsilon K_1-K_2)E\left(\|\mathbf{n}(k)\|^{\alpha}\right) &\leq E(V(1))-E(V(M+1))+MK_3 \\ &\leq E(V(1))+MK_3. \end{aligned}$$

Finally,

$$\limsup_{M\to\infty}\frac{1}{M}\sum_{k=1}^{M}E\left(\|\mathbf{n}(k)\|^{\alpha}\right)\leq\frac{K_3}{\varepsilon K_1-K_2}.$$

Since the Markov chain is ergodic, the bound also applies to the steady-state $\alpha^{\text{th}}$ moment. Note that, as the load on any link gets close to the link capacity, $\varepsilon$ decreases and the corresponding value of $B$ required to ensure that $K_2-\varepsilon K_1$ is positive increases. In fact, this means that $K_2-\varepsilon K_1$ decreases and therefore, the bound on the $\alpha^{\text{th}}$ moment increases, which is reasonable when the system load increases.

The above result has been obtained under the assumption that the file arrival processes are Poisson and the file-size distributions are exponential. There is experimental evidence which suggests that the file-arrival process in the Internet is indeed Poisson. On the other hand, the file-size distribution is not as easily characterized. It is well-known that the file-sizes in the Internet have a large variance. However, the files that we consider here are only those that are large enough so that it reasonable to assume that compared to the file arrival and departure time scale, the congestion-control algorithms that control the file transfers converge to the solution of (24). It is difficult to characterize the file-size distribution of such files alone. Thus, it is of much interest to extend the above stability result to general file-size distributions.

Extending the result to hyper-exponential file-size distributions is immediate. Suppose that the file-size distribution on the $r^{\text{th}}$ route is exponential with mean $1/\mu_{ri}$ with probability $p_{ri}$, for $i=1,,2,\ldots I(r)$ for some $I(r)>1$. Then, the arrival process to the $r^{\text{th}}$ route can be thought as $I(r)$ independent Poisson processes with mean

rates $\lambda_r p_{ri}$, and the $i^{\text{th}}$ such process generating files from an exponential distribution with mean $1/\mu_{ri}$. This is no different from the stability problem that we have already considered and therefore, the Markov chain describing the number of files in the network is still positive recurrent. A more complicated extension is for the case where the resource allocation parameter $\alpha$ is chosen to be equal to 1 and the weights $w_r$ are all equal to 1, i.e., $U_r(x_r) = \log x_r$, $\forall r$. For this special case and for certain network topologies, using a reversibility argument, it has been shown in [4, 5, 26] that the connection-level system has a steady-state distribution which is independent of the file-size distribution. Extensions to general file-size distributions, general topologies and general resource allocation parameters is still an open problem. For simple topologies and Coxian file-size distributions with a small number of phases, numerical construction of Lyapunov functions suggests that the stability condition may be the same irrespective of the file-size distribution [19].

## 6 Conclusions

In this paper, we have surveyed recent research on the topic of Internet congestion control. We have emphasized the role of optimization, control theory and stochastic models in developing algorithms that perform well even in the presence of delays and stochastic disturbances such as arrivals and departures of flows.

Fairness and stability are two important considerations in designing congestion control mechanisms that allow resource-sharing in a network with many competing users. By associating a utility function with each user and defining fairness to mean the maximization of the sum utility of the network, the fair resource sharing problem can be viewed as an optimization problem. In the language of convex optimization, the congestion controllers can be thought as numerical techniques (such as gradient search) to solve an optimization problem whose objective is social welfare. The congestion feedback from the network can be thought of as shadow prices or Lagrange multipliers computed by the links in the network to ensure that each link in the network is not over-utilized.

Designing the step sizes for the gradient procedure is difficult since the controllers that implement these algorithms are distributed in the network. Each controller's congestion feedback from the network is subject to propagation delay, and possibly queueing delay, and the challenge is to design the control gains (which are the step sizes of the gradient search procedure) to ensure stability in the presence of heterogeneous feedback delays. We used Razumikhin's theory to perform the controller design.

The randomness in the network due to file arrivals and departures was captured using a Markov chain model at the connection level. We then used stochastic Lyapunov analysis to show that the resource allocation formulation leads to efficient network operation.

**Acknowledgments.**
It is a pleasure to thank Ashvin Lakshmikantha for providing useful comments on an earlier draft of this paper.

## References

[1] T. Alpcan and T. Başar (2003) *A utility-based congestion control scheme for internet-style networks with delay*, Proceedings of IEEE Infocom (San Francisco, California)
[2] S. Asmussen (2003) *Applied Probability and Queues*, Springer-Verlag, New York
[3] D. Bertsekas (1995) *Nonlinear programming*, Athena Scientific, Belmont, MA, 1995
[4] T. Bonald and L. Massoulie (2001) *Impact of fairness on Internet performance*, Proceedings of ACM Sigmetrics
[5] T. Bonald and A. Proutiere (2003) *Insensitive bandwidth sharing in data networks*, Queueing Systems **44**, 69–100
[6] G. de Veciana, T.-J. Lee, and T. Konstantopoulos (2000) *Stability and performance analysis of networks supporting elastic services*, IEEE/ACM Transactions on Networking **9** (2001), no. 1, 2–14
[7] J. Hale and S. M. V. Lunel (1991) *Introduction to functional differential equations*, 2nd edition, Springer Verlag, New York, NY
[8] C.V. Hollot, V. Misra, D. Towsley, and W. Gong (2001) *On designing improved controllers for AQM routers supporting TCP flows*, Proceedings of IEEE Infocom (Anchorage, Alaska, pp. 1726–1734
[9] V. Jacobson (1988) *Congestion avoidance and control*, ACM Computer Communication Review **18**, 314–329
[10] R. Johari and D. Tan (2001) *End-to-end congestion control for the Internet: Delays and stability*, IEEE/ACM Transactions on Networking **9**, no. 6, 818–832
[11] F. P. Kelly (1997) *Charging and rate control for elastic traffic*, European Transactions on Telecommunications **8** (1997), 33–37
[12] F. P. Kelly (2003) *Fairness and stability of end-to-end congestion control*, European Journal of Control **9**, 149–165
[13] F. P. Kelly, A. Maulloo, and D. Tan (1998) *Rate control in communication networks: shadow prices, proportional fairness and stability*, Journal of the Operational Research Society **49**, 237–252
[14] P.R. Kumar and P. Varaiya (1986) *Stochastic Systems, Estimation, Identification and Adaptive Control*, Prentice Hall, 1986
[15] S. Kunniyur and R. Srikant (2000) *End-to-end congestion control: utility functions, random losses and ECN marks*, Proceedings of IEEE Infocom (Tel Aviv, Israel)

[16] S. Kunniyur and R. Srikant (2003) *End-to-end congestion control: utility functions, random losses and ECN marks*, IEEE/ACM Transactions on Networking **7**, no. 5, 689–702

[17] S. Kunniyur and R. Srikant (2003) *Stable, scalable, fair congestion control and AQM schemes that achieve high utilization in the internet*, IEEE Transactions on Automatic Control **48**, 2024–2028

[18] R. La, P. Ranjan, and E. Abed (2004) *Global stability conditions for rate control with arbitrary communication delays*, University of Maryland CSHCN TR 2003-25

[19] A. Lakshmikantha, C. Beck, and R. Srikant (2004) *Connection level analysis of the Internet using the sum of squares technique*, Proceedings of the Conference on Information Sciences and Systems (Princeton, NJ)

[20] S. Liu, T. Başar, and R. Srikant (2003) *Exponential RED: A stabilizing AQM scheme for low and high speed TCP*, University of Illinois Tech Report. An earlier version appeared in the Proceedings of IEEE Conference on Decision and Control, December 2003, under the title "Controlling the Internet: A survey and some new results"

[21] S. H. Low and D. E. Lapsley (1999) *Optimization flow control, I: Basic algorithm and convergence*, IEEE/ACM Transactions on Networking (1999), 861–875

[22] L. Massoulie (2002) *Stability of distributed congestion control with heterogenous feedback delays*, IEEE Transactions on Automatic Control **47** (2002), 895–902

[23] V. Misra, W. Gong, and D. Towsley (2000) *A fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED*, Proceedings of ACM Sigcomm (Stockholm, Sweden)

[24] F. Paganini, J. Doyle, and S. Low (2000) *Scalable laws for stable network congestion control*, Proceedings of the IEEE Conference on Decision and Control (Orlando, FL), pp. 185–190.

[25] G. Raina (2003) *Personal communication*

[26] J. Roberts and L. Massoulie (1998) *Bandwidth sharing and admission control for elastic traffic*, Proc. ITC specialists seminar (Yokohama, Japan)

[27] R. Srikant (2004) *The Mathematics of Internet Congestion Control*, Birkhauser

[28] G. Vinnicombe (2001) *On the stability of end-to-end congestion control for the Internet*, University of Cambridge Technical Report CUED/F-INFENG/TR.398. Available at http://www.eng.cam.ac.uk/˜gv

[29] G. Vinnicombe (2002) *On the stability of networks operating TCP-like congestion control*, Proceedings of the IFAC World Congress (Barcelona, Spain)

[30] J.T. Wen and M. Arcak (2003) *A unifying passivity framework for network flow control*, Proceedings of IEEE Infocom

[31] H. Yaiche, R. R. Mazumdar, and C. Rosenberg (2000) *A game-theoretic framework for bandwidth allocation and pricing in broadband networks*, IEEE/ACM Transactions on Networking **8**, no. 5, 667–678.

[32] L. Ying, G. Dullerud, and R. Srikant (2004) *Global stability of Internet congestion controllers with heterogeneous delays*, To appear in the Proceedings of the American Control Conference (Boston, MA)

# Delay Effects on the Asymptotic Stability of Various Fluid Models in High-Performance Networks

Silviu-Iulian Niculescu[1], Wim Michiels[2], Daniel Melchor-Aguillar[3], Tatyana Luzyanina[2], Frédéric Mazenc[4], Keqin Gu[5], and Fabien Chatté[1]

[1] HEUDIASYC (UMR CNRS 6599), Université de Technologie de Compiègne, Centre de Recherche de Royallieu, BP 20529, 60205, Compiègne, cedex, France.
E-mail: silviu,fabien.chatte@hds.utc.fr

[2] Department of Computer Science, K.U. Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium.
E-mail: Wim.Michiels, Tatyana.Luzyanina@cs.kuleuven.ac.be

[3] Department of Applied Mathematics and Computer Science IPICyT, San Luis Potosí, México
E-mail: dmelchor@ipicyt.edu.mx

[4] Projet MERE INRIA-INRA, UMR Analyse des systèmes et Biométrie, INRA, 2, pl. Viala, 34060, Montpellier, France.
E-mail: mazenc@helios.ensam.inra.fr

[5] Department of Mechanical and Industrial Engineering, Southern Illinois University at Edwarsville, Edwardsville, IL 62026, USA.
E-mail: kgu@siue.edu

## 1 Introduction

Roughly speaking, a communication network consists of a collection of (network users/sources) elements interconnected to transfer information or data from one (network) *node* to another node through some (transmission or communication) *links*. A high-performance network represents a communication network able to support a large variety of applications, which can be transferred at high-speed and with low (communication) delay, and its dominant feature is the extremely large scale of the system [32].

There exists basically two ways to model communication networks systems using (discrete- or continuous-time) stochastic representations and/or deterministic (continuous-time) approximations, and we shall not insist on the difficulty of the task. Each modeling presents advantages and inconveniences, function of the difficulty in handling and/or analyzing the corresponding qualitative properties.

In the sequel, we shall consider only *continuous-time fluid approximations* due to their simplicity of representation as dynamical systems. For the brevity of the presentation, we do not insist on the way to derive such models or on their explicit interest for congestion control analysis, but we point out the discussions proposed

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 87–110, 2004.

in [32, 39] (see also [5, 6] for the connections between fluid-continuous and discrete-time representations).

The aim of the chapter is to discuss the asymptotic stability of various nonlinear fluid approximation models involving time-delays. We are interested in analyzing the delay effects on the asymptotic stability of the corresponding models using a wide panorama of methods and techniques encountered in *robust feedback control* theory. The presentation will be as simple as possible, and our approach is based on an *uncertainty* interpretation of the delay terms. We shall focus more on *methodological* aspects (matrix pencil construction, IQCs, form of the Lyapunov candidates, etc.), and less on detailed proofs.

More explicitly, we shall analyze three distinct fluid approximation models. The first model introduced by Izmailov [19, 20] is a second-order linear system, and describes the dynamics of a single connection between a source (controlled by an access regulator) and a distant node with a constant transmission capacity. The particularity of the model is the existence of two (independent) delays: round-trip time and control-time interval, respectively. The stability analysis is performed in two cases: constant, and time-varying round-trip times, respectively. In the first case, we shall point out some robustness/sensitivity properties of the control scheme, and in the second case, appropriate IQCs and related interconnected schemes are introduced, and discussed (Section 2).

The second model was introduced by Kelly in [22], and describes the dynamics of a collection of flows, all using a single resource, and sharing the same gain parameter. The model is a nonlinear first-order system including a single delay (round-trip time). The particular form of the model allows a very simple interpretation of the delayed state, an interpretation which yields various stability conditions including delay information (Section 3). Note that the stability is performed in two cases: differentiable or only continuous packet marking functions, i.e. functions which indicate potential congestion presence. Furthermore, both constant and time-varying delays are considered. In both cases, appropriate *model transformations* of the original model are proposed, and discussed. The interest of using such transformations of the original dynamics as well as their limitations can be found in [33].

Next, the third model was introduced by Misra, Gong and Towsley in [31], and is of TCP congestion-avoidance type. Note that this last model is a nonlinear second-order system including only a single delay (round-trip time): the first differential equation describes the TCP window dynamics, and the second equation models the bottleneck queue behavior. As in the previous case, we shall use appropriate model transformations combined with Lyapunov-Krasovskii functionals for deriving local asymptotic stability. Furthermore, we shall emphasize also the *chaotic behavior* of the fluid model.

The chapter is organized as follows: Section 2 addresses the Izmailov's models, as well as various frequency-domain techniques to handle the stability analysis for constant, and time-varying round-trip times. Section 3 is devoted to the Kelly's model, and to some simple *model transformations* combined with an appropriate construction of Lyapunov-Krasovskii functionals to handle the corresponding stability problem. Section 4 deals with a TCP-like model introduced by Misra, Gong

and Towsley, and its *sensitivity* with respect to the delays using both frequency- and time-domain methods. Section 5 is devoted to a description of software for simulation and stability analysis of time-delay models, as those considered in the text. Some concluding remarks end the chapter. The notations are standard.

## 2 Izmailov's models

### 2.1 Model

Izmailov proposed in [19, 20] the following deterministic models of a single connection between a source controlled by an access regulator and a distant node with a constant transmission capacity $\mu$:

$$\begin{cases} \dot{x}_1(t) = x_2(t-\tau_1) - \mu \\ \dot{x}_2(t) = -a(x_1(t-\tau_2) - \bar{X}) - b(x_1(t-\tau_2-r) - \bar{X}), \end{cases} \tag{1}$$

and:

$$\begin{cases} \dot{x}_1(t) = x_2(t-\tau_1) - \mu \\ \dot{x}_2(t) = -a(x_1(t-\tau_2) - \bar{X}) - b(x_1(t-\tau_2-r) - \bar{X}) \\ \qquad\qquad -c\left(x_1\left(t-\tau_2-\frac{r}{2}\right) - \bar{X}\right), \end{cases} \tag{2}$$

where $x_1$ represents the buffer contents, $x_2$ the current input rate, and $\bar{X}$ the target value, respectively. Using the new variable $y(t) = x_1(t) - \bar{X}$, (1) and (2) lead to the following second-order delay equations with two discrete and independent delays $\tau$ and $r$:

$$\ddot{y}(t) + ay(t-\tau) + by(t-\tau-r) = 0, \tag{3}$$

and respectively:

$$\ddot{y}(t) + ay(t-\tau) + by(t-\tau-r) + cy\left(t-\tau-\frac{r}{2}\right) = 0, \tag{4}$$

where the 'total' delay $\tau = \tau_1 + \tau_2$ represents the *round-trip time*, and $r$ is the *control time-interval*.

In the sequel, we shall focus on the stability analysis of (3), and (4). More explicitly, we shall consider a more general model including both (3) and (4) as particular examples. As mentioned in the Introduction, the extension to a particular time-varying round-trip time is also considered.

### 2.2 Linear stability analysis: frequency-domain methods

As seen in Niculescu [34, 35], the delays $\tau$ and $r$ have a *stabilizing* effect if the gains $a, b$ and $c$ satisfy some appropriate assumptions, even if the system free of delays is not asymptotically stable. In this sense consider the case $a > \mid b \mid, b < 0$ in (3), which corresponds to a system that is an *oscillator* if it is free of delays, but which is stable for sufficiently *small* delays $\tau, r \neq 0$.

The underlying idea is to analyze first the stability of (3) if $\tau = 0$, but $r \neq 0$, and second to assume also $\tau \neq 0$. The interesting property of this system is that *any sufficiently small delay r* can stabilize (3) if $\tau = 0$ (oscillator subject to a delay output feedback). Note that the assumption $a > 0$ and $b < 0$ was already encountered in the works of Izmailov (see, for instance, [20]), but the argument was completely different.

More precisely, we shall consider the following more general second-order delay system:

$$\ddot{y}(t) + \sum_{k=0}^{n} a_k y\left(t - \frac{k}{n}r - \tau_0 - \delta(t)\right) = 0, \tag{5}$$

with appropriate defined initial condition (see, for instance, [10, 33]). We assume $\delta(t)$ to be a continuously differentiable function of time, with a bounded derivative, such that:

$$0 \leq \delta(t) \leq \varepsilon, \quad \forall t \in \mathbf{R}^+ \tag{6}$$

$$\dot{\delta}(t) \leq \rho, \quad \forall t \in \mathbf{R}^+, \quad \rho < 1. \tag{7}$$

Simple computation allows us to rewrite the system (5) with $\delta(t) = 0$ as follows:

$$\dot{x}(t) = Ax(t) + \sum_{k=0}^{n} A_k x\left(t - \frac{k}{n}r - \tau_0\right), \tag{8}$$

where:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad A_k = \begin{bmatrix} 0 & 0 \\ -a_k & 0 \end{bmatrix}, \qquad k = 0, \ldots, n \tag{9}$$

and $x = [y\ \dot{y}]^T$. Introduce now the following matrix pencils $\Lambda_1 \in \mathbf{C}^{2n \times 2n}$ and $\Lambda_2 \in \mathbf{C}^{n \times n}$:

$$\Lambda_1(z) = z \begin{bmatrix} 1 & & 0 & 0 \\ & \ddots & & \\ 0 & & 1 & 0 \\ 0 & \ldots & 0 & a_n \end{bmatrix} + \begin{bmatrix} 0 & -1 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ & & & \ddots & & & & \\ 0 & & & & & & & -1 \\ -a_n & -a_{n-1} & \ldots & -a_1 & 0 & a_1 & \ldots & a_{n-1} \end{bmatrix} \tag{10}$$

$$\Lambda_2(z) = z \begin{bmatrix} 1 & & 0 & 0 \\ & \ddots & & \\ 0 & & 1 & 0 \\ 0 & \ldots & 0 & a_n \end{bmatrix} + \begin{bmatrix} 0 & -1 & & 0 \\ & & \ddots & \\ 0 & 0 & & -1 \\ a_0 & a_1 & \ldots & a_{n-1} \end{bmatrix}, \tag{11}$$

and define the following set:

$$\Lambda_{0,+} = \left\{ (r_{h_i}, \alpha_h) :\ r_{h_i} = \frac{\alpha_h}{\omega_{hi}} > 0 : \quad e^{-j\alpha_h} \in \sigma(\Lambda_1) - \sigma(\Lambda_2), \right.$$
$$\left. j\omega_{hi} \in \sigma\left(\sum_{k=0}^{n} A_k e^{-j\frac{\alpha_h k}{n}}\right) - \{0\},\ 1 \leq h \leq 2n,\ 1 \leq i \leq n \right\}, \tag{12}$$

where $\sigma(\Lambda_i)$ denotes the set of generalized eigenvalues of the corresponding matrix pencil $\Lambda_i$. As seen in [33], the distribution of generalized eigenvalues of matrix pencils $\Lambda_1$, $\Lambda_2$ with respect to the unit circle of the complex plane combined with some explicit computation of the eigenvalues of the imaginary axis of some complex matrices give the complete asymptotic behavior of the original (8) with respect to the parameter $r$, when $\tau_0$ is assumed 0.

In the sequel, we reduce our analysis only to the first delay intervals guaranteeing stability with respect to $r$, and $\tau$. First, in the case of *small delays*, we have the following stability result, which simply extends the remarks concerning the stabilization of oscillatory systems by using delays in the output feedback control laws:

**Proposition 1 (Small delay).** *[35] Assume that*

$$\sum_{k=0}^{n} a_k > 0, \sum_{k=1}^{n} k a_k < 0. \tag{13}$$

*Then there exists a sufficiently small positive value* $\varepsilon > 0$*, such that (8) with* $r = \varepsilon$*,* $\tau_0 = 0$ *is asymptotically stable.*

The next result gives the complete characterization of the *first switch*, that is the case when the roots of the characteristic equation associated to the original system cross the imaginary axis towards instability when the delay parameter $r$ is varying from 0 to $+\infty$.

**Proposition 2 (First switch).** *[35] The system (8) with* $\tau_0 = 0$*, satisfying the condition (13), is asymptotically stable for all delay values r satisfying*

$$0 < r < r_1(a_0, \ldots, a_n),$$

*where*

$$r_1(a_0, \ldots a_n) = \mathbf{inf}\{\gamma : \quad (\gamma, \alpha) \in \Lambda_{0,+}\}. \tag{14}$$

*Furthermore, if* $r = 0$ *or* $r = r_1(a_0, \ldots, a_n)$*, the corresponding associated characteristic equation has at least one pair of complex conjugate roots on the imaginary axis.*

**Proposition 3 (Delay bounds).** *[35] The system (8) satisfying the constraints (13) is asymptotically stable for all delays r and* $\tau_0$ *satisfying the following conditions:*

$$\begin{cases} 0 < r < r_1(a_0, \ldots, a_n) \\ 0 < \tau_0 < \tau_{0,r} = \min\limits_{\omega_s} \left\{ \dfrac{1}{\omega_s} \cdot \mathbf{arctan} \dfrac{-\sum\limits_{k=1}^{n} a_k \sin \frac{\omega_s k r}{n}}{\sum\limits_{k=0}^{n} a_k \cos \frac{\omega_s k r}{n}} \right\}, \end{cases} \tag{15}$$

*where* $\omega_s$ *belongs to the set of positive solutions of the equation:*

$$\omega^4 = \sum_{k=0}^{n} a_k^2 + 2 \sum_{k=1}^{n} \sum_{h=0}^{k-1} a_k a_h \cos\left(\frac{(k-h)\omega r}{n}\right). \tag{16}$$

*Furthermore, assume that the chosen delay r and the solution $\tilde{\omega}_s$ defining the corresponding upper bound $\tau_{0,r}$ in (15) satisfy the condition:*

$$\tilde{\omega}_s^3 > \frac{1}{2}\sum_{k=1}^{n}\sum_{h=0}^{k-1}\frac{a_k a_h (k-h) r}{n}\sin\left(\frac{(k-h)\tilde{\omega}_s r}{n}\right). \tag{17}$$

*Then:*

*i) if $\tau_0 = \tau_{0,r} + \varepsilon$, with $\varepsilon > 0$ sufficiently small, the system (5) is unstable.*
*ii) if the equation (16) has only one positive solution, then there does not exist any $\tau > \tau_{0,r}$ such that the system (5) with $\delta(t) = 0$ is asymptotically stable.*

*Remark 1 (Instability).* It seems clear that the same delay interval $[0, \tau_{0,\hat{r}})$ leads to *instability* of the system (5) with $\delta(t) = 0$ for a given $(n+1)$-tuple $(a_0, \ldots, a_n)$ satisfying the constraints (13) if the original system with $r = \hat{r} > r_1(a_0, \ldots, a_n)$ and $\tau = 0$ is *unstable*.
Furthermore, the instability is *delay-independent* if the equation (16) has *only* one positive solution, but we have always roots crossing the imaginary axis.

*Remark 2 (Instability persistence).* It is important to note that the equation (16) always has at least one positive solution for $\omega$ in the interval

$$\left[\min_{\mathbf{i}\in\mathcal{I}_n}\sqrt{\sum_{k=1}^{n}(-1)^{i_k}a_k^2}, \max_{\mathbf{i}\in\mathcal{I}_n}\sqrt{\sum_{k=1}^{n}(-1)^{i_k}a_k^2}\right], \tag{18}$$

where

$$\mathcal{I}_n = \{i = (i_1, \ldots i_n) : i_k \in \{1,2\}, \forall k = \overline{1,n}\}$$

is an appropriate index family. We can, therefore, conclude that the *upper* bound on $\tau$ will *always be finite*, that is one may expect a sequence of stability/instability delay intervals in terms of $\tau_0$, with *instability persistence* for sufficiently large delays (see also [33]). In other words, there exists a finite value $\bar{\tau}_{0,r}$ such that the system is unstable for $\tau > \bar{\tau}_{0,r}$.

*Remark 3 (Robustness/sensitivity of the control algorithm).* Based on the results above, it is easy to see that *reducing* the *control-time interval* will increase the *sensitivity* of the control algorithm with respect to the *round-trip time* (see, for instance, Figures 1 and 2, below).

If necessary or desirable, more precise stability information can be obtained by directly computing and monitoring the rightmost characteristic roots of the closed-loop system. This is illustrated in Figure 1. The characteristic roots of the system (4) with $a = 2.25,\ b = c = -1,\ \tau = 0$ are shown as a function of the control interval $r$, computed with the software package DDE-BIFTOOL [11]. By automatic continuation in the two-parameter space $(r, \tau)$ of solutions, for which characteristic roots lie on the imaginary axis, the stability region in this parameter space is obtained, see Figure 2 (above). For the optimal value of the control interval (w.r.t. robustness in the `rtt`), the rightmost characteristic roots are shown in Figure 2 (below).

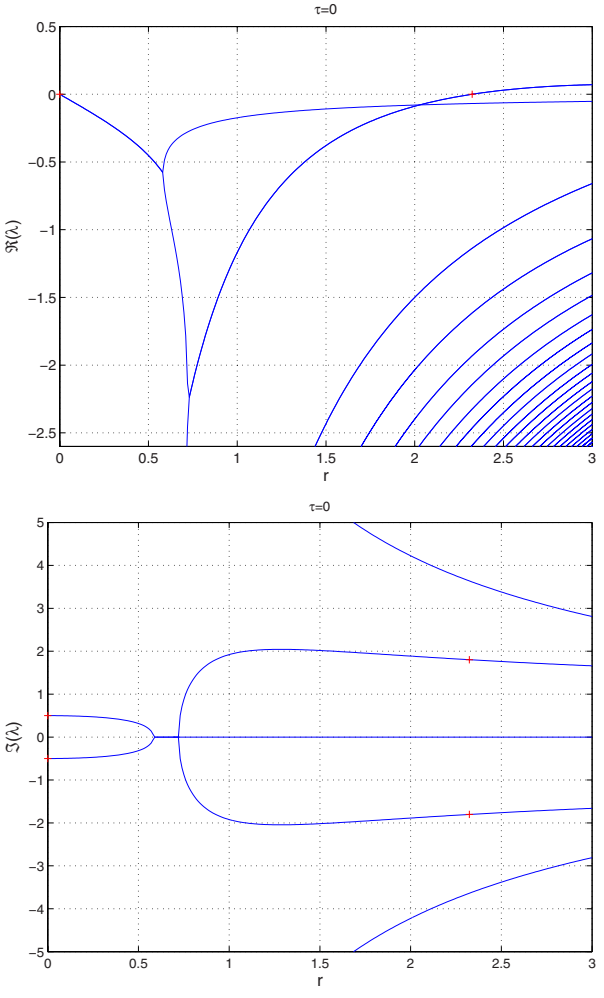



**Fig. 1.** Real and imaginary parts of the rightmost characteristic roots of Equation (4) as a function of $r$ when $\tau = 0$. Characteristic roots on the imaginary axis are indicated with ('+'). The parameter values are $a = 2.25,\ b = c = -1$.

Consider now the *time-varying delay* case.

For the sake of convenience, denote: $h_k = \dfrac{k}{n} r + \tau_0$. Then, the original system can be rewritten as follows:

$$\dot{x}(t) = Ax(t) + \sum_{k=0}^{n} A_k x(t - h_k - \delta(t)). \tag{19}$$

Using the relation:

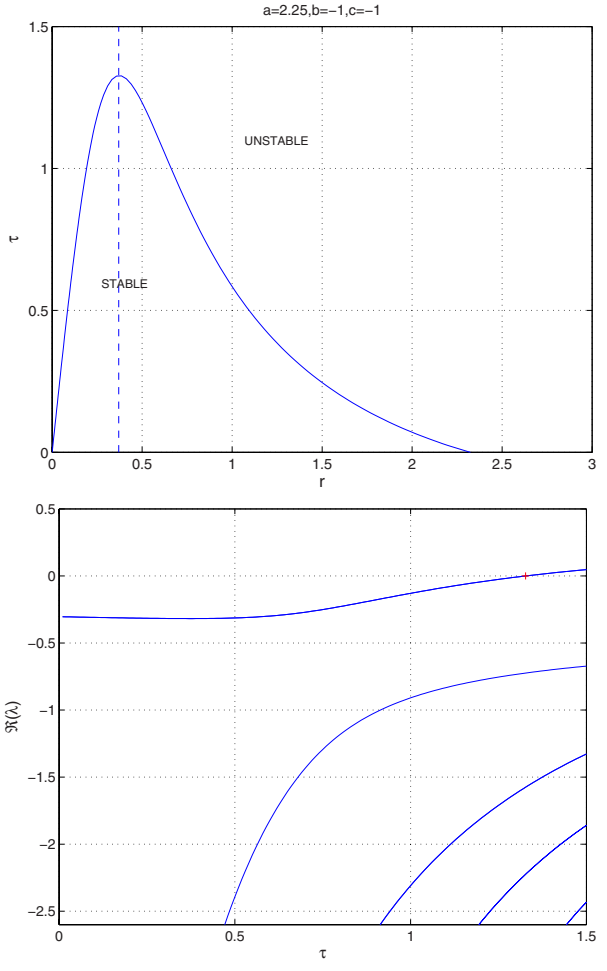



**Fig. 2.** (above) Stability region in the $(r,\tau)$-plane, whose boundary is computed with the software DDE-BIFTOOL [11] as a branch of Hopf bifurcations. For the optimal control interval $r \approx 0.37$, indicated with the dashed line, the real parts of the rightmost characteristic roots are shown as a function of the round-trip time $\tau$ (below).

$$x(t-h_k-\delta(t)) = x(t-h_k) - \int_0^{\delta(t)} \frac{\partial x}{\partial t}(t-h_k-\xi)d\xi = x(t-h_k)$$
$$- \int_0^{\delta(t)} \left[ Ax(t-h_k-\xi) + \sum_{l=0}^{n} A_l x(t-h_k-\xi-h_l-\delta(t-h_k-\xi)) \right] d\xi. \qquad (20)$$

the above system can be rewritten as one with time-invariant delay subject to uncertain feedback [14]:

$$\dot{x}(t) = Ax(t) + \sum_{k=0}^{n} A_k x(t-h_k) + \sum_{k=0}^{n+1} A_k u_k(t), \tag{21}$$

where $A_{n+1} = I$, and $u_k$ is defined as

$$u_k(t) = \begin{cases} -\int_0^{\delta(t)} \sum_{l=0}^{n} A_l x(t-h_k-\xi-h_l-\delta(t-h_k-\xi))\,d\xi, & 0 \le k \le n, \\ -\int_0^{\delta(t)} \sum_{l=0}^{n} A_l A x(t-h_l-\xi)d\xi, & k = n+1. \end{cases} \tag{22}$$

Using the Jensen inequality combined with Fubini's theorem, one gets:

$$\int_0^t u_k(\theta)^T u_k(\theta)d\theta \le \varepsilon^2 c_k^2 \int_0^t x(\theta)^T x(\theta)d\theta, \tag{23}$$

for all $1 \le k \le n$, where:

$$c_k = \sqrt{\frac{n+1}{1-\rho} \sum_{l=0}^{n} \|A_l\|^2}. \tag{24}$$

Therefore, the system can be written in the following form:

$$\begin{cases} \dot{x}(t) = Ax(t) + \sum_{k=0}^{n} A_k x(t-h_k) + \sum_{k=0}^{n+1} A_k u_k(t) \\ y_k(t) = \varepsilon c_k x(t), \quad 0 \le k \le n+1, \end{cases} \tag{25}$$

under the feedback:

$$u_k = \Delta_k y_k, \quad 0 \le k \le n+1 \tag{26}$$

where the gain of the "uncertainty" $\Delta_k$ is bounded by 1.

In conclusion, we have the following result:

**Theorem 1.** *The original system (5) is uniformly asymptotically stable for all delays $r$ and $\tau_0$ satisfying Propositions 2 and 3 above and for any time-varying delay uncertainty $\delta(t)$ satisfying (6) and (7) if there exists scalars $\alpha_k$, $0 \le k \le n+1$ such that:*

$$\|\Lambda H(j\omega)\Lambda^{-1}\|_\infty < \frac{1}{\varepsilon}, \tag{27}$$

*where: $\Lambda = \mathrm{diag}(\alpha_0 I, \alpha_1 I \ldots, \alpha_{n+1} I)$, and:*

$$H(s) = \begin{bmatrix} c_0 I \\ c_1 I \\ \vdots \\ c_{n+1} I \end{bmatrix} \left( sI - A - \sum_{k=0}^{n} A_k e^{-s\left(\frac{k}{n} r + \tau_0\right)} \right)^{-1} [A_0 \ \ldots \ A_n \ I]. \tag{28}$$

*Remark 4 (computing estimations).* The result above can be also used to *estimate* a bound $\varepsilon_{max} > 0$ of the uncertain time-varying delay $\delta(t)$ such that the corresponding fluid model is uniformly asymptotically stable. Furthermore, this bound on $\varepsilon$ can be seen as a *measure* of the degree of robustness of the corresponding delay system.

*Remark 5 (frequency-sweeping tests).* Condition (27) can be checked by a *frequency-sweeping* method. As discussed in Chapter 8 of [14], the robust stability conditions for the system described by (25) and (26) with the gain of $\Delta_k$ bounded by 1 (or equivalently, checking Condition (27)) can also be written in a Lyapunov-Krasovskii functional form, which can be reduced to an LMI through a discretized Lyapunov functional method.

## 3 Kelly's model

### 3.1 Model

The first model to be considered is:

$$\dot{x}(t) = k\left(w - x(t-\tau)p(x(t-\tau))\right), \tag{29}$$

under appropriate initial conditions defined on the interval $[-\tau, 0]$, and with $k, w$ positive real.

First, we shall consider $p(\cdot)$ as a $\mathcal{C}^1$ continuous function, strictly increasing, and bounded by 1, and second we shall relax the assumptions above to the Lipschitz continuity of $xp(x)$ under the constraint that $p(\cdot)$ is still strictly increasing. Note that a slight modification of (29) can be encountered in [8], where $kw$ is replaced by $\Delta = \dfrac{1}{h^2}$, and $k$ by some positive real $\beta$.

To the best of the authors' knowledge, such a model was proposed by Kelly in [22] for describing the dynamics of a collection of flows, all using a single resource, and sharing the same gain parameter $k$. The delay $\tau$ represents the round-trip time, and is assumed constant. The function $p(\cdot)$ can be interpreted as the fraction of packets indicating (potential) congestion (presence) [8]. Furthermore, the assumptions on the function $p(\cdot)$ are natural in the context of TCP behavior (see, for instance, [8, 22, 39]).

As specified in the Introduction, we shall consider first the asymptotic stability of (29). The stability of the nonlinear delay system above will be given using an appropriate construction of a Lyapunov-Krasovskii functional [15] combined with the local (stability) analysis of the linearized model. Two model transformations (derived by integration over one delay interval) will be introduced, and discussed. The uncertainty interpretation of the delay term will allow the extension of the methodology to the case of time-varying delays.

### 3.2 Stability analysis: model transformations and Lyapunov's second method

First, we restrict our analysis to the case where the function is linear, $p(s) = s$. In the context of the flows, this consideration is the most simple approximation for the real function $p(\cdot)$, see, for instance, [22]. Thus, (29) becomes:

$$\begin{aligned} \dot{x}(t) &= \alpha - \beta x^2(t-\tau), \\ x(t) &= \varphi(t), \forall t \in [-\tau, 0], \end{aligned} \tag{30}$$

where $\alpha, \beta, \tau$ are positive constants and $\varphi(\cdot)$ is the initial function with the following norm $|\varphi| = \sup_{\theta \in [-\tau, 0]} |\varphi(\theta)|$ (see, e.g. [15]). It is clear that the equilibrium points of system (30) are $x^* = \pm\sqrt{\frac{\alpha}{\beta}}$.

Let $y(t) = x(t) - x^*$, then system (30) becomes

$$\dot{y}(t) = -2\beta x^* y(t-\tau) - \beta y^2(t-\tau). \tag{31}$$

It is well known from the theory of differential and difference equations that if the trivial solution of the system

$$\dot{y}(t) = -2\beta x^* y(t-\tau), \tag{32}$$

is asymptotically stable, then the trivial solution of (31) and hence the equilibrium point $x^*$ of (30) is locally asymptotically stable, see [15]. On the other hand, we know that system (32) is asymptotically stable if and only if $-2\beta x^* \in \left(-\frac{\pi}{2\tau}, 0\right)$, see [33]. Therefore, the unique equilibrium point of system (30) which is asymptotically stable is $x^* = \sqrt{\frac{\alpha}{\beta}}$. In conclusion, it is clear that the equilibrium point $x^*$ of (30), is (locally) asymptotically stable when the norm of the initial function $|\varphi|$ is such that the quantity $|\tilde{\varphi}|$ (where $\tilde{\varphi} = \varphi - x^*$) is *sufficiently small*.

The next step is to *interpret* the nonlinear term $y(t-\tau)^2$ in (31) as the product between some *uncertainty* $\delta(t)$, and the delayed state $y(t-\tau)$. In this case, the system (31) rewrites as:

$$\dot{y}(t) = -(a + \delta(t))\, y(t-\tau), \tag{33}$$

where $a = 2\beta x^*$,and $\delta(t)$ $(= y(t-\tau))$ denotes the uncertainty. It is clear that the class (33) includes the original system (31). Assume the uncertainty $\delta(t)$ bounded by some positive real $\rho$: $|\delta(t)| \leq \rho, \forall t$. The particular way to construct the solution of the original system ('step-by-step') by integration from one delay interval $(t_1 - \tau, t_1]$ to $(t_1, t_1 + \tau)$, for some positive $t_1$ allows the coherence of the interpretation[6]. Thus, one needs to guarantee that if $y(t)$ is sufficiently small on some $[t-\tau, t)$, then we have the same property for the next delay interval $[t, t+\tau)$, and by iteration for all next delay intervals. In other words, if there exists some Lyapunov-Krasovskii functional with negative derivative from some interval of delay size to the next one, its monotonicity will define appropriate *bounds* $\rho$, by iteration, for the whole class (33).

Consider now (33), but free of uncertainty, that is:

$$\dot{y}(t) = -a y(t-\tau). \tag{34}$$

As seen above, this system is asymptotically stable if $a\tau < \frac{\pi}{2}$. Then, there exists a Lyapunov-Krasovskii functional [23] of the form:

[6] in terms of local asymptotic stability

$$V(y_t) = mu(0)y^2(t) - 2amy(t)\int_{-\tau}^{0} u(\tau+\theta)y(t+\theta)d\theta$$
$$+ma^2\int_{-\tau}^{0}\int_{-\tau}^{0} u(\theta_1-\theta_2)y(t+\theta_1)y(t+\theta_2)d\theta_1 d\theta_2$$
$$+\int_{-\tau}^{0}(\mu_1+(\tau+\theta)\mu_2)y(t+\theta)^2 d\theta, \tag{35}$$

where $y_t(\eta) = y(t+\eta)$ (for all $\eta$), and $m = \mu_0+\mu_1+\tau\mu_2$, with $\mu_i$, $i = 1,2,3$ positive real numbers.

Here, $u(\cdot)$ is defined by the following relation:

$$u(\theta) = \int_0^{\infty} k(t)k(t+\theta)dt, \tag{36}$$

where $Kk$ is the fundamental solution associated to the nominal system (34), which is assumed exponentially stable. It is important to note that the exponential stability of (34) guarantees the existence of the integral (36). Furthermore, it is easy to see that $u(0) > 0$, and that $u$ can be appropriately defined as the solution of some differential equation derived from the original dynamics [23] (see also [24] for some applications).

The functional has the following properties:

1) for some positive $\gamma_i$, $i = 1,2$

$$\gamma_1|y(t)|^2 \le V(y_t) \le \gamma_2|y_t|^2.$$

2) and along the solutions of (34),

$$\dot{V}(y_t) = -\omega(y_t), \tag{37}$$

where:

$$\omega(y_t) = \mu_0 y(t)^2 + \mu_1 y(t-\tau)^2 + \mu_2\int_{-\tau}^{0} y(t+\theta)^2 d\theta \tag{38}$$

for any given $\mu_0,\mu_1,\mu_2$ positive.

3) Based on the remarks above, the scalar function $u(\cdot)$ satisfies the following ordinary differential equation $\ddot{u}(t)+a^2u(t) = 0$, with the following additional conditions:
   i) $\dot{u}(t) = -au(t-\tau)$, for all $t \ge 0$;
   ii) $u(-t) = u(t)$, for all $t \ge 0$, and
   iii) $u(\tau) = \frac{1}{2a}$.

Deriving (35) along the trajectories of (33), and using the properties mentioned above, we obtain:

**Theorem 2.** *[28] The trivial solution of the nonlinear system (31) is locally asymptotically stable if* $\tau\sqrt{\alpha\beta} < \dfrac{\pi}{4}$, *and the initial function* $\tilde{\varphi}$ *belongs to the set:*

$$U_l = \{\psi : \quad V(\psi) < l\},$$

*where* $l = \sqrt{\frac{\gamma_1}{\gamma_2}}\frac{\rho}{2\beta}$, *with* $\rho > 0$ *satisfying the following inequalities:*

$$\begin{cases} \mu_0 > m\rho u(0), \\ \mu_1 > m\rho\left(u(0) + a\tau \max_{\theta\in[0,\tau]} u(\theta)\right) \\ \mu_2 > m\rho a\tau \max_{\theta\in[0,\tau]} u(\theta). \end{cases} \tag{39}$$

*Remark 6.* The conditions in the Theorem 2 are easy to check. Indeed, the scalar function $u$ can be easily derived from the conditions (i)–(iii) above, and thus, its maximum on the interval $[0,\tau]$ can be computed. All the other quantities will define appropriate upper bounds for $\rho$.

Next, we assume the delay *time-varying*, that is:

$$\dot{y}(t) = \alpha - \beta x(t - \tau + \eta(t))^2, \tag{40}$$

with an appropriate initial condition, and such that $\eta$ is a continuous and bounded function satisfying the following inequalities:

$$|\,\eta(t)\,| \le \eta_0 < \tau, \;\; |\,\dot{\eta}(t)\,| \le \eta_1 < 1. \tag{41}$$

Using the same idea as in the previous case, we shall define:

$$\dot{y}(t) = -(a + \Delta(t))y(t - \tau + \eta(t)), \tag{42}$$

with $\Delta(t) = x(t - \tau + \eta(t))$. In order to derive stability conditions for (40), we shall integrate (42) over one delay interval, that is:

$$\dot{y}(t) = -(a + \Delta(t))\left(y(t-\tau) + \int_0^{\eta(t)} \dot{y}(t - \tau + \theta)d\theta\right). \tag{43}$$

Using the same idea as in the constant delay case, but the Lyapunov-Krasovskii candidate adapted to (43), we obtain:

**Theorem 3.** *[28] The trivial solution of the nonlinear system (40) with time-varying delay* $\tau + \eta(t)$*, satisfying the conditions (41) is locally asymptotically stable if* $\tau\sqrt{\alpha\beta} < \frac{\pi}{4}$*, and the initial function* $\tilde{\psi}$ *belongs to the set:*

$$U_l = \{\psi : \quad V(\psi) < L\},$$

*where* $L = \sqrt{\frac{\gamma_1}{\gamma_2}}\frac{\sigma}{2\beta}$, *with* $\sigma > 0$ *satisfying for some* $k_j > 0$*,* $j = 1,2,3,4$ *the following inequalities:*

$$\begin{cases} \mu_0 > & u(0)m\left(k_1\sigma + k_2\eta_0(a+\sigma)^2\right) \\ \mu_1 > & m\sigma\left(\frac{1}{k_1}u(0) + k_3 ha\left(\max_{\theta\in[0,\tau]}|u(\theta)|\right)\right) \\ \frac{\mu_2}{2} > & ma\left(\max_{\theta\in[0,\tau]}|u(\theta)|\right)\left(\frac{1}{k_3}\sigma + k_4(a+\sigma)^2\eta_0\right) \\ \frac{\mu_2}{2} > & \frac{m(a+\sigma)^2}{1-\eta_1}\left(\frac{1}{k_2}u(0) + \frac{1}{k_4}ha\left(\max_{\theta\in[0,\tau]}|u(\theta)|\right)\right) \end{cases}. \tag{44}$$

Here, the Lyapunov-Krasovskii functional candidate is:

$$V(y_t) = mu(0)y^2(t) - 2amy(t)\int_{-\tau}^{0} u(\tau+\theta)y(t+\theta)d\theta$$
$$+ma^2\int_{-\tau}^{0}\int_{-\tau}^{0} u(\theta_1-\theta_2)y(t+\theta_1)y(t+\theta_2)d\theta_1 d\theta_2$$
$$+\int_{-\tau}^{0}(\mu_1+(3\tau+\theta)\mu_2)y(t+\theta)^2 d\theta,$$

and the functional $\omega$:

$$\omega(y_t) = \mu_0 y(t)^2 + \mu_1 y(t-\tau)^2 + \mu_2\int_{-\tau}^{0} y(t+\theta)^2 d\theta,$$

where $m = \mu_0+\mu_1+3\tau\mu_2$, and $\mu_0,\mu_1,\mu_2$ positive real numbers, and $u(\cdot)$ defined by the formula (36).

Consider now more general *probability functions* $p$. The construction above still works, but it is more computationally involved. However, the methodology cannot be directly applied when $p$ is *not differentiable*. To treat this case as well, we describe below a new way to handle the problem: a different *model transformation* combined with a *simpler* Lyapunov-Krasovskii functional candidate, but with a more conservative delay bound.

Define the function $F : \mathbf{R} \mapsto \mathbf{R}$ by:

$$F(y) = -k\left(w-(y+x^*)p(y+x^*)\right),$$

where $x^*$ denotes the equilibrium of (29) for the system free of delay, that is the *unique (positive) solution* of the equation $w = xp(x)$. Thus, the original system rewrites as follows:

$$\dot{y}(t) = -F(y(t-\tau)). \tag{45}$$

In order to understand the model transformation to be used, assume, as in the previous case, that $p(\cdot)$ is $\mathcal{C}^1$. Next, an integration over one delay interval leads to the following *model transformation*:

$$\dot{y}(t) = -F(y(t)) + \int_{t-\tau}^{t} F'(y(\theta))d\theta, \tag{46}$$

where $F'$ is well-defined and continuous, since $p(\cdot)$ is $\mathcal{C}^1$. Next, introduce now the following Lyapunov-Krasovskii functional candidate:

$$V(y_t) = V_1(y(t)) + V_2(y_t), \tag{47}$$

where:

$$V_1(y(t)) = \int_0^y F(\theta)d\theta,$$
$$V_2(y_t) = \frac{1}{8\tau}\int_{t-2\tau}^{t}\left(\int_{\theta}^{t} F(y(\eta))d\eta\right)d\theta,$$

Since the function $xp(x)$ is 0 for $x = 0$, and $+\infty$, when $x \to +\infty$, then it is not difficult to prove that: $\lim_{y\to+\infty} V_1(y) = +\infty$. Furthermore, $F(y)$ is positive (negative) when $y$ is positive (negative). Thus, $V_1$ and $V_2$ are both positive.

In conclusion, $V$ is positive-definite and has an infinitesimal upper limit (see, e.g. [33] for the definitions). Next, with all the arguments given above, the stability can be easily performed by computing the derivative of $V$ over the system's trajectories, and imposing its negativity. The corresponding condition will depend on the size of $\tau$.

The novelty of the method lies first in the way to define the model transformation, and second in the construction itself of the Lyapunov candidate.

*Remark 7.* The construction of the model transformation is similar to the construction of (43) above, but it uses directly the nonlinearity involved in the system's representation, and not only an appropriate (linear) *"part"* of it.

Now, if the function $p(\cdot)$ is only continuous, but concave, then its concavity combined with the properties of $F$ (globally Lipschitz on $[x^*, +\infty)$, with $2k$ as Lipschitz constant) leads to the following:

**Theorem 4.** *[27] Consider the system (29) and assume that $p(\cdot)$ is continuous, nonnegative, nondecreasing, and bounded by 1 and such that $\lim_{x\to+\infty} p(x) = 1$.*

*Furthermore, assume that for all $\beta > 0$ and for all $\gamma \in [0, \beta)$,*

$$\frac{p(\beta)}{\beta} \geq \frac{p(\beta) - p(\gamma)}{\beta - \gamma}, \tag{48}$$

*and that the delay $\tau$ satisfies*

$$0 \leq \tau \leq \frac{1}{4k}. \tag{49}$$

*Then all the solutions of the system (29), nonnegative for all $t \geq -\tau$, converge to the unique equilibrium point of the system as the time goes to the infinity.*

## 4 TCP-like model (Misra, Gong, Towsley)

### 4.1 Model

In the sequel, we briefly introduce some fluid model of *TCPs congestion-avoidance mode*, as proposed in [31]:

$$\begin{cases} \dot{w}(t) = \frac{1}{\tau(t)} - \frac{w(t)}{2}\frac{w(t-\tau(t))}{\tau(t-\tau(t))}p(t-\tau(t)) \\ \dot{q}(t) = \begin{cases} -c + \frac{n(t)}{\tau(t)}w(t), & q > 0 \\ \mathbf{max}\left\{0, -c + \frac{n(t)}{\tau(t)}w(t)\right\}, & q = 0, \end{cases} \end{cases} \tag{50}$$

where $w(t)$ denotes the average TCP window size (packets), $q$ the average queue length (packets), $n$ the load factor (number of TCP sessions), $p$ the probability of

packet mark (taking values in $[0,1]$), and $\tau(t)$ the round trip-time (seconds), denoted `rtt`, and defined as follows:

$$\tau(t) = \frac{w(t)}{c} + \tau_p, \tag{51}$$

with $c$ denoting the link capacity (packets/second), and $\tau_p$ the propagation delay (seconds).

As seen in [18, 31], the first differential equation describes the *TCP window control dynamic*. Indeed, the first term describes the window's additive increase phase, and the second term the multiplicative decreasing phase (including the packet marking probability). Different AIMD continuous-time models can be found in [22, 27].

The second differential equation describes the bottleneck queue length as the difference between the packet arrival rate: $\frac{nw}{\tau}$ and the link capacity $c$, and assuming that there are no internal dynamics in the bottleneck.

In order to reduce the degree of conservatism of such a marking probability function, we need some further assumptions which are natural with respect to the TCP behavior. Thus, for example, the profile of the marking probability function to be considered is given by [18]:

$$p(q) = \begin{cases} 0, & q(t) \le q_{min} \\ \frac{p_{max}}{q_{max}-q_{min}}(q(t)-q_{min}), & \\ & q_{min} < q(t) < q_{max} \\ 1, & q(t) \ge q_{max}, \end{cases} \tag{52}$$

where the values $q_{min}$, $q_{max}$ and $p_{max} \in (0,1)$ are to be specified and represent the corresponding RED (random early detection) parameters. Based on the remarks and simulations in Misra *et al.* [31], the model (50)-(51) accurately captures the dynamics of the TCP with the assumption that the *TCP timeout mechanism is ignored* (see also [18]).

### 4.2 Linear stability analysis

Let us consider a model of the form (50) with the assumption that the TCP load $n(t)$ and the round-trip time $\tau(t)$ are time-invariant, and that $p(t) = K_p\, q(t)$. Under these conditions a scaling of state and time allows to normalize (50) to

$$\begin{cases} \dot{W}(t) = 1 - \frac{W(t)W(t-1)}{2} KQ(t-1) \\ \dot{Q}(t) = \begin{cases} W(t) - C, & Q > 0 \\ \max(W(t)-C, 0), & Q = 0 \end{cases}, \end{cases} \tag{53}$$

where

$$W = w,\ Q = q/n,\ t^{(\text{new})} = t^{(\text{old})}/\tau,\ C = \frac{\tau c}{n},\ K = K_p n.$$

The importance of this transformation lies in the fact that the four model parameters $(K_p, n, c, \tau)$ are reduced to only two parameters $(K, C)$, which facilitates the study of

the dependence of the attractors and their stability properties on the system's parameters.

In the normalized coordinates $(W,Q)$ the unique equilibrium point is given by

$$(W^*,Q^*) = (C, \frac{2}{KC^2}). \tag{54}$$

Linearization around it results in the second order differential equation in $\tilde{Q} \triangleq Q - Q^*$,

$$\ddot{\tilde{Q}}(t) + \frac{1}{C}\dot{\tilde{Q}}(t) + \frac{1}{C}\dot{\tilde{Q}}(t-1) + \frac{KC^2}{2}\tilde{Q}(t-1) = 0,$$

whose characteristic equation is given by

$$H(\lambda) \triangleq \lambda^2 + \frac{1}{C}\lambda + \frac{1}{C}\lambda e^{-\lambda} + \frac{KC^2}{2}e^{-\lambda} = 0. \tag{55}$$

To characterize the stability region in the $(K,C)$-plane we first fixed $C > 0$ and consider the stability region as a function of $k$. We have the following result:

**Proposition 4.** *[30] For each value of C, there exists exactly* one *stability interval as a function of K, i.e.* $K \in (0,\ \bar{K}(C))$ *with* $\bar{K}(C) \in \bar{\mathbf{R}}^+$.

While a zero eigenvalue occurs for $K = 0$, the value $\bar{K}(C)$ in Proposition 4 corresponds to a subcritical *Hopf bifurcation* of the original nonlinear system. By numerical continuation of such Hopf bifurcation in the two-parameter space $(K,C)$, the stability region can be computed. See [38] for theory on continuation and bifurcation analysis and [11] for a numerical tool.

While the technique of numerical continuation of Hopf bifurcations to separate stability/instability regions of a steady state solution in a two-parameter space is applicable in general, the method of *D*-subdivision only applies to specific problems, as the example above, but allows to obtain *analytical* expressions for the boundary: when substituting $\lambda = j\omega$ in (55), some simple computations yield an implicit expression of the relation $\bar{K}(C)$:

$$\begin{cases} C = \frac{1+\cos\omega}{\omega\sin\omega}, \\ \bar{K} = \frac{2\omega^4(\sin\omega)^2}{(1+\cos\omega)^2}, \quad \omega \in (0,\ \pi). \end{cases}$$

### 4.3 Attraction domain of the equilibrium point

In [29] the attraction domain of the equilibrium point (54) was investigated under the additional assumption that $W \gg 1$. As seen in [17] this allows to further simplify the system (53) to:

$$\begin{cases} \dot{W}(t) = 1 - \frac{W(t)^2}{2}KQ(t-1) \\ \dot{Q}(t) = \begin{cases} W(t) - C, & Q > 0 \\ \max(W(t) - C, 0), & Q = 0 \end{cases}, \end{cases} \tag{56}$$

which corresponds to the following second-order delay-differential equation in $\tilde{Q}$:

$$\ddot{\tilde{Q}}(t)+K\left(\frac{1}{2}(\dot{\tilde{Q}}(t))^2+C\dot{\tilde{Q}}(t)\right)(\tilde{Q}(t-1)+Q^*)+\frac{KC^2}{2}\tilde{Q}(t-1)=0. \quad (57)$$

Let us apply the *model transformation* idea combined with an appropriate construction of a Lyapunov-Krasovskii functional candidate. Therefore, $\tilde{Q}(t-\tau)$ in the last term will be rewritten using the well-known Leibniz-Newton formula ($\tilde{Q}(t)-\tilde{Q}(t-\tau)=\int_{t-\tau}^{t}\dot{\tilde{Q}}(\theta)d\theta$).

Now, using the same construction idea as in the previous section devoted to Kelly's model, we shall introduce the following Lyapunov-Krasovskii functional:

$$V\left(\tilde{Q}_t,\dot{\tilde{Q}}_t\right)=V_1(\tilde{Q}(t),\dot{\tilde{Q}}(t))+V_2(\dot{\tilde{Q}}_t), \quad (58)$$

with:

$$V_1(\tilde{Q},\dot{\tilde{Q}})=\frac{KC^2}{2}\,\tilde{Q}^2(t)+\dot{\tilde{Q}}^2(t)$$
$$V_2(\dot{\tilde{Q}}_t)=\frac{KC^2}{2}\int_{-1}^{0}\int_{\theta}^{0}\dot{\tilde{Q}}^2(t+\xi)d\xi d\theta,$$

where $V_1$ is exactly the Lyapunov function needed to perform the analysis of the system free of delay, and $V_2$ is needed to "complete" the "squares" in the derivative of $V$. Some simple, but tedious computations lead to:

**Theorem 5.** *[29] The trivial solution of (57) is asymptotically stable if*

$$K<\frac{2}{C^3}\quad\left(i.e.\ K_p<\frac{2n^2}{(\tau c)^3}\right). \quad (59)$$

*A region of attraction is determined by*

$$U_l=\left\{(\tilde{Q}_t,\dot{\tilde{Q}}_t):\quad V(\tilde{Q}_t,\dot{\tilde{Q}}_t)<l\right\}, \quad (60)$$

*where* $(\tilde{Q}_t,\dot{\tilde{Q}}_t)\in\left\{(\tilde{Q}_t,\dot{\tilde{Q}}_t):\|(\tilde{Q}_t,\dot{\tilde{Q}}_t)\|<Q^*-C\right\}$. *Thus, a suitable choice for* $l$ *is* $l=\sqrt{\frac{\gamma_1}{\gamma_2}}(Q^*-C)$, *where* $\gamma_1=\min\left\{\frac{KC^2}{2},1\right\}$, *and* $\gamma_2=\max\left\{\frac{KC^2}{2},1+\frac{KC^2}{4}\right\}$.

### 4.4 Other attractors in the model

An important observation is that the solutions of (50) cannot grown unbounded, even when the steady state solution is (locally) exponentially unstable:

**Proposition 5.** *[30] All the solutions of the system (53) are bounded.*

As a consequence, there exist other *attractors* than the equilibrium point. We now provide some qualitative and quantitative information on such attractors.

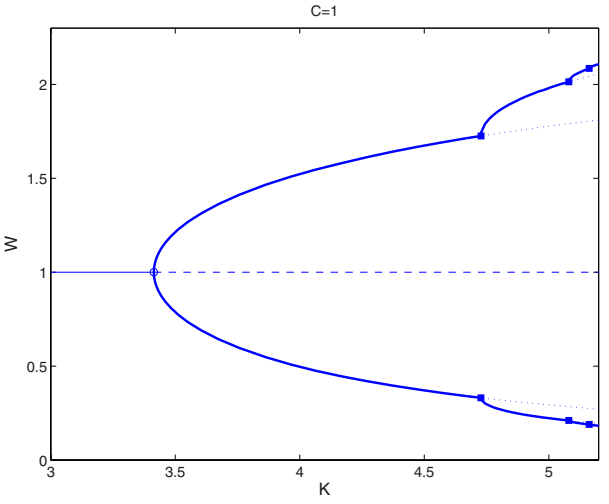



**Fig. 3.** Bifurcation diagram of the system (53) with $C = 1$ and free parameter $K$. Stable (thin line) and unstable $(--)$ steady state solution. Stable (thick lines) and unstable $(\cdots)$ branches of periodic solutions (maximum and minimum values of $W(t)$ are shown). Connections are formed by a Hopf Bifurcation ($\circ$) and by Period Doubling Bifurcations ($\square$).

**Chaotic attractors.** Figure 3 shows the bifurcation diagram of the system (53), when $K$ is the free parameter and $C = 1$ is fixed, as computed with the help of DDE-BIFTOOL [11]. For small $K > 0$ the unique steady state solution is locally asymptotically stable and stability is lost in a subcritical Hopf bifurcation as $K$ is increased. In the Hopf bifurcation a branch of stable *periodic solutions* emanates. The latter become unstable after a period doubling bifurcation, where a new branch of stable, *period doubled* periodic solutions emanates. A sequence of period doubling bifurcations ultimately leads to chaos. In Figure 4 the chaotic attractor is depicted for $K = 5.3$.

Notice that chaotic behavior is inherent to TCP/IP traffic. The work of Veras and Boda [42], where chaos was detected and analyzed in simulations with the *ns-2 simulator* [36], showed that TCP itself can cause or contribute to chaotic behavior as a *deterministic system*, while in previous works a large number of ON-OFF sources with *random* periods were rather seen as the (only) source of chaos in TCP. The analysis above shows that already the simple second-order deterministic model (53) exhibits chaos, which thus supports this proposition.

*Remark 8.* The chaotic behavior of the solutions of (53) is caused by the nonlinear delayed damping term in the first equation, which is proportional to $-W(t)W(t-1)$ (notice that $Q$ is strictly larger than zero along the attractor shown in Figure 4, hence the discontinuity in the right-hand size of (53) does not contribute). Therefore, it is expected that also the model (29) may exhibit chaotic behavior for particular choices of the function $p(.)$.

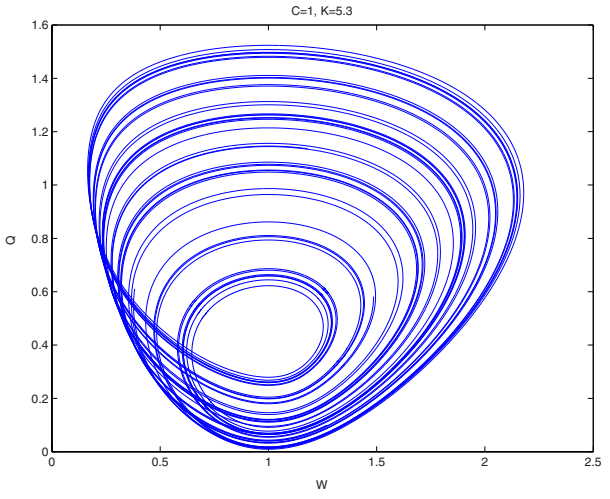



**Fig. 4.** As $K$ is increased a periodic doubling route to chaos occurs. The chaotic attractor for $K = 5.3$.

An analogous instability mechanism leading to chaos occurs in the *delayed logistic equation* [41].

**Superstable limit cycles.** In [30] it is shown that, due the discontinuity in the right-hand side of (53), so-called superstable limit cycles may occur, i.e. having the property that small perturbations around them disappear in a *finite time*.

## 5 Software for numerical analysis of delay differential equations

Due to the complexity of delay differential equations, which are of a infinite-dimensional nature, the analytical studies of mathematical models can be complemented with a *numerical analysis* to a chieve a better qualitative and quantitative understanding of the model dynamics. Concerning numerical methods we can distinguish two major directions: *time integration* and *stability and bifurcation analysis*.

**Time integration** (simulation) of differential equations can be used to study both the transient and long-term dynamics and is a basic tool for parameter estimation. Numerical techniques for time integration of DDEs are based on dense-output methods for *ordinary differential equations* (ODEs) and include certain facilities to cope with special features of DDEs, cf. [2]. In particular, numerical methods have been developed to track the propagation of discontinuities, to evaluate delay terms with a required accuracy, to deal with vanishing delays, to detect stiffness, etc, see, e.g. [1, 3, 4] for an extensive bibliography on numerical methods for time integration of DDEs. At present, a number of software packages for solving different types of DDEs is available, such as ARCHI [37], DKLAG6 [7], XP-

PAUT [13] and dde23 [40]. A more complete list (with links) can be found at the *delay differential equations homepage* [11].

**Stability and bifurcation analysis** of a dynamical system is used to understand how solutions and their stability change as the parameters in the system vary. In particular, it can be used for the computation, stability analysis and continuation of equilibria (steady state solutions), periodic and quasi-periodic oscillations. Much research has been done on the development of numerical methods for bifurcation analysis of ODEs. This research has led to well developed and publicly available software packages. A survey on methods for numerical bifurcation analysis of ODEs and software packages can be found in [9].

Numerical methods for bifurcation analysis of ODEs, as opposed to methods for time integration, cannot be extended to DDEs due to a significant difference in the relevant stability theory. Namely, the characteristic matrix, appearing in the stability theory for DDEs, has an infinite number of eigenvalues because of the infinite-dimensional nature of DDEs. Some numerical methods related to bifurcation analysis of DDEs were developed many years ago, see [12] for a review. However, until the year 2000, only two packages existed which deal with some form of stability analysis of DDEs: the code XPPAUT allows limited stability analysis of steady state solutions of DDEs using an approach based on the argument principle [25] and the program BIFDD [16] allows the normal form analysis of Hopf bifurcation points. The recently developed package DDE-BIFTOOL [11] is the first general purpose package for stability and bifurcation analysis of DDEs. This package can be used to compute and analyze the local stability of steady state and periodic solutions of a given system with constant and state-dependent delays as well as to study the dependence of these solutions on system parameters via continuation. The package also allows one to compute homoclinic and heteroclinic orbits of DDEs with constant delays. DDE-BIFTOOL has been used successfully in a number of biological and physical applications. As an illustration, also the numerical examples of the previous sections have been carried out with the package. Information on how to obtain it is given in [11].

## 6 Concluding remarks

In this paper, we addressed the stability analysis of three continuous fluid approximation models for describing flow and congestion dynamics. All the systems considered include *delays* in their representation, delays which can be constant or time-varying.

The methodology proposed here allows an *unitary* treatment of such problems, and is based on some *uncertainty* interpretations of the delay terms. All the conditions derived in the paper include *delay* information and are relatively easy to check for application purposes. Furthermore, the ideas developed in the chapter can be easily applied to other examples in the network control literature.

**Acknowledgements.**
The research of Silviu-Iulian Niculescu was supported in part by a PICS CNRS-NSF Grant (2002-2004): "Time-delay Systems: Analysis, Computer Aided Design, and Applications," ACI Jeunes Chercheurs (2002-2004): "Applications de l'Automatique en algorithmique des télécommunication," and an "Abondement AN-VAR GRADIENT-UTC (2002-2003): "Contrôle des congestions dans des réseaux." Wim Michiels is a postdoctoral fellow of the Fund for Scientific Research - Flanders (Belgium). His research is supported by the Belgian programme on Interuniversity Poles of Attraction, initiated by the Belgian Federal Science Policy Office (IAP P5).

## References

[1] Baker, C.T.H.: Retarded differential equations. *J. Comput. Appl. Math.*, 125:309–335, 2000.

[2] Baker, C.T.H., Paul, C.A.H. and Willé, D.R.: Issues in the numerical solution of evolutionary delay differential equations. Numerical Analysis Report 248, Manchester Centre for Computational Mathematics, Manchester, England, 1994.

[3] Baker, C.T.H., Paul, C.A.H. and Willé, D.R.: A bibliography on the numerical solution of delay differential equations. Numerical Analysis Report 269, Manchester Centre for Computational Mathematics, Manchester, England, 1995.

[4] Bellen, A. and Zennaro, M.: *Numerical methods for delay differential equations*. Oxford University Press, 2003.

[5] Bolot, J. -C. and Shankat, A. U.: Analysis of a fluid control approximation to flow control dynamics, *Proc. IEEE Infocom'92*, Florence, Italy (1992) 2398-2407.

[6] Chatté, F., Ducourthial, B., Nace, D. and Niculescu, S.-I.: Results on fluid modelling packet switched networks. in *Proc. 3rd IFAC Workshop on Time-Delay Systems*, Santa Fe, NM, USA (2001).

[7] Corwin, S.P., Sarafyan, D. and Thompson, S.: DKLAG6: A code based on continuously imbedded sixth order Runge-Kutta methods for the solution of state dependent functional differential equations. *Appl. Numer. Math.*, 24(2–3):319–330, 1997.

[8] Deb, S. and Srikant, R.: *Global stability of congestion controllers for the Internet*. (2002).

[9] Doedel, E.: Nonlinear numerics. *Internat. J. Bifur. Chaos*, 7:2127–2143, 1997.

[10] El'sgol'ts, L. E. and Norkin, S. B.: *Introduction to the theory and applications of differential equations with deviating arguments* (Mathematics in Science and Eng., **105**, Academic Press, New York, 1973).

[11] Engelborghs, K., Luzyanina, T. and Samaey, G.: DDE-BIFTOOL v. 2.00: a Matlab package for numerical bifurcation analysis of delay differential equations. *Report TW 330*, Dept. Computer Science, K.U.Leuven, Leuven, Belgium, 2001. Available from http://www.cs.kuleuven.ac.be/∼koen/delay/ddebiftool.shtml.

[12] Engelborghs, K., Luzyanina, T. and Roose, D.: Numerical bifurcation analysis of delay differential equations. *J. Comput. Appl. Math.*, 125(1–2):265–275, 2000.

[13] Ermentrout, B.: *XPPAUT3.91 - The differential equations tool*. University of Pittsburgh, Pittsburgh, 1998. (http://www.pitt.edu/∼phase/).

[14] Gu, K., Kharitonov, V.L. and Chen, J.: *Stability of time-delay systems* (Birkhauser: Boston, 2003).

[15] Hale, J. K. and Verduyn Lunel, S. M.: *Introduction to Functional Differential Equations* (Applied Math. Sciences, **99**, Springer-Verlag, New York, 1993).

[16] Hassard, B.D.: A code for Hopf bifurcation analysis of autonomous delay-differential systems. In F. V. Atkinson, W. F. Langford, and A. B. Mingarelli, editors, *Oscillations, Bifurcations and Chaos*, volume 8 of *Can. Math. Soc. Conference Proceedings*, pages 447–463. Amer. Math. Soc., Providence, RI, 1987.

[17] Hollot, C.V. and Chait, Y: Nonlinear stability analysis for a classof TCP/AQM networks. in *Proc. 40th IEEE Conf. Dec. Contr.*, Orlando, FL, USA (2001).

[18] Hollot, C. V., Misra, V., Towsley, D. and Gong, W.: Analysis and design of controllers for AQM routers supporting TCP flows. in *IEEE Trans. Automat. Contr.*, **47** (2002) 945-959.

[19] Izmailov, R.: Adaptive feedback control algorithms for large data transfer in high-speed networks. *IEEE Trans. Automat. Contr.* **40** (1995) 1469-1471.

[20] Izmailov, R.: Analysis and optimization of feedback control algorithms for data transfers in high-speed networks. in *SIAM J. Contr. Optimiz.* **34** (1996) 1767-1780.

[21] Jacobson, V.: Congestion avoidance and control. *ACM Communic. Rev.* **18** (1988) 314-329.

[22] Kelly, F. P.: Mathematical modelling of the internet, in *Mathematics unlimited - 2001 and beyond* (Eds. B. ENGQUIST, W. SCHMID), Springer-Verlag: Berlin (2001) 685-702.

[23] Kharitonov, V.L. and Zhabko, A.P.: Lyapunov-Krasovskii approach to robust stability of time delay systems, in *Proc. 2nd IFAC Simposium on System Structure and Control*, 23:185-196, 1999.

[24] Kharitonov, V.L. and Niculescu, S.-I.: On the stability of linear systems with uncertain delay, *IEEE Trans. Automat. Contr.*, January 2003.

[25] Luzyanina, T. and Roose, D.: Numerical stability analysis and computation of Hopf bifurcation points for delay differential equations. *J. Comput. Appl. Math.*, 72:379–392, 1996.

[27] Low, S., Paganini, F. and Doyle, J. C.: Internet congestion control. in *IEEE Contr. Syst. Mag.* **22** (2002) 28-43.

[27] Mazenc, F. and Niculescu, S. -I.: Remarks on the stability of a class of TCP-like congestion control models. in *Proc 42nd IEEE Conf. Dec. Contr.*, Maui, Hawaii (2003).

[28] Melchor-Aguilar, D. and Niculescu, S.-I.: Stability analysis of some class of nonlinear time delay systems with applications. in *Proc. 7th European Control Conf.*, Cambridge, UK, September 2003.

[29] Melchor-Aguilar, D. and Niculescu, S.-I.: Remarks on nonlinear stability analysis for a class of TCP/AQM networks, *Proc. 4th IFAC Wshop Time Delay Syst.*, Rocquencourt, France, September 2003.

[30] Michiels, W. and Niculescu, S.-I.: Stability and bifurcation analysis of some fluid flow model for TCP behavior, *Proc. MTNS 2004*, Leuven, Belgium, July 2004.

[31] Misra, V., Gong, W. and Towsley, D.: Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. in *Proc. ACM/SIGCOMM'00* (2000).

[32] Murray, R.M. (Eds.): *Control in an information rich world*. Panel on Future Directions in Control, Dynamics, and Systems, Pasadena, April 2002 (see also `http://www.cds.caltech.edu/murray/cdspanel`).

[33] Niculescu, S. -I.: *Delay effects on stability: A robust control approach* (Springer-Verlag: Heidelberg, Germany, LNCIS, vol. **269**, May 2001).

[34] Niculescu, S.-I.: On delay robustness analysis of a simple control algorithm in high-speed networks. in *Automatica* (May 2002).

[35] Niculescu, S.-I.: On some stability regions of a multiple delays linear systems with applications. in *Proc. 15th IFAC World Congress*, Barcelona, Spain (July 2002).

[36] The NS-2 simulator. Software and documentation available at `http://www-mash.cs.berkeley.edu/ns`

[37] Paul C. A. H.,: A user-guide to Archi - an explicit Runge-Kutta code for solving delay and neutral differential equations. Technical Report 283, The University of Manchester, Manchester Center for Computational Mathematics, 1997.

[38] Seydel, R.: *Practical bifurcation and stability analysis: from equilibrum to chaos*, volume 5 of *Interdisciplinary Applied Mathematics*. Springer New-York, 1994.

[39] Shakkottai, S. and Srikant, R.: How good are deterministic fluid models of internet congestion control. in Proc. IEEE INFOCOM 2002, New York, NY, USA.

[40] Shampine, L.F. and Thompson, S.: *Solving DDEs in Matlab*. Southern Methodist University and Radford University, Dallas, Radford, 2000. Available from http://www.runet.edu/∼thompson/webddes/.

[41] Tian, Y. and Gao, F.: Adaptive control of chaotic continuous-time systems with delay, *Physica*, **D117**, 1-12.

[42] Veras, A. and Boda, M.: The chaotic nature of TCP congestion control. in *Proc. IEEE Infocom'00* (2000).

# A Sliding Mode Approach to Traffic Engineering in Computer Networks

Bernardo A. Movsichoff[1], Constantino M. Lagoa[1], and Hao Che[2]

[1] Department of Electrical Engineering. The Pennsylvania State University
bernardo@gandalf.ee.psu.edu, lagoa@engr.psu.edu
[2] Department of Computer Science and Engineering. University of Texas at Arlington
hche@cse.uta.edu

## 1 Introduction

In this chapter, we address the problem of optimal Traffic Engineering (TE) in computer networks; i.e., determining an optimal traffic allocation in the presence of both multiple paths and multiple Classes of Service (CoSs). More precisely, we aim at designing data rate adaptation laws that maximize the utilization of the network resources (as measured by a given utility function) subject to link capacity constraints and call service requirements.

This problem, as it is formulated here, is an "easy" problem; i.e., the problem of optimal traffic allocation can be stated as maximizing a concave function subject to linear constraints. Hence, if global information is available, one could use well-known algorithms, such as gradient descent, to determine the optimal traffic allocation.

However, in most computer networks, it is not possible to obtain an accurate measure of the network status. Even if this is possible, there would be a delay in the propagation of the information. Furthermore, obtaining such information would require the propagation of large amounts of data in the network, leading to a significant increase in the network traffic and, hence, a decrease on the resources available for user utilization. Given this, the objective of the work presented in this chapter is to develop decentralized adaptation laws that converge to the optimal traffic allocation using the least possible amount of feedback from the network. The algorithms presented here allow for the distribution of the traffic among several paths. Furthermore, they allow for multiple CoSs.

To accomplish these objectives, we use results from nonlinear control theory to design control laws that endow the network with the "right" dynamic behavior. More precisely, by treating calls as *flows*, we develop data rate adaptation algorithms that result in an asymptotically stable system whose equilibrium points are optimal traffic allocations. In other words, we endow the network with the necessary dynamics to "solve" the optimal traffic engineering problem mentioned above.

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 111–136, 2004.

Central to the results presented here is Sliding Mode theory and its use in mathematical programming; e.g., see [14]. A first motivation for the use of the approach presented here is the fact that currently used Transport Control Protocol (TCP) congestion control algorithms can be thought of, in their essence, as sliding mode control laws. More precisely, in the TCP congestion avoidance phase, the data rate is increased linearly if no congestion is detected and decreased exponentially when congestion occurs. In other words, one has a congestion control law which is discontinuous along the surface that represents the "congestion barrier". Using tools other than sliding modes, it has been proven that the TCP congestion control law is optimal in the single path case ([2, 11]). However, this work does not extend to the multiple path case.

The realization that currently used adaptation laws can be studied using nonlinear control theory provides the motivation for the approach taken in this chapter. We show that one can use sliding mode theory to develop rate control laws for a class of service enabled network where several paths are available for each call. Moreover, we show that this can be done with minimal feedback from the network.

### 1.1 Literature Background

There is extensive literature on distributed traffic control. It includes both empirical algorithms (for example, see [4, 5]) and algorithms based on control theory; e.g., see [1, 2, 12]. These algorithms are designed for single path rate adaptation and the approach taken is not optimization-based.

Recently, several methodologies have been proposed which address the optimization-based rate adaptation problem using nonlinear optimization techniques. Their starting point is similar to the one in this chapter; i.e., maximization (minimization) of a utility (cost) function, subject to network resource constraints, where the constraints represent the interaction between different types of traffic; i.e., traffic with different ingress/egress nodes and/or different service requirements.

In the paper by Golestani, et al. [6], instead of using link resource constraints, a link congestion cost is incorporated into the overall utility function. The optimization problem was then solved using a gradient type algorithm. Iterative algorithms were proposed where individual sources periodically adjust their sending rates based on the congestion cost information periodically fed back from each of the links along the flow forwarding paths.

Kelly, et al. [7] use a Lagrangian multiplier technique to solve the optimization problem at hand. This results in a separation between the rate control executed at individual sources and the calculation of a "price" for each link in the network. The rate control at individual sources is based on the "prices" fed back from all the links in the data paths. The computation of the "price" is in turn based on the sending rate information fed back from all the sources using this link. Since only a relaxation of the original optimization problem is solved, it is not proven that the algorithm converges to the optimal traffic allocation. The same formulation is used in the work by La, et al. [8] where the algorithm provided is proven to converge to the optimum rate allocation in the single congested link case (low traffic networks).

Low [10] uses a technique which converts a constrained problem into a non-constrained dual problem. This reformulation results in a similar distributed control scheme to the one presented in [7]. Iterative algorithms were also proposed and their convergence is proven for the single-path case. A similar approach is taken by Sarkar, et al. [13] to address the optimization-based multi-rate multicast. A distributed control scheme is proposed and it degenerates to the one in [10] when there is only one path in the multicast tree.

## 2 Notation and Assumptions

In our model, the traffic flows are assumed to be described by a fluid flow model and the only resource considered is link bandwidth. In the remainder of this chapter we will use the terms call, session and flow interchangeably.

Consider a computer network where calls with different service requirements are present. We divide these calls into *types*. Types are aggregate of calls that, from the point a view of the traffic engineering algorithm, can be treated as a *unit*; i.e., these are calls with the same ingress and egress nodes and the service requirements are to be applied to the aggregate, not the individual calls. One can have call types with just one call. Moreover, one can have different call types with the same ingress/egress node pair. We assume that each call type can have several paths available. The objective is to find the allocation of the resources that leads to the maximization of a given utility function subject to the network resource constraints and CoS requirements.

More precisely, consider a computer network whose set of links is denoted by $\mathcal{L}$, with cardinality $\text{card}(\mathcal{L})$. Let $c_l$ be the capacity of link $l \in \mathcal{L}$, $n$ be the number of types of calls, $n_i$ be the number of paths available for calls of type $i$ and $\mathcal{L}_{i,j}$ be the set of links used by calls of type $i$ taking path $j$. Given calls of type $i$, let $x_{i,j}$ be the total data rate of calls of type $i$ using path $j$. Also, let

$$\mathbf{x}_i \doteq [x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}]^T \in \mathbf{R}^{n_i}$$

denote the vector containing the data rates allocated to the different paths taken by calls of type $i$ and

$$\mathbf{x} \doteq \left[\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_n^T\right]^T \in \mathbf{R}^{n_1+n_2+\cdots+n_n}$$

the vector containing all the data rates allocated to different call types and respective paths. Now, a link $l \in \mathcal{L}$ is said to be congested if the aggregated data rate of the calls using the link reaches its capacity $c_l$. Given this, define $b_{i,j}(\mathbf{x})$ as the number of congested links along the $j$-th path of calls of type $i$. Finally, let $u(x)$ be the unit step function; i.e., $u(x) = 1$ if $x \geq 0$ and $u(x) = 0$ otherwise.

### 2.1 Classes of Service

Each call in the network is associated to some service requirement or Class of Service (CoS). We assume that five categories of CoSs share the network: Calls of type $i$,

$i = 1, 2, \ldots, s_1$, are assumed, without loss of generality, to be of Assured Service (AS) CoS category. By AS we mean that a target rate for the call should be guaranteed in average sense. More precisely, assuming that the target rate for $\mathbf{x}_i$ is $\Lambda_i$, the objective is to allocate the data rates in such a way that

$$\sum_{j=1}^{n_i} x_{i,j} = \Lambda_i,$$

for all $i = 1, 2, \ldots, s_1$. Calls of type $i$, $i = s_1 + 1, s_1 + 2, \ldots, s_2$, are assumed to be of the Minimum Rate Guaranteed Service category (MRGS). More precisely, this type of calls should satisfy the following requirement

$$\sum_{j=1}^{n_i} x_{i,j} \geq \theta_i,$$

for some $\theta_i > 0$ and all $i = s_1 + 1, s_1 + 2, \ldots, s_2$.

Calls of type $i$, $i = s_2 + 1, s_2 + 2, \ldots, s_3$, belong to the Upper Bounded Rate Service (UBRS) category; i.e., there is an upper bound on the maximum bandwidth that can be used. More precisely, traffic should be allocated in such a way that calls of type $i$ satisfy

$$\sum_{j=1}^{n_i} x_{i,j} \leq \Theta_i,$$

for some $\Theta_i > 0$ and all $i = s_2 + 1, s_2 + 2, \ldots, s_3$.

Next, calls of type $i$, $i = s_3 + 1, s_3 + 2, \ldots, s_4$, are defined to be of the CoS category consisting traffic with both a Minimum Service Guarantee and an Upper Bounded Rate (MRGUBS); i.e., traffic should be allocated so that

$$\theta_i \leq \sum_{j=1}^{n_i} x_{i,j} \leq \Theta_i,$$

for some $\theta_i > 0$, $\Theta_i > 0$ and all $i = s_3 + 1, s_3 + 2, \ldots, s_4$.

Finally, calls of type $i$, $i = s_4 + 1, s_4 + 2, \ldots, n$ are assumed to be of the Best Effort (BE) class. Calls of this class do not have any service requirements.

## 3 The Network Optimization Problem

The results in this chapter aim at solving the problem of maximizing utility functions of the form

$$U(\mathbf{x}) \doteq \sum_{i=1}^{n} f_i(\mathbf{x}_i) \doteq \sum_{i=1}^{n} f_i\left(x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}\right),$$

subject to the network constraints and CoS requirements, where $f_i(\cdot)$, $i = 1, 2, \ldots, n$, are differentiable increasing concave functions.

Given this, the problem of optimal resource allocation can be formulated as the following optimization problem:

$$\max_{\mathbf{x}} U(\mathbf{x})$$

subject to the network capacity constraints

$$\sum_{i,j:\, l\in\mathcal{L}_{i,j}} x_{i,j} - c_l \le 0; \qquad l \in \mathcal{L},$$

the AS requirements

$$\sum_{j=1}^{n_i} x_{i,j} = \Lambda_i; \qquad i = 1,2,\ldots,s_1,$$

the MRGS requirements

$$\sum_{j=1}^{n_i} x_{i,j} \ge \theta_i; \qquad i = s_1+1, s_1+2, \ldots, s_2,$$

the UBRS requirements

$$\sum_{j=1}^{n_i} x_{i,j} \le \Theta_i; \qquad i = s_2+1, s_2+2, \ldots, s_3,$$

the MRGUBS requirements

$$\theta_i \le \sum_{j=1}^{n_i} x_{i,j} \le \Theta_i; \qquad i = s_3+1, s_3+2, \ldots, s_4$$

and all data rates are non-negative $x_{i,j} \ge 0$, for $i = 1,2,\ldots,n$ and $j = 1,2,\ldots,n_i$.

Obviously, the optimization problem above is a convex problem; i.e., maximizing a concave function over a convex set. Algorithms such as gradient descent could be used to solve it provided global information is available. However, this is not generally the case. The objective of this work is then to provide decentralized adaptation laws that converge to the solution of the problem stated above.

## 4 Sliding Mode Control Laws

In this section the proposed solution to the optimization problem above is presented. This solution consists of a family of Sliding Mode control laws that achieve optimal utilization of network resources.

### 4.1 A Family of Optimal Control Laws

Define the following family of control laws: For $i = 1,2,\ldots,s_1$ (AS calls), let

$$\dot{x}_{i,j} = z_{i,j}\big(t, \mathbf{x}_i(t), b_{i,j}(t)\big)\left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha b_{i,j}(\mathbf{x}) - \beta_i r_i(\mathbf{x}_i) + \xi_{i,j} u(-x_{i,j})\right],$$

where

$$r_i(\mathbf{x}_i) = \begin{cases} 1 \text{ if } \sum_{j=1}^{n_i} x_{i,j} > \Lambda_i \\ -1 \text{ if } \sum_{j=1}^{n_i} x_{i,j} < \Lambda_i \end{cases}.$$

For $i = s_1+1, s_1+2, \ldots, s_2$ (MRGS calls), let

$$\dot{x}_{i,j} = z_{i,j}\big(t, \mathbf{x}_i(t), b_{i,j}(t)\big)\left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha b_{i,j}(\mathbf{x}) + \beta_i^m r_i^m(\mathbf{x}_i) + \xi_{i,j}u(-x_{i,j})\right],$$

where

$$r_i^m(\mathbf{x}_i) = \begin{cases} 0 \text{ if } \sum_{j=1}^{n_i} x_{i,j} > \theta_i \\ 1 \text{ if } \sum_{j=1}^{n_i} x_{i,j} < \theta_i \end{cases}.$$

For $i = s_2+1, s_2+2, \ldots, s_3$ (UBRS calls), let

$$\dot{x}_{i,j} = z_{i,j}\big(t, \mathbf{x}_i(t), b_{i,j}(t)\big)\left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha b_{i,j}(\mathbf{x}) - \beta_i^M r_i^M(\mathbf{x}_i) + \xi_{i,j}u(-x_{i,j})\right],$$

where

$$r_i^M(\mathbf{x}_i) = \begin{cases} 1 \text{ if } \sum_{j=1}^{n_i} x_{i,j} > \Theta_i \\ 0 \text{ if } \sum_{j=1}^{n_i} x_{i,j} < \Theta_i \end{cases}.$$

For $i = s_3+1, s_3+2, \ldots, s_4$ (MRGUBS calls), let

$$\dot{x}_{i,j} = z_{i,j}\big(t, \mathbf{x}_i(t), b_{i,j}(t)\big)\left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha b_{i,j}(\mathbf{x}) + \beta_i^m r_i^m(\mathbf{x}_i) - \beta_i^M r_i^M(\mathbf{x}_i) + \xi_{i,j}u(-x_{i,j})\right],$$

where

$$r_i^m(\mathbf{x}_i) = \begin{cases} 0 \text{ if } \sum_{j=1}^{n_i} x_{i,j} > \theta_i \\ 1 \text{ if } \sum_{j=1}^{n_i} x_{i,j} < \theta_i \end{cases} \quad \text{and} \quad r_i^M(\mathbf{x}_i) = \begin{cases} 1 \text{ if } \sum_{j=1}^{n_i} x_{i,j} > \Theta_i \\ 0 \text{ if } \sum_{j=1}^{n_i} x_{i,j} < \Theta_i \end{cases}.$$

For $i = s_4+1, s_4+2, \ldots, n$ (BE calls), let

$$\dot{x}_{i,j} = z_{i,j}\big(t, \mathbf{x}_i(t), b_{i,j}(t)\big)\left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha b_{i,j}(\mathbf{x}) + \xi_{i,j}u(-x_{i,j})\right].$$

In the equations above $z_{i,j}(\cdot)$, $\alpha$, $\beta_i$, $\beta_i^m$, $\beta_i^M$ and $\xi_{i,j}$ are design parameters and $b_{i,j}$ is the number of congested links encountered by calls of type $i$ taking path $j$.

We now formally establish the optimality of these control laws; i.e., the control laws presented above converge to the optimal traffic allocation for the problem at hand. The proof of this result is presented in Section 10.1.

**Theorem 1 (Optimal Control Laws).** *Assume that all data rates are bounded; i.e., there exists an* $\rho \in \mathbf{R}$ *such that the data rate vector* $\mathbf{x}$ *always belongs to the set*

$$\mathcal{X} \doteq \left\{ \mathbf{x} \in \mathbf{R}^{n_1+n_2+\cdots+n_n} : x_{i,j} \leq \rho;\ l \in \mathcal{L}_{i,j};\ j = 1,2,\ldots,n_i;\ i = 1,2,\ldots,n \right\}.$$

*Also, assume that at the optimal traffic allocation, each congested link has at least one non-binding class of service or a BE call with a nonzero data rate. Furthermore, assume that the components of the gradient of the utility function; i.e.,* $\nabla U(\mathbf{x})$*, are bounded in* $\mathcal{X}$*.*

*Let* $\zeta > 0$ *be a given (arbitrarily small) constant and let* $z_{i,j}\big(t, \mathbf{x}_i(t), b_{i,j}(t)\big)$ *be scalar functions continuous in* $t$ *for all choices of* $\mathbf{x}_i(t) \in \mathcal{X}$ *and* $b_{i,j}(t) \in \{0,1\}$*, satisfying* $z_{i,j}\big(t, \mathbf{x}_i(t), b_{i,j}(t)\big) > \zeta$*, for all* $t > 0$*. Moreover, let* $\alpha > \alpha_{min}$*,* $\beta_i > \beta_{min}$*,* $\beta_i^m > \beta_{min}$*,* $\beta_i^M > \beta_{min}$ *and* $\xi_{i,j} > \xi_{min}$ *be positive constants, with*

$$\alpha_{min} \doteq \max_{i,j,\mathbf{x}\in\mathcal{X}} \frac{\partial U(\mathbf{x})}{\partial x_{i,j}}, \qquad \beta_{min} \doteq \alpha_{min} \max_{i,j} B_{i,j}, \qquad \xi_{i,j,min} \doteq \alpha B_{i,j} + \beta_i,$$

*where* $\beta \in \{\beta_i, \beta_i^m, \beta_i^M\}$ *and* $B_{i,j}$ *is the number of links used by calls of type* $i$ *taking path* $j$*.*

*Then, the control laws presented in Section 4.1 converge to the maximum of the utility function*

$$U(\mathbf{x}) \doteq \sum_{i=1}^{n} f_i(\mathbf{x}_i) \doteq \sum_{i=1}^{n} f_i\left(x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}\right), \tag{1}$$

*subject to the network link capacity constraints and AS, MRGS, UBRS, MRGUBS and non-negativity requirements.*

### 4.2 A Family of Quasi-Optimal Control Laws

It should be noted that the proposed algorithm might converge to a non-optimal equilibrium if the true number of congested links is not known. Nevertheless, experiments suggest that if a large value of $\alpha$ is used then the algorithm becomes more robust with respect to this loss of information. However, there is a trade-off in doing so, since larger values of $\alpha$ will result in larger oscillations.

To be more precise, let us consider the case of a single path per call type and only BE service being provided. Furthermore, assume that all data rates are bounded away from zero and that the utility function that one is trying to maximize is

$$U(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i).$$

In the case of a single path, the control law proposed in this work becomes

$$\dot{x}_i = z_i(t,\mathbf{x})\left[d_i(x_i) - \alpha b_i(\mathbf{x})\right],$$

where $d_i(x_i) = df_i/dx_i$. Assume that the only information available is whether the path is congested or not in which case $b_i$ is either zero or one. Then, according to [11], the control law converges to the maximum of the utility function

$$\tilde{U}(\mathbf{x}) = \sum_{i=1}^{n} \int_0^{x_i} \log\left(\frac{\alpha}{\alpha - d_i(u_i)}\right) du_i.$$

This result shows that we do not converge to the desired point. However, if $\alpha$ is large, then the two utility functions are approximately the same.

### Quasi-Optimal Laws

Given the results above, a new family of control laws is proposed that only uses less feedback from the network. The only information required is whether a path is congested or not, as opposed to the number of congested links in it. These laws are only quasi-optimal. However, they require much less information and therefore, their implementation is much simpler.

These control laws are similar to the optimal ones and are obtained by replacing the quantity $b_{i,j}$ in the expressions of Section 4.1 by $bin_{i,j}$ defined as

$$bin_{i,j} \doteq \begin{cases} 0 & \text{if no links are congested in the path} \\ 1 & \text{if at least one link in the path is congested} \end{cases}.$$

We now show that this relaxed version of the control laws converges to a small neighborhood of the optimal solution to the problem at hand. This is formally stated through the following theorem whose proof is presented in Section 10.2.

**Theorem 2 (Quasi-Optimal Laws).** *Assume that the hypothesis in Theorem 1 are satisfied. Furthermore, assume that a similar set of conditions are also satisfied when replacing $b_{i,j}$ by $bin_{i,j}$.*

*Then, given any $\varepsilon > 0$ there exists $\alpha^* > \alpha_{min}$ such that for all $\alpha > \alpha^*$ satisfying the conditions above, the control laws in Section 4.2 converge to an $\varepsilon$-neighborhood of $\mathbf{x}^*$, where $\mathbf{x}^*$ achieves the maximum of the utility function* (1) *subject to the network link capacity constraints, assured service requirements and non-negativity of all the data rates.*

## 5 Reducing Oscillation

The families of control laws presented in Section 4 can lead to excessive oscillation when data optimal data rates are close to zero. The structure of the non-negativity

constraints can be exploited to address this issue. More precisely, a simple truncation procedure can be used to reduce oscillation and still preserve optimality.

The following Lemma introduces this improved control laws and establishes their optimality properties. For a proof see the Appendix.

**Lemma 1.** *Define the following family of control laws: For* $i = 1,2,\ldots,s$*; i.e., AS calls, let*

$$p_{i,j} = z_{i,j}\big(t,\mathbf{x}_i(t),cg_{i,j}(t)\big)\left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha c g_{i,j}(t) - \beta_i(t) r_i\right],$$

*where* $r_i(t)$ *is defined as before. For* $i = s+1,s+2,\ldots,n$*; i.e., BE calls, let*

$$p_{i,j} = z_{i,j}\big(t,\mathbf{x}_i(t),cg_{i,j}(t)\big)\left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha c g_{i,j}(t)\right].$$

*The family of control laws is then given by*

$$\dot{x}_{i,j} = \begin{cases} p_{i,j} & \text{if } x_{i,j} > 0 \\ \max\{0, p_{i,j}\} & \text{if } x_{i,j} = 0 \end{cases}.$$

*Then under the same assumptions of Theorems 1 and 2, the control laws above converge to the optimal (respectively the given* $\varepsilon$*-neighborhood of the optimal) traffic allocation* $\mathbf{x}^*$.

*Proof.* See Section 1.

## 6 Discrete Time Control Laws and Oscillation Reduction

The implementation of the above control laws in this chapter has to be performed in discrete time in a "real" network. Therefore, we now describe a discrete-time approximation of the continuous-time control laws.

As is the case with any discrete time controller design, there are different ways of obtaining difference state equations from the differential state equations. The approach used here is the forward rule approximation. This method avoids computational delays inherent to other discretization techniques such as trapezoidal or backward rule approximation.

Let $\dot{x}_{i,j} = g_{i,j}(\mathbf{x},t)$ denote the continuous time control law described in Section 4. The discrete approximation that we propose is

$$x^d_{i,j}\big[(k+1)t_d\big] = x^d[kt_d] + t_d g_{i,j}\big(\mathbf{x}(kt_d),kt_d\big);\ k = 0,1,\ldots$$

where $t_d$ is the sampling period. Obviously, since this is not a continuous time law, Sliding Mode theory does not apply. However, one has the following result.

**Proposition 1.** *Let* $\mathbf{x}(t)$ *be the trajectory obtained using the control laws in Section 4 and let* $\mathbf{x}^d(t)$ *be the corresponding discrete time trajectory obtained using the discretization algorithm above. Define the set* $\mathcal{X}$ *as in Theorem 1.*

*Given any time interval* $[t_0,t_1]$ *and constant* $\varepsilon > 0$*, there exists a* $\delta > 0$ *such that if* $t_d z_{i,j}(t,\mathbf{x}) < \delta$*, for all* $t > 0$ *and* $\mathbf{x} \in \mathcal{X}$*, then* $\left\|\mathbf{x}(t) - \mathbf{x}^d(t)\right\| < \varepsilon$ *for all* $t \in [t_0,t_1]$.

*Proof.* Direct application of result 2, page 95 of [3]. □

When implementing the control laws developed in this paper, one is faced with several issues: First, one has to implement a discrete time version of the control algorithms, such as the one described above. Second, usually one uses finite word length which leads to a quantization of the possible data rate values. Finally, there is delay in the propagation of the congestion information. All of these lead to a well known phenomenon: Oscillation. Even in this case, the discretization of the control laws presented in this paper is approximately optimal. We now state the precise result.

**Proposition 2.** *Let* $\mathbf{x}(t)$ *be the trajectory obtained using the control laws in Section 4 and let* $\mathbf{x}^r(t)$ *be the corresponding discrete time trajectory obtained using the discretization algorithm above and in the presence of delays in the propagation of the congestion information. Let* $t_r$ *be an upper bound on the largest delay. Again, define* $\mathcal{X}$ *as in Theorem 1.*

*Given any time interval* $[t_0,t_1]$ *and constant* $\varepsilon > 0$*, there exists a* $\delta > 0$ *such that if* $\max\{t_d,t_r\} z_{i,j}(t,\mathbf{x}) < \delta$*, for all* $t > 0$ *and* $\mathbf{x} \in \mathcal{X}$*, then* $\|\mathbf{x}(t) - \mathbf{x}^r(t)\| < \varepsilon$ *for all* $t \in [t_0,t_1]$.

*Proof.* Direct application of result 2, page 95 of [3]. □

### 6.1 Adaptive Oscillation Reduction

Although we do have approximate optimality, the performance might degrade if the delays are too large and/or the value of $z_{i,j}(\cdot)$ is too low. Therefore, we now provide a method for choosing the functions $z_{i,j}(\cdot)$ which reduces the amplitude of oscillation of the data rates. The main idea is the following: Oscillation occurs when a link is congested. Hence, when there is no congestion, one would like the rates to increase at a reasonably fast rate. Once congestion is about to occur, one would like to decrease the rate of change in order to reduce the amplitude of the oscillations.

Hence, we propose the following scheme for choosing the value of $z_{i,j}(\cdot)$: Let $T > 0$ be given. Initialize $z_{i,j}(0,\mathbf{x}) = k$, where $k > 0$ is a constant. If congestion is detected at time $t_0$ for calls of type $i$ taking path $j$ let

$$z_{i,j}(t,\mathbf{x}) = \omega(t - t_0)\,; \quad t_0 \le t < t_0 + T,$$

where $\omega : [0,T] \to [\zeta,k]$ is a decreasing function and $\zeta$ is some positive constant. Now, at $t_0 + T$ repeat the same reasoning. If there is no congestion, let $z_{i,j}(t,\mathbf{x}) = k$ until congestion is detected. Once congestion is detected let $z_{i,j}(t,\mathbf{x})$ be equal to a shifted version of $\omega(t)$. Examples of such functions are provided in the next section

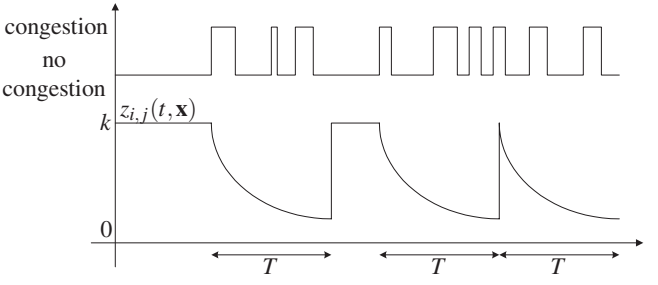



**Fig. 1.** Example of a scaling function $z_{i,j}(\cdot)$

and the desired behavior is depicted in Fig. 1. Notice that the value $z_{i,j}(\cdot)$ is continuously reset every $T$ seconds. This ensures that the network is able to adapt to changes in the traffic demand. Also, although we reduce the rate of change of $x_{i,j}$ when congestion is detected, the algorithm still converges to the optimum. This is a consequence of the fact that the functions $z_{i,j}(\cdot)$ mentioned above satisfy the conditions of Theorem 1.

Now, this choice for the functions $z_{i,j}(\cdot)$ leaves one with three free parameters that have to be selected. The period $T$ of $z_{i,j}(\cdot)$ should be chosen taking into account the behavior of the demand on the network. More precisely, it should be equal to the interval of time in between substantial changes in the demand. Resetting $z_{i,j}(\cdot)$ in this form will enable the network to more rapidly adapt to the demand. Now, the parameters $k$ and $\zeta$ of $z_{i,j}(\cdot)$ should be inversely proportional to the round trip time of the path $j$ of calls of type $i$, which can be easily estimated. The reasoning behind this particular choice has to do with the fact that the largest delay for the propagation of the congestion information is the round trip time. Now, the exact value of these parameters depends on how one would like the network to behave. If they are high, then the network will quickly react to changes in the demand, but the oscillations will be large and the network might be operating far away from the optimal behavior. If the values $k$ and $\zeta$ are low, then one will have small oscillations and a better steady state operating point. However, there will be a much larger transient behavior.

## 7 Robust Load Sharing

In practice, in order to distribute traffic among the multiple paths available for a given call, the utility function should reflect the fact be it is not important how traffic is distributed among these paths, but only how much traffic of a given type the network can handle. That is, the utility function should be of the form

$$U(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_{i,1} + x_{i,2} + \cdots + x_{i,n_i}).$$

For this kind of utility functions, the control laws that are obtained in Section 4 provide a crucial added benefit: Robustness with respect to failures. In the case of

a link failure, the algorithm tries to reroute the traffic to the other available paths and provides an optimal traffic allocation for the new network configuration. In other words, in the case of a link failure, the algorithm provides a systematic (optimal) way of redistributing traffic.

## 8 Simulation Examples

In this Section, some simulation examples are presented that exemplify the behavior of the control laws presented in this chapter. A brief discussion of the observed behavior and the tradeoffs incurred in the design of the parameters is also presented. These simulations address the case of only AS and BE sharing the network, since the other CoSs do not include any additional complexity from a mathematical standpoint.

The model of the network used here is taken from [8] and is shown in Fig. 2, where also all the links' bandwidths and delays, as well as source and destination nodes are shown. Here, however, several paths and CoSs are available. Overall, $n = 8$ types of calls are considered and the paths available are shown in Table 1. Recall that $n_i$ indicates the number of paths available to calls of type $i$. Furthermore, calls of types 3 and 5 are assumed to be of AS type with target rates $\Lambda_3 = \Lambda_5 = 1\,\text{Mbps}$.

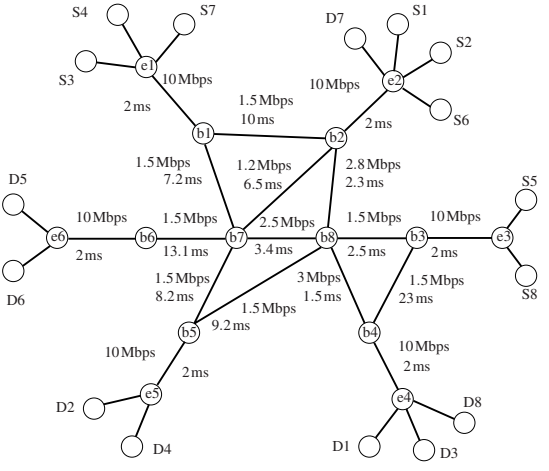



**Fig. 2.** Topology of the network

The utility function proposed belongs to the family presented in Section 4; i.e.,

$$U(\mathbf{x}) \doteq \sum_{i=1}^{8} \log\left(0.5 + \sum_{j=1}^{n_i} x_{i,j}\right); \qquad \begin{matrix} i = 1,\ldots,8 \\ j = 1,\ldots,n_i \end{matrix}.$$

Given the utility function and service requirement above, Theorems 1 and 2 are applied to obtain the following families of control laws: For $i = 1,3$ and $j = 1,\ldots,n_i$;

**Table 1.** Paths available for each type of calls

| | |
|---|---|
| type 1 $x_{1,1} : e_2b_2b_8b_4e_4$<br>$n_1 = 4$ $x_{1,2} : e_2b_2b_8b_3b_4e_4$<br>$x_{1,3} : e_2b_2b_7b_8b_3b_4e_4$<br>$x_{1,4} : e_2b_2b_7b_8b_4e_4$ | type 5 $x_{5,1} : e_3b_3b_8b_7b_6e_6$<br>$n_5 = 2$ $x_{5,2} : e_3b_3b_4b_8b_5b_7b_6e_6$ |
| type 2 $x_{2,1} : e_2b_2b_8b_5e_5$<br>$n_2 = 3$ $x_{2,2} : e_2b_2b_7b_5e_5$<br>$x_{2,3} : e_2b_2b_1b_7b_5e_5$ | type 6 $x_{6,1} : e_2b_2b_1b_7b_6e_6$<br>$n_6 = 3$ $x_{6,2} : e_2b_2b_8b_7b_6e_6$<br>$x_{6,3} : e_2b_2b_7b_6e_6$ |
| type 3 $x_{3,1} : e_1b_1b_7b_8b_4e_4$<br>$n_3 = 2$ $x_{3,2} : e_1b_1b_2b_8b_4e_4$ | type 7 $x_{7,1} : e_1b_1b_2e_2$<br>$n_7 = 3$ $x_{7,2} : e_1b_1b_7b_2e_2$<br>$x_{7,3} : e_1b_1b_7b_8b_2e_2$ |
| type 4 $x_{4,1} : e_1b_1b_7b_5e_5$<br>$n_4 = 4$ $x_{4,2} : e_1b_1b_7b_8b_5e_5$<br>$x_{4,3} : e_1b_1b_2b_7b_5e_5$<br>$x_{4,4} : e_1b_1b_2b_8b_5e_5$ | type 8 $x_{8,1} : e_3b_3b_4e_4$<br>$n_8 = 2$ $x_{8,2} : e_3b_3b_8b_4e_4$ |

i.e., AS calls

$$\dot{x}_{i,j} = z_{i,j}(t,\mathbf{x}) \left[ \frac{1}{0.5 + \sum_{j=1}^{n_i} x_{i,j}} - \alpha c g_{i,j}(\mathbf{x}) - \beta_i r_i(\mathbf{x}_i) + \xi_{i,j} u(-x_{i,j}) \right]$$

and for $i = 1,2,4,6,7,8$ and $j = 1,\ldots,n_i$; i.e., BE calls

$$\dot{x}_{i,j} = z_{i,j}(t,\mathbf{x}) \left[ \frac{1}{0.5 + \sum_{j=1}^{n_i} x_{i,j}} - \alpha c g_{i,j}(\mathbf{x}) - \xi_{i,j} u(-x_{i,j}) \right],$$

where $cg_{i,j}$ denotes the congestion information used; i.e., $b_{i,j}$ for the optimal control laws and $bin_{i,j}$ for the quasi-optimal ones. The design parameters $\alpha$, $\beta_i$ and $\xi_{i,j}$ and the functions $z_{i,j}(t,\mathbf{x})$ are chosen to satisfy the convergence conditions set forth by Theorem 1. It is clear from the last equation, that the term 0.5 in the denominator imposes an upper bound on the derivatives, that otherwise will tend to infinity as all the data rates go to zero.

### 8.1 Ideal Conditions

As a first step and in order to have a starting point for comparison, the optimal control laws are simulated for an almost ideal case where one has very small delays and sampling intervals. The discrete time version of the control laws are obtained using a sampling interval of $t_d = 0.1$ ms, while all delays were chosen equal to 0.1 ms. The remaining parameters were taken as $\alpha = 4$, $\beta_3 = \beta_5 = 22$ and $\xi_{i,j} = B_{i,j}\alpha + \beta_i + 0.0001$, while the oscillation reducing function was kept constant at 0.375.

Fig. 3 shows that, under these ideal conditions, the utility function converges to the optimal value. The plots of the data rates show a clear sliding motion and demonstrate that the AS requirements are being met.

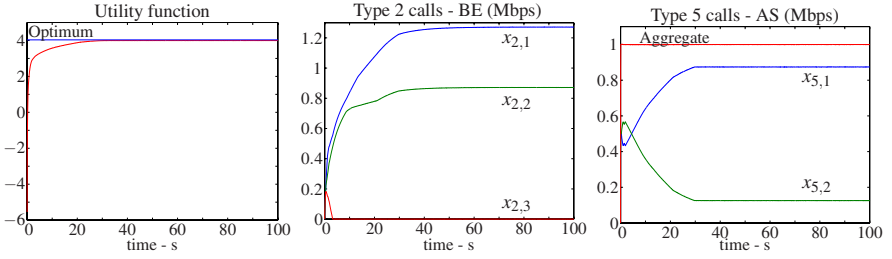
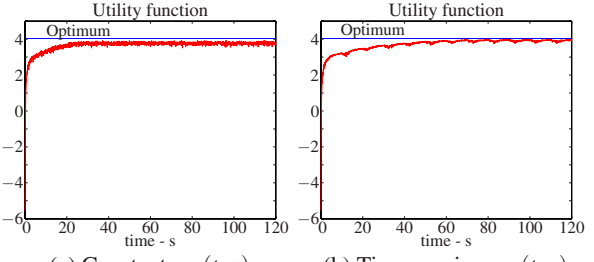



**Fig. 3.** Ideal Conditions: small delays and $t_d$

## 8.2 Non-ideal Conditions

The existence of delays in the propagation of information as well as the discretization of the continuous time laws introduce undesired oscillations in the trajectories of the data rates. This phenomenon can be explained using Sliding Modes theory. Indeed, Sliding Modes can be derived as a limiting procedure of the motion in a boundary layer around the discontinuity surfaces. These boundary layers can occur due to non-ideal switching caused by delays, hysteresis, etc. It is then only natural to observe an oscillatory behavior in this case. Moreover, the larger the magnitude of the derivatives, the larger the boundary layer will be. Hence, the effects of $\alpha$ and the parameters of the oscillation reduction function $k$, $\zeta$ and $T$ can all be explained in a similar way and the simulations all show the same type of behavior. For more thorough simulation examples than the ones presented in this chapter, the reader is referred to [9].

Fig. 4 shows the effect of larger delays and sampling interval on the utility function for both the cases of constant and time-varying $z_{i,j}$. It can be seen that under




(a) Constant $z_{i,j}(t,\mathbf{x})$ (b) Time varying $z_{i,j}(t,\mathbf{x})$

**Fig. 4.** Oscillation reduction on the utility functions for larger delays and $t_d$

these conditions the control laws converge to a value close to the optimal one and not to the optimal itself. Moreover, with a constant $z_{i,j}$ (Fig. 4(a)) there is excessive oscillation as opposed to the case of varying $z_{i,j}$ (Fig. 4(b)) where oscillations are milder.

However, the use of this oscillation reduction scheme introduces further undesired effects. Namely, the speed of convergence may be reduced and, as a consequence, a slower reaction to changing conditions can result. On the other hand, by having milder oscillations the network spends less time working above link capacity with the effect that less packets are dropped and fewer retransmissions are needed. Finally, Fig. 5 shows the trajectories obtained for the case of varying $z_{i,j}$, where the AS requirements are met.

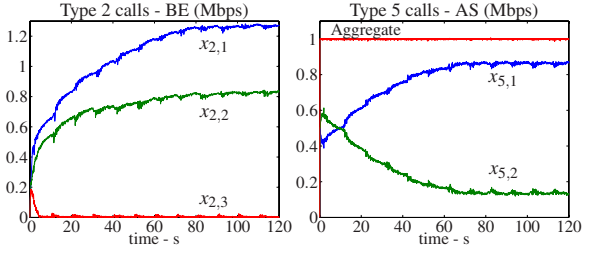



**Fig. 5.** Data rates for time-varying $z_{i,j}(t,\mathbf{x})$ with larger delays and $t_d$

### 8.3 Robustness

One of the most important features of the control laws presented in this chapter is the following: They endow the network with robustness against link and/or node failures. We claim that since the data rates are being updated adaptively, as soon as link failure is detected, the control law will reroute all the traffic away from the paths using this link. The control laws running at the sending nodes can handle this situation by treating link failure as congestion; i.e., the adaptation laws are oblivious to the failure. Instead, they simply reduce the sending rates because congestion is detected along the paths that include the broken link.

In order to test this feature, the link connecting nodes $b_7$-$b_8$ was opened at time $t = 120\,\mathrm{s}$. This link is particularly problematic because both type 3 and 5 (the AS calls) lose one of the two paths available to them. Furthermore, the path that remains for type 5 shares links with almost every other type, making it necessary for the other call types to also reroute some of their traffic and reduce their aggregated data rates. Finally, all the variables involved were set to their nominal values; i.e., $\alpha = 4$, $\beta_3 = \beta_5 = 22$, $T = 10\,\mathrm{s}$ and delays from Fig. 2.

The simulations in Fig. 6 show that the control laws excel at this task: The network reacts almost immediately to the failure. Furthermore, the utility function converges to its new optimal value, while satisfying the AS requirements. The link failure forces the network to reroute the AS traffic to the available paths. This means transferring the AS traffic to alternative paths that were almost unused by these types of traffic. This, in turn, forces the network to substantially reduce the resources allocated to BE traffic. For example, in Fig. 6 we see that data rates of BE traffic of type 2 had to be greatly reduced to make sure that AS traffic requirements were met. In

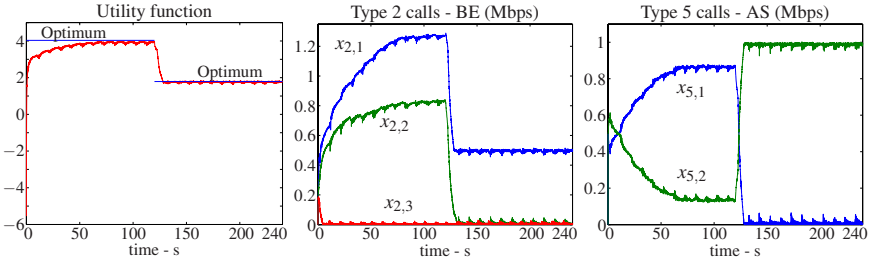



**Fig. 6.** Response to a link failure

other words, the control laws presented here endow the network with the capacity to quickly react to failures, always enforcing the AS requirements and distributing both the AS traffic and BE traffic among the available paths in such a way as to maximize the utilization of the network resources.

### 8.4 Quasi-optimal Control Laws

In this Section, simulations of the quasi-optimal control laws are presented. These laws become more relevant in light of the fact that under non-ideal conditions, the optimal laws converge only to a neighborhood of the optimal; i.e., the optimality obtained by using more information is lost but the complexity is not.

These quasi-optimal laws are also used to show an improved version that exploits the form of the non-negativity constraints on the data rates to further reduce oscillations around zero. Indeed, by removing the term $\xi_{i,j}$ in the adaptation laws and truncating the data rates, these exhibit much less oscillation. It can be seen in Fig. 7, that both control laws (with and without $\xi$) converge to a value close to the optimal one. However, the ones without $\xi$ do so with a less oscillatory behavior.

If one compares the behavior of these control laws against the optimal ones in the previous sections, one sees that they provide comparable performance. The quasi-optimal ones however, require much less information that allows for a much simpler implementation.

## 9 Conclusions

In this chapter, a unified framework for traffic engineering is presented. The approach presented enables one to address the problem of rate adaptation and load balancing in computer networks. Moreover, the algorithms presented can be applied to the case where several CoS are to be provided to network users and several paths are available between each pair of source/destination nodes. Furthermore, the algorithms endow the network with the ability of optimally adjust to changing conditions such as link or node failures. The issue of oscillation mitigation is also addressed. Namely, an adaptive scheme is provided that reduces oscillation while keeping the ability to react

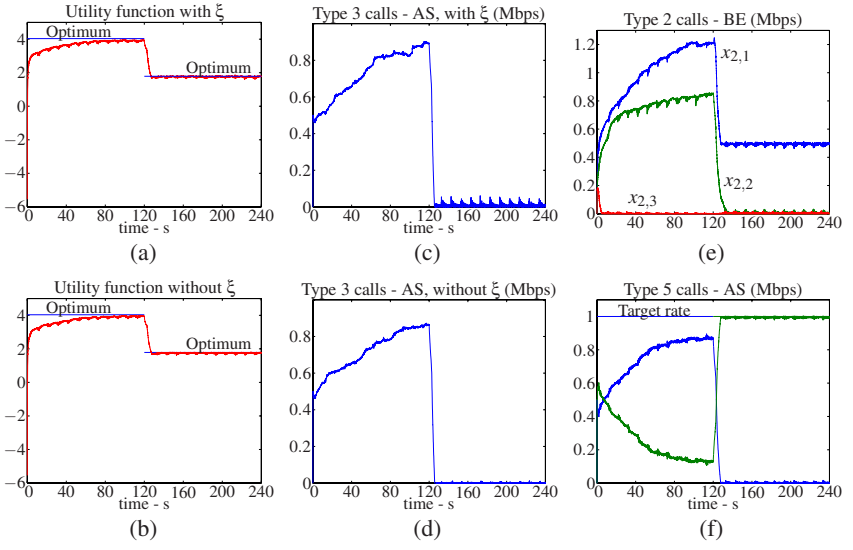



**Fig. 7.** (a,b) Utility function, (c,d) Oscillation of data rates close to zero, (e) Example of Best Effort calls without ξ, (f) Assured service calls without ξ.

to changes in demand and/or network configuration. To the best of our knowledge, this is the first comprehensive solution to this problem.

Effort is now being put on the implementation and further development of rate control laws. The proposed laws have several free parameters and different choices can lead to very different transient behaviors. Hence, we are now developing rules for determining these parameters when one is to deploy the proposed algorithms in a "real" network. Furthermore, we aim at modifying the rate adaptation laws so that one achieves optimal traffic allocation with less feedback from the network and, hence, with less traffic overhead.

## 10 Proof of the Main Results

In this Section, the proof of the main results in this chapter are presented but first some preliminaries, including additional notation is presented, that will enable us to present them in a more intuitive way.

The problem at hand can be represented in the following standard form

$$\max_{\mathbf{x}} U(\mathbf{x})$$

subject to the inequality and equality constraints of the form

$$h_k(\mathbf{x}) \leq 0;\ \ k = 1, 2, \ldots, m \qquad \text{and} \qquad h_k(\mathbf{x}) = 0;\ \ k = m+1, \ldots, L,$$

where $U(\mathbf{x})$ is a concave differentiable increasing function and $h_k(\mathbf{x})$ are affine functions for all $k$. The inequality constraints correspond to the link capacity constraints,

the restriction that the data rates have to be nonnegative and MRGS, UBRS and MRGUBS CoS constraints. On the other hand, the equality constraints correspond to the AS requirements. Now, define the matrix

$$Z(t,\mathbf{x}) \doteq \operatorname{diag}\big(z_{1,1}(t,\mathbf{x}),\ldots,z_{1,n_1}(t,\mathbf{x}),\ldots,z_{n,1}(t,\mathbf{x}),\ldots,z_{n,n_n}(t,\mathbf{x})\big).$$

Note that, given the special form of the constraints it is easy to see that the adaptation laws presented in Section 4.1 can be represented in the following form

$$\dot{\mathbf{x}} = Z(t,\mathbf{x})\big[\nabla U(\mathbf{x}) - H(\mathbf{x})\,\mathbf{v}(\mathbf{x})\big],$$

where $\nabla U(\cdot)$ denotes the gradient of the function $U(\cdot)$, $H(\cdot)$ is the following matrix

$$H(\cdot) \doteq \big[\nabla h_1(\cdot)\ \nabla h_2(\cdot)\ \cdots\ \nabla h_L(\cdot)\big]$$

and $\mathbf{v}(\cdot) = \big[v_1(\cdot), v_2(\cdot),\ldots,v_L(\cdot)\big]^T$ is an $L$-dimensional vector whose entries are of the form

$$v_k(\mathbf{x}) = \begin{cases} \gamma_k & \text{if } h_k(\mathbf{x}) > 0 \\ 0 & \text{if } h_k(\mathbf{x}) < 0, \end{cases}$$

for $k = 1,2,\ldots,m$, where $\gamma_k = \alpha$ for the constraints associated with link capacity; i.e., $k = 1,2,\ldots,\operatorname{card}(\mathcal{L})$, $\gamma_k = \xi_{i,j}$ for the constraints associated with non-negativity of $x_{i,j}$, $\gamma_k = \beta_i^M$ for the maximum allowed rate constraints on calls of type $i$ and $\gamma_i = \beta_i^m$ for the minimum service guarantee constraints on calls of type $i$. Also,

$$v_k(\mathbf{x}) = \begin{cases} \xi_k & \text{if } h_k(\mathbf{x}) > 0 \\ -\xi_k & \text{if } h_k(\mathbf{x}) < 0 \end{cases}$$

for $k = m+1, m+2,\ldots,L$, where $\xi_k = \beta_i$ for the AS constraint $k$ associated with calls of type $i$. Also, let the admissible domain be the set

$$\mathcal{D} \doteq \Big\{\mathbf{x} \in \mathbf{R}^{\sum_{i=1}^n n_i} :\ h_k(\mathbf{x}) \le 0 \text{ for } k = 1,2,\ldots,m; h_k(\mathbf{x}) = 0 \text{ for } k = m+1,\ldots,L\Big\}.$$

Essentially, the proof requires 4 steps. First, we prove that the adaptation law converges to the maximum of the function

$$\widehat{U}(\mathbf{x}) \doteq U(\mathbf{x}) - \Xi(\mathbf{x}),$$

where $\Xi(\mathbf{x}) \doteq \big[h_1(\mathbf{x}), h_2(\mathbf{x}),\ldots,h_L(\mathbf{x})\big]\mathbf{v}(\mathbf{x})$. Second, we provide necessary and sufficient conditions for the maximum of $\widehat{U}(\mathbf{x})$ to coincide with the maximum of $U(\mathbf{x})$. Third, we show that under this conditions this procedure converges to the solution of the optimization problem at hand. The final step is the realization that, under the conditions of Theorem 1, the necessary and sufficient conditions mentioned above are satisfied.

**Lemma 2.** *The function $\widehat{U}(\mathbf{x})$ does not decrease along the trajectories.*

*Proof.* If a sliding mode does not occur then

$$\frac{d\widehat{U}}{dt} = \left[\nabla U - H(\mathbf{x})\mathbf{v}(\mathbf{x})\right]^T \dot{x} = \left[\nabla U - H(\mathbf{x})\mathbf{v}(\mathbf{x})\right]^T Z(t,\mathbf{x})\left[\nabla U - H(\mathbf{x})\mathbf{v}(\mathbf{x})\right] \geq 0$$

since the matrix $Z(t,\mathbf{x})$ is positive definite.

Now, assume that a sliding mode occurs in the intersection of the surfaces $h_k(\mathbf{x}) = 0$, $k \in \mathcal{I}$. Let $H_1(\mathbf{x})$ be the matrix whose columns are $\nabla h_k(\mathbf{x})$ for $k \in \mathcal{I}$ (and in the same order as in $H(\mathbf{x})$). Also, let $H_2(\mathbf{x})$ be the matrix with columns $\nabla h_k(\mathbf{x})$ for $k \notin \mathcal{I}$ (again in the same order as in $H(\mathbf{x})$). Then, given that a sliding mode occurs in the intersection of the surfaces $h_k(\mathbf{x}) = 0$, $k \in \mathcal{I}$, we have

$$H_1(\mathbf{x})^T Z(t,\mathbf{x})\left[\nabla U - H_1(\mathbf{x})\mathbf{v}_1(\mathbf{x}) - H_2(\mathbf{x})\mathbf{v}_2(\mathbf{x})\right] = 0$$

where $\mathbf{v}_1(\mathbf{x})$ is the vector containing $v_k(\mathbf{x})$, for $k \in \mathcal{I}$ and $\mathbf{v}_2(\mathbf{x})$ is the vector containing $v_k(\mathbf{x})$, for $k \notin \mathcal{I}$. Now, assume that $\det\left[H_1(\mathbf{x})^T Z(t,\mathbf{x}) H_1(\mathbf{x})\right] \neq 0$ (a reasoning similar to the one in [14] can be done to address the case where this does not happen). From now on, to simplify the exposition, we drop the dependency on $\mathbf{x}$. Then, the equivalent control is

$$\mathbf{v}_{1,\text{eq}} = \left(H_1^T Z H_1\right)^{-1}\left(H_1^T Z \nabla U - H_1^T Z H_2 \mathbf{v}_2\right)$$

and replacing in the expression for $\dot{\mathbf{x}}$, the resulting sliding motion is

$$\dot{\mathbf{x}} = \sqrt{Z} P \sqrt{Z}\left(\nabla U - H_2 \mathbf{v}_2\right),$$

where $\sqrt{Z}$ is well defined since $Z > 0$ and $P$ is given by

$$P \doteq I - \sqrt{Z} H_1 \left(H_1^T \sqrt{Z}\sqrt{Z} H_1\right)^{-1} H_1^T \sqrt{Z}.$$

Now, let $\Xi_1$ be the elements $h_k$ of $\Xi$ with $k \in \mathcal{I}$ (in the same order as in $\Xi$). Also, let $\Xi_2$ be the elements $h_k$ of $\Xi$ with $k \notin \mathcal{I}$ (again in the same order as in $\Xi$). Since a sliding mode occurs, during this motion we have

$$\widehat{U} = U - \Xi_2 \mathbf{v}_2.$$

Now, since $U$ and $\Xi_2$ are continuously differentiable and along this sliding motion $\mathbf{v}_2$ is constant, we have

$$\frac{d\widehat{U}}{dt} = \left(\nabla U - H_2 \mathbf{v}_2\right)^T \dot{\mathbf{x}}.$$

Now, notice that $P = P^T = P^2$. Hence,

$$\frac{d\widehat{U}}{dt} = \left(\nabla U - H_2 \mathbf{v}_2\right)^T \sqrt{Z} P \sqrt{Z}\left(\nabla U - H_2 \mathbf{v}_2\right) = \left\| P\sqrt{Z}\left(\nabla U - H_2 \mathbf{v}_2\right)\right\|^2 \geq 0.$$

□

**Lemma 3.** *The time derivative of $\widehat{U}$ is zero only when $\dot{\mathbf{x}} = 0$.*

*Proof.* If a sliding mode does not occur we have $d\widehat{U}/dt = \left[\nabla U - H\mathbf{v}\right]^T Z\left[\nabla U - H\mathbf{v}\right]$ and since $Z$ is positive definite

$$\frac{d\widehat{U}}{dt} = 0 \Rightarrow \nabla U - H\mathbf{v} = 0 \Rightarrow Z\left[\nabla U - H\mathbf{v}\right] = 0 \Rightarrow \dot{\mathbf{x}} = 0.$$

Now assume that a sliding mode occurs in the intersection of the surfaces $h_k(\mathbf{x}) = 0$, $k \in \mathcal{I}$. In this case,

$$\frac{d\widehat{U}}{dt} = \left\| P\sqrt{Z}\left(\nabla U - H_2\mathbf{v}_2\right)\right\|^2.$$

Hence,

$$\frac{d\widehat{U}}{dt} = 0 \Rightarrow P\sqrt{Z}\left(\nabla U - H_2\mathbf{v}_2\right) = 0 \Rightarrow \sqrt{Z}P\sqrt{Z}\left(\nabla U - H_2\mathbf{v}_2\right) = 0 \Rightarrow \dot{\mathbf{x}} = 0.$$

□

**Lemma 4.** *The stationary points of $\widehat{U}$ are the maximum points of $\widehat{U}$.*

*Proof.* Let $\mathbf{x}_0$ be a stationary point on the intersection of surfaces given by $H_1 = 0$. In this case, we have

$$\nabla U(\mathbf{x}_0) - H_1(\mathbf{x}_0)\mathbf{v}_{1,\text{eq}}(\mathbf{x}_0) - H_2(\mathbf{x}_0)\mathbf{v}_2(\mathbf{x}_0) = 0.$$

Now, consider the function $\widehat{U}^*(\mathbf{x}) = U(\mathbf{x}) - H_1(\mathbf{x})\mathbf{v}_{1,\text{eq}}(\mathbf{x}_0) - H_2(\mathbf{x})\mathbf{v}_2(\mathbf{x})$. Given that the components of the equivalent control satisfy

$$\begin{aligned} 0 \le v_{1,\text{eq},k}(\mathbf{x}_0) \le \gamma_k \qquad & \text{for } 1 \le k \le m \text{ and} \\ -\xi_k \le v_{1,\text{eq},k}(\mathbf{x}_0) \le \xi_k \qquad & \text{for } m < k \le L, \end{aligned}$$

it holds that $H_1(\mathbf{x})\mathbf{v}_{1,\text{eq}}(\mathbf{x}_0) \le H_1(\mathbf{x})\mathbf{v}_1(\mathbf{x})$ and, as a consequence $\widehat{U}^*(\mathbf{x}) \ge \widehat{U}(\mathbf{x})$. Now, since $U$ is a concave function, $h_k$ are convex functions for $1 \le k \le m$ and $h_k$ are linear functions for $m+1 \le k \le L$ then $\widehat{U}^*$ is a concave function and hence it has a unique maximum. Furthermore, $\widehat{U}^*$ is continuously differentiable and

$$\nabla\widehat{U}^*(\mathbf{x}_0) = 0.$$

Therefore, $\max_{\mathbf{x}} \widehat{U}^*(\mathbf{x}) = \widehat{U}^*(\mathbf{x}_0)$. Now, since $\widehat{U}^*(\mathbf{x}) \ge \widehat{U}(\mathbf{x})$ and $\widehat{U}^*(\mathbf{x}_0) = \widehat{U}(\mathbf{x}_0)$, we conclude that $\widehat{U}(\mathbf{x})$ reaches its maximum at $\mathbf{x}_0$. Therefore, any stationary point of the optimization procedure is a maximum of $\widehat{U}(\mathbf{x})$. Now, assume that a maximum point $\mathbf{x}^*$ of $\widehat{U}(\mathbf{x})$ is not a stationary point. Then, we have $d\widehat{U}(\mathbf{x}^*)/dt > 0$ and so $\widehat{U}$ will increase along the trajectory which contradicts the fact that $\mathbf{x}^*$ is a maximum point of $\widehat{U}(\mathbf{x})$. □

**Lemma 5.** *If the set of all maximum points is bounded (which is our case) then* $\mathbf{x}$ *will converge to this set from any initial condition.*

*Proof.* The proof is very similar to the one in [14], Chapter 15, Section 3. It makes use of the results from Lemmas 2, 3 and 4. Therefore, we refer the reader to it. □

**Theorem 3.** *Let* $\mathbf{v}^0$ *be a vector whose entries are of the form*

$$\begin{aligned} 0 \le v_k \le \gamma_k\,; \quad & k = 1,2,\ldots,m \\ -\xi_k \le v_k \le \xi_k\,; \quad & k = m+1, m+2, \ldots, L, \end{aligned}$$

*where* $v_k = 0$ *for non-binding constraints. Then, the maximum of* $\widehat{U}(\mathbf{x})$ *coincides with the optimal* $U(\mathbf{x}^*)$ *if and only if there exists* $\mathbf{x}^*$ *such that* $\nabla U(\mathbf{x}^*) = H(\mathbf{x}^*)\mathbf{v}^0$.

*Proof.* See [14], Chapter 15, Section 4. □

**Theorem 4.** *The control laws presented above converge to the set of maximum points of the utility function* $U(\mathbf{x})$ *if this set is bounded, the condition of Theorem 3 is satisfied and vector* $\mathbf{v}^0$ *is an inner point of the set defined in Theorem 3, except for the non-binding constraints.*

*Proof.* See [14], Chapter 15, Section 4. The proof follows by using Theorem 3 and Lemma 5. □

*Remark 1.* The components of vector $\mathbf{v}^0$ are the Lagrange multipliers of the optimization problem at hand.

### 10.1 Proof of Theorem 1

The conditions on the parameters $\alpha$, $\beta_i$, $\beta_I^m$, $\beta_i^M$ and $\xi_{i,j}$ imposed in Theorem 1 imply that the necessary and sufficient conditions of Theorem 3 are satisfied for the optimization problem at hand. Indeed, if each congested link is used by a non-binding CoS or a BE call, then in $\nabla U(\mathbf{x}^*) = H(\mathbf{x}^*)\mathbf{v}^0$ the components of $\mathbf{v}^0$ associated with capacity constraints; i.e., $v_k^0$ for $k = 1,2,\ldots,\mathrm{card}\,(\mathcal{L})$, appear in a set of equations decoupled from the remaining components of $\mathbf{v}^0$. Then, the worst case (larger) value of $v_k^0$, $k = 1,2,\ldots,\mathrm{card}\,(\mathcal{L})$ is

$$v_{k,max}^0 = \max_{i,j,\mathbf{x}\in\mathcal{X}} \frac{\partial U(\mathbf{x})}{\partial x_{i,j}}$$

Now, using this information in the remaining equations, it is possible to solve for $v_{k,max}^0$, $k = m+1,\ldots .L$. Since, $U(\mathbf{x})$ is an increasing function in all its variables $x_{i,j}$, the worst case for $v_k^0$ associated with CoS constraints is

$$v_{k,max}^0 = \sum_{\kappa\in\mathcal{K}\subseteq\{1,2,\ldots,\mathrm{card}(\mathcal{L})\}} v_{\kappa,max}^0 = \max_{i,j} B_{i,j} \max_{i,j,\mathbf{x}\in\mathcal{X}} \frac{\partial U(\mathbf{x})}{\partial x_{i,j}} \doteq \alpha_{min} \max_{i,j} B_{i,j}.$$

Once these are determined, all that remains is to pick the worst case value $v_{k,max}^0$ associated with non-negativity constraints. Since each one of these appears in a single equation, where all the other multipliers are already determined, the worst case value is given by

$$v_{k,max}^0 = \alpha_{min} \max_{i,j} B_{i,j} + \beta, \qquad \beta \in \{\beta_i, \beta_i^m, \beta_i^m\}.$$

Hence, in order to satisfy the conditions in Theorem 3

$$\begin{aligned} v_k \leq v_{k,max}^0 < \gamma_k\,, \quad k = 1,2,\ldots,m \\ |v_k| \leq |v_{k,max}^0| < \xi_k\,, \quad k = m+1,\ldots,L. \end{aligned}$$

Therefore, the family of adaptation laws proposed in this paper converge to the maximum of the utility function $U(\mathbf{x})$ subject to $\mathbf{x} \in \mathcal{D}$. In other words, they converge to the optimum of our optimization problem. □

### 10.2 Proof of Theorem 2

In this section we present the proof for Theorem 2, but first we set the stage by introducing some notation that is needed.

The control laws presented in Section 4.2 can be written in terms of a *modified control* as follows: Let $\mathcal{I}_{i,j}$ and $\mathcal{I}_i$ be the set of indices $k \in \{1,2,\ldots,L\}$ such that the inequality, respectively equality, constraints $h_k(\mathbf{x})$ involve the data rate $x_{i,j}$. Furthermore, let $\mathcal{I}_{i,j}$ be classified into $\mathcal{I}_{i,j}^{\alpha}$ and $\mathcal{I}_{i,j}^{\xi}$ for the capacity and non-negativity constraints respectively. Note that each of the sets $\mathcal{I}_{i,j}^{\xi}$ and $\mathcal{I}_i$ consist of a single point. Now, define the *modified control* $u_k$ as

$$u_k = \begin{cases} 0 & \text{if } h_k > 0 \\ u_{k,\text{eq}} & \text{if } h_k = 0 \\ 1 & \text{if } h_k < 0 \end{cases} \qquad k \in \{1,2,\ldots,m\}$$

$$u_k = \begin{cases} (1-r_i) & \text{if } h_k \neq 0 \\ u_{k,\text{eq}} & \text{if } h_k = 0 \end{cases} \qquad k \in \{m+1,\ldots,L\} \cap \mathcal{I}_i,$$

where $u_{k,\text{eq}}$ applies when a sliding mode occurs in the surface $h_k(\mathbf{x}) = 0$ and is defined as the convex combination of the maximum ($\overline{u}_k$) and minimum ($\underline{u}_k$) values of $u_k$; i.e., $u_{k,\text{eq}} \doteq \lambda \underline{u}_k + (1-\lambda)\overline{u}_k$, with $\lambda \in [0,1]$. Finally, to simplify the notation let $z_{i,j}$ denote

$$z_{i,j}\left(t, \mathbf{x}_i(t), \{u_k\}_{k \in \mathcal{I}_{i,j}^{\alpha}}\right).$$

Then, the following *modified control* laws are equivalent to the ones in Section 4.2: For $i = 1,2,\ldots,s$; i.e, AS calls, let

$$\dot{x}_{i,j} = z_{i,j}\left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha\left(1 - \prod_{k \in \mathcal{I}_{i,j}^{\alpha}} u_k\right) - \beta_i(1-u_l) + \xi_{i,j}(1-u_p)\right],$$

where $l \in \mathcal{I}_i$ and $p \in \mathcal{I}_{i,j}^{\xi}$. For $i = s+1, s+2, \ldots, n$; i.e., BE calls, let

$$\dot{x}_{i,j} = z_{i,j}\left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha\left(1 - \prod_{k \in \mathcal{I}_{i,j}^{\alpha}} u_k\right) + \xi_{i,j}(1-u_p)\right].$$

*Remark 2.* There are as many parameters $u_k$ as there are constraints in the optimization problem at hand, as opposed to one parameter $r_i$ per type of calls and one parameter $bin_{i,j}$ and $\xi_{i,j}$ per path.

In a similar manner, the *convergent* control laws presented in Section 4.2 can also be recast as a *modified control* form: For $i = 1,2,\ldots,s$; i.e, AS calls, let

$$\dot{x}_{i,j}^{int} = z_{i,j}\left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha\sum_{k\in\mathcal{I}_{i,j}^{\alpha}}(1-u_k^{int}) - \beta_i(1-u_l) + \xi_{i,j}(1-u_p)\right],$$

where $l \in \mathcal{I}_i$ and $p \in \mathcal{I}_{i,j}^{\xi}$. For $i = s+1, s+2,\ldots,n$; i.e., BE calls, let

$$\dot{x}_{i,j}^{int} = z_{i,j}\left[\left.\frac{\partial f_i}{\partial x_{i,j}}\right|_{\mathbf{x}_i} - \alpha\sum_{k\in\mathcal{I}_{i,j}^{\alpha}}(1-u_k^{int}) + \xi_{i,j}(1-u_p)\right].$$

Finally, let $\mathrm{ord}(\alpha^{-1})$ and $\mathrm{Ord}(\alpha^{-1})$ denote terms of the order of $\alpha^{-1}$ in the sense

$$\lim_{\alpha\to\infty}\alpha\,\mathrm{ord}(\alpha^{-1}) = 0 \qquad \text{and} \qquad \lim_{\alpha\to\infty}\alpha\,\mathrm{Ord}(\alpha^{-1}) = \mathrm{cte}.$$

*Proof.* The following proof stems from the fact that the control laws in Section 4.2 provide convergence in finite time to the admissible region $\mathcal{C}$, where

$$\mathcal{C} = \left\{\mathbf{x}\in\mathbf{R}^{n_1+\cdots+n_n} : h_k(\mathbf{x}) \le 0,\ k = 1,2,\ldots,m\right\}.$$

Given the assumptions on the utility function $U(\mathbf{x})$, the conditions on $\alpha$, $\beta_i$ and $\xi_{i,j}$ can be satisfied. Therefore, the derivatives $\dot{x}_{i,j}$ are bounded away from 0 for all $\mathbf{x}\notin\mathcal{C}$ and convergence to $\mathcal{C}$ in finite time is guaranteed.

For simplicity, the case where $z_{i,j}(\cdot) = 1$ is shown here. However, a straightforward modification can be done to address the case of a general $z_{i,j}(\cdot)$.

Let $\mathbf{x}^*$ denote the optimal solution to the problem posed in Section 3 and let $\mathbf{x}^{eq}$ denote any equilibrium point of the laws above that might exist.

Without loss of generality, let a sliding mode occur in the admissible region along the intersection of a set of surfaces $h_k(\mathbf{x}) = 0$, $k\in\mathcal{I}$. Since a sliding mode occurs along these surfaces it follows that $\dot{h}_k(\mathbf{x}) = 0$, for all $k\in\mathcal{I}$ and all $\alpha$. Hence, given the specific linear dependence of the previous expression on $\dot{x}_{i,j}$, the term

$$\alpha\left(1 - \prod_{k\in\mathcal{I}\cap\mathcal{I}_{i,j}^{\alpha}} u_{k,\mathrm{eq}}\right)$$

is bounded for all $i$, $j$ and $\alpha$. Moreover, as $\alpha\to\infty$, it holds that $u_{k,\mathrm{eq}}\to 1$, $\forall k\in\mathcal{I}$. Therefore, for all $i$, $j$, the following Taylor expansion holds around $u_k = 1$, $\forall k\in\mathcal{I}$

$$\alpha\left(1 - \prod_{k\in\mathcal{I}\cap\mathcal{I}_{i,j}^{\alpha}} u_{k,\mathrm{eq}}\right) = \alpha\sum_{k\in\mathcal{I}\cap\mathcal{I}_{i,j}^{\alpha}}\Big((1-u_{k,\mathrm{eq}}) + \mathrm{Ord}\big((1-u_{k,\mathrm{eq}})^2\big)\Big).$$

Since the left hand side is bounded for every $\alpha$ it holds that

$$\lim_{\alpha\to\infty} \alpha \sum_{k\in\mathcal{I}\cap\mathcal{I}_{i,j}^{\alpha}} \mathrm{Ord}\big((1-u_{k,\mathrm{eq}})^2\big) = 0, \quad \text{and} \quad \alpha \sum_{k\in\mathcal{I}\cap\mathcal{I}_{i,j}^{\alpha}} \big(1-u_{k,\mathrm{eq}}\big) < M \quad M\in\mathbf{R}$$

Equivalently we can write,

$$1 - \prod_{k\in\mathcal{I}\cap\mathcal{I}_{i,j}^{\alpha}} u_{k,\mathrm{eq}} \;=\; \sum_{k\in\mathcal{I}\cap\mathcal{I}_{i,j}^{\alpha}} (1-u_{k,\mathrm{eq}}) + \mathrm{ord}(\alpha^{-1}).$$

Now, using the expression for the equivalent control (10)

$$\big(H_1^T H_1\big)\big(\mathbf{v}_{1,\mathrm{eq}}\big) = \big(H_1^T\nabla U + H_1^T H_2 \mathbf{v}_2\big)$$
$$\big(H_1^T H_1\big)\big(\mathbf{v}_{1,\mathrm{eq}}^{bin} + \overline{\mathrm{ord}}(\alpha^{-1})\big) = \big(H_1^T\nabla U + H_1^T H_2 \mathbf{v}_2^{bin}\big),$$

where $\dot{\mathbf{v}}_2 = \dot{\mathbf{v}}_2^{bin} = 0$ and $\mathbf{v}_2 = \mathbf{v}_2^{bin}$. Subtracting these, $\big(\mathbf{v}_{1,\mathrm{eq}} - \mathbf{v}_{1,\mathrm{eq}}^{bin}\big) = \overline{\mathrm{ord}}(\alpha^{-1})$, where $\overline{\mathrm{ord}}(\alpha^{-1})$ is a vector whose components are of the form $\mathrm{ord}_k(\alpha^{-1})$. Hence,

$$\dot{x}_{i,j} = \dot{x}_{i,j}^{int} + \alpha\,\mathrm{ord}(\alpha^{-1}).$$

Furthermore, since $\dot{U}(\mathbf{x}) = \nabla U(\mathbf{x})^T\dot{\mathbf{x}}$,

$$\dot{U}(\mathbf{x}) = \nabla U(\mathbf{x})^T\dot{\mathbf{x}}^{int} + \nabla U(\mathbf{x})\alpha\,\mathrm{ord}(\alpha^{-1}) = \dot{U}^{int}(\mathbf{x}) + \mathrm{Ord}(\alpha^{-1}).$$

Now, given $\varepsilon > 0$ define $\mathcal{B}_\varepsilon(\mathbf{x}^*)$ to be an open neighborhood of radius $\varepsilon$ around the optimal $\mathbf{x}^*$.

Therefore, since $dU^{int}(\mathbf{x})/dt > 0$ for all $\mathbf{x}\neq\mathbf{x}^*$, and the set $\mathbf{R}$ is dense; there exists $\alpha^*$ such that for all $\alpha > \alpha^*$, and all $\mathbf{x}\notin\mathcal{B}_\varepsilon(\mathbf{x}^*)$, it holds that $\dot{U}(\mathbf{x}) > 0$. Hence, there are no stationary points of the control law outside $\mathcal{B}_\varepsilon(\mathbf{x}^*)$ and since for every trajectory $U(\cdot)$ is strictly increasing outside this neighborhood, then $\mathbf{x}\to\mathcal{B}_\varepsilon(\mathbf{x}^*)$.

Finally, since the choice of $\varepsilon$ is arbitrary, it holds that for all $\varepsilon > 0$ there exists $\alpha^*$, such that the above control laws converge to $\mathcal{B}_\varepsilon(\mathbf{x}^*)$, for all $\alpha > \alpha^*$. □

### 10.3 Sketch of the Proof of Lemma 1

The proof essentially requires two steps. First, we show that a sliding motion on the surfaces $x_{i,j} = 0$ is obtained. Second, we show that the utility function does not decrease along these trajectories, in essence the same reasoning used in Section 10.1.

Since the proposed laws reduce to the ones in Theorem 1 for the case $x_{i,j} > 0$, we concentrate without loss of generality, on the case $x_{i,j} = 0$.

Let $x_{i,j} = 0$ at time $t = t_0$, for some $i$, $j$. If at $t_0$, $p_{i,j} > 0$, then the adaptation laws reduce to the ones in Theorem 1 and convergence is guaranteed.

On the other hand, if $p_{i,j} < 0$, a sliding mode will occur on the surface $x_{i,j} = 0$, with $\dot{x}_{i,j} = 0$. Indeed, since the constraints of the optimization problem at hand are affine, by forcing $\dot{x}_{i,j} = 0$ a sliding mode on the aforementioned surface occurs, as is the case with the laws in Theorem 1. Furthermore, the equivalent control is also

the same. This is the case because the equivalent control has the effect of keeping the trajectories tangent to the sliding surface with $\dot{x}_{i,j} = 0$. Moreover, all other derivatives $\dot{x}_{k,l}$ such that $x_{k,l} \neq 0$ at time $t_0$ are the same.

Therefore using the same reasoning as in Section 10.1, the utility function does not decrease along the trajectories generated by the above control laws and a near-optimum is achieved. □

## References


[1] Flavio Bonomi, Debasis Mitra, and Judith B. Seery. Adaptive algorithms for feedback-based flow control in high-speed, wide-area networks. *IEEE J. Select. Areas Commun.*, 13(7):1267–1283, September 1995.

[2] D. Chiu and R. Jain. Analysis of the Increase/Decrease algorithms for congestion avoidance in computer networks. *J. Comp. Networks and ISDN systems*, 17(1):1–14, June 1989.

[3] Aleksei Fedorovich Filippov. *Differential Equations with Discontinuous Right-hand Sides*. Mathematics and Its Applications. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1988.

[4] Sally Floyd and Tom Henderson. The new Reno modification to TCP's fast recovery algorithm. IETF, RFC 2582, April 1999.

[5] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. Networking*, 1(4):397–413, August 1993.

[6] S. Jamaloddin Golestani and Supratik Bhattacharyya. A class of end-to-end congestion control algorithms for the Internet. In *Proc. Int. Conf. Network Protocols, ICNP*, pages 137–150, October 1998.

[7] Frank P. Kelly, Aman K. Maulloo, and Dacid K. H. Tan. Rate control in communication networks: Shadow prices, proportional fairness and stability. *J. Oper. Res. Soc.*, 49(3):237–252, March 1998.

[8] Richard J. La and Venkat Anantharam. Charge-sensitive TCP and rate control in the Internet. In *Proc. IEEE INFOCOM'2000*, pages 1166–1175, March 2000.

[9] Constantino M. Lagoa, Hao Che, and Bernardo Adrián Movsichoff. Adaptive control algorithms for decentralized optimal traffic engineering in the Internet. To appear in IEEE/ACM Trans. Networking, 2004.

[10] Steven H. Low. Optimization flow control with on-line measurement or multiple paths. In *Proc. 16th Int. Teletraffic Cong.*, Edinburgh, U.K., June 1999.

[11] Laurent Massoulié and James Roberts. Bandwidth sharing: Objectives and algorithms. In *Proc. IEEE INFOCOM '99*, pages 1395–1403, March 1999.

[12] Gopalakrishnan Ramamurthy and Aleksandar Kolarov. Application of control theory for the design of closed loop rate control for ABR service. In *Proc. Int. Test Conf.,ITC*, pages 751–760, Washington, USA, 1997.

[13] Saswati Sarkar and Leandros Tassiulas. Distributed algorithms for computation of fair rates in multirate multicast trees. In *Proc. IEEE INFOCOM'2000*, volume 1, pages 52–61, Tel Aviv, Israel, March 2000.

[14] Vadim I. Utkin. *Sliding Modes in Control and Optimization*, volume 66 of *Communications and Control Engineering Series*. Spriger-Verlag, Berlin, Heidelberg, 1992.

# Modeling and Designing the Internet Congestion Control

Saverio Mascolo[1]

Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, via Orabona, 4, Italy.
mascolo@poliba.it

## 1 Introduction

The Internet is a set of interconnected packet switching networks each of them having its own protocol. It was soon recognized the importance of internetworking these different networks. To this purpose, a cross-network protocol called the Transmission Control Protocol (TCP) was proposed [7],[9],[36], [34].

The Transmission Control Protocol implements a set of functionalities that enable the reliable internetworking of heterogeneous networks. One of these functionalities is the congestion control algorithm that has been introduced in 1988 by Van Jacobson after the network was suffering from congestion collapse [21]. Since that time, the TCP congestion control algorithm has been of crucial importance and quite effective in preventing congestion collapse.

TCP congestion control is one of the most complex piece of the TCP stack since it implements dynamic feedback control. TCP has two feedback mechanisms to tackle congestion: the flow control and the congestion control. The TCP flow control aims at avoiding the overflow of the receiver's buffer and is based on explicit feedback. In particular, the TCP receiver sends to the source the Receiver Advertised Window, which is the buffer available at the receiver. The TCP congestion control aims at avoiding the flooding of the network and is based on implicit feedback such as timeouts, duplicate acknowledgments (DUPACKs), round trip time measurements.

The original TCP congestion control, called TCP Tahoe [21] introduced a probing mechanism made of two phases, the slow-start phase and the congestion avoidance phase. The probing phase aims at increasing the flow input rate until the network available capacity is hit and a congestion episode happens, after which the window is reduced to one. Later two other basic mechanisms, fast retransmit and fast recovery, were introduced to form the Reno version of TCP [2],[21],[22]. The sender becomes aware of congestion via the reception of duplicate acknowledgments (DUPACKs) or the expiration of a timeout. Then it reacts to light congestion (i.e. 3 DUPACKs) by halving the congestion window (fast recovery) and sending again the missing packet (fast retransmit), and to heavy congestion (i.e. timeout) by reducing the congestion

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 137–158, 2004.

window to one. Reno TCP proved to be much more efficient than Tahoe since it reduces the number of timeouts and, consequently, reduces network underutilization during slow-start phases. Both the flow and congestion control implements the self-clocking principle, that is, when a packet exits a new one enters the network. The described mechanisms form the core of the classic Internet congestion control algorithm known as Tahoe/Reno TCP [21],[2], [35]. It is interesting to notice that these mechanisms are still the core of all enhanced TCP congestion control algorithms that have been proposed in the literature.

Research on TCP congestion control is today still active in order to improve its efficiency and fairness, especially in new environments such as the wireless Internet [4, 8, 29] or the high-speed Internet [10, 23, 24, 37]. We briefly summarize the most significant modifications that have been proposed up to now [27].

The New Reno feature is an enhancement of Reno that has been proposed to avoid multiple window reductions in a window of data [13, 18]. TCP Vegas estimates the expected connection rate as $cwnd/RTT_m$, where $RTT_m$ is the minimum round trip time $RTT$, and the actual connection rate as $cwnd/RTT$; when the difference $\Delta$ between the expected and the actual rate is less than a threshold $a > 0$, the *cwnd* is additively increased. When the difference is greater than a threshold $b > a$ then the *cwnd* is additively decreased. When $a \leq \Delta \leq b$, *cwnd* is kept constant [6].

TCP Westwood proposes an end-to-end estimation of the available bandwidth to adaptively set the control windows after congestion [29]. Westwood+ TCP [15, 16, 18] proposes a slight modification of the bandwidth estimation algorithm that is robust with respect to ACK compression. Both Vegas and Westwood preserve the standard multiplicative decrease behavior after a timeout.

TCP Santa Cruz proposes to employ an estimate of the forward path delay, rather than the round trip delay, to reach a target operating point for the number of packets in the bottleneck of the connection [33]. In [14] the concept of generalized advertised window has been proposed to provide an explicit indication of the network congestion status. The concept of explicit feedback has been exploited also in [10]. A comparison of the control algorithms proposed in [14] and [10] appears an issue valuable to be investigated.

Fast TCP congestion control algorithm has been recently proposed by researchers at Caltech . In authors' words, "Fast TCP is a sort of high-speed version of Vegas" [24]. At the time of this paper Fast TCP is still in a trial phase and authors do not have released any kernel code or ns-2 implementation. Being based on RTT measurements to infer congestion, it could inherit all drawbacks of Vegas that will be illustrated in this paper, mainly the incapacity to grab bandwidth when coexisting with Reno traffic or in the presence of reverse traffic [18].

Recently, non linear stochastic differential equations have been proposed to model the dynamics of the TCP congestion window (*cwnd*) [15, 20, 25, 32]. In these models, the dynamics of the expected value of the *cwnd* depends on packet drop probability through a non-linear differential stochastic equation. These models have been used to predict the long-term throughput of the TCP and to design control laws for throttling the packet drop probability of routers implementing Active Queue Management. In particular, the mentioned nonlinear stochastic differential equation

of the TCP window dynamics has been linearized around the equilibrium point to derive a transfer function from the packet drop probability to the bottleneck queue length. The linearized model has been employed to design a control law for throttling the packet drop rate aiming at stabilizing the queue average length in [20]. A remaining issue is to investigate how effective and robust are these methods to deal with real-time dynamics of the TCP in the presence of multi-bottleneck topologies with reverse and cross traffic [18].

This paper proposes a classical control theoretic approach to model the TCP flow and congestion control, along with its variants such as for example Reno and Westwood, in a unified framework. The model is general and captures multi-bottlenecks as well as moving bottleneck. The work is organized as follows: Section 2 outlines the TCP flow and congestion control algorithm; Section 3 models a generic TCP going over a store-and-forward IP network using buffers and integrators; Section 4 models the TCP flow and congestion control using a Smith predictor and a proportional controller; Section 5 models different TCP algorithms, such as Reno and Westwood TCP, by properly setting the controller reference input; Section 6 proposes an application of the developed model to design a TCP friendly rate-based control algorithm; Section 7 shows that a standard *PID* controller cannot be used in data networks because it would provide a too sluggish closed loop behavior; finally, Section 8 draws the conclusions.

## 2 Background: today TCP/IP flow and congestion control algorithm

A TCP connection establishes a virtual pipe between the send socket buffer and the receive socket buffer (see Fig. 1). The Sender Application can send arbitrary bytes into the Send Socket Buffer; TCP guarantees that the Receiver Application will receive all the data in the correct order. The objective of TCP control algorithm, which is executed at the sender side, is to avoid network congestion and overflow of buffer at the receiver side. Today's TCP has two feedback mechanisms to tackle congestion: the flow control mechanism that prevents the sender from overflowing the receiver's buffer, and the congestion control mechanism that prevents the sender from overloading the network [34].

### 2.1 The TCP Flow Control Algorithm

TCP *flow control* is based on *explicit feedback*. In particular, the TCP receiver sends the *Receiver Advertised Window*, which is the buffer available at the receiver, to the source. Letting *MaxRcvBuffer* be the size of the receiver buffer in bytes, *LastByteRcvd* be the last byte received and *NextByteRead* the next byte to be read, the TCP on the receive side must keep

$$LastByteRcvd - NextByteRead \leq MaxRcvBuffer$$

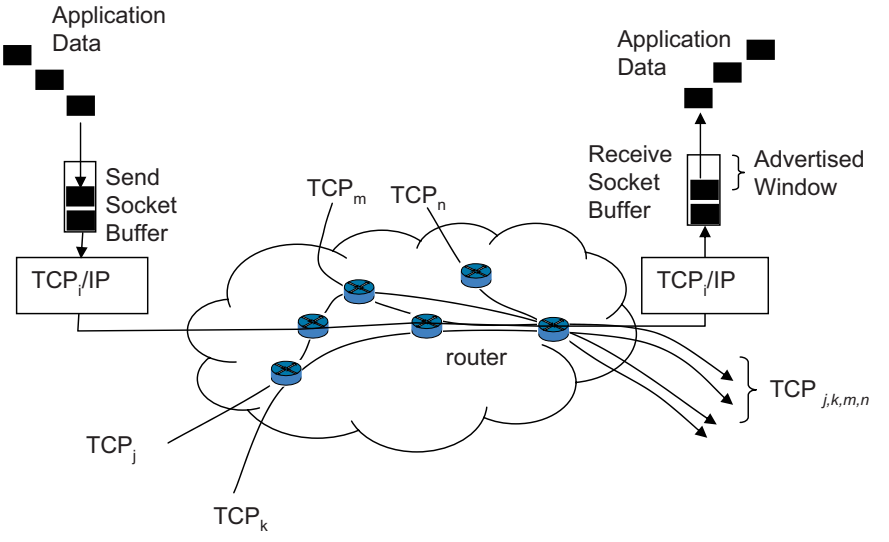



**Fig. 1.** Schematic of a TCP flow sharing an IP network

to avoid the overflow. Therefore receiver advertises a window size of

$$AdvertisedWindow = MaxRcvBuffer - (LastByteRcvd - NextByteRead)$$

which represents the amount of free space remaining in the receiver buffer. The TCP on the send side computes an *Effective Window* ($W$)

$$W = AdvertisedWindow = -(LastByteSent - LastByteACKed) = AdvertisedWindow = -OutstandingPackets \quad (1)$$

which limits how much outstanding data it can send.

### 2.2 The Standard TCP Tahoe/Reno Congestion Control Algorithm

The TCP *congestion control* algorithm aims at avoiding network flooding. Today's TCP estimates the best effort capacity of the network by *increasing/decreasing* a variable called Congestion Window (*cwnd*). This method can be viewed as a "trial and error" procedure to obtain a "rough" but robust measurement of the best effort available bandwidth.

The congestion window plays, in congestion control, the role of the *advertised window* in flow control. Van Jacobson's cornerstone paper [21] defines an $additive - increase$ and $multiplicative - decrease$ ($AIMD$) mechanism to throttle the size of the congestion window. The key idea is to probe the network capacity by increasing the load until packet loss is experienced. When a loss is experienced

the window size is reduced. More precisely, during the slow start phase *cwnd* is exponentially increased until the slow start threshold (*ssthresh*) value is reached. This phase is intended to quickly grab available bandwidth. After the *ssthresh* value is reached, the *cwnd* is linearly increased to gently probe for extra available bandwidth. This phase is called congestion avoidance. At some point the TCP connection starts to lose packets. After a timeout is experienced the window is drastically reduced to one and the slow start, congestion avoidance cycle repeats. This behavior is known as TCP Tahoe. During the slow start phase the network can be strongly underutilized. To minimize the number of timeouts, there were introduced two other mechanisms: fast retransmit and fast recovery. Implementation of these mechanisms forms what is currently known as the Reno version of TCP [2, 21, 22, 35]. Fast retransmit and fast recovery are both triggered after the sender receives three duplicate acknowledgments (ACKs). The mechanism of fast retransmit sends again the packet that was acknowledged three times. The mechanism of fast recovery reduces the congestion window to half and enters into congestion avoidance phase. After a time out, the Congestion window is reduced to one.

Fig. 2 shows the typical behavior of the Tahoe/Reno *cwnd*, whereas a pseudocode of TCP congestion window dynamics for Tahoe and Reno is reported below (for details see, f.i., [2, 21, 35]):

**1. Tahoe/Reno**

```
After every non-repeated ACK:
if cwnd<ssthresh set cwnd=cwnd+1; / slow start phase
else set cwnd=cwnd+1/cwnd; / congestion avoidance phase
```

**2. Tahoe**

```
After a packet loss is detected (when timer expires or the
number of repeated ACKs exceeds a threshold):
set ssthresh = cwnd/2;
set cwnd=1.
```

**2. Reno**

```
When the number of repeated ACKs exceeds a threshold,
retransmit "next expected" packet;
set ssthresh =cwnd/2, then set cwnd= ssthresh+3;
resume congestion avoidance phase using new window once
retransmission is acknowledged.
```

**3. Tahoe/Reno**

```
Upon timer expiry, the algorithm goes into slow start phase
as before: set ssthresh =cwnd/2;
set cwnd=1.
```

The TCP sender computes the minimum of the congestion window and the advertised window and computes the Effective Window W

$$W = MIN(Cwnd, AdvertisedWindow) - OutstandingPackets \tag{2}$$

The Effective Window corresponds to the amount of data that are injected into the network [2, 21, 34, 35].

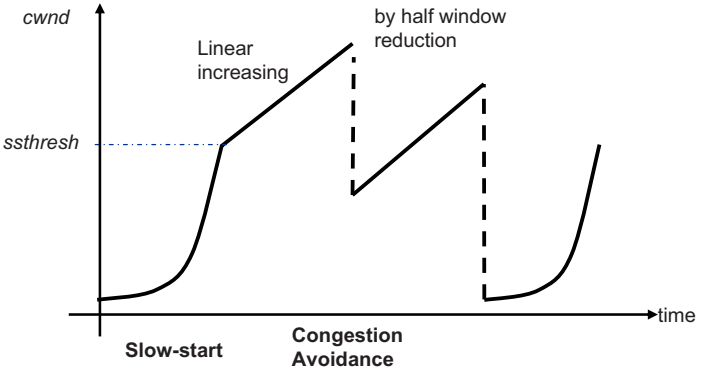



**Fig. 2.** *cwnd* dynamics of TCP Reno: typical behavior

### 2.3 The TCP Westwood+ Congestion Control Algorithm

The main idea of Westwood+ TCP is to use the rate of acknowledgment packets to estimate the available bandwidth $B$ and adaptively set the control windows after congestion. In particular, the stream of ACK packets are used to infer a measurement of the bandwidth used by the TCP connection as shown in Fig. 3. At the end of the probing phase, this measurement, which is, by definition, an estimate of the *best effort* available bandwidth, is employed to properly set the congestion window and the slow-start threshold as shown in Fig. 4.

The rationale of TCP Westwood+ is simple: in contrast with TCP Reno, which implements a *blind* multiplicative decrease algorithm after congestion, TCP Westwood+ sets a slow start threshold and a congestion window which are consistent with the effective bandwidth used at the time congestion is experienced. A pseudo-code of the algorithm is reported below:

***When 3 DUPACKs are received by the sender:***
*set* $ssthresh = max(2, B * RTTmin)$;
*set* $cwnd = ssthresh$;
***When coarse timeout expires:***
*set* $ssthresh = max(2, B * RTTmin)$;
$cwnd = 1$;
***When ACKs are successfully received:***
*cwnd is increased following the Reno algorithm*

Basically, after a timeout or 3 DUPACKs, the bandwidth estimate is used to set the control windows *cwnd* and *ssthresh* equal to the bandwidth estimate $B$, expressed in [packets/s], times the minimum round trip time $RTTmin$ expressed in seconds. This setting follows the known principle of *keeping the pipe full* [34].

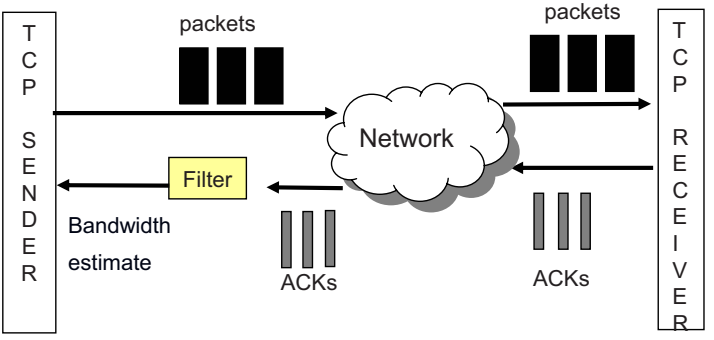



**Fig. 3.** Bandwidth estimation

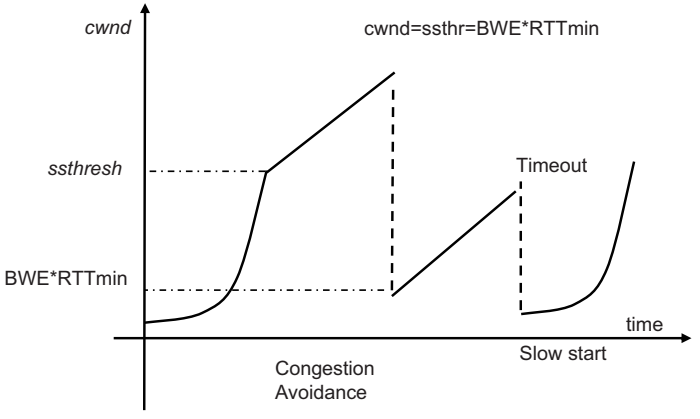



**Fig. 4.** *cwnd* dynamics of TCP Westwood: typical behavior

TCP Westwood+ congestion control heavily relies on the bandwidth estimate; thus, it is of crucial importance to obtain a good estimate of the bandwidth that is available when congestion is experienced.

In the original version of Westwood TCP [29], a sample of available bandwidth

$$b_k = d_k/(t_k - t_{k-1}) = d_k/\Delta_k$$

is computed every time $t_k$ the sender receives an ACK, where the amount $d_k$ of data acknowledged by an ACK is determined by a proper counting procedure that takes into account delayed ACKs, duplicate and cumulative ACKs [29]. Samples $b_k$ are low-pass filtered using the time-varying filter

$$\hat{b}_k = \frac{2\tau_f - \Delta_k}{2\tau_f + \Delta_k}\hat{b}_{k-1} + \frac{\Delta_k}{2\tau_f + \Delta_k}(b_k + b_{k-1}) \tag{3}$$

where $\tau_f$ is the filter time constant (a typical value is $\tau_f = 0.5s$), because congestion is due to the low frequency content of the load [26]. Later it was found that

this estimate provides large overestimate in the presence of ACK compression due to aliasing phenomena [15, 16]. As a consequence, a slightly modified version of the filter used in [29] has been proposed and originates the version called Westwood+. The novelty of Westwood+ is that bandwidth samples are now computed every $RTT$. More precisely, all data $D_k$ acknowledged during the last $RTT$ are counted to compute bandwidth samples

$$b_k = D_k/\Delta_k \tag{4}$$

where $\Delta_k$ is the last $RTT$. Again, samples (4) are filtered using Eq. 3. Westwood+ TCP has been recently implemented in the kernel of Linux 2.4 and experimental results have been published in [11, 30]. A deep investigation and comparison of Westwood+, New Reno and Vegas TCP congestion control is reported in [18].

## 3 Modelling a generic TCP flow

In his milestone paper [21], Van Jacobson clearly states that: "A packet network is to a very good approximation a linear system made of gains, delays and integrators". In this paper we propose a model of a TCP/IP connection using (a) integrators, to model network and receiver buffers and (b) delays, to model propagation times.

A data network is a set of store-and-forward nodes connected by communication links. A generic TCP flow goes through a communication path made of a series of buffers and communication links. The number of packets of the considered TCP flow that are stored at the generic $i-th$ buffer along the communication path is given by the following dynamic equation:

$$x_i(t) = \int_{-\infty}^{t} [u_i(\tau) - b_i(\tau) - o_i(\tau)]d\tau \tag{5}$$

where $u_i(t) \geq 0$ models the data arrival rate, $b_i(t) \geq 0$ models the data depletion rate, i.e. the used bandwidth, and $o_i(t) \geq 0$ models the overflow data rate, i.e. the data that are lost when the buffer is full and the input rate exceeds the output rate. The dynamic equation of the generic communication link $(i-1)$ connecting the $(i-1)-th$ buffer to the next $i-th$ buffer is a pure delay. In particular, letting $b_{i-1}(t)$ be the link input rate at the $(i-1)-th$ buffer and $u_i(t)$ be the link output rate at the next $i-th$ buffer, it results:

$$u_i(t) = b_{i-1}(t - T_{i-1}) \tag{6}$$

where $T_{i-1}$ is the link propagation time. Starting from the basic Equations 5 and 6, we propose to model a generic TCP flow over an IP network as it is shown in Fig. 5. In particular, Fig. 5 is a functional block diagram showing:

1. The TCP connection receiver buffer of length $x_r(t)$, which is modeled using an integrator with Laplace transfer function $1/s$. The receiver buffer receives the inputs $u_r(t)$, $b_r(t)$, $o_r(t)$, which represent the input rate, the depletion rate and the overflow data rate, respectively;

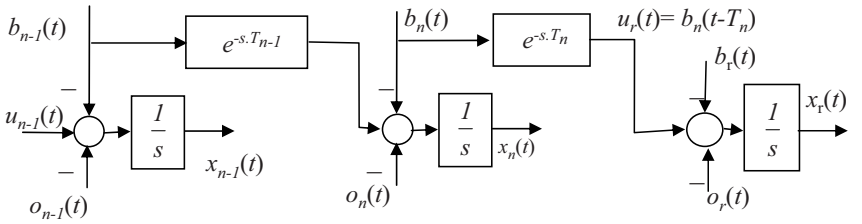



**Fig. 5.** Functional block diagram of a generic TCP/IP flow

2. The $n-th$ buffer that the TCP connection goes through before reaching the receiver buffer, which is modeled using an integrator with output $x_n(t)$. The $n-th$ buffer receives the inputs $u_n(t)$, $b_n(t)$, $o_n(t)$, which, again, represent the input rate, the depletion rate and the overflow data rate, respectively. It is important to notice that the depletion rate $b_n(t)$ reaches the next buffer $(n+1)$, which is the receiver buffer, after the propagation time $T_n$, i.e. $u_r(t) = b_n(t-T_n)$. Moreover, it should be noted that the input rate $u_n(t)$ is equal to the depletion rate $b_{n-1}(t)$ at the previous $(n-1)-th$ buffer, i.e. $b_{n-1}(t-T_{n-1}) = u_n(t)$, where $T_{n-1}$ is the propagation time from the $(n-1)-th$ buffer to the $n-th$ buffer. Depletion rates are unpredictable because they model the best effort bandwidth available for a TCP connection when going over statistically multiplexed IP networks.

The series of buffers shown in Fig. 5 can be recursively augmented both in the left direction, to model up to the first buffer node encountered by the TCP connection, and in the right direction to model buffers $n+j$, with $j = 2, p$, that are encountered by ACK packets when going back from the receiver to the sender. By considering a closed surface that contains the TCP path going from the first to the last buffer $m$, where $m = n+p$, that is encountered by the TCP flow in its round trip, we can invoke the flow conservation principle for the unique input rate, which is the TCP input rate $u_1(t)$, and the output rates that are: (a) $b_m(t)$, which models the bandwidth used by the TCP connection, i.e. the best-effort bandwidth as viewed by the considered TCP flow through the ACK stream; and (b) the overflow rates $o_i(t)$, for $i = 1, m$, which represent packets that are lost along the path due to congestion.

In equations, we can write the number $x(t)$ of packets belonging to the considered TCP flow and stored into the network by summing packets stored at each buffer along the path:

$$x(t) = \sum_{i=1}^{m} x_i(t) \tag{7}$$

Substituting Eq. 5 in Eq. 7 and considering Eq. 6 it turns out

$$x(t) = \int_{-\infty}^{t} [u_1(\tau) - b_m(\tau) - \sum_{i=1}^{m} o_i(\tau) - \sum_{i=1}^{m-1} (b_i(\tau) - b_i(\tau - T_i))] d\tau$$

that can be rewritten as

$$x(t) = \int_{-\infty}^{t} [u_1(\tau) - b_m(\tau) - \sum_{i=1}^{m} o_i(\tau)]d\tau - \sum_{i=1}^{m-1} \int_{t-T_i}^{t} b_i(\tau)d\tau \quad (8)$$

Eq. 8 states that the network storage is equal to the integral of the TCP input rate $u_1(t)$ minus the output rate $b_m(t)$ leaving the last buffer of the path, minus the sum of the overflow rates $o_i(t)$, minus the sum of packets that are in flight over each link $i$.

Since the TCP implements an end-to-end congestion control that does not receive any explicit feedback from the network, it is not possible for the controller to know terms in Eq. 8. Thus, we consider the sum of the in flight packets plus the stored packets, which we call the total network storage $x_t$:

$$x_t(t) = x(t) + \sum_{i=1}^{m-1} \int_{t-T_i}^{t} b_i(\tau)d\tau = \int_{-\infty}^{t} [u_1(\tau) - b_m(\tau) - \sum_{i=1}^{m} o_i(\tau)]d\tau$$

and the sum of overflow rates $o_t$:

$$o_t(t) = \sum_{i=1}^{m} o_i(t)$$

Thus, we can write

$$x_t(t) = \int_{-\infty}^{t} [u_1(\tau) - b_m(\tau) - o_t(\tau)]d\tau \quad (9)$$

By considering that the TCP establishes a *circular flow*, i.e. that the data input rate enters the network and comes back to the sender as an ACK rate, it can be said that $b_m(t)$ models the rate of ACK packets. Thus we can write:

$$b_m(t) = u_1(t - T) - o_t(t) \quad (10)$$

By substituting Eq. 10 into Eq. 9 it turns out:

$$x_t(t) = \int_{-\infty}^{t} [u_1(\tau) - u_1(\tau - T)]d\tau = \int_{t-T}^{t} u_1(\tau)d\tau = outstandingpackets \quad (11)$$

Equation 11 states that the network total storage is equal to the integral of the input during the last round trip time $T = RTT$.

## 4 Modelling the TCP flow and congestion control

This section aims at showing that the closed loop control system depicted in Fig. 6 implements both the TCP flow and congestion control. In details, the following variables and blocks are shown:

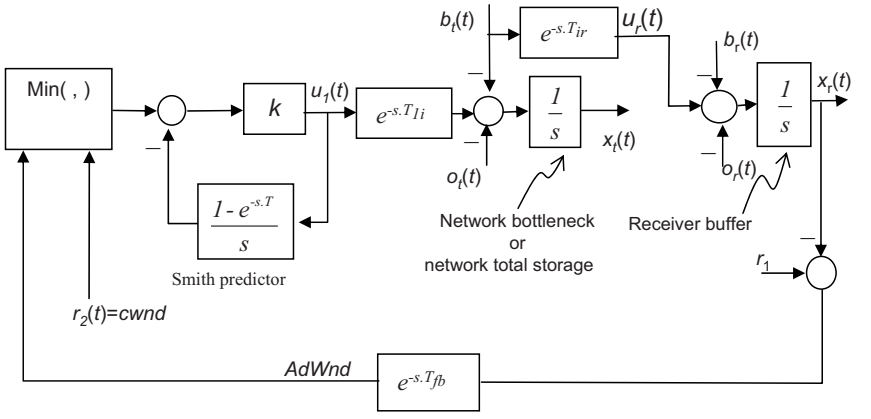



**Fig. 6.** Functional block diagram of the TCP flow and congestion control

1. The receiver queue length $x_r(t)$ and the receiver capacity $r_1(t)$ provide the term $r_1(t) - x_r(t)$ (i.e. the *Advertised Window*), which reaches the sender after the propagation time $T_{fb}(t)$ that is modeled in the Laplace domain by the transfer function $e^{-sT_{fb}}$;
2. The set point $r_2(t)$ represents a threshold for the total network storage, which is modeled by the queue $x_t(t)$;
3. The minimum block takes the minimum between the *Advertised Window* and $r_2(t)$;
4. Delays $T_{1i}$ and $T_{ir}$ model the time delay from the sender to the generic node $i$ and from the node $i$ to the receiver, respectively; the forward delay from the sender to the receiver is $T_{fw} = T_{1i} + T_{ir}$;
5. The controller transfer function

$$G(s) = \frac{k}{1 + \frac{k}{s}(1 - e^{-sT})} \tag{12}$$

which contains the proportional gain $k$ and the Smith predictor $\frac{(1-e^{-sT})}{s}$, where $T$ is the round trip time sum of the forward delay $T_{fw}$ and the backward delay $T_{fb}$. The role of the Smith predictor is to overcome the delay $T$, which is inside the feedback loop and is harmful for the stability of the closed-loop control system [28]. Notice that the buffer $x_t$ in Fig. 6 can model both the total network storage of packets but also it can model the generic buffer $x_i$ that is the bottleneck of the TCP connection at time $t$; moreover, a moving bottleneck is easily captured by the model through delays $T_{1i}$ and $T_{ir}$ where $i$ is the generic moving bottleneck.

In order to show that the block diagram in Fig. 6 models the TCP/IP flow and congestion control, first we will assume that the bottleneck is located at the receiver and then that the bottleneck is located inside the network.

### 4.1 Modelling the TCP Flow Control

By assuming that the bottleneck is at the receiver, it results:

$$min(Adwnd, r_2(t)) = Adwnd,\ u_r(t) = u_1(t - T_{fw}) \text{ and } o_t(t) = 0$$

In other words, the connection is constrained by the receiver, and the input rate reaches the receiver after the forward delay , that is $b_r(t) = u_1(t - T_{fw})$. Under these conditions, the block diagram in Fig. 6 can be transformed into the block diagram in Fig. 7 in order to model the TCP flow control.

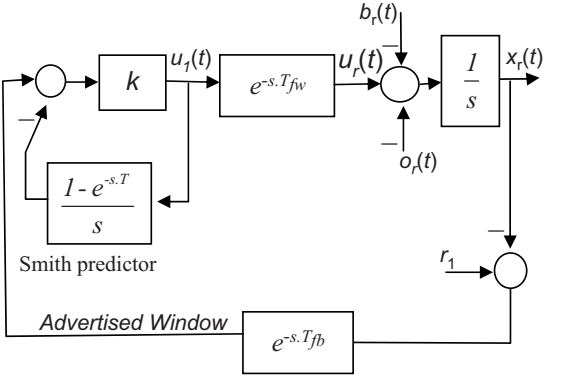



**Fig. 7.** Functional block diagram of the TCP flow control

The following proposition can be shown.

**Proposition 1** *The Smith controller (12) implements the TCP flow control Equation 1.*

**Proof:** To derive the input rate $u_1(t)$ computed by the TCP sender we use standard Laplace techniques, that is, we compute the Laplace transform of the input rate:

$$U_1(s) = [R_1(s) - X_r(s)]e^{-sT}\frac{k}{1 + k\frac{1-e^{-sT}}{s}}$$

that can be written as

$$U_1(s) = -kU_1(s)\frac{1 - e^{-sT}}{s} + k[R_1(s) - X_r(s)]e^{-sT_{fb}}$$

By transforming back to time domain it results:

$$\frac{u_1(t)}{k} = r_1(t - T_{fb}) - x_r(t - T_{fb}) - \int_{t-T}^{t} u_1(\tau)d\tau \tag{13}$$

By considering that

$$r_1(t - T_{fb}) - x_r(t - T_{fb}) = AdvertisedWindow$$

and that

$$\int_{t-T}^{t} u_1(\tau)d\tau = OutstandingPackets$$

Equation 13 reduces to the classic window-based flow control (i. e. to the Equation 1), where $W = u_1(t)/k$. By considering that $W = u_1(t)/k$ relates the rate and the window of a window-based control, it results $1/k = T$. This concludes the proof.

Notice that the *outstanding packets* automatically take into account the round trip time $T$ that in general can be time varying due to queuing delays. In the case of flow control $T$ is constant since it is assumed that there is no congestion inside the network which implies that network queuing delay is zero and round trip time is pure propagation delay.

### 4.2 The TCP Congestion Control

By assuming that the bottleneck is localized inside the network, we can ignore the outer feedback loop. Therefore, block diagram in Fig. 6 can be transformed into the equivalent one shown in Fig. 8, which models the TCP congestion control. The following proposition can be shown:

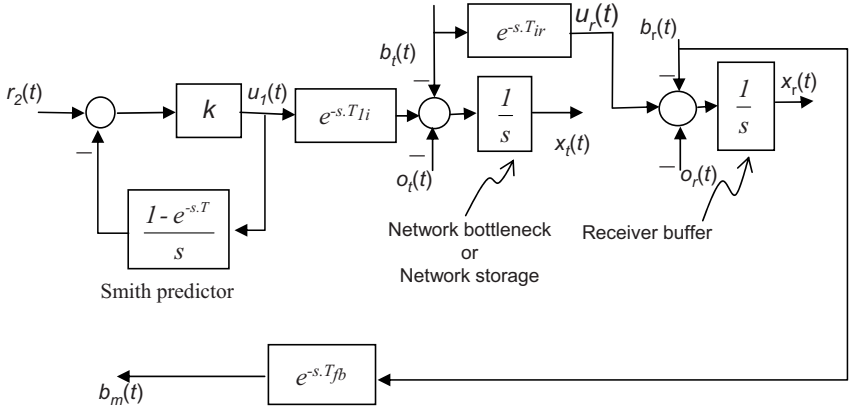



**Fig. 8.** Functional block diagram of the TCP congestion control

**Proposition 2** *The Smith controller (12) implements the TCP congestion control Equation (2).*

**Proof:** By assuming that the bottleneck is located inside the network, there results:

$$min(Adwnd, r_2(t)) = r_2(t)$$

From Fig. 8, it results that the output of the Smith predictor in the Laplace domain is:

$$Q(s) = U_1(s)\frac{1 - e^{-sT}}{s}$$

which transformed back to time domain turns out :

$$q(t) = \int_{t-T}^{t} u_1(\tau)d\tau$$

and, by considering Eq. 11, it turns out

$$q(t) = x_t(t) = outstandingpackets$$

Therefore the output of the controller is:

$$u_1(t) = k(r_2(t) - outstandingpackets) \tag{14}$$

that can be rewritten as

$$\frac{u_1(t)}{k} = r_2(t) - outstandingpackets \tag{15}$$

Equation 15 is the classic window-based congestion control equation 2, where $W = u_1(t)/k$, and $r_2(t) = cwnd$. This concludes the proof.

**Remark 1:** It should be noted that 14 and 15 are the rate-besed and window-based versions of the same control equation.

## 5 Modeling Reno or Westwood TCP

In this section we show that the dynamic model depicted in Fig. 6 is able to model successful variants of TCP congestion control, such as for example Tahoe/Reno [21] or the recent Westwood TCP [29]. Other TCP variants, such as Vegas or Santa Cruz, could also be easily modeled in the same unified framework proposed in this paper.

We have seen that the congestion control algorithm aims at estimating the available bandwidth using a probing mechanism. The classic TCP probing mechanism, which is currently used in all successful variants of the TCP such as Tahoe/Reno, New Reno or Westwood, comprises two mechanisms: the slow-start phase, which exponentially increase the congestion window up to the *ssthresh*, and the congestion avoidance phase which linearly increase the *cwnd* when $cwnd \geq ssthresh$. Now we show that both these mechanisms can be modeled in the control theoretical framework reported in 6 by properly shaping the controller input $r_2(t) = cwnd$.

### 5.1 The Reno Algorithm

By considering the Proposition 2, where it has been shown that $cwnd = r_2(t)$, the TCP Reno slow-start phase can be modeled by setting the reference input $r_2(t)$ as follows:

$$r_2(t) = r_0 \cdot 2^{\frac{t}{T}} \text{ while } r_2(t) < ssthresh$$

where the initial window $r_0$ is generally equal to 1 or 2 [1]. TCP Reno enters the congestion avoidance phase when $r_2(t) = ssthresh$ at $t_1 = Tlog_2(sstresh - r_0)$. This phase can be modelled by setting the reference input $r_2(t)$ as follows:

$$r_2(t) = ssthresh + \frac{t-t_1}{T} \text{ when } r_2(t) \geq ssthresh$$

The TCP probing phase ends when 3 DUPACKs are received or a timeout happens at $t = t_k$, which indicate that the network capacity has been hit. In these cases the *cwnd* behavior can be modeled using the following settings for $r_2(t)$:

***After a timeout at*** $t_k$
$ssthresh \leftarrow max(\frac{r_2(t)}{2}, r_0)$
$r_2(t) \leftarrow r_0$
$r_2(t) \leftarrow r_0 \cdot 2^{\frac{t-t_k}{T}}$ if $r_2(t) < ssthresh$
$r_2(t) \leftarrow ssthresh + \frac{t-t_k}{T}$ if $r_2(t) \geq ssthresh$

***After 3 DUPACKs at*** $t_k$
$ssthresh \leftarrow max(\frac{r_2(t)}{2}, r_0)$
$r_2(t) \leftarrow ssthresh + \frac{t-t_k}{T}$

### 5.2 The Westwood algorithm

TCP Westwood employs the same probing mechanism of Reno. It differs from Reno because of the behavior after congestion. In fact, Westwood sets the *cwnd* and *ssthresh* using an end-to-end estimate of the network bandwidth $b_m(t_k)$ available at time of congestion $t_k$. In particular, the Westwood TCP window behavior after congestion can be modeled as follows:

***After a timeout at*** $t_k$
$ssthresh \leftarrow max(r_0, b_m(t_k) \cdot RTT_{min})$
$r_2(t) \leftarrow r_0$
$r_2(t) \leftarrow r_0 \cdot 2^{\frac{t-t_k}{T}}$ if $r_2(t) < ssthresh$
$r_2(t) \leftarrow ssthresh + \frac{t-t_k}{T}$ if $r_2(t) \geq ssthresh$

***After 3 DUPACKs at*** $t_k$
$ssthresh \leftarrow max(r_0, b_m(t_k) \cdot RTT_{min})$
$r_2(t) \leftarrow ssthresh + \frac{t-t_k}{T}$ if $r_2(t) \geq ssthresh$

## 6 Designing a TCP-friendly rate control

The TCP implements a window-based control equation. As a consequence, it sends packets in bursts. Burstiness degrades the performance of the control algorithm and makes it unsuitable for applications such as audio or video streaming where a relative smooth rate is of importance [5, 12].

To overcome the mentioned problem, rate-based control algorithms have been proposed [17, 19]. A key requirement that a new rate-based congestion control algorithm must satisfy is friendliness toward TCP, i.e. it must share network bandwidth with TCP fairly. The idea of TFRC is to enforce and guarantee friendliness by using the TCP long-term throughput equation to compute the input rate [32]. This approach could reveal to be unfriendly since a TFRC sets instantaneously the rate that, in similar conditions, a TCP flow would reach only in long-term conditions [17]. This section sketches how the analysis developed in this paper can be used to design a TCP friendly rate-based congestion control. For that purpose we start from the TCP congestion control equation in rate-based form, which is

$$u_1(t) = k(cwnd - outstanding) \tag{16}$$

In order to propose a rate-based congestion control that is friendly to TCP, we can build on results of Proposition 1 and 2 to design the *exact* rate-based version of the TCP Reno or Westwood congestion control. In particular, by properly setting $r_2(t) = cwnd$ in Eq. 16, it is possible to match the corresponding increasing/decreasing mechanism of Reno or Westwood TCP.

### 6.1 Exponential probing corresponding to the slow-start phase

This phase aims at quick probing of network capacity and corresponds to Reno slow-start phase. It is obtained by setting the controller input $r_2(t) = cwnd$ in Eq. 16 as follows:

$$cwnd = r(t_0) \cdot 2^{\frac{t-t_0}{T}} \text{ when } cwnd \leq ssthresh$$

where $t_0$ is the starting time, $T$ is the round trip time and $r(t_0)$ is equal to one or two [1].

### 6.2 Linear probing corresponding to the congestion avoidance phase

This phase aims at gentle probing of network capacity and corresponds to Reno congestion avoidance phase. It is obtained by setting the controller input $r_2(t)$ in Eq. 16 as follows:

$$cwnd = ssthresh + \frac{t-t_k}{T} \text{ if } r_2(t) \geq ssthresh$$

### 6.3 Shrinking phase

After a congestion episode (timeout or DUPACK) at $t = t_k$, the controller input $r_2(t) = cwnd$ in (14) can be set by following Reno or Westwood TCP behavior as follows:

**After a timeout at** $t = t_k$
$ssthresh \leftarrow max(r_0, \frac{cwnd(t_k)}{2})$ /*Reno behavior
or
$ssthresh = max(r_0, b_m(t_k) \cdot RTTmin)$ /*Westwood behavior
$r_2(t) \leftarrow r_0$
$cwnd = r_2(t)$
*Enter the probing phase*

**After 3 DUPACKs at** $t_k$
$ssthresh \leftarrow max(r_0, \frac{cwnd(t_k)}{2})$ /* Reno behavior
or
$ssthresh = max(r_0, b_m(t_k) \cdot RTTmin)$ /*Westwood behavior
$r_2(t) \leftarrow ssthresh + \frac{t-t_k}{T}$
$cwnd = r_2(t)$

In [17] a detailed implementation of the rate-based algorithm sketched in this section has been called Adaptive Rate Control (ARC) and tested against the Transport Friendly Rate Control (TFRC) [5, 19] and the standard TCP using Network Simulator-2 [31]. Simulations results have shown that TFRC is not friendly toward TCP; on the other hand, ARC improves intra-protocol fairness and is friendly toward TCP because it mimics the dynamic behavior of the TCP and not only the long-term behavior.

## 7 Why a standard PID controller cannot be used in data networks

The PID controller is by far the most common control algorithm and performs satisfactorily well in many practical cases [3]. People could be tempted to use a PID controller also in data networks due to its simplicity. In this section we show that a standard PID cannot satisfactorily be employed for congestion control in data networks. In fact, congestion control in data networks reduces to control a process that is an integral mode with a time-delay. In order to provide stability for the closed-loop system using a PID, it turns out that it is necessary to set a low gain loop that turns out an unacceptable sluggish system.

To start the discussion we consider the simple proportional controller shown in Fig. 9(a) and we study the stability of the closed-loop system by plotting the polar diagram of the loop gain $\frac{k}{s}e^{sT_3}$ which is shown in Fig. 9(b).

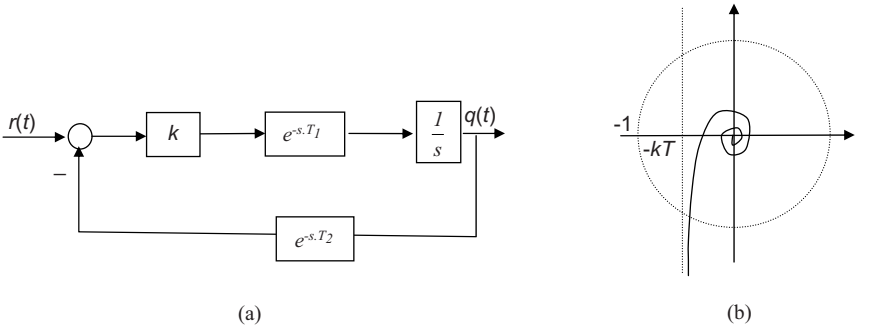



**Fig. 9.** (a) Closed-loop proportional controller; (b) Nyquist plot of the loop gain

The crossover frequency $\omega$ for which $\left|\frac{k}{j\omega}e^{-j\omega}\right| = 1$ is $\omega_0 = k$. By invoking the Nyquist stability criterion, it results that the closed loop system is stable if the process phase margin is positive, which gives the stability condition

$$k < k_u = \pi/2T \tag{17}$$

where $k_u$ is the ultimate gain. The inequality (17) states that the gain k must be chosen lower and lower with increasing round trip delay $T$, which turns out a too sluggish closed loop system in the presence of connections with large delays.

The use of a PD controller does not help much [3]. In fact, the open-loop transfer function is now: $\frac{k(1+s\tau_D)}{s}e^{-sT}$. The crossover frequency is now $\omega_1 = k/\sqrt{(\overline{1-(k\tau_D)^2} > \omega_0}$. Thus, even though the derivative action adds the positive contribute $\arctan(\omega_1\tau_D)$ to the phase margin, it happens that at the new cross-over frequency $\omega_1$ the negative contribute to the phase margin due to the time delay is now augmented of $(\omega_1 - \omega_2)T$. Thus, in the presence of large delay $T_3$, the lead action of a PD controller may not be useful or may be even pejorative of the stability margin because it may happen that $(\omega_1 - \omega_2)T > \arctan(\omega_1\tau_D)$. Finally, the integral action of the standard PID controller is surely not recommended for the system we are considering because it would reduce the phase margin of $\pi/2$ at any frequency thus making the system stability even more critical.

To provide a further insight, we compare a proportional controller with respect to a proportional controller plus Smith predictor using computer simulations. We consider a connection with $RTT = 100$ units of time. From Eq. 17, the ultimate gain is $k_u = 0.01571$ and the corresponding oscillating period is $T_u = 4RTT = 400$. Following the Ziegler-Nichols rules, we choose the proportional gain $k = k_u/2 = 0.00785$. We have also tried the Ziegler-Nichols rules for tuning a PI or a PID controller and we have found that, in this case, they do not provide a stable system, which confirms the difficulty of controlling a system with large delays using a *PI* or a *PID* controller [3]. The reference signal $r(t) = 500 \cdot 1(t)$ is chosen, which corresponds to a queue threshold of 500 packets. We assume an available bandwidth of 2 pack-

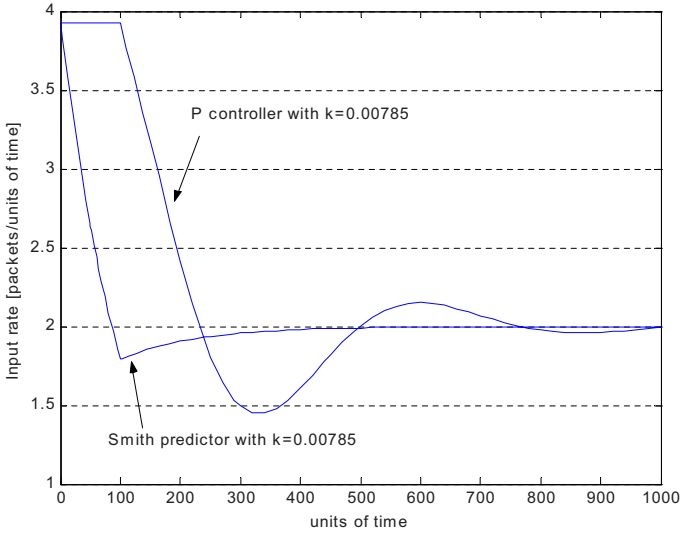



**Fig. 10.** Dynamics of the input rate

ets/units of time. Fig. 10 reports the dynamics of the input data rate obtained using a proportional controller with $k = 0.00785$ and a proportional controller $k = 0.00785$ with dead-time compensation. The input rate obtained with dead-time compensation reaches the steady state value of 2 packets/units of time faster and with smaller oscillations. Fig. 11 shows that dead-time compensation provides much smaller queuing, i.e. much smaller delays.

## 8 Conclusions

This paper has proposed a classical control theoretic approach to model the closed-loop dynamic of TCP flow and congestion control. It has been shown that a proportional controller ($P$) plus a Smith predictor ($SP$) provides an exact model of the Internet flow and congestion control and that successful TCP congestion control algorithms, such as classic Reno or recent Westwood TCP, can be modeled in a unified control framework by properly shaping the controller reference signal. The utility of the proposed model has been illustrated by sketching criteria to design a novel TCP friendly rate control. Finally it has been shown that a $PID$ controllers cannot be used in data networks because they would provide a too sluggish closed-loop behavior.

## References

[1] M. Allman, S. Floyd, and C. Partridge. Increasing initial TCP's initial window. RFC 3390, October 2002.

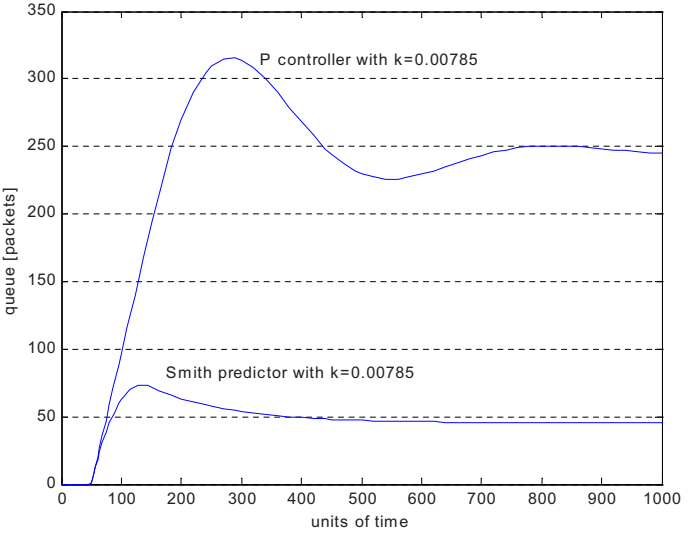



**Fig. 11.** Dynamics of the queue

[2] M. Allman, V. Paxson, and W. R. Stevens. TCP congestion control. RFC 2581, April 1999.

[3] K. Astrom and T. Hagglund. *PID Controllers: Theory, Design, and Tuning*. ISA, 1995.

[4] H. Balakrishnan, V. N. Padmanabhan, and S. Seshan andR. H. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, December 1997.

[5] D. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker. Dynamic Behavior of Slowly-Responsive Congestion Control Algorithms. In *ACM SIGCOMM 2001*, pages 263–273, UC San Diego, CA, USA, August 2001.

[6] L. S. Brakmo and L. Peterson. TCP Vegas: End-to-end congestion avoidance on a global Internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–1480, October 1995.

[7] V. C. Cerf and R. E. Kahn. A Protocol for packet Network Interconnections. *IEEE Transactions on Communications*, 22(5):637–648, May 1974.

[8] H. M. Chaskar, T. V. Lakshman, and U. Madhow. TCP over wireless with link level error control: Analysis and design methodology. *IEEE/ACM Transactions on Networking*, 7(5):605–615, October 1999.

[9] D. Clark. The design philosophy of the DARPA Internet protocols. In *ACM Sigcomm '88*, pages 106–114, Stanford, CA, USA, August 1988.

[10] M.Handley D. Katabi and C. Rohrs. Internet congestion control for high bandwidth-delay product networks. In *ACM Sigcomm '02*, Pittsburgh, Usa, August 2002.

[11] A. Dell'Aera, L. A. Grieco, and S. Mascolo. Linux 2.4 Implementation of Westwood+ TCP with Rate-halving: A Performance Evaluation over the Internet. In *IEEE International Conference on Communications (ICC)*, Paris, France, June 2004.

[12] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast application. In *ACM SIGCOMM 2000*, pages 43–56, Stockholm, Sweden, August 2000.

[13] S. Floyd and T. Henderson. New Reno modification to TCP's fast recovery. RFC 2582, April 1999.

[14] M. Gerla, R. Locigno, S. Mascolo, and R. Weng. Generalized window advertising for TCP congestion control. *European Transactions on Telecommunications*, (6), November/December 2002.

[15] L. A. Grieco and S. Mascolo. Westwood TCP and easy RED to Improve Fairness in High Speed Networks. In *IFIP/IEEE Seventh International Workshop on Protocols For High-Speed Networks*, pages 130–146, Berlin, Germany, April 2002.

[16] L. A. Grieco and S. Mascolo. End-to-End Bandwidth Estimation for Congestion Control in Packet Networks. In *Second International Workshop, QoS-IP 2003*, pages 645–658, Milano, Italy, February 2003.

[17] L.A. Grieco and S. Mascolo. Adaptive Rate Control for Streaming Flows over the Internet. *ACM Multimedia Systems Journal*, 4, April 2004.

[18] L.A. Grieco and S. Mascolo. Performance evaluation and comparison of Westwood+ TCP, New Reno and Vegas TCP congestion control. *ACM Computer Communication Review*, 34(2), 2004. April.

[19] M. Handley, S. Floyd, J. Padhye, and J. Widmer. TCP friendly rate control (TFRC): Protocol specification. RFC 3448, January 2003.

[20] C. V. Hollot, V.Misra, Donald F. Towsley, and Wei-Bo Gong. Analysis and design of controllers for AQM Routers supporting TCP flows. *IEEE Trans. Automatic Control*, 47(6):945–959, June 2002.

[21] V. Jacobson. Congestion avoidance and control. In *ACM Sigcomm '88*, pages 314–329, Stanford, CA, USA, August 1988.

[22] V. Jacobson. Berkeley TCP evolution from 4.3-Tahoe to 4.3 Reno. In *18th Internet Engineering Task Force*, University of British Colombia, Vancouver, BC, September 1990.

[23] V. Jacobson, R. Braden, and D. Borman. TCP extensions for high performance. RFC 1323, May 1992.

[24] Cheng Jin, David X. Wei, and Steven H. Low. Fast TCP: Motivation, architecture, algorithms, performance. Proc. of Infocom, March 2004.

[25] F. Kelly. Mathematical modeling of the internet. In *Fourth International Congress on Industrial and Applied Mathematics*, Edinburgh, July 1999.

[26] S. Q. Li and C. Hwang. Link capacity allocation and network control by filtered input rate in high speed networks. *IEEE/ACM Transactions on Networking*, 3(1):10–25, February 1995.

[27] S. H. Low, F. Paganini, and J. C. Doyle. Internet congestion control. *IEEE Contr. Syst. Mag.*, 22(1):28, January 2002.

[28] S. Mascolo. Congestion control in high-speed communication networks using the Smith principle. *Automatica, Special Issue on Control methods for communication networks*, 35:1921–1935, December 1999.

[29] S. Mascolo, C. Casetti, M. Gerla, M. Sanadidi, and R. Wang. TCP Westwood: End-to-End Bandwidth Estimation for Efficient Transport over Wired and Wireless Networks. In *ACM Mobicom 2001*, Rome, Italy, July 2001.

[30] S. Mascolo, L. A. Grieco, R. Ferorelli, P. Camarda, and G. Piscitelli. Performance Evaluation of Westwood+ TCP Congestion Control. *Performance Evaluation, Special Issue with Selected papers from Globecom 02*, 55(1):93–111, January 2004.

[31] Ns-2. network simulator, 2002.

[32] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. In *ACM Sigcomm '98*, pages 303–314, Vancouver BC, Canada, 1998.

[33] Christina Parsa and J. J. Garcia-Luna-Aceves. Improving TCP congestion control over internets with heterogeneous transmission media. In *Proceedings of the 7th IEEE International Conference on Network Protocols (ICNP)*, 1999.

[34] L. Peterson, B. Davie, and D. Clark. *Computer Networks a Systems Approach*. Morgan Kaufmann, 2 edition, 2000.

[35] W. Stevens. *TCP/IP illustrated*, volume 1. Addison Wesley, Reading, MA, July 1997.

[36] P. Varaiya and J. Walrand. *HighPerformance Communications Networks*. Morgan Kaufmann Publishers, 2 edition, January 2000.

[37] C. Villamizar and C. Song. High performance TCP in ANSNET. *ACM Computer Communication Review*, 24(5):45–60, October 1995.

# Saturated Controller Design of an ABR Explicit Rate Algorithm for ATM Switches*

Sophie Tarbouriech[1], Marco Ariola[2], and Chaouki T. Abdallah[3]

[1] LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse cedex 4, FRANCE
tarbour@laas.fr
[2] Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, Napoli, ITALY ariola@unina.it
[3] Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131, USA chaouki@eece.unm.edu

## 1 Introduction

The transmission of multimedia traffic on the broadband integrated service digital networks (B-ISDN) has created the need for new transport technologies such as Asynchronous Transfer Mode (ATM). Briefly, because of the variability of the multimedia traffic, ATM networks seek to guarantee an end-to-end quality of service (QoS) by dividing the varying types of traffic (voice, data, etc.) into short, fixed-size cells (53 bytes each) whose transmission delay may be predicted and controlled. ATM is thus a *Virtual Circuit* (VC) technology which combines advantages of circuit-switching (all intermediate switches are alerted of the transmission requirements, and a connecting circuit is established) and packet-switching (many circuits can share the network resources). In order for the various VC's to share network resources, flow and congestion control algorithms need to be designed and implemented. The congestion control problem is solved by regulating the input traffic rate. In addition, because of its inherent flexibility, ATM traffic may be served under one of the following service classes:

1. The *constant bit rate* (CBR) class: it accommodates traffic that must be received at a guaranteed bit rate, such as telephone conversations, video conferencing, and television.
2. The *variable bit rate* (VBR): it accommodates bursty traffic such as industrial control, multimedia e-mail, and interactive compressed video.
3. The *available bit rate* (ABR): it is a best-effort class for applications such as file transfer or e-mail. Thus, no service guarantees (transfer delay) are required, but the source of data packets controls its data rate, using a feedback signal provided by switches downstream which measure the congestion of the network.

* This works has been supported by an international agreement CNRS (France) - NSF (USA). The research of C.T. Abdallah was partially supported by NSF INT-9818312

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 159–176, 2004.

Due to the presence of this feedback, many classical and advanced control theory concepts have been suggested to deal with the congestion control problem in the ATM/ABR case [2, 12].

4. The *unspecified bit rate* (UBR): it uses any leftover capacity to accommodate applications such as e-mail.

Note that the CBR and VBR service categories, a traffic contract is negotiated at the initial stage of the VC setup, and maintained for the duration of the connection. This contract will guarantee the following QoS parameters:

1. Minimum cell rate (MCR),
2. Peak cell rate (PCR),
3. cell delay variation (CDV),
4. maximum cell transfer delay (maxCTD), and
5. cell loss ratio (CLR).

This then forces CBR and VBR sources to keep their rate constant regardless of the congestion status of the network. The ABR sources on the other hand, are only required to guarantee an MCR and an PCR, and thus can adjust their rates to accommodate the level available after all CBR and VBR traffic has been accommodated. In order to avoid congestion, the ATM Forum adopted a rate-based ABR control algorithm as opposed to a credit approach whereby the number of incoming cells as opposed to their rate is controlled [8]. This chapter will then concentrate on the ABR service category since ABR sources are the ones to adjust their rates using explicit network feedback. In the original ATM forum specification, an ATM/ABR source is required to send one cell called a resource management (RM) cell for every 32 data cells. Switches along the path from the source to the destination then write into the RM cell their required data rate to avoid congestion. The destination switch then has information about the minimum rate required by all switches along the VC which is then relayed back to the ATM/ABR source as a feedback signal which serves to adjust its own data rate.

The earliest control algorithms for ABR consisted of setting a binary digit in the RM cell by any switch along the VC when its queue level exceeds a certain threshold [2]. This was then shown to cause oscillations in the closed-loop system. Other controllers were then suggested by various authors [5, 6], to address this problem. Most of these controllers are either complex or did not guarantee the closed-loop stability (in a sense defined later).

In addition, one of the limiting factors of these earlier proposed controllers was that the ABR bandwidth needed to be known in the implementation of the control algorithm. This however poses a problem in multimedia applications where the ABR bandwidth is bursty and is effectively the remaining available bandwidth after the CBR and VBR traffic have been accommodated. In [12] this particular issue was dealt with using a Smith predictor which then considered the available ABR bandwidth as an unknown disturbance. While this controller had many desirable properties, it only guaranteed stability in an appropriately defined sense but had no optimality guarantees. In addition, the delays encountered along with the number of ABR

sources were assumed known, although the earlier tech report [8] did not require the delays to be exactly known. In [11], robust controllers were designed when both the number of ABR sources and the delays were uncertain.

In this chapter, we consider a discrete-time model for an ATM/ABR switch and source which was presented in [5] and attempt to control the resulting nonlinear system. The technique developed is based on the use of Finsler's lemma and a generalized sector nonlinearity description. Considering quadratic Lyapunov functions, the use of Finsler's lemma allows us to express stability and invariance conditions without explicitly substituting the dynamic system equations into the Lyapunov function. Such an approach corresponds to weighing the dynamic system equations through multipliers which represent additional variables [14], [13]. The introduction of these new variables increases the degrees of freedom in the problem, and relaxes the conditions of applicability..

**Notations.** $\Re^+$ is the set of non-negative real numbers. For any vector $x \in \Re^n$, $x \succeq 0$ means that all the components of $x$, denoted by $x_{(i)}$, are nonnegative. For two vectors $x, y$ of $\Re^n$, the notation $x \succeq y$ means that $x_{(i)} - y_{(i)} \geq 0, \forall i = 1, \ldots, n$. $A_{(i)}$ denotes the $i$th row of matrix $A$. For two symmetric matrices, $A$ and $B$, $A > B$ means that $A - B$ is positive definite. $A'$ denotes the transpose of $A$. $1_m \stackrel{\triangle}{=} [1 \ldots 1]' \in \Re^m$. $I_m$ denotes the $m$-order identity matrix. $\lambda_{\max}(P)$ and $\lambda_{\min}(P)$ denote respectively the maximum and minimum eigenvalues of matrix $P$. $\mathrm{sat}_{\{A_1,A_2\}}(\nu)$ denotes the scalar saturation function defined as:

$$\mathrm{sat}_{\{A_1,A_2\}}(\nu) = \begin{cases} A_1 \text{ if } \nu < A_1 \\ \nu \quad \text{if } A_1 \leq \nu \leq A_2 \\ A_2 \text{ if } \nu > A_2 \end{cases} \tag{1}$$

## 2 The network model and the control problem

### 2.1 The network model

As in [5] and [1], we consider the closed-loop discrete-time system:

$$Q(n+1) = \mathrm{sat}_{\{0,B\}}(Q(n) + \lambda(n - d^f) - \mu(n)) \tag{2}$$

$$R(n+1) = \mathrm{sat}_{\{0,C\}}(R(n) - \sum_{j=0}^{J} \alpha_j (Q(n-j) - Q_0) - \sum_{k=0}^{K} \beta_k R(n-k)) \tag{3}$$

where $R$ denotes the explicit rate (ER) computed by a switch for a given $VC$ and $Q$ denotes the buffer occupancy of this $VC$ at the switch. Furthermore, $\lambda(n)$ and $\mu(n)$ are respectively, the rate at the ABR source and the service rate at the switch during the interval $[n, n+1]$, $d^f$ is the forward delay from the source to the switch and the saturation level $B$ represents the buffer size. The saturation level $C$ is the maximum $ER$ and $Q_0$ is the desired buffer occupancy. The ABR source is greedy if the source's rate $\lambda(n - d^f)$ is equal to $R(n+1-d)$, where $d = d^f + d^b$ is the round trip delay ($d^b$ is the feedback delay from the switch back to the source).

Note that equation (2) describes the plant dynamics, i.e. the state of the buffer, and equation (3) defines a particular controller structure. In (2)-(3), the numbers $J$, $K$ and the parameters $\alpha_j$, $\beta_k$ have to be found such that closed-loop stability and some performance levels are attained [4]. Such objectives will be more precisely described later in the chapter.

In [4], the author showed that when considering the linearized model of system (2)-(3) it is sufficient to consider $J = 1$ and $K = d$ in order to completely place the closed-loop poles. Hence, we consider (2)-(3) with $J = 1, K = d$ but without removing the saturation functions as was done in [5]. Moreover we assume that $\lambda(n - d^f)$ is equal to $R(n+1-d)$.

Let us now define both the extended state and disturbance vectors

$$X(n) = \begin{bmatrix} Q(n) \\ Q(n-1) \\ R(n) \\ R(n-1) \\ \vdots \\ R(n+1-d) \\ R(n-d) \end{bmatrix} \in \Re^{d+3} \; ; \; W(n) = \begin{bmatrix} \mu(n) \\ Q_0 \end{bmatrix} \in \Re^2 \tag{4}$$

and define the following matrices

$$\mathbb{A} = \begin{bmatrix} 0 & 0 & \dots & \dots & 0 & 0 & 0 \\ 1 & 0 & \dots & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & \dots & 0 \\ \vdots & \dots & \dots & \ddots & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \dots & \ddots & \dots & \vdots \\ 0 & \dots & \dots & \dots & \dots & 1 & 0 \end{bmatrix} \; ; \; \mathbb{B}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \mathbb{B}_{11} & \mathbb{B}_{12} \end{bmatrix} \; ; \; \mathbb{B}_2 = \begin{bmatrix} -1 & 0 \\ 0 & \alpha_0 + \alpha_1 \end{bmatrix}$$

$$\mathbb{G} = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 & 1 & 0 \\ -\alpha_0 & -\alpha_1 & 1-\beta_0 & -\beta_1 & \dots & \dots & -\beta_d \end{bmatrix} = \begin{bmatrix} \mathbb{R} \\ \mathbb{K} \end{bmatrix} \tag{5}$$

Hence, from (4) and (5), the system (3) reads:

$$X(n+1) = (\mathbb{A} + \mathbb{B}_1\mathbb{G})X(n) + \mathbb{B}_1\mathbb{B}_2 W(n) + \mathbb{B}_1\Psi(X(n), W(n)) \tag{6}$$

with

$$\Psi(X(n), W(n)) = \mathrm{sat}_{\{0,B\},\{0,C\}}(\mathbb{G}X(n) + \mathbb{B}_2 W(n)) - (\mathbb{G}X(n) + \mathbb{B}_2 W(n))$$

where

$$\mathrm{sat}_{\{0,B\},\{0,C\}}(\nu) = \begin{bmatrix} \mathrm{sat}_{\{0,B\}}(\nu_{(1)}) \\ \mathrm{sat}_{\{0,C\}}(\nu_{(2)}) \end{bmatrix} \tag{7}$$

Furthermore, by definition, $\Psi(X(n), W(n))$, which will be simply denoted by $\Psi(n)$ in the sequel, is a decentralized dead-zone nonlinearity and satisfies the following sector condition [10]:

$$\Psi(n)'[\Psi(n)+\mathbb{E}_1 X(n)+\mathbb{E}_2 W(n)] \leq 0,\ \forall n=0,1,...,\ \text{for some } X \text{ and } W$$

The domain of suitable $X$ and $W$ is the polyhedral set $S(\mathbb{E}_1,\mathbb{E}_2)$ defined by:

$$S(\mathbb{E}_1,\mathbb{E}_2)=\left\{X\in\Re^{d+3}, W\in\Re^2; 0\preceq(\mathbb{G}-\mathbb{E}_1)X+(\mathbb{B}_2-\mathbb{E}_2)W\preceq\begin{bmatrix}B\\C\end{bmatrix}\right\}$$

It is important to note that the closed-loop matrix $\mathbb{A}+\mathbb{B}_1\mathbb{G}$ will be asymptotically stable if and only if matrix $\mathbb{K}$ is suitably chosen. Furthermore the $\mathbb{A}+\mathbb{B}_1\mathbb{G}$ matrix has an eigenvalue equal to zero, since the sum of its first and second rows is equal to the last row.

### 2.2 Problem formulation

Considering the nonlinear system (6), the problem we want to solve throughout this chapter may be summarized as follows.

**Problem 1.** Determine a matrix $\mathbb{K}$, a set of admissible initial conditions $S_0$, and a set of admissible disturbances $D_0$ such that:

1. The closed-loop matrix $\mathbb{A}+\mathbb{B}_1\mathbb{G}$ is asymptotically stable.
2. The closed-loop trajectories remain bounded for any $X(0)\in S_0$ and any admissible disturbance $W(n)\in D_0$, $\forall n$.
3. The steady state buffer occupancy is equal to the desired threshold $Q_0$.

By solving this problem, we consider that the saturation functions are effectively taken into account, that is, the nonlinear behavior of the closed-loop system is studied. The implicit objectives in Problem 1 consist of optimizing the size of the region of stability and/or the size of the region of admissible disturbances. Throughout the chapter, the criteria allowing us to optimize the size of the region of stability of the closed-loop system, and the trade-off between the stability region and the region of admissible disturbances will be discussed.

## 3 Preliminary results

It is important to note that satisfying point 3 in Problem 1 allows us to study the existence of possible equilibrium points corresponding to the case $W(n)=W_e$ where $W_e$ is a constant value.

**Lemma 1.** *[18] Suppose that there exists an equilibrium point $X_e$ for system (6). Then this equilibrium point satisfies:*

$$R_e(n-k)=R_e(n-k+1)=R_e=\mu_e(n)=\mu_e,\ k=0,1,...,d \tag{8}$$

$$Q_e(n)=Q_e(n-1)=Q_e=Q_0 \tag{9}$$

$$\sum_{k=0}^{d} \beta_k R_e = \sum_{k=0}^{d} \beta_k \mu_e = 0 \tag{10}$$

*provided that the following two conditions hold*

$$0 \leq Q_0 \leq B \tag{11}$$

$$0 \leq \mu_e \leq C \tag{12}$$

*Remark 1.* The conditions of Lemma 1 are consistent with those given in [5]. Moreover, in general $\mu_e$ is not equal to 0, therefore condition (10) implies that

$$\sum_{k=0}^{d} \beta_k = 0 \tag{13}$$

Equality (13) implies that $\beta_d$ is computed from the last $d$ entries. In other words, one can write:

$$\beta_d = -\sum_{k=0}^{d-1} \beta_k \tag{14}$$

*Remark 2.* If we do not require that $\beta_d$ verifies (14) then the equilibrium point $Q_e$ is such that

$$Q_e = Q_0 - \frac{\sum_{k=0}^{d} \beta_k \mu_e}{\alpha_0 + \alpha_1} \neq Q_0 \tag{15}$$

Hence, the steady-state buffer occupancy will be equal to $Q_e$ (i.e. different from $Q_0$). Furthermore, in this case, condition (11) reads:

$$0 \leq Q_0 - \frac{\sum_{k=0}^{d} \beta_k \mu_e}{\alpha_0 + \alpha_1} \leq B \tag{16}$$

From Remark 1, we have that the vector $\mathbb{K}$ in (5) can be written as

$$\mathbb{K} = \left[ -\alpha_0 \ -\alpha_1 \ 1-\beta_0 \ -\beta_1 \ \ldots \ \ldots \ \sum_{k=0}^{d-1} \beta_k \right] \tag{17}$$

Thus, from Lemma 1 and Remark 1, we can consider another representation of model (2)-(3) and therefore of model (6). Towards this aim, consider the following vectors

$$Y(n) = \begin{bmatrix} Q(n) - Q_0 \\ Q(n-1) - Q_0 \\ R(n) - \mu_e \\ R(n-1) - \mu_e \\ \vdots \\ R(n-d) - \mu_e \end{bmatrix} \in \Re^{d+3} \text{ and } \nu(n) = \mu(n) - \mu_e \in \Re \tag{18}$$

which correspond to a change of variables around the equilibrium point $X_e$, and define the matrix

$$\mathbb{B}_3 = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \Re^{d+3} \tag{19}$$

From (18) and (19), the closed-loop system under consideration reads:

$$\begin{aligned} Y(n+1) &= (\mathbb{A} + \mathbb{B}_1\mathbb{G})Y(n) + \mathbb{B}_3\nu(n) + \mathbb{B}_1\Psi(Y(n), \nu(n)) \\ &= (\mathbb{A} + \mathbb{B}_{11}\mathbb{R} + \mathbb{B}_{12}\mathbb{K})Y(n) + \mathbb{B}_3\nu(n) + \mathbb{B}_1\Psi(Y(n), \nu(n)) \\ &= (\mathbb{A}_0 + \mathbb{B}_{12}\mathbb{K})Y(n) + \mathbb{B}_3\nu(n) + \mathbb{B}_1\Psi(Y(n), \nu(n)) \end{aligned} \tag{20}$$

with

$$\begin{aligned} \Psi(Y(n), \nu(n)) = \text{sat}_{\{-Q_0, B-Q_0\}, \{-\mu_e, C-\mu_e\}} &\left( \mathbb{G}Y(n) + \begin{bmatrix} -1 \\ 0 \end{bmatrix} \nu(n) \right) \\ &- \left( \mathbb{G}Y(n) + \begin{bmatrix} -1 \\ 0 \end{bmatrix} \nu(n) \right) \end{aligned} \tag{21}$$

where the saturation function is defined similarly to (7). The nonlinearity $\Psi(Y(n), \nu(n))$, simply denoted by $\Psi(n)$, satisfies the following sector condition

$$\Psi(n)' [\Psi(n) + \mathbb{E}_1 Y(n) + \mathbb{E}_3 \nu(n)] \leq 0, \ \forall n = 0, 1, ..., \tag{22}$$

for some $Y$ and $\nu$ belonging to the polyhedral set $S(\mathbb{E}_1, \mathbb{E}_3)$ defined by:

$$\begin{aligned} S(\mathbb{E}_1, \mathbb{E}_3) = \Bigg\{ &Y \in \Re^{d+3}, \nu \in \Re; \begin{bmatrix} -Q_0 \\ -\mu_e \end{bmatrix} \preceq (\mathbb{G} - \mathbb{E}_1)Y \\ &+ \left( \begin{bmatrix} -1 \\ 0 \end{bmatrix} - \mathbb{E}_3 \right) \nu \preceq \begin{bmatrix} B - Q_0 \\ C - \mu_e \end{bmatrix} \Bigg\} \end{aligned} \tag{23}$$

*Remark 3.* By considering that the behavior of the closed-loop system is linear (without taking into account the saturation limitations, i.e., $\Psi(n) = 0$) and by considering that the service rate $\mu(n)$ is constant (which corresponds to $\nu(n) = 0$), one obtains the same system studied in [5].

*Remark 4.* The constraints described in (23) will be symmetrical if and only if $B = 2Q_0$ and $C = 2\mu_e$.

*Remark 5.* If we work in the context of Remark 2, the closed-loop system defined in (20) is obtained by replacing $Q_0$ in (18) with $Q_e$, (21) and (23).

For notational simplicity, in the sequel, we define the following vectors in $\Re^2$:

$$\mathbb{B}_0 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}; \rho_1 = \begin{bmatrix} B - Q_0 \\ C - \mu_e \end{bmatrix}; \rho_2 = \begin{bmatrix} Q_0 \\ \mu_e \end{bmatrix} \tag{24}$$

## 4 Main results

The proposed conditions are expressed in terms of matrix inequalities. In order to simplify the statement of the results, we assume that the constraints in (23) are symmetrical, that is, we let $B = 2Q_0$ and $C = 2\mu_e$ and therefore $\rho_1 = \rho_2 = \rho_0$. Some comments will also be given in the non-symmetrical case (i.e. $\rho_1 \neq \rho_2$).

### 4.1 Results without structural conditions on $\mathbb{K}$

Suppose first that the gain $\mathbb{K}$ does not satisfy the structural condition $\sum_{k=0}^{d} \beta_k = 0$ and therefore is not defined from (17) but just from (5). In other words, we suppose that we are working in the context of Remarks 2 and 5.

**Proposition 1.** *If there exist matrices of appropriate dimensions $P = P' > 0$, $F$, $G$, $H$, $L$, $\mathbb{K}$, $\mathbb{E}_1$, $\mathbb{E}_3$, diagonal positive matrix $T$, positive scalars $\gamma$, $\nu_0$, $\delta$ and $\omega$ verifying*[4]

$$\begin{bmatrix} -\delta P + F(\mathbb{A}_0 + \mathbb{B}_{12}\mathbb{K}) + (\mathbb{A}_0 + \mathbb{B}_{12}\mathbb{K})'F' & \star & \star & \star \\ G(\mathbb{A}_0 + \mathbb{B}_{12}\mathbb{K}) - F' & P - G - G' & \star & \star \\ H(\mathbb{A}_0 + \mathbb{B}_{12}\mathbb{K}) + \mathbb{B}_1'F' - T\mathbb{E}_1 & -H + \mathbb{B}_1'G' & -2T + H\mathbb{B}_1 + \mathbb{B}_1'H' & \star \\ \mathbb{B}_3'F' + L(\mathbb{A}_0 + \mathbb{B}_{12}\mathbb{K}) & -L + \mathbb{B}_3'G' & L\mathbb{B}_1 + \mathbb{B}_3'H' - \mathbb{E}_3'T & L\mathbb{B}_3 + \mathbb{B}_3'L' - \omega \end{bmatrix} < 0 \tag{25}$$

$$-1 + \delta + \omega\gamma\nu_0^2 \leq 0 \tag{26}$$

$$\begin{bmatrix} P & \star \\ \mathbb{R} - \mathbb{E}_{1(1)} & \gamma\eta_{(1)}^2 \end{bmatrix} \geq 0 \tag{27}$$

$$\begin{bmatrix} P & \star \\ \mathbb{K} - \mathbb{E}_{1(2)} & \gamma\eta_{(2)}^2 \end{bmatrix} \geq 0 \tag{28}$$

$$\eta = \begin{bmatrix} Q_e \\ \mu_e \end{bmatrix} - |\mathbb{B}_0 - \mathbb{E}_3|\nu_0 \succeq 0 \tag{29}$$

*then*

[4] The symbol $\star$ stands for a symmetric block in the matrix.

*(a) the gain* $\mathbb{K}$ *is such that* $\mathbb{A}+\mathbb{B}_1\mathbb{G}$ *is asymptotically stable,*
*(b) the closed-loop trajectories remain bounded in the set* $\mathcal{E}(P,\gamma)=\{Y\in\Re^{d+3};\ Y'PY\leq\gamma^{-1}\}$ *for any admissible disturbance satisfying* $-\nu_0\leq\nu(n)\leq\nu_0$.
*(c) the steady-state occupancy is equal to* $Q_0-\frac{(\mathbb{K}e-1)\mu_e}{\mathbb{K}(e-1_{d+3})}$ *with* $e=\begin{bmatrix}0&0&1&\ldots&1\end{bmatrix}'$.

**Proof.** The satisfaction of relations (27) and (28) means that, from the definition of $\mathbb{G}$, the set $\mathcal{E}(P,\gamma)=\{Y\in\Re^{d+3};\ Y'PY\leq\gamma^{-1}\}$ is included in the set $\{Y\in\Re^{d+3};-\eta\leq(\mathbb{G}-\mathbb{E}_1)Y\leq\eta\}$. Furthermore, if relation (29) is satisfied, the bounds of the set $\begin{bmatrix}-Q_e\\-\mu_e\end{bmatrix}-(\mathbb{B}_0-\mathbb{E}_3)\nu(n)\preceq\mathbb{G}Y(n)\preceq\begin{bmatrix}Q_e\\\mu_e\end{bmatrix}-(\mathbb{B}_0-\mathbb{E}_3)\nu(n)$ under the constraints $-\nu_0\leq\nu(n)\leq\nu_0$ are well defined. By considering that any admissible disturbance satisfying $-\nu_0\leq\nu(n)\leq\nu_0$ and the vector $\eta$ defined from (29), it follows that the satisfaction of relations (27) and (28) means that the set $\mathcal{E}(P,\gamma)$ is included in the set $S(\mathbb{E}_1,\mathbb{E}_3)$ defined in (23). It follows that the sector condition (22) is satisfied for any $Y\in\mathcal{E}(P,\gamma)$ and any admissible disturbance satisfying $-\nu_0\leq\nu(n)\leq\nu_0$.

Consider the quadratic function $V(n)=Y(n)'PY(n)$ with $P=P'>0$ and the following definitions:

$$\xi\stackrel{\triangle}{=}\begin{bmatrix}Y(n)\\Y(n+1)\\\Psi(n)\\\nu(n)\end{bmatrix};\mathcal{Q}\stackrel{\triangle}{=}\begin{bmatrix}-\delta P&0&0&0\\0&P&0&0\\0&0&0&0\\0&0&0&-\omega\end{bmatrix};\mathcal{R}\stackrel{\triangle}{=}\begin{bmatrix}F\\G\\H\\L\end{bmatrix}$$
$$\mathcal{B}\stackrel{\triangle}{=}\begin{bmatrix}\mathbb{A}+\mathbb{B}_1\mathbb{G}&-I_{d+3}&\mathbb{B}_1&\mathbb{B}_3\end{bmatrix}$$

If

$$\begin{cases}V(n+1)-\delta V(n)-\omega\nu(n)'\nu(n)\leq 0\\-1+\delta+\omega\gamma\nu_0^2\leq 0\end{cases}\tag{30}$$

are satisfied, then all the closed-loop trajectories initiated in $\mathcal{E}(P,\gamma)=\{Y\in\Re^{d+3};\ Y'PY\leq\gamma^{-1}\}$ remain in it for any admissible disturbance satisfying $\nu(n)'\nu(n)=\nu(n)^2\leq\nu_0^2$. Indeed, letting $n=0$ in (30) we have that if $V(0)=Y'(0)PY(0)\leq\gamma^{-1}$ then also $V(1)=Y'(1)PY(1)\leq\gamma^{-1}$. Then we can go on and show that $V(n)=Y'(n)PY(n)\leq\gamma^{-1}$ for all $n>0$.

Then the first inequality in (30) is equivalent to:

$$\xi'\mathcal{Q}\xi\leq 0\text{ such that }\mathcal{B}\xi=0,\xi\neq 0\tag{31}$$

Now, applying Finsler's lemma [14], it follows by using inequality (22) that if there exists a matrix $\mathcal{R}$ such that

$$\xi'\left(\mathcal{Q}+\mathcal{R}\mathcal{B}+\mathcal{B}'\mathcal{R}'\right)\xi\leq\xi'\begin{bmatrix}0&\star&\star&\star\\0&0&\star&\star\\T\mathbb{E}_1&0&2T&\star\\0&0&\mathbb{E}_3'T&0\end{bmatrix}\xi\tag{32}$$

where $T$ is a positive definite, diagonal matrix, then for all $Y,\nu\in S(\mathbb{E}_1,\mathbb{E}_3)$ the first inequality in (30) is verified. Thus, provided that relations (25) and (26) are verified and $Y,\nu\in S(\mathbb{E}_1,\mathbb{E}_3)$, inequality (32), and as a consequence, (30) is verified.

Hence if all the inequalities of Proposition 1 are verified, we can conclude that:

- When $\nu \neq 0$, $\mathcal{E}(P,\gamma) \subset S(\mathbb{E}_1,\mathbb{E}_3)$ is a positively invariant set with respect to the trajectories of the closed-loop system for $Y(0) \in \mathcal{E}(P,\gamma) \subset S(\mathbb{E}_1,\mathbb{E}_3)$ and any admissible $\nu(n)$.
- When $\nu = 0$ (constant service rate), we have that for any $Y(0) \in \mathcal{E}(P,\gamma) \subset S(\mathbb{E}_1,\mathbb{E}_3)$, $V(n+1) < V(n)$ and therefore $Y(n) \to 0$ as $n \to \infty$. Hence, $X(n) \to Q_e$ which, from Remark 5, implies that point (c) is satisfied. □

In the matrix inequalities of Proposition 1, some nonlinearities appear due to the product between the multipliers $(F, G, H, L)$ and the gain $\mathbb{K}$, to the product between $T$, $\mathbb{E}_1$ and $\mathbb{E}_3$, and due to the product involving $\nu_0$. However, it is important to note that from (25), that $G+G' > P > 0$ and therefore matrix $G$ must be nonsingular. Hence, we can investigate a suitable choice of multipliers with an adequate change of variables in order to simplify a major part of these nonlinearities. The following corollary is provided in this case.

**Corollary 1.** *If there exist matrices of appropriate dimensions $V = V' > 0$, $S$, $Z$, $\mathcal{Y}$, $\mathbb{E}_3$, diagonal positive matrix $D$, positive scalars $\gamma$, $\nu_0$, $\delta$ and $\omega$ verifying*

$$\begin{bmatrix} -\delta V + \mathbb{A}_0 S' + S\mathbb{A}_0' + \mathbb{B}_{12}Z + Z'\mathbb{B}_{12}' & \star & \star & \star \\ \mathbb{A}_0 S' + \mathbb{B}_{12}Z - S & V - S - S' & \star & \star \\ D\mathbb{B}_1' - \mathcal{Y} & D\mathbb{B}_1' & -2D & \star \\ \mathbb{B}_3' & \mathbb{B}_3' & -\mathbb{E}_3 & -\omega \end{bmatrix} < 0 \tag{33}$$

$$-1 + \delta + \omega\gamma\nu_0^2 \leq 0 \tag{34}$$

$$\begin{bmatrix} V & \star \\ \mathbb{R}S' - \mathcal{Y}_{(1)} & \gamma\eta_{(1)}^2 \end{bmatrix} \geq 0 \tag{35}$$

$$\begin{bmatrix} V & \star \\ Z - \mathcal{Y}_{(2)} & \gamma\eta_{(2)}^2 \end{bmatrix} \geq 0 \tag{36}$$

$$\eta = \begin{bmatrix} Q_e \\ \mu_e \end{bmatrix} - |\mathbb{B}_0 - \mathbb{E}_3|\nu_0 \succeq 0 \tag{37}$$

*then*

*(a) the gain $\mathbb{K} = Z(S')^{-1}$ is such that $\mathbb{A} + \mathbb{B}_1\mathbb{G}$ is asymptotically stable,*
*(b) the closed-loop trajectories remain bounded in set $\mathcal{E}(S^{-1}V(S')^{-1},\gamma) = \{Y \in \mathfrak{R}^{d+3};\ Y'S^{-1}V(S')^{-1}Y \leq \gamma^{-1}\}$ for any admissible disturbance satisfying $-\nu_0 \leq \nu(n) \leq \nu_0$.*
*(c) the steady state occupancy is equal to $Q_e = Q_0 - \frac{(\mathbb{K}e-1)\mu_e}{\mathbb{K}(e-1_{d+3})}$ with $e = \begin{bmatrix} 0 & 0 & 1 & \ldots & 1 \end{bmatrix}'$.*

**Proof.** This corollary is a particular application of Proposition 1. We choose $F = G$, $H = 0$, $L = 0$, $S = G^{-1}$, $D = T^{-1}$ and $\mathbb{K} = ZG'$. By pre- and post-multiplying by $\mathrm{diag}(\begin{bmatrix} S & S & D & 1 \end{bmatrix})$ and $\mathrm{diag}(\begin{bmatrix} S' & S' & D & 1 \end{bmatrix})$, relation (25) becomes (33) with $V = SPS'$.

A similar procedure is applied to relations (27) and (28). Note that we can define $\mathbb{E}_1 = \mathcal{Y}(S')^{-1}$. □

The following comments apply to the previous results:

- As opposed to the classical approach [17] where $\mathbb{K} = ZW^{-1}$ with $W^{-1} = P$, here the state feedback gain $\mathbb{K}$ is computed from matrix $S$ ($\mathbb{K} = Z(S')^{-1}$) which need not to be positive definite.
- The matrix $V$ which allowed us to define the Lyapunov matrix $P$ ($P = S^{-1}V(S')^{-1}$) is a decision variable. Such a fact allows us to consider different matrices $V$ (and therefore implicitly different matrices $P$) in other LMI constraints, for example, to deal with the regional assignment of the poles of the closed-loop system (see [16] in the continuous-time case).

In Proposition 1 and Corollary 1, we have exhibited the gain $\mathbb{K}$ without considering the required structural conditions given in (17). Hence, if we want to impose on the gain $\mathbb{K}$ the structural condition given in (17) during the synthesis procedure, we have the three following choices:

- After computing the gain $\mathbb{K}$ from Corollary 1, the actual gain $\tilde{\mathbb{K}}$ applied is obtained as follows: keep the first $d+2$ entries of $\mathbb{K}$ and change the last one. Hence, by defining each component of $\mathbb{K}$ by $\mathbb{K}_i$, $i = 1, ..., d+3$, one gets: $\tilde{\mathbb{K}}_i = \mathbb{K}_i$, $i = 1, ...d+2$ and $\tilde{\mathbb{K}}_{d+3} = 1 - \sum_{i=3}^{d+2} \mathbb{K}_i$. This is actually the solution in the linear case obtained in [5]. However, we have to prove for this new gain $\tilde{\mathbb{K}}$, that the required domains of stability and disturbance remain valid. This may be achieved by testing the conditions of Proposition 1 with respect to this new gain and by replacing $Q_e$ by $Q_0$ in relations (27) and (29).
- Consider an optimization criterion:

$$\min \| \mathbb{K}e - 1 \|_2$$

  with $e = \begin{bmatrix} 0 & 0 & 1 & \dots & 1 \end{bmatrix}'$. Such a solution however may be numerically complex.
- A third approach as described in the next section.

Moreover, if we suppose that the suitable gain $\mathbb{K}$ with the desired structural condition is given a priori, Proposition 1 can be used to determine the admissible domain of stability in $Y$ and the admissible domain of disturbances $\nu$.

Finally, Proposition 1 presents a local stability condition for the closed-loop saturated system (20). The global asymptotic stability of the closed-loop system could be considered in the case $\nu(n) = 0$, that is in the case $\mu(n) = \mu_e$ (i.e., the service rate is supposed to be constant).

**Proposition 2.** *If there exist matrices of appropriate dimensions $V = V' > 0$, $S$, $Z$, diagonal positive matrix $D$ verifying*

$$\begin{bmatrix} -V+\mathbb{A}_0S'+S\mathbb{A}_0'+\mathbb{B}_{12}Z+Z'\mathbb{B}_{12}' & \star & \star \\ \mathbb{A}_0S'+\mathbb{B}_{12}Z-S & V-S-S' & \star \\ D\mathbb{B}_1'-\begin{bmatrix}1\\0\end{bmatrix}\mathbb{R}S'-\begin{bmatrix}0\\1\end{bmatrix}Z & D\mathbb{B}_1' & -2D \end{bmatrix} < 0 \tag{38}$$

*then*

*(a) the gain* $\mathbb{K}=Z(S')^{-1}$ *globally stabilizes the closed-loop system (20),*
*(b) the steady state occupancy is equal to* $Q_e=Q_0-\frac{(\mathbb{K}e-1)\mu_e}{\mathbb{K}(e-1_{d+3})}$ *with* $e=\begin{bmatrix}0&0&1&\dots&1\end{bmatrix}'$.

**Proof.** It suffices to consider $\mathbb{E}_1=\mathbb{G}$, with $\mathbb{E}_{1(1)}=\mathbb{R}$ and $\mathbb{E}_{1(2)}=\mathbb{K}$, which corresponds to set $\mathcal{Y}_{(1)}=\mathbb{R}S'$ and $\mathcal{Y}_{(2)}=Z$. In this case the sector condition (22), in which the part due to $\nu(n)$ is removed, is globally satisfied. Hence, one can prove that one has $V(n+1)-V(n)<0$, for all $Y(n)\in\mathfrak{R}^{d+3}$. □

### 4.2 Results with structural conditions on $\mathbb{K}$

Consider now the case where gain $\mathbb{K}$ satisfies relation (17). The first step consists in rewriting the gain $\mathbb{K}$ as:

$$\mathbb{K}=\mathbb{K}_0+\mathbb{K}_1\mathbb{C} \tag{39}$$

with

$$\begin{aligned} \mathbb{K}_0&=\begin{bmatrix}0&0&1&0&\dots&\dots&0\end{bmatrix}\in\mathfrak{R}^{1\times(d+3)}\\ \mathbb{K}_1&=\begin{bmatrix}-\alpha_0&-\alpha_1&-\beta_0&-\beta_1&\dots&-\beta_{d-1}\end{bmatrix}\in\mathfrak{R}^{1\times(d+2)}\\ \mathbb{C}&=\begin{bmatrix} 1&0&0&0&\dots&\dots&0&0\\ 0&1&0&0&\dots&\dots&0&0\\ 0&0&1&0&\dots&\dots&0&-1\\ \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\ 0&0&0&\dots&\dots&1&0&-1\\ 0&0&\dots&\dots&\dots&0&1&-1 \end{bmatrix}\in\mathfrak{R}^{(d+2)\times(d+3)} \end{aligned} \tag{40}$$

Thus from the definition (39) of $\mathbb{K}$, the problem of designing the gain $\mathbb{K}$ consists of designing the gain $\mathbb{K}_1$, since $\mathbb{K}_0$ is fixed. A very important fact is that the synthesis problem is now a static output design problem and no more a state feedback design problem due to the presence of matrix $\mathbb{C}$. Note that in this case the closed-loop system reads:

$$\begin{aligned} Y(n+1)&=(\mathbb{A}_0+\mathbb{B}_{12}\mathbb{K}_0+\mathbb{B}_{12}\mathbb{K}_1\mathbb{C})Y(n)+\mathbb{B}_3\nu(n)+\mathbb{B}_1\Psi(n)\\ &=(\mathbb{A}_1+\mathbb{B}_{12}\mathbb{K}_1\mathbb{C})Y(n)+\mathbb{B}_3\nu(n)+\mathbb{B}_1\Psi(n) \end{aligned} \tag{41}$$

A result equivalent to Proposition 1 in the current case could be stated by replacing $\mathbb{K}$ by $\mathbb{K}_0+\mathbb{K}_1\mathbb{C}$.

The presence of matrix $\mathbb{C}$ implies that an equivalent result to Corollary 1 cannot be directly stated. In order to overcome this difficulty, we consider the following modifications. First, since matrix $\mathbb{C}\in\mathfrak{R}^{(d+2)\times(d+3)}$ is full row rank ($rank(\mathbb{C})=d+2$) there always exists a matrix $N\in\mathfrak{R}^{1\times(d+3)}$ such that the following matrix

$$M = \begin{bmatrix} \mathbb{C} \\ N \end{bmatrix}$$

is non-singular. For example a trivial solution is $N = \begin{bmatrix} 0\ 0\ \ldots\ \ldots\ \ldots\ 0\ 0\ 1 \end{bmatrix} \in \Re^{1\times(d+3)}$. From this non-singular matrix $M$ we can use the following change of variables:

$$\tilde{Y}(n) = \begin{bmatrix} \mathbb{C} \\ N \end{bmatrix} Y(n) = MY(n) \tag{42}$$

which gives the augmented closed-loop system:

$$\begin{aligned} \tilde{Y}(n+1) &= M(\mathbb{A}_1 + \mathbb{B}_{12}\mathbb{K}_1\mathbb{C})M^{-1}\tilde{Y}(n) + M\mathbb{B}_3\nu(n) + M\mathbb{B}_1\Psi(n) \\ &= (\tilde{\mathbb{A}}_1 + \tilde{\mathbb{B}}_{12}\begin{bmatrix}\mathbb{K}_1\ 0\end{bmatrix})\tilde{Y}(n) + \tilde{\mathbb{B}}_3\nu(n) + \tilde{\mathbb{B}}_1\Psi(n) \end{aligned} \tag{43}$$

Hence, relative to this system our objective consists in being able to compute a state feedback $\tilde{\mathbb{K}}_1$ with a structural constraint since one wants $\tilde{\mathbb{K}}_1 = \begin{bmatrix}\mathbb{K}_1\ 0\end{bmatrix}$. Such a constraint can be linearly treated. At this stage, we can consider an extension of Corollary 1 with respect to system (43).

**Corollary 2.** *If there exist matrices of appropriate dimensions* $V = V' > 0$, $S = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}$, $Z = \begin{bmatrix} Z_1\ 0 \end{bmatrix}$, $\mathcal{Y} = \begin{bmatrix} \mathcal{Y}_1\ 0 \end{bmatrix}$, $\mathbb{E}_3$, *diagonal positive matrix* $D$, *positive scalars* $\gamma$, $\nu_0$, $\delta$ *and* $\omega$ *verifying relations of Corollary 1, in which* $\mathbb{A}_0$, $\mathbb{B}_{12}$, $\mathbb{B}_1$, $\mathbb{B}_3$ *and* $Q_e$ *have been replaced by* $\tilde{\mathbb{A}}_1$, $\tilde{\mathbb{B}}_{12}$, $\tilde{\mathbb{B}}_1$, $\tilde{\mathbb{B}}_3$ *and* $Q_0$, *respectively, then*

*(a) the gain* $\tilde{\mathbb{K}}_1 = Z(S')^{-1}$ *is such that* $\tilde{\mathbb{A}}_1 + \tilde{\mathbb{B}}_{12}\tilde{\mathbb{K}}_1$ *(an therefore matrix* $\mathbb{A} + \mathbb{B}_1\mathbb{G}$*) is asymptotically stable,*

*(b) the closed-loop trajectories remain bounded in set* $\mathcal{E}(S^{-1}V(S')^{-1}, \gamma) = \{\tilde{Y} \in \Re^{d+3};\ \tilde{Y}'S^{-1}V(S')^{-1}\tilde{Y} \leq \gamma^{-1}\}$ *for any admissible disturbance satisfying* $-\nu_0 \leq \nu(n) \leq \nu_0$.

*(c) the steady state occupancy is equal to* $Q_0$.

Of course, the structural constraints imposed on the matrices $Z$ and $S$ induce more conservatism than in the state feedback case. Indeed the problem arises from the fact that the output feedback gain is dependent on a state transformation whose choice is still an open problem. In fact, there is no particular difficulty in choosing a matrix $N$ such that $M$ is non-singular, but in no way guarantees that a feasible solution to the set of matrix inequalities (of Corollary 2) will be obtained [3]. Moreover, other approaches could be investigated to deal with this output feedback problem in a less conservative way [15].

Moreover, by the same way Proposition 2 can be extended by considering system (43) in the case where the service rate is supposed to be constant (i.e., $\nu(n) = 0$).

### 4.3 Non-symmetrical case

In the case where $\rho_1 \neq \rho_2$, that is, $B \neq 2Q_0$ (or $B \neq 2Q_e$) and $C \neq 2\mu_e$, only relations (27), (28) and (29) are modified as follows

$$\begin{bmatrix} P & \star \\ \mathbb{R} - \mathbb{E}_{1(1)} & \gamma \min\{\eta_{1(1)}^2, \eta_{2(1)}^2\} \end{bmatrix} \geq 0 \tag{44}$$

$$\begin{bmatrix} P & \star \\ \mathbb{K} - \mathbb{E}_{1(2)} & \gamma \min\{\eta_{1(2)}^2, \eta_{2(2)}^2\} \end{bmatrix} \geq 0 \tag{45}$$

$$\begin{cases} \eta_1 = \begin{bmatrix} B - Q_e \\ C - \mu_e \end{bmatrix} - |\mathbb{B}_0 - \mathbb{E}_3|\nu_0 \succeq 0 \\ \eta_2 = \begin{bmatrix} Q_e \\ \mu_e \end{bmatrix} - |\mathbb{B}_0 - \mathbb{E}_3|\nu_0 \succeq 0 \end{cases} \tag{46}$$

In this case, an interesting solution could be to study the possibility of considering an ellipsoid $\mathcal{E}(P,\gamma)$ not centered in zero, the center $Y_c$ of the new ellipsoid having to be searched. Such an approach however requires more research.

### 4.4 Robustness issues

An interesting problem consists of studying the robustness of the closed-loop system with respect to the delay $d$. For this, we suppose that $d$ is uncertain but known to lie in a given interval defined as follows:

$$d_{\min} \leq d \leq d_{\max} \tag{47}$$

where $d_{\min}$ and $d_{\max}$ are known positive integers. It is important to note that $d$ is also a constant but unknown integer. Hence, to say that $d$ satisfies (47), means that the delay $d$ considered to construct the closed-loop system (6), (20), (41) or (43) can take $d_{\max} - d_{\min} + 1$ integer values (the first value being $d_{\min}$ and the last one $d_{\max}$). For each of these $d_{\max} - d_{\min} + 1$ values, we can associate $d_{\max} - d_{\min} + 1$ closed-loop systems. In this case the problem consists in being able to simultaneously stabilize the $d_{\max} - d_{\min} + 1$ systems and therefore one has to find a controller that simultaneously stabilize these $d_{\max} - d_{\min} + 1$ systems (see, for example, [7]). In other words, we want to solve Problem 1 with respect to the $d_{\max} - d_{\min} + 1$ systems. By building the vector $Y(n)$ (as defined in (18)) as follows:

$$Y(n) = \begin{bmatrix} Q(n) - Q_0 \\ Q(n-1) - Q_0 \\ R(n) - \mu_e \\ \vdots \\ R(n - d_{\min}) - \mu_e \\ \vdots \\ R(n - d_{\max}) - \mu_e \end{bmatrix} \in \Re^{d_{\max}+3}$$

and by denoting each admissible value of $d$ by $d_i$, $i = 1, \ldots, d_{\max} - d_{\min} + 1$, we consider $d_{\max} - d_{\min} + 1$ different matrices $\mathbb{A}_0$ and $\mathbb{R}$, while matrices $\mathbb{B}_1$, $\mathbb{B}_3$ and $\mathbb{K}$ remain the same in all cases.

*Example 1.* Consider a simple example with $d_{\max} = 3$ and $d_{\min} = 1$. We have to consider three systems defined as in (20) as follows:

$$Y(n+1) = (\mathbb{A}_{01} + \mathbb{B}_{12}\mathbb{K})Y(n) + \mathbb{B}_3\nu(n) + \mathbb{B}_1\Psi_1(n) \tag{48}$$

$$Y(n+1) = (\mathbb{A}_{02} + \mathbb{B}_{12}\mathbb{K})Y(n) + \mathbb{B}_3\nu(n) + \mathbb{B}_1\Psi_2(n) \tag{49}$$

$$Y(n+1) = (\mathbb{A}_{03} + \mathbb{B}_{12}\mathbb{K})Y(n) + \mathbb{B}_3\nu(n) + \mathbb{B}_1\Psi_3(n) \tag{50}$$

with

$$\mathbb{A}_{01} = \begin{bmatrix} 1&0&1&0&0&0\\ 1&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0 \end{bmatrix} ; \mathbb{A}_{02} = \begin{bmatrix} 1&0&0&1&0&0\\ 1&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0 \end{bmatrix} ; \mathbb{A}_{03} = \begin{bmatrix} 1&0&0&0&1&0\\ 1&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0 \end{bmatrix}$$

The nonlinearities $\Psi_i(n)$, $i = 1,2,3$ are defined as in (21), that is:

$$\begin{aligned} \Psi_i(n) = \mathrm{sat}_{\{-Q_0, B-Q_0\},\{-\mu_e, C-\mu_e\}} &\left( \begin{bmatrix} \mathbb{R}_i \\ \mathbb{K} \end{bmatrix} Y(n) + \begin{bmatrix} -1 \\ 0 \end{bmatrix} \nu(n) \right) \\ &- \left( \begin{bmatrix} \mathbb{R}_i \\ \mathbb{K} \end{bmatrix} Y(n) + \begin{bmatrix} -1 \\ 0 \end{bmatrix} \nu(n) \right), \; i = 1,2,3 \end{aligned} \tag{51}$$

with

$$\mathbb{R}_1 = \begin{bmatrix} 1&0&1&0&0&0 \end{bmatrix} ; \mathbb{R}_2 = \begin{bmatrix} 1&0&0&1&0&0 \end{bmatrix} ; \mathbb{R}_3 = \begin{bmatrix} 1&0&0&0&1&0 \end{bmatrix}$$

Systems (48), (49) and (50) correspond to the cases $d = d_{\min} = 1$, $d = d_{\min} + 1 = 2$ and $d = d_{\max} = 3$, respectively. The considered system (20) may be in the configuration (48), (49) or (50). Hence to prove that these three systems can be stabilized, we can use Propositions 1 or 2, Corollaries 1 or 2 with respect to each of these systems.

## 5 Numerical issues

Relations in Corollaries 1 and 2 have some nonlinearities due to the product between $\delta$ and $V$, the product between $\omega, \gamma$ and $\nu_0$ or still the product between $\mathbb{E}_3$ and $\mu_0$. Note that these relations have less nonlinearities than those described in [18], for which matrices $\mathbb{E}_1$ and $\mathbb{E}_3$ had particular fixed structure.

The implicit objectives are to maximize the region of stability of the closed-loop system and/or the region of admissible disturbances. From Propositions 1, Corollaries 1 or 2, the region of stability associated to the closed-loop system (20) is the ellipsoid $\mathcal{E}(P,\gamma) = \{Y \in \Re^{d+3}; Y'PY \leq \gamma^{-1}\}$, whereas the set of admissible disturbances $\nu(n)$ is given through the positive scalar $\nu_0$. By noting that the volume of the ellipsoid is proportional to $\sqrt{\det(\frac{P^{-1}}{\gamma})}$, it is then possible to maximize its size by minimizing the function $\log(\det(\gamma P))$. By definition of $P$ it follows:

$$\det(\gamma P) = \det(\gamma S^{-1} V (S')^{-1})$$
$$= \gamma^{d+3} \det(S^{-1}) \det(V) \det((S')^{-1}) = \gamma^{d+3} \frac{\det(V)}{\det(S)^2}$$

or still

$$\log(\det(\gamma P)) = (d+3)\log(\gamma) + \log(\det(V)) - 2\log(\det(S))$$

Hence, depending on the weight that we want to give to the set $\mathcal{E}(P,\gamma)$ or to the set of admissible disturbances $\nu(n)$, one can consider the following optimization problem

$$\min\{\beta_1((d+3)\log(\gamma) + \log(\det(V)) - 2\log(\det(S))) + \beta_2 \nu_0\}$$

for which relations of Corollaries 1 or 2 are the constraints and the $\beta_i$'s are tuning parameters.

*Example 2.* Let us consider an ATM network with a bandwidth of 100 Mbps and let us apply our controller to a switch located at a distance of 500 Km from the source; with these values of the parameters the round trip delay $d = d^f + d^b$ amounts to 5ms [4]. We assume that the state of the switch is updated every 200 cells, which corresponds[5] to 0.848ms; therefore the delay in terms of computation cycles is $d = 6$. The buffer size was chosen $B = 400$, whereas the maximum explicit rate $C$ was set equal to 20.

In this example we used the results of Corollary 1. The conditions were implemented in terms of LMI's using the relaxation technique, by means of the LMI Toolbox [9].

We fixed $Q_0 = 200$, $\mu_e = 10$. The controller guarantees a steady-state buffer occupancy $Q_e = 125$ and a region of admissible disturbances with $\nu_0 = 10$. This example has been already studied in [18]; the new conditions of Corollary 1, where the matrices $\mathbb{E}_1$ and $\mathbb{E}_3$ have no particular fixed structure, has lead to a significant improvement of the performance guaranteed by the designed controller in terms of both buffer occupancy and amplitude of the region of admissible disturbances.

Figure 1 shows the controller performance in the presence of a constant $\mu = \mu_e$ in terms of queue length (QL) and explicit rate (ER).

## 6 Conclusions

This chapter presented a nonlinear, discrete-time model of an ABR/ATM switch and showed how queue saturation may be incorporated directly into the control design step. The resulting fixed-structure controller is simple, implementable, and robust.

## References

[1] C.T. Abdallah, M. Ariola, R. Byrne. Statistical-learning control of an ABR explicit rate algorithm for ATM switches. *Proc. of the 39th IEEE Conference on Decision and Control (CDC)*, Sydney (Australia), December 2000.

[5] The length of each cell in the ATM networks is equal to 53 bytes.

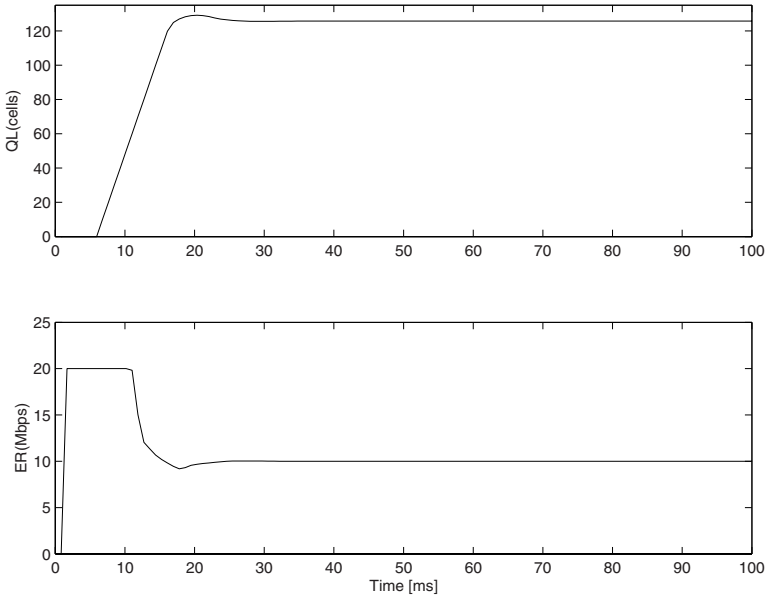



**Fig. 1.** Controller performance

[2] ATM Forum Traffic Management Working Group AF-TM-0056.000. *ATM Forum Traffic Management Specification*, April 1996. Version 4.0.

[3] H.X. de Araujo, I. Queinnec, S. Tarbouriech. *Robust control of linear uncertain systems subject to state and actuator limitations*, J.E.S.A, vol.35, no.1-2, 2001.

[4] L. Benmohamed. Control-theoric applications in congestion control. *NSF/ONR Workshop on Future Directions in Systems and Control: Research in Communication Networks*, November 6-7, 1998.

[5] L. Benmohamed and Y.T. Wang. A Control-Theoretic ABR Explicit Rate Algorithm for ATM Switches with Per-VC Queuing. *Proc of Conference on Computer Communication (Infocom98)*, pp.183-191, San Francisco (USA) 1998.

[6] F. Blanchini, R. Lo Cigno, and R. Tempo. *Control of ATM Networks: Fragility and Robustness Issues*, Proc. of American Control Conference, pages 2847–2851, Philadelphia, PA, 1998.

[7] V. Blondel. *Simultaneous stabilization of linear systems*, Springer-Verlag, London, 1994.

[8] D. Cavendish, S. Mascolo, and M. Gerla, *SP-EPRCA: an ATM rate Based Congestion Control Scheme Based on a Smith Predictor*, 1996 UCLA CS Tech Report 960001, available at: ftp://ftp.cs.ucla.edu/tech-report/

[9] Gahinet P., A. Nemirovski, A. J. Laub and M. Chilali, *LMI Control Toolbox*, The Mathworks Inc, Natick, MA, 1995.

[10] J.M. Gomes da Silva Jr. and S. Tarbouriech, *Anti-windup design with guarateed regions of stability: an LMI-based approach*, Accepted at 42th IEEE Conference on Decision and Control (CDC'03), Hawaii, USA, December 2003.

[11] O.C. Imer, S. Compans, T. Başar, and R. Srikant. *ABR Control in ATM Networks*, IEEE Control Systems Magazine, vol. 21, No.1, pp. 38–56, 2001.

[12] S. Mascolo, *Smith's Principle for Congestion Control in High-Speed Data Networks*, IEEE Trans. on Automatic Control, vol.45, no.2, pp.358-364, 2000.

[13] M.C de Oliveira. *On Stability Tests for Linear Systems*, Proc. of the 15th World IFAC Congress, Barcelona, Spain, July 2002.

[14] M.C de Oliveira and R.E. Skelton. *Stability tests for constrained linear systems*, pp.241-257, in Perspectives in Robust Control, Lecture Notes in Control and Information Sciences, Springer-Verlag, 2001.

[15] D. Peaucelle and D. Arzelier. *An efficient numerical solution for $H_2$ static output feedback synthesis*, Proc. of the European Control Conference (ECC), Porto, Portugal, July 2001.

[16] C. Paim, S. Tarbouriech, J.M. Gomes da Silva Jr., E.B. Castelan. *Control for linear systems with saturating actuators and $L_2$-bounded disturbances*, accepted at the IEEE Conference on Decision and Control (CDC'02), Las Vegas, USA, December 2002.

[17] S. Tarbouriech and G. Garcia. *Control of uncertain systems with bounded inputs*, Lectures Notes in Control and Information Sciences, Springer-Verlag, vol.227, 1997.

[18] S. Tarbouriech, M. Ariola, C.T. Abdallah. *Bounded controller design of an ABR explicit rate algorithm for ATM switches*, Accepted at 42th IEEE Conference on Decision and Control (CDC'03), Hawaii, USA, December 2003.

# State-Space Models for Control and Identification

Henri-François Raynaud, Caroline Kulcsár, and Rim Hammi

Laboratoire de Traitement et Transport de l'Information (L2TI)
Institut Galilée, Université Paris 13, raynaud@l2ti.univ-paris13.fr

## 1 Why a state-space approach?

In the last two decades, design and performance evaluation of efficient congestion control methods for packet-switching computer communication networks has emerged as a major engineering challenge. When congestion control is considered in the framework of control theory, communication networks are viewed as dynamical systems whose inputs and outputs are rates of data transmission.

Most attempts to formulate congestion control problems in the language of control engineering have focused on an input-output approach, where network elements such as buffers and transmission lines are modeled by differential, delay-differential or difference equations. If these equations are linear or can be realistically linearized around some equilibrium point or trajectory, standard transfer function techniques can be applied to perform controller design and to evaluate open and closed-loop performance criteria such as stability, robustness margins, transient response or disturbance attenuation.

That line of research has been quite productive. It has led to innovative proposals for new or improved feedback control laws, notably in the technologically more flexible framework of ATM networks, and also to a better understanding of existing algorithms (see, *e.g.,* [3, 7, 11, 12, 15, 17, 18, 20]). Other approaches used to control the input-output behavior of network elements include neural networks [13] and fuzzy control [5]. Also, an intriguing line of research is feedback-control design using concepts from classical economic theory, namely utility functions, the resulting control algorithm being interpreted as the embodiment of a Walrasian auctioneer [9, 14].

However, as the recent history of control engineering theory and practice amply demonstrates, in most domains of application much is to be gained by combining input-output and state-space approaches. In particular, a well-known merit of state-space methods is that they enable to separate the problem of designing a control action based on available measurements into a full information control problem and a state estimation problem, both of which can then be tackled using standard and constructive methods. This is especially attractive in communication networks, where

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 177–197, 2004.

congestion control algorithms generally have to make the best of partial and delayed measurements.

The purpose of this contribution is to explore the application of state-space methods to congestion control problems in computer networks. We deal here with the case of a single router connected to a buffered link. Clearly, real networks are much more complex. However, this system can be seen as an elementary building block with which more complicated networks are constructed. Thus, a good understanding of its dynamics, and of the associated control problems, is a necessary first step towards the design of control methods of practical interest. Also, in any actual network, congestion control is only required upstream of links where demand exceeds available bandwidth, *i.e.* for a sub-network of reduced complexity. This explains why this model has been extensively studied from an input-output point of view [3, 15, 17, 18], both in the single and multi-source cases.

The analysis of this system using standard state-space concepts leads to interesting conclusions, which would remain pertinent for more complex network elements. In our opinion, the most relevant, and somewhat surprising, is that the congestion-induced delay experienced by the sources, together with the data flows delivered to the various destinations, can be expressed as non-linear functions of the available bandwidth and of an internal state whose basic, saturation-free dynamics, are linear. Consequently, it is possible to achieve both congestion, delay and output flow control using an appropriate linear feedback plus feedforward scheme, associated if need be with a linear observer. The state-space analysis also suggests that the ubiquitous concept of round-trip time (RTT) may be deceptive, since it lumps together three qualitatively different phenomena, namely a control delay, a congestion-induced delay and a measurement delay.

On a more practical level, another important motivation for this work is the need to elaborate congestion control procedures able to meet the quality of service (QoS) requirements of real-time applications such as video transmission [8, 10]. One of the most important constraints for this type of applications is that bandwidth should remain stable and superior to video data throughput, while maintaining a stable and acceptable quality level. Congestion control thus includes rate control, which aims at regulating video stream rate according to available bandwidth.

Rate control schemes include probe-based and model-based approaches [24]. In a probe-based approach, the source rate is adjusted to stay below a certain threshold, depending on the available bandwidth probed on the network [22]. Model-based schemes are based on a model of data throughput, as seen for example with TCP-friendly [16], which uses informations sent back by the destination for source rate computation. The latter approach, as in fact many real-time applications, currently implement source-level rate control using feedback information relayed according to the Real time Transport Control Protocol (RTCP), see for example [19, 21]. However, RTCP provides only end-to-end aggregated feedback information such as mean loss rate and transmission delay, therefore imposing severe limitations on the structure of the rate control algorithm.

It is hoped that the so-called active network technologies currently being developed can remove these limitations. Indeed, active networks probe for necessary

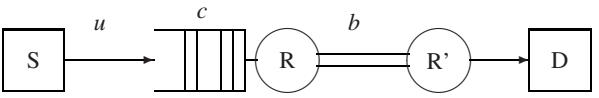



**Fig. 1.** Elementary network configuration with one source, one buffer, one link

feedback informations to be sent to the congestion control algorithm, which can be implemented in active routers. Therefore, active networks provide a technological framework for implementing a wider range of alternative control procedures [1, 4, 6, 23].

The approach presented here uses discrete-time models. In practice, the sampling period needs not coincide with the clocking frequency of any particular network component. Since computer networks are usually made up of a number of interacting but unsynchronized automata handling packetized chunks of data, the sampling period should be chosen large enough in order to average the haphazard high-frequency variations resulting from the interaction of these elemental machines. Obviously, it should also be large enough to perform the control action computation. On the other hand, the period should be small enough so that the variables to be controlled, and especially buffer occupations, do not fluctuate too much between two consecutive sampling times. In other words, one only need to apply to this particular class of problems the same commonsensical criteria which are routinely used to select a suitable sampling rate in standard control applications.

This chapter is organized as follows. Sect. 2 presents a model made up of one source, one router, one destination, with no control delay. In Sect. 3, the problem of controlling congestion-induced delay for this model, in the presence of input saturation, is solved under the ideal *full information hypothesis*, in other words assuming that full state and disturbance measurements are available at source level. In Sect. 6, this result is extended to cope with a control delay by using the discrete-time version of a Smith predictor. In Sect. 7, the congestion estimation problem resulting from (possible time-varying) measurement delays is solved by using the dual observer of a Smith predictor, yielding a comprehensive feedforward plus observer-based output feedback control presented in Sect. 8. In Sect. 9, the results are extended to the case of several source-destination pairs. Some perspectives for future research are presented in Sect. 10.

## 2 Elementary network configuration

Let us consider first a network made up of a router connected to a single link, with one source S and one destination D (Fig. 1). The packets arriving at the router are stored in a first-in, first-out (FIFO) buffer. The time unit is defined as the number of sampling periods. Assuming for the moment that i/ the time needed by the router to transfer incoming packets to the buffer and outgoing packets to the link is significantly smaller than the sampling period and that ii/ the source is capable of instantly adjusting its data rate according to the solicitations of the control procedure, the basic

equation governing the buffer dynamics should be

$$c(t+1) = c(t) + u(t) - l(t) - \big(b(t) - v(t)\big), \tag{1}$$

where $c(t)$, $u(t)$ and $b(t)$ denote respectively the buffer congestion, source rate and advertised available link bandwidth (measured in data bytes per sampling interval); $l(t)$ is the number of packets lost between $t$ and $t+1$, and $v(t)$ is the bandwidth depletion resulting from buffer underflow. It is also assumed that the buffer has a maximum capacity $c_M$, while the source rate has a maximum desired emission rate $u_M$, so that the control $u$ is constrained by $0 \leq u \leq u_M$. Clearly, if the link bandwidth remains superior to $u_M$, the system will then reach the (non-linear) stable equilibrium point $c = 0$ no matter what the source chooses to emit, and the appropriate control policy is therefore $u = u_M$. Consequently, we need only to consider here situations where $b$ satisfies a constraint in the form $0 \leq b \leq b_M < u_M$.

The variables $l(t)$ and $v(t)$ account for the fact that buffer over- or underflows, should they occur, cannot be predicted with complete accuracy based on the sole knowledge of the average rates of packet arrivals and departures.

Let us consider first $l(t)$. Between $t$ and $t+1$, the arrival rate of packets in the buffer fluctuates. As long as the maximum buffer capacity $c_M$ is not exceeded, no packet loss happens. The buffer state at time $t+1$ is then simply $c(t+1) = c(t) + u(t) - b(t)$ and $c(t+1) \leq c_M$. If the maximum buffer capacity happens to be exceeded at some point between $t$ and $t+1$, some incoming packets will be lost, and, at $t+1$, the buffer state thus corresponds to the sum of all packets arrived between $t$ and $t+1$, minus the lost ones denoted by $l(t)$, minus those sent during the same time interval. This leads to $c(t+1) = c(t) + u(t) - l(t) - b(t) \leq c_M$. Finally, when saturation has not been activated between $t$ and $t+1$, we have $l(t) = 0$. To sum it up, while $l(t)$ is not completely predictable, it is bounded by the constraints

$$\max\big(0, c(t) + u(t) - b_M - c_M\big) \leq l(t) \leq \max\big(0, c(t) + u(t) - c_M\big).$$

Let us now consider $v(t)$, which corresponds to the effect of buffer underflow. If $b(t) \leq c(t)$, underflow cannot occur and in this case $v(t) = 0$. Let us now assume that between $t$ et $t+1$ the bandwidth happens to be bigger than buffer congestion. All the packets contained in the buffer are sent, but the bandwidth actually used is meanwhile lower than the available average bandwidth $b(t)$. The total average bandwidth used between $t$ and $t+1$ is thus less than $b(t)$, that is in the form $b(t) - v(t)$. This leads to $c(t+1) = c(t) + u(t) - \big(b(t) - v(t)\big) \leq c_M$, and similarly to $l(t)$ we have

$$\max\big(0, c(t) + u(t) - b_M - c_M\big) \leq l(t) \leq \max\big(0, c(t) + u(t) - c_M\big).$$

The buffer dynamics, taking both phenomena into account, should therefore be written in the form (1). However, if the variations of $u$ and $b$ are small between $t$ and $t+1$, their instantaneous value will slightly differ from their mean value on $[t, t+1]$. In this case, the buffer dynamics equation amounts to the classical equation proposed in [3]:

$$c(t+1) = \mathrm{Sat}_{c_M}\big(c(t) + u(t) - b(t) - v(t)\big), \tag{2}$$

where the saturation function Sat is defined as

$$\mathrm{Sat}_{c_M}(c) \triangleq \begin{cases} 0 & \text{if} \quad c < 0, \\ c & \text{if} \quad 0 \le c \le c_M, \\ c_M & \text{if} \quad c > c_M. \end{cases} \tag{3}$$

In order to avoid buffer overflow (*i.e.* to get $l(t) = 0$), one should seek to drive the congestion to a desired value $c^r$ significantly smaller than $c_M$, assuming $c_M$ to be sufficiently bigger than $u_M$. Similarly, buffer underflow can be avoided if $c^r$ is larger than $b_M$, leading to $v(t) = 0$. Therefore, one can state that if $b_M << c_M$ and $c_M >> u_M$, we can, at least initially, neglect saturation and buffer over- and underflow altogether and reduce the dynamics to the linear regime

$$c(t+1) = c(t) + u(t) - b(t). \tag{4}$$

Obviously, this linear regime is unstable, for it defines $c$ as the output of a discrete-time integrator driven by the difference between the control $u$ and the link bandwidth $b$, which should be treated as a disturbance signal. Equally obviously, should one desire to drive the congestion to some desired value $c^r$, an appropriate choice of $u$ would be the feedforward plus feedback control

$$u(t) = b(t) - k\left(c(t) - c^r(t)\right), \tag{5}$$

with $0 < k \le 1$. Combining (4) and (5) yields the stable closed-loop dynamics:

$$c(t+1) = (1-k)\,c(t) + kc^r(t), \tag{6}$$

which guarantees that the tracking error $\Delta c \triangleq c - c^r$ converges monotonously towards zero, while $u$ converges towards $b$.

A question worth asking is why one should want to drive $c$ towards a strictly positive target value $c^r$. Obviously, the larger $c$, the greater the congestion-induced delay. On the other hand, as noted above, driving $c$ either to zero or towards a too small $c^r$ would result in buffer underflow, and therefore under-utilization of available link bandwidth. The choice of an appropriate reference value $c^r$ should therefore represent a compromise between these two valid concerns. As we shall see in the next section, a sensible choice of $c^r$ can be deduced from a study of the congestion-induced delay.

## 3 State-space model for the congestion-induced delay

Let us now turn our attention towards the congestion-induced transmission delay $d(t)$ experienced by the destination at time $t$, which is the difference between $t$ and the time at which the packets presently exiting from the buffer have been emitted. The justification for this concern is that keeping this delay constant is, in some important respects, a more relevant control objective than maintaining the congestion at a prespecified level. To begin with, it means something for the user, while buffer

congestion does not. Secondly, in real-time applications such as video transmission, an important QoS criterion is delay fluctuation (jitter).

In the absence of fixed delays, and assuming that no buffer overflow has occurred at least in the recent past, the delay incurred by packets percolating through a FIFO buffer is

$$d(t) \triangleq \min \left\{ d > 0 \ \text{ such that } \ \sum_{s=1}^{d} u(t-s) \geq c(t) \right\}. \tag{7}$$

We now proceed to define the state of the system as the collection of all the information that, together with $u(t)$ and $b(t)$, needs to be stored at time $t$ in order to compute both $c(t+1)$ and $d(t)$. Assuming with no loss of generality that $d \leq d_M$, the adequate choice of state is the vector with $n \triangleq d_M + 1$ coordinates

$$x(t) \triangleq \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ \vdots \\ x_n(t) \end{pmatrix} \triangleq \begin{pmatrix} c(t) \\ u(t-1) \\ u(t-2) \\ \vdots \\ u(t+1-n) \end{pmatrix}. \tag{8}$$

Equations (4) and (7) can be rewritten as a *linear state transition* equation associated with a *non-linear observation* equation:

$$x(t+1) = Ax(t) + Bu(t) + \Gamma b(t), \tag{9}$$

$$d(t) = h(x(t)), \tag{10}$$

with

$$A \triangleq \begin{pmatrix} 1\,0\,0 \cdots 0\,0 \\ 0\,0\,0 \cdots 0\,0 \\ 0\,1\,0 \cdots 0\,0 \\ 0\,0\,1 \cdots 0\,0 \\ \vdots\,\vdots\,\vdots\,\ddots\,\vdots\,\vdots \\ 0\,0\,0 \cdots 1\,0 \end{pmatrix}, \ B \triangleq \begin{pmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ \Gamma \triangleq \begin{pmatrix} -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{11}$$

$$h(x) \triangleq \min \left\{ d > 0 \ \text{ such that } \ \sum_{s=1}^{d} x_{s+1}(t) \geq x_1(t) \right\}. \tag{12}$$

Note that if the deterministic variables $u$ and $c$ are replaced by stationary processes with known distributions, the non-linear function $h$ enables to compute a corresponding stationary distribution for $d$. In the special case where the distributions of both $u$ and $c$ depend on a single parameter (typically the mean of a Poisson distribution), the mean delay can therefore be expressed as a function of the mean buffer congestion, so that one recovers the classical relations used in queuing theory (see [18] for a description of such averaged models and their utilization for controller design).

## 4 Constant-delay reference trajectory

Using these formula, and for any constant desired delay $d^r$, one can construct a reference trajectory $(u^r, x^r)$ for the state transition equation (9) which will result in $d = d^r$.

**Proposition 1.** *Let $d^r$ be any fixed arbitrary target value for the congestion delay, and*

$$u^r(t) \triangleq \tfrac{1}{2}\left(b(t+d^r-1)+b(t+d^r)\right), \tag{13}$$

$$x^r(t) \triangleq \begin{pmatrix} c^r(t) \\ u^r(t-1) \\ u^r(t-2) \\ \vdots \\ u^r(t+1-n) \end{pmatrix}, \tag{14}$$

*with $n \triangleq d_M + 1$ and*

$$c^r(t) \triangleq \frac{1}{2}b(t+d^r-1) + \sum_{s=0}^{d^r-2} b(t+s). \tag{15}$$

*Assume that for all $t \geq 0$ the link bandwidth $b(t)$ is non-zero. Then $(u^r, x^r, d^r)$ is a reference trajectory for the state-space representation (9)-(10), in the sense that for all $t \geq 0$,*

$$x^r(t+1) = Ax^r(t) + Bu^r(t) + \Gamma b(t), \tag{16}$$

$$d^r = h(x^r(t)). \tag{17}$$

*Proof.* The first step is to check that $(u^r, x^r)$ is a reference trajectory for (9), which means that $c^r$ and $u^r$ must verify $c^r(t+1) = c^r(t) + u^r(t) - b(t)$:

$$\begin{aligned} c^r(t+1) - c^r(t) &= \sum_{s=1}^{d^r-1} b(t+s) + \frac{1}{2}b(t+d^r) - \sum_{s=0}^{d^r-2} b(t+s) - \frac{1}{2}b(t+d^r-1) \\ &= b(t+d^r-1) - b(t) + \frac{1}{2}b(t+d^r) - \frac{1}{2}b(t+d^r-1) \\ &= \frac{1}{2}\left(b(t+d^r-1)+b(t+d^r)\right) - b(t) = u^r(t) - b(t). \end{aligned} \tag{18}$$

Then, from the definitions of $x^r$ and $h$, the identity $h(x^r(t)) = d^r$ is equivalent to

$$\sum_{s=1}^{d^r-1} u^r(t-s) < c^r(t) \leq \sum_{s=1}^{d^r} u^r(t-s). \tag{19}$$

From the definition of $u^r(t)$ in equation (13), we get

$$\begin{aligned}\sum_{s=1}^{d^r-1} u^r(t-s) &= \sum_{s=1}^{d^r-1} \frac{1}{2}\left(b\left(t+d^r-s-1\right)+b\left(t+d^r-s\right)\right)\\ &= \sum_{s'=1}^{d^r-1} \frac{1}{2}\left(b\left(t-1+s'\right)+b\left(t+s'\right)\right)\\ &= \sum_{s=1}^{d^r-2} b(t+s)+\frac{1}{2}\left(b(t)+b\left(t+d^r-1\right)\right). \end{aligned} \tag{20}$$

Thus, using equation (15) for $c^r$, we get

$$\begin{aligned}c^r(t)-\sum_{s=1}^{d^r-1} u^r(t-s) &= \sum_{s=0}^{d^r-2} b(t+s)+\frac{1}{2}b\left(t+d^r-1\right)-\sum_{s=1}^{d^r-2} b(t+s)-\frac{1}{2}b(t)\\ &\quad -\frac{1}{2}b\left(t+d^r-1\right)\\ &= \frac{1}{2}b(t)>0. \end{aligned} \tag{21}$$

Similarly,

$$\begin{aligned}\sum_{s=1}^{d^r} u^r(t-s) &= \sum_{s=1}^{d^r} \frac{1}{2}\left(b\left(t+d^r-s-1\right)+b\left(t+d^r-s\right)\right)\\ &= \sum_{s'=0}^{d^r-1} \frac{1}{2}\left(b\left(t+s'\right)+b\left(t-1+s'\right)\right)\\ &= \sum_{s=0}^{d^r-2} b(t+s)+\frac{1}{2}\left(b(t-1)+b\left(t+d^r-1\right)\right). \end{aligned} \tag{22}$$

$$\begin{aligned}\sum_{s=1}^{d^r} u^r(t-s)-c^r(t) &= \sum_{s=0}^{d^r-2} b(t+s)+\frac{1}{2}b(t-1)+\frac{1}{2}b\left(t+d^r-1\right)\\ &\quad -\sum_{s=0}^{d^r-2} b(t+s)-\frac{1}{2}b\left(t+d^r-1\right)\\ &= \frac{1}{2}b(t-1)>0. \end{aligned} \tag{23}$$

Provided that $b(t)>0$, the identity $h\left(x^r(t)\right)=d^r$ shall therefore hold for all $t\geq 0$, which completes the proof. □

## 5 Feedforward plus feedback control with input saturation

Consider now the tracking errors defined by

$$\Delta u \triangleq u-u^r, \quad \Delta x \triangleq x-x^r, \quad \Delta d \triangleq d-d^r. \tag{24}$$

The state transition equation for the tracking error $\Delta x$ in computed by subtracting (16) from (9). This manipulation removes the disturbance $b$ from the state transition equation in the new variables, so that for any choice of the control $u$,

$$\Delta x(t+1) = A\Delta x(t) + B\Delta u(t). \tag{25}$$

In addition, from the definition of the non-linear but piecewise-constant function $h$, the identity $h(x(t)) = h(x^r(t) + \Delta x(t)) = d^r$ holds if and only if the two inequalities

$$\frac{1}{2}b(t) + \Delta x_1(t) - \sum_{s=1}^{d^r-1} \Delta x_{s+1}(t) \geq 0, \tag{26}$$

$$\frac{1}{2}b(t-1) - \Delta x_1(t) + \sum_{s=1}^{d^r} \Delta x_{s+1}(t) > 0 \tag{27}$$

are verified. Thus, under the mild assumption that $b(t) \geq b_m > 0$ for all $t \geq 0$, the identity $h(x(t)) = d^r$ is guaranteed to be verified as soon as $\|\Delta x(t)\|_1 = \sum_j |\Delta x(t)| < b_m/2$.As a consequence, if the control $u$ is chosen so that $\Delta x$ converges towards zero, then $d$ will converge towards the target value $d^r$ in finite time.

In the absence of input saturation, such a convergence can of course be contrived by applying a standard state-feedback control in the form $\Delta u = -K\Delta x$, or equivalently

$$u = u^r - K(x - x^r), \tag{28}$$

where the control gain $K$ is chosen so that $A - BK$ is a stability matrix, *i.e.* has all its eigenvalues of modulus strictly less than one, resulting in the closed-loop dynamics

$$\Delta x(t+1) = (A - BK)\Delta x(t). \tag{29}$$

Because the linear open-loop dynamics are obtained as the series connection of an integrator and a delay line, the congestion feedback described in Sect. 2 suffices to stabilize the closed-loop. In addition, the following proposition guarantees that, under fairly mild assumptions, this control law will guarantee asymptotic convergence of both $\Delta x$ and $\Delta d$ towards zero in the presence of input saturation:

**Proposition 2.** *Assume that c is governed by the linear regime (4), that the link bandwidth at all $t \geq 0$ is constrained by*

$$0 < b_m \leq b(t) \leq b_M < u_M, \tag{30}$$

*and that the source rate control is*

$$u^l(t) = u^r(t) - k\left(c(t) - c^r(t)\right), \tag{31}$$

$$u(t) = Sat_{u_M}\left(u^l(t)\right), \tag{32}$$

*where $u^r$, $x^r$ are defined by (13)-(15) and $0 < k \leq 1$. Then for any initial congestion $c(0)$, both $\Delta x$ and $\Delta d$ converge towards zero. Furthermore, the convergence of $\Delta x$ is exponential with guaranteed rate $1 - k_m$, where*

$$k_m \triangleq \min\left\{k, \frac{b_m}{|\Delta c(0)|}, \frac{u_M - b_M}{|\Delta c(0)|}\right\}, \tag{33}$$

*whereas $\Delta d$ converges to zero in finite time.*

*Proof.* Should $u^l$ remain at all times in the interval $[0, u_M]$, the closed-loop dynamics of $c$ would be governed by

$$\Delta x_1(t) = \Delta c(t) = (1-k)^t \Delta c(0), \tag{34}$$
$$\Delta x_j(t) = \Delta u(t-j) = -k(1-k)^{t-j} \Delta c(0) \tag{35}$$

for $j > 1$. In order to derive similar relations for the saturated control, it suffices to note that (31)-(32) can be rewritten as

$$\Delta u(t) = -k_S(t) \Delta c(t), \tag{36}$$

with

$$k_S(t) = k \quad \text{if} \quad -u^r(t) \leq -k\Delta c(t) \leq u_M - u^r(t), \tag{37}$$
$$k_S(t) = \frac{u_M - u^r(t)}{-\Delta c(t)} \quad \text{if} \quad -k\Delta c(t) > u_M - u^r(t), \tag{38}$$
$$k_S(t) = \frac{u^r(t)}{\Delta c(t)} \quad \text{if} \quad k\Delta c(t) > u^r(t). \tag{39}$$

Since $0 \leq k_S(t) \leq k$ for all $t \geq 0$, we get

$$|\Delta c(t+1)| = (1 - k_S(t)) |\Delta c(t)| \leq |\Delta c(0)|. \tag{40}$$

Using this uniform boundedness of $|\Delta c(t)|$, the definition of $u^r$ in (13) and equation (30), we can guarantee that for all $t \geq 0$, $k_S(t) \geq k_m$, so that (34)-(35) can be replaced by

$$|\Delta x_1(t)| \leq (1-k_m)^t |\Delta c(0)|, \tag{41}$$
$$\left|\Delta x_j(t)\right| \leq k(1-k_m)^{t+1-j} |\Delta c(0)| \tag{42}$$

for $j > 1$. This establishes that $\Delta x$ converges exponentially towards zero with guaranteed rate $1 - k_m$, and therefore that $\Delta x$ converges towards zero in finite time. □

## 6 Dealing with control delays

In the previous sections, it has been assumed that the source was able to immediately adjust its rate as requested by the congestion control mechanism, and that packets travel from the source to the router buffer in a small fraction of the sampling period. In this section, we show how to modify the control algorithm in the case where, because of the time needed for the source to adjust its rate and/or incompressible transmission and processing time, changes in the value of $u$ will affect the data flow

into the buffer after $T_c$ additional sampling periods. In such situations, the linear buffer regime (4) should be replaced by

$$c(t+1) = c(t) + u(t - T_c) - b(t), \tag{43}$$

where the total control delay $T_c$ is the sum of the time $T_s$ needed for the source to adjust its rate and of an incompressible processing time $T_p$ incurred by packets before they enter the router buffer. The total delay experienced by the packets is the sum of $T_p$ and of the congestion delay

$$d(t) = \min\left\{ d > 0 \text{ such that } \sum_{s=1}^{d} u(t - T_c - s) \geq c(t) \right\}. \tag{44}$$

To construct a state-space representation for this system, one can retain the state vector $x$ defined by (8), albeit with $n \triangleq d_M + T_c + 1$. Clearly, this representation can be written in the form (9)-(10). For example, if $T_c = 4$ and $d_M = 2$,

$$A \triangleq \begin{pmatrix} 1&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0 \\ 0&0&1&0&0&0&0 \\ 0&0&0&1&0&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0 \end{pmatrix}, \; B \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \; \Gamma \triangleq \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{45}$$

while the non-linear function $h$ becomes

$$h(x) \triangleq \min\left\{ d > 0 \text{ such that } \sum_{s=1}^{d} x_{s+T_s+1}(t) \geq x_1(t) \right\}. \tag{46}$$

It is immediately checked that in order to obtain a reference trajectory corresponding to a constant congestion delay $d^r$, the only modification to (13)-(15) should be

$$u^r(t) \triangleq \frac{1}{2}\left( b(t + d^r + T_c - 1) + b(t + d^r + T_c) \right). \tag{47}$$

The following proposition states that for this choice of reference trajectory, asymptotic convergence of both $\Delta x$ and $\Delta d$ towards zero can be guaranteed in the presence of input saturation for a suitably modified version of the control law defined in proposition 2:

**Proposition 3.** *Assume that $c$ is governed by the linear regime (43), that the link bandwidth at all $t \geq 0$ is constrained by (30), and that the source rate control is*

$$u^l(t) = u^r(t) - k\left(c(t) - c^r(t)\right) - k\sum_{s=1}^{T_c}\left( u(t-s) - u^r(t-s) \right), \tag{48}$$

$$u(t) = Sat_{u_M}\left( u^l(t) \right), \tag{49}$$

*where $u^r$, $x^r$ are defined by (47), (14)-(15), and where $0 < k \leq 1$. Then for any initial congestion $c(0)$, both $\Delta x$ and $\Delta d$ converge towards zero. Furthermore, the convergence of $\Delta x$ is exponential with guaranteed rate $1 - k_m$, where*

$$k_m \triangleq \min\left\{k, \frac{b_m}{\delta}, \frac{u_M - b_M}{\delta}\right\}, \tag{50}$$

$$\delta \triangleq \left|\Delta c(0) + \sum_{j=1}^{T_c} \Delta u(-j)\right|, \tag{51}$$

*whereas $\Delta d$ converges to zero in finite time.*

*Proof.* For the sake of clarity, we shall only deal here with the case $T_c = 3$, the adaptations to the general situation being obvious. Denoting as $x^c$ the vector of the $T_c + 1 = 4$ first coordinates of $x$, the dynamics of this subsystem are

$$x^c(t+1) = A_c x^c(t) + B_c u(t) + \Gamma_c b(t), \tag{52}$$

where

$$A_c \triangleq \begin{pmatrix} 1&0&0&1 \\ 0&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \end{pmatrix}, \quad B_c \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \Gamma_c \triangleq \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{53}$$

Let $\Delta x^c \triangleq x^c - x^{cr}$, where $x^{cr}$ is the vector of the $T_c + 1 = 4$ first coordinates of $x^r$. Then (52) can be rewritten as

$$\Delta x^c(t+1) = A_c \Delta x^c(t) + B_c \Delta u(t). \tag{54}$$

The matrix $A_c$ has two eigenvalues: $z = 1$ (simple) and $z = 0$ (multiplicity $T_c = 3$). Since the modes associated with $z = 0$ are stable, we can apply feedback gain design strategy suggested in [2]: map the state transition equation (54) in a system of coordinates corresponding to the Jordan blocs decomposition of $A_c$, and apply a stabilizing feedback action based on the unstable mode(s), taking advantage of the parallel structure of the modal decomposition. In this case, the appropriate change of coordinates is $\gamma \triangleq P^{-1}\Delta x^c$, where

$$P \triangleq \begin{pmatrix} 1&-1&-1&-1 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix}. \tag{55}$$

In the new coordinates, the state transition equation (54) becomes

$$\gamma(t+1) = \widetilde{A}_c \gamma(t+1) + \widetilde{B}_c \Delta u(t), \tag{56}$$

with

$$\widetilde{A}_c \triangleq \begin{pmatrix} 1&0&0&0 \\ 0&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \end{pmatrix}, \quad \widetilde{B}_c \triangleq \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \tag{57}$$

This state transition equation is in the form

$$\gamma_1(t+1) = \gamma_1(t) + \Delta u(t), \tag{58}$$

$$\gamma^{[2]}(t+1) = \widetilde{A}_2 \gamma^{[2]}(t) + \widetilde{B}_2 \Delta u(t), \tag{59}$$

where $\gamma^{[2]}$ is the vector of the $T_c = 3$ last coordinates of $\gamma$, and

$$\widetilde{A}_2 \triangleq \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \widetilde{B}_2 \triangleq \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \tag{60}$$

We now choose $\Delta u(t) = -k\gamma_1(t)$ as the non-saturated control, which translates in the initial coordinates as $\Delta u(t) = -K_c \Delta x^c(t)$, with

$$\begin{aligned} K_c &= \begin{pmatrix} k & 0 & 0 & 0 \end{pmatrix} P^{-1} = \begin{pmatrix} k & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} k & k & k & k \end{pmatrix}. \end{aligned} \tag{61}$$

Applying this control would result in the stable closed-loop dynamics

$$\gamma_1(t+1) = (1-k)\gamma_1(t), \tag{62}$$

$$\gamma^{[2]}(t+1) = \widetilde{A}_2 \gamma^{[2]}(t) - \widetilde{B}_2 k \gamma_1(t). \tag{63}$$

The saturation can now be dealt with as in the proof of proposition 2, noting that here too the saturated input can always be expressed as $\Delta u(t) = -k_S(t)\gamma_1(t)$, and that $\Delta c(0)$ should be replaced by

$$\gamma_1(0) = \Delta c(0) + \sum_{j=1}^{T_c} \Delta u(-j). \tag{64}$$

Finally, since the coordinates of $\Delta x(t)$ not included in $\Delta x^c$ are in the form $\Delta u(t-j)$, the exponential convergence of $\Delta x^c$ towards zero implies the exponential convergence of $\Delta x$ with the same exponential rate, and therefore also that $\Delta d \to 0$ in finite time. □

It is worth noting that this controller is simply the discrete-time version of a Smith predictor corresponding to the proportional feedback $\Delta u = -k\Delta c$. Thus, in the absence of input saturation, and for the special choice of initial conditions $\Delta u(-j) = 0$ for $1 < j \leq T_c$, this control would result in the closed-loop dynamics

$$\Delta c(t) = (1-k)^{t-T_c} \Delta c(0). \tag{65}$$

This control action can also be viewed as a linear constrained feedback on the unique unstable mode of the open-loop regime, which belongs to the class of so-called *neutrally stable linear systems, i.e.* systems whose transition matrix has no eigenvalue of modulus strictly greater than one. As a result, this particular choice of feedback gain is a simple application of the control strategy proposed in [2], which in the general case is guaranteed to result in stable closed-loop behavior in the presence of input saturation at least for small enough values of $k$.

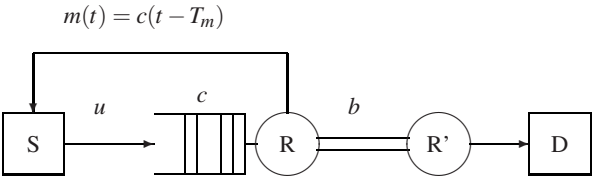



**Fig. 2.** Simple network with delayed measurements of buffer congestion

## 7 Estimating buffer congestion from delayed measurements

As noted in the introduction, the control actions proposed above would require that measurements of both the current value of $c$ and future values of $b$ are available at all times at source level. Clearly, this full information hypothesis is highly unrealistic in the context of communication networks.

In this contribution, we have chosen to deal only with the problem of estimating $c(t)$ from delayed measurements, assuming that future values of $b$ are known in advance. While this hypothesis encompasses several cases of practical interest, including constant link bandwidth, it is nevertheless restrictive. The obvious remedy, namely to estimate future values of $b$, needs to be based on a model of the dynamics of $b$, such as the autoregressive (AR) stochastic process proposed in [12].

Let us consider here the most favorable scenario where measurements of $c$ have to travel back from the buffer to the source using another network route, resulting in a measurement delay (Fig. 2). In the absence of further data loss or degradation, we therefore assume that the congestion measurement available at time $t$ is

$$m(t) \triangleq c(t - T_m), \tag{66}$$

where the measurement delay $T_m$ is (at least for the moment) assumed to be both constant and known. Note that in practice, this last assumption is tantamount to the requirement that a single time reference of sufficient precision (relatively to the sampling period) can be distributed through the network. Provided that the sampling period is large enough (typically, greater than one millisecond), such a synchronization can be achieved using the standard Network Time Protocol (NTP).

We now proceed to construct an estimate $\widehat{c}(t)$ of $c(t)$ based on the knowledge of past and present values of $m$, $b$ and $u$. In order to formulate this problem, we introduce the partial state vector

$$\eta(t) \triangleq \begin{pmatrix} c(t) \\ c(t-1) \\ \vdots \\ c(t+1-T_m) \end{pmatrix}. \tag{67}$$

Using this notation, and assuming that the dynamics of $c$ are described by the linear regime (43), we obtain the linear state-space representation

$$\eta(t) = A_m \eta(t-1) + B_m u(t - T_c - 1) + \Gamma_m b(t-1), \tag{68}$$

$$m(t) = C_m \eta(t-1), \tag{69}$$

with

$$A_m \triangleq \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}, \; B_m \triangleq \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \; \Gamma_m \triangleq \begin{pmatrix} -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{70}$$

$$C_m \triangleq \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix}. \tag{71}$$

A standard solution to the estimation problem is to use an observer in the form

$$\widehat{\eta}(t) = A_m \widehat{\eta}(t-1) + B_m u(t - T_c - 1) + \Gamma_m b(t-1) + L\left(m(t) - \widehat{m}(t)\right) \tag{72}$$

where $L$ is an observer gain and

$$\widehat{m}(t) \triangleq C_m \widehat{\eta}(t-1). \tag{73}$$

The dynamics of the estimation error $\widetilde{\eta} \triangleq \eta - \widehat{\eta}$ then become

$$\widetilde{\eta}(t) = (A_m - LC_m)\widetilde{\eta}(t-1). \tag{74}$$

Thus, an appropriate choice of $L$ will yield a convergent estimate of $c$ in the form $\widehat{c}(t) = \widehat{\eta}_1(t)$. Since the state transition equation (68) corresponds to an integrator followed by a delay line, a suitable solution is to use the dual version of the feedback gain selection approach in Sect. 6. This leads to the following result:

**Proposition 4.** *For the linear congestion regime (43) and the measurement equation (66), the observer (72)-(73), with the gain*

$$L = \begin{pmatrix} g \\ \vdots \\ g \end{pmatrix}, \tag{75}$$

*where* $0 < g \leq 1$*, and with an initial condition in the form*

$$\widehat{\eta}(0) = \begin{pmatrix} \widehat{c}(0) \\ \widehat{c}(0) + b(-1) - u(-1) \\ \vdots \\ \widehat{c}(0) + \sum_{j=1}^{T_m - 1} \left( b(t-j) - u(t-j) \right) \end{pmatrix}, \tag{76}$$

*guarantees that for any trajectory of the control u, the identification error* $\widetilde{\eta}$ *shall converge exponentially towards zero with rate* $1-g$*, and that, for* $\widehat{c}(t) = \widehat{\eta}_1(t)$*,*

$$\widetilde{c}(t) \triangleq c(t) - \widehat{c}(t) = (1-g)^t \widetilde{c}(0). \tag{77}$$

*Proof.* Since this observer design is the dual of the control feedback in Sect. 6, we shall only indicate here the main modifications that should be made to the first part of the proof of proposition 3, noting that in this case one needs not worry about input saturation. For the sake of clarity, we consider here the special case $T_m = 4$, so that

$$A_m \triangleq \begin{pmatrix} 1\,0\,0\,0 \\ 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \end{pmatrix}, \quad B_m \triangleq \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad C_m \triangleq \begin{pmatrix} 0\,0\,0\,1 \end{pmatrix}.$$

The change of coordinates corresponding to a Jordan block decomposition of $A_m$ is $\gamma \triangleq P^{-1}\eta$, with

$$P \triangleq \begin{pmatrix} 1\,0\,0\,0 \\ 1\,1\,0\,0 \\ 1\,0\,1\,0 \\ 1\,0\,0\,1 \end{pmatrix}. \tag{78}$$

In these new coordinates, (75) corresponds to

$$\widetilde{L} = P^{-1}L = \begin{pmatrix} 1 & 0\,0\,0 \\ -1\,1\,0\,0 \\ -1\,0\,1\,0 \\ -1\,0\,0\,1 \end{pmatrix} \begin{pmatrix} g \\ g \\ g \\ g \end{pmatrix} = \begin{pmatrix} g \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{79}$$

Thus,

$$\begin{aligned} P^{-1}(A_m - LC_m)P &= \begin{pmatrix} 1\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \end{pmatrix} - \begin{pmatrix} g\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \end{pmatrix} \\ &= \begin{pmatrix} (1-g) & 0\,0\,0 \\ 0 & 0\,0\,0 \\ 0 & 1\,0\,0 \\ 0 & 0\,1\,0 \end{pmatrix}. \end{aligned} \tag{80}$$

This proves that $A_m - LC_m$, is a stability matrix with spectral radius $1-g$, which establishes the exponential convergence claim. In addition, (76) corresponds to $\widehat{\gamma}(0) = \gamma(0)$, so that $\widetilde{\gamma}_j(0) = 0$ for $j > 1$. Because of the particular shape of $P^{-1}(A_m - LC_m)P$, this choice of initial condition for the observer will result at all $t \geq 0$ and $j > 1$ in $\widetilde{\gamma}_j(t) = 0$ and

$$\widetilde{c}(t) = \widetilde{\gamma}_1(t) = (1-g)^t \widetilde{\gamma}_1(0) = (1-g)^t \widetilde{c}(0). \tag{81}$$

The proof, which is given in Appendix B, is a straightforward adaptation of the proof of proposition 3. □

It is interesting to note that this observer design can be adapted almost effortlessly to the case of a variable measurement delay $T_m(t)$. In order to achieve this, one needs only to define $\eta$ so as to accommodate the maximum anticipated value of $T_m(t)$, and to turn both $C_m$ and $L$ into time-varying vectors.

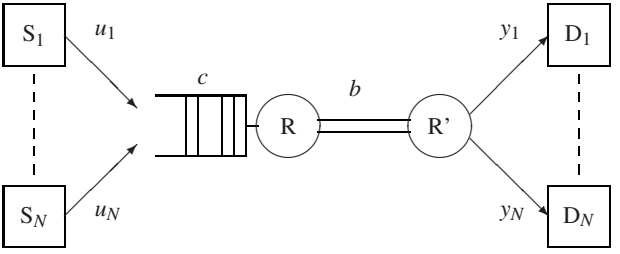



**Fig. 3.** *N* sources sharing one buffer and one link

## 8 Observer-based output feedback delay control

The feedback delay control law proposed in Sect. 6 may be combined with the observer in Sect. 7 into a standard observer-based control design. Indeed, the exponential stability of the closed-loop system with input saturation and direct state feedback will be preserved for the observer-based feedback. Thus, as a special case of the main theorem in [2], we can state the following property:

**Proposition 5.** *Let the source rate control be defined as*

$$u^l(t) = u^r(t) - k\left(\widehat{c}(t) - c^r(t)\right) - k\sum_{s=1}^{T_c}\left(u(t-s) - u^r(t-s)\right), \tag{82}$$

$$u(t) = Sat_{u_M}\left(u^l(t)\right), \tag{83}$$

*where* $\widehat{c}(t) = \widehat{\eta}_1(t)$ *is obtained using the observer in proposition 4. Then under the assumptions of propositions 3 and 4, for any initial congestion* $c(0)$, *both* $\Delta x$ *and* $\Delta d$ *converge towards zero, respectively exponentially and in finite time.*

## 9 Extension to several sources sharing a single link

All the results presented above can be extended to the scenario where $N$ source-destination pairs, possibly experiencing different control and/or measurement delays, share a single buffer/link (Fig. 3). For the sake of clarity, we shall deal here only with the case $N = 2$, the extension to larger values of $N$ being self-evident.

As we shall see, the $N$ sources scenario differs from the single source case mainly in one important qualitative respect: the data flows received by the different sources now depend not only on the link rate $b(t)$ but also on the internal state itself.

If $N = 2$, the linear congestion regime is now described by

$$c(t+1) = c(t) + u_1(t - T_{c1}) + u_2(t - T_{c2}) - b(t), \tag{84}$$

where $u_1$, $u_2$ and $T_{c1}$, $T_{c2}$ are the rates and control delays for the two sources. The appropriate state vector (assuming for the moment that there is no measurement problem) should be

$$x(t) \triangleq \begin{pmatrix} c(t) \\ u_1(t-1) \\ \vdots \\ u_1(t-d_M-T_{c1}) \\ u_2(t-1) \\ \vdots \\ u_2(t-d_M-T_{c2}) \end{pmatrix}, \tag{85}$$

where $u_1$, $u_2$ are the rates for the two sources and $d_M$ is the maximum value of the congestion delay.

It is immediately checked that the state dynamics are linear, in the form (9). The congestion delay, which is necessarily the same for the two sources, is still a non-linear function of $x$:

$$d(t) = \min \left\{ \begin{array}{c} d > 0 \text{ such that } \sum_{s=1}^{d} u_1(t-T_{c1}-s) \\ +\sum_{s=1}^{d} u_2(t-T_{c2}-s) \geq c(t) \end{array} \right\}. \tag{86}$$

Let now $y_1(t)$, $y_2(t)$ be the data rates received by the two sources, or more precisely the total number of packets received at each of the two destinations during the time slot $[t-1,t]$. The exact values of these rates are difficult to predict with absolute accuracy, since they depend on the way the packets emitted by the sources during a particular sampling interval are inserted into the buffer. However, if in any given slice of the buffer the two classes of packets are randomly mixed and small enough compared with the total number of packets emitted during one sampling interval, one can confidently assume that $y_1(t)/y_2(t)$ is close to the corresponding ratio of the input rates at time $t-d(t)$. Then, one can use the approximation

$$y_j(t) = b(t) \times \frac{u_j(t-d(t))}{u_1(t-d(t)) + u_2(t-d(t))}. \tag{87}$$

Thus, $y_1(t)$ and $y_2(t)$ are non-linear, somewhat complicated but nevertheless piecemeal continuous, functions of $x(t)$ and $b(t)$. As a consequence, any control procedure ensuring that $d$ and $u_1$, $u_2$ converge towards prespecified values will also enable to control the output rates $y_1$, $y_2$. Keeping in mind this important remark, we now proceed to adapt the control procedures of Sect. 6 to the multi-source case.

This can be achieved in a relatively straightforward manner by defining a suitable scalar feedforward plus feedback control, then splitting it between the various sources. To begin with, we define a scalar reference control as

$$u_0^r(t) \triangleq \frac{1}{2}\left(b(t+d^r-1) + b(t+d^r)\right). \tag{88}$$

Assuming that the sources are constrained by $0 \leq u_1 \leq u_{M1}$ and $0 \leq u_2 \leq u_{M2}$, the reference trajectory $x^r$ will be defined through (85), using (15) and

$$u_j^r(t) \triangleq \frac{u_{Mj}}{u_{M1}+u_{M2}} \times u_0^r(t). \tag{89}$$

Routine calculations show that this defines a proper reference trajectory for the system. The next step is to adapt to the multi-source case the feedback gain design in Sect. 6. It turns out that the only (mild) difficulty is to properly adapt the change of coordinates associated with the control delays $T_{c1}$ and $T_{c2}$. Finally, one obtains the control

$$\begin{aligned} u_0^l(t) &= -u_0^r(t) - k\left(c(t) - c^r(t)\right) \\ &\quad -k\left(\sum_{s=1}^{T_{c1}}\left(u_1(t-s) - u_1^r(t-s)\right) + \sum_{s=1}^{T_{c2}}\left(u_2(t-s) - u_2^r(t-s)\right)\right) \end{aligned} \tag{90}$$

$$u_j(t) \triangleq \mathrm{Sat}_{u_{Mj}}\left(\frac{u_{Mj}}{u_{M1}+u_{M2}} \times u_0^l(t)\right). \tag{91}$$

Because the controls $u_1, u_2$ necessarily enter and leave their respective lower and upper saturations simultaneously, this control retains all the convergence properties laid out in proposition 3. Thus, under the assumption that $b_M < u_{M1} + u_{M2}$, it can be guaranteed that as $t$ increases, $\Delta x \to 0$ and $\Delta d \to 0$, so that

$$y_j(t) \to \frac{u_{Mj}}{u_{M1}+u_{M2}} \times b(t). \tag{92}$$

This control action can be associated with an appropriate observer providing an estimate of $c(t)$.

An issue of some practical interest is whether this algorithm can be transformed into a fully decentralized control structure, where each source would compute its data rate $u_j(t)$ separately. In order to achieve this while preserving all the guaranteed convergence properties of the resulting algorithm, each source should be capable of predicting accurately what its peers are emitting. From (90)-(91), this condition can be translated in the sole requirement that all sources should be able to compute the same estimate of $c(t)$, and thus use observers based on the same measurement delay. Also, the model can be augmented to accommodate the data flow generated by non-controlled sources. This variable would enter the model as an additional disturbance. Interestingly, from the standpoint of congestion control alone a non-controlled rate is completely equivalent to a corresponding drop in the link bandwidth, and can therefore be amalgamated into $b$. However, because the non-controlled rate and the link rate affect the state dynamics in different ways, their effects on congestion-induced delay will not be identical.

## 10 Perspectives

In this chapter, standard state-space methods have been used to investigate the congestion and delay control problem presented by an elementary network element, namely a router connected to a single link. In order to tackle this problem, we have proposed a disturbance feedforward plus observer-based state feedback approach.

Even in this case, however, several issues of practical interest still warrant further investigation.

As noted above, one such issue is on-line identification of the link bandwidth and competing non-controlled traffic. We should also mention the ability of the control scheme to reject additional disturbances, for example those resulting from the packetized nature of the data flows. (Obviously, since closed-loop exponential stability, even in the presence of input saturation, translates into bounds on the various disturbance to output operators, the proposed scheme can be confidently expected to offer at least a minimal level of performance in this respect.)

Another important concern is robustness *vis-à-vis* uncertainties/variations in the system's parameters, and especially control and measurement delays. As usual, one would expect that such robustness concerns lead one to impose limitations to the feedback and observer gains, and thus to limit closed-loop performance. Finally, there is the non trivial problem of evaluating the impact of buffer under- or overflow on the transient responses of the controlled system.

Besides the obvious issue of implementation and performance evaluation of this algorithm over a real network, the perspectives for further developments include "scaling up" this state-space approach to more complex network combinations involving several routers. As an ultimate, yet perhaps unrealistic, goal, one can dream of a comprehensive set of rules which would permit to combine and coordinate elementary "control agents" operating at different locations so as to be able to control any possible network configuration. As indicated in the introduction, such a control architecture could hopefully be deployed through a network, or a portion of it, using the emerging active network technology.

## References

[1] Amir E, McCanne S, Katz R (1998) An active service framework and its applications to real-time multimedia transcoding. Proc. ACM SIGCOMM'98:178–189.

[2] Bao X, Zongli L, Sontag ED (2000) Finite gain stabilization of discrete-time linear systems subject to actuator saturation. Automatica 36:269–277.

[3] Benmohamed L, Meerkov SM (1993) Feedback control of congestion in packet switching networks: the case of a single congested mode. IEEE/ACM Transactions on Networking 6(1):693–708.

[4] Calvert K (1999) Architectural framework for active networks, version 1.0., Active Network Working Group, http://www.cc.gatech.edu/projects/-canes/papers/arch-1-0.ps.gz.

[5] Cheng RG, Chang CJ (1996) Design of a fuzzy traffic controller for ATM networks. IEEE/ACM Transactions on Networking 4(3):460–469.

[6] Clark D, Tennehouse DL (1999) Architectural considerations for new generation of protocols. ACM symposium on communication architectures & protocols:2000–2008.

[7] Delli Priscoli F, Pietrabissa A (2002) Resource management for ATM-based geostationary satellite networks with on-board processing. Computer Networks 39:43–60.
[8] Gayraud T, Berthou P, Owezarski P, Diaz M (2000) M3POC: a multimedia multicast transport protocol for cooperative applications. ICME'2000.
[9] Gibbens RJ, Kelly FP (1999) Resource pricing and the evolution of congestion control. Automatica 35:1969–1985.
[10] Hammi R, Chen K, Blin JP, Loras F (2002) A framework for responsive control of video communication using active networks. 18th World Telecommunication Congress.
[11] Hollot CV, Misra V, Towsley D, Gong W. Analysis and design of controllers for AQM routers supporting TCP flows. IEEE Transactions on Automatic Control 47(6):945–959.
[12] Imer OÇ, Compans S, Başar T, Skikant R (2001) ABR congestion control in ATM networks. IEEE Control Systems Magazine 21(1):38–56.
[13] Jagannathan S, Talluri J (2002) Predictive congestion control of ATM networks: multiple sources/single buffer scenario. Automatica 38:815–820.
[14] Kelly FP, Maulloo AK, Tan DKH (1998) Rate control for communication networks: shadow prices, proportional fairness and stability. Journal of the Operational Research Society 49:237–252.
[15] Kolarov A, Ramamurthy G (1999) A control-theoretic approach to the design of an explicit rate controller for ABR service. IEEE/ACM Transactions on Networking 7(5):741–753.
[16] Mahdavi J, Floyd S (1997) TCP-Friendly Unicast Rate Based Flow Control. Technical note sent to the end2end-interest mailing list, 8 January 1997.
[17] Mascolo S (1999) Congestion control in high-speed communication networks using the Smith principle. Automatica 35:1921–1935.
[18] Mounier H, Bastin G (2003) Compartmental modelling for traffic control in communication networks. Internal Report, École des Mines de Paris.
[19] Rejai R, Handley M, Estrin D (1999) RAP: An End-to-End Rate-based Congestion Control Mechanism for Real-time Streams in the Internet. Proc. of IEEE Infocom'99:1337–1345.
[20] Schwiebert L, Wang LY (2001) Robust control and rate coordination for efficiency and fairness in ABR with explicit rate marking. Computer Communications 24:1329–1340.
[21] Sisalem S, Schulzrinne H (1998) The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme. Proc. NOSSDAV'98.
[22] Tang J, Morabito G, Akyildiz Ian F, Johnson M (2001) RCS: A Rate Control Scheme for Real-Time Traffic in Networks with High Bandwidth-Delay Products and High Bit Error Rates. Proc. IEEE Infocom'01.
[23] Wetherall D, Tennenhouse DL (1996) The active IP option. 7th ACM SIGOPS European Workshop.
[24] Wu D, Hou Y, Wenwu Zhu, Zhang YQ and Peha J (2001) Streaming Video over the Internet: Approaches and Directions. IEEE Transactions on Circuits and Systems for Video Technology 11(3):282–300.

# Global Stability of Nonlinear Congestion Control with Time-Delay

Zhikui Wang[1] and Fernando Paganini[1]

Electrical Engineering Department, University of California, Los Angeles
zkwang@ee.ucla.edu, paganini@ee.ucal.edu

## 1 Introduction

In recent years large strides have been taken in analytical approaches to network congestion control. These methods, originating in the academic community [1, 2, 3, 4, 5, 6, 7] are quickly evolving into viable alternatives to correct deficiencies of current Internet protocols [8, 9]. Using a framework proposed in [1, 2] based on fluid-flow models and utility functions, tools from convex optimization and control theory have come into play to propose new control laws that achieve equilibrium objectives of efficient network utilization, fair resource allocation and low delay, and local stability. Control can be included in a decentralized way either at network sources or at links; following [1], it has become customary to denote by "primal" control laws those that contain dynamics at sources, but static functions at links, and "dual" laws those where the opposite holds. In this vein, the case where both control laws are dynamic is named "primal-dual".

If one neglects network delay, global stability results for various families of control laws can be established in very general networks; indeed, [1, 10] show Lyapunov proofs for both "primal" and "dual" type laws, and "primal-dual" laws can be reduced to the above by a time-scale analysis [11], or in some cases treated directly through a passivity framework [12]. When feedback delay is included in the problem (an essential feature in large-scale networks) the designs become harder, especially if one wants to achieve stability by a decentralized control in a *scalable* way over arbitrary networks. The most successful approach has been to start from linearized designs in which such scalable results can be obtained [4, 6, 13], and build from these linearizations the global, nonlinear control laws. The question that remains after this is whether the resulting control laws are *globally* stabilizing. In this chapter we address this issue for the control laws , the "dual" control and "primal-dual" control developed in [4] and [13] respectively. They are reviewed in Sect. 2.

For the global stability question, involving both nonlinearity and delay, it appears very difficult to tackle the case of an arbitrary network. For this reason, here and in other recent contributions [14, 15, 16, 17], attention is restricted to the case of a

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 199–221, 2004.

single bottleneck network. One available set of tools are extensions of Lyapunov stability theory [18, 19, 20, 21, 22, 23], either by finding a Lyapunov-Krasovskii functional (usually difficult to do in a constructive way), or using Lyapunov-Razuminhin functions. Another approach, as those pursued in [16, 24], is to invoke methods of absolute stability theory such as the Small Gain theorem.

It turns out that significantly sharper results, with parameter conditions which are very close to those of the linear analysis, can be obtained through a direct analysis of the system trajectories. We build on the methods used in [16] and [17] to obtain bounds for the system trajectories, and use these bounds to generate tighter bounds at a later time. We thus generate recursively a sequence of bounds, and prove its convergence based on a contractive mapping argument, under appropriate parameter conditions.

In Sect. 3, this analysis is done for the "dual" control laws of [4]; Section 3.1 reproduces the bounds already obtained in [16], but then these are used to generate the recursion, proven to converge in Sect. 3.2. For the "primal-dual" laws, Sect. 4 focuses on the single link-source case, as in [17], and also starts with the bounds given there, with some refinement. Here there is an extra state so the mapping between bounds is two-dimensional, and its proof of convergence is more intricate. Conclusions are given in Sect. 5.

## 2 Problem Formulation

In this section we briefly review the congestion control problems, presenting the "dual" control, the "primal-dual" control and their local scalable stability developed in [4] and [13] respectively. They are also included in [25].

We are concerned with a system of $L$ communication links shared by a set of $S$ sources. The routing matrix $R$, of dimensions $L \times S$, is defined by

$$R_{li} = \begin{cases} 1 \text{ if source } i \text{ uses link } l \\ 0 \text{ otherwise} \end{cases}.$$

Each source $i$ has an associated transmission rate $x_i(t)$; the set of transmission rates determines the aggregate flow $y_l(t)$ at each link, by the equation

$$y_l(t) = \sum_i R_{li} x_i(t - \tau^f_{li}), \tag{1}$$

in which the forward transmission delays $\tau^f_{li}$ between sources and links are accounted for. We assume each link has a capacity $c_l$ in packets per second.

To model the feedback mechanism which communicates to sources the congestion information about the network, we associate with each link $l$ a congestion measure or *price* $p_l(t)$ [1, 2] and assume sources have access to the *aggregate* price of all links in their route,

$$q_i(t) = \sum_l R_{li} p_l(t - \tau^b_{li}). \tag{2}$$

Here again we allow for backward delays $\tau_{li}^b$ in the feedback path from links to sources. The total round-trip-time (RTT) for the source thus satisfies

$$\tau_i = \tau_{li}^f + \tau_{li}^b$$

for every link in the source's path. This quantity can be estimated by sources in real time.

A "dual" control is developed in [4], with dynamics at the links and static control at the sources. On the link side, the "price" is controlled as

$$\dot{p}_l = \left( \frac{y_l - c_{0l}}{c_{0l}} \right)_{p_l}^+, \tag{3}$$

in which the *virtual* capacity $c_{0l} < c_l$ guarantees almost full utilization and empty queue at equilibrium. Equation (3) uses the "positive-projection" notation [12], namely

$$(f(x))_x^+ = \begin{cases} f(x), \text{ if } x > 0 \text{ or } x = 0 \text{ and } f(x) > 0; \\ 0 \qquad \text{if } x = 0 \text{ and } f(x) < 0. \end{cases}$$

A static control law is applied in the sources

$$x_i = x_{\max,i}\, e^{-\frac{\alpha_i q_i}{M_i \tau_i}}. \tag{4}$$

Here $M_i$ is the number of the bottleneck links the flow from this source goes through, $\tau_i$ is the RTT, and $x_{\max,i}$ is a maximum rate parameter, which can vary for each source, and can also depend on $M_i$, $\tau_i$ (but not on $q_i$). The following local stability result holds, where $\bar{R}$ denotes the reduced routing matrix obtained by eliminating non-bottleneck links.

**Theorem 1 ([4]).** *Suppose the matrix $\bar{R}$ is of full row rank, and that $\alpha_i < \frac{\pi}{2}$. Then the "dual" link and source control laws (1-4) give a locally stable system for arbitrary delays and link capacities.*

In order to obtain the very general scalable stability in the "dual" control, some restrictions apply to the sources' demand curves (or their utility functions). To leave the utility functions completely up to the sources, new *dynamics* are added at the sources. Such a "primal-dual" solution is developed in [13], by taking the same link control (3) as that of the "dual", which maintains high utilization and small equlibrium queues, and a new source control as

$$\tau_i \dot{\xi}_i = \beta_i (U_i'(x_i) - q_i), \tag{5}$$

$$x_i = x_{m,i} e^{(\xi_i - \frac{\alpha_i q_i}{M_i \tau_i})}. \tag{6}$$

Here $U(x)$ is a strictly concave increasing utility function which leads to "elastic" flows [1]; st equilibrium, $q_i^* = U_i^{'}(x_i^*)$, maximizing the local profit $U_i(x_i) - q_i x_i$. When the parameter $\beta_i$ is small, the system works in two different time scales. In the fast time scale when $\xi$ varies slowly, the system works like the dual to track the utilization; the slow time scale is applied to track the resource allocation. A local stability theorem is given in [13].

**Theorem 2 ([13]).** *Assume that for every source i, $\tau_i \leq \bar{\tau}$. With the rank assumptions of Theorem 1, the "primal-dual" control, with $\alpha_i \leq \alpha < \frac{\pi}{2}$ and $\beta = \eta \frac{\alpha_i}{M_i \bar{\tau}}$, is linearly stable for a small enough $\eta \in (0,1)$ depending only on $\alpha$.*

Simulations show that such control laws seem also globally stabilizing under the same conditions as for local stability; however this is difficult to establish analytically at this level of generality. In what follows, we analyize global stability for simple network scenarios with a single bottleneck link.

## 3 Global Stability of "Dual" Control

In [16], the global stability question for the "dual" laws on a single link is pursued. One scheme is proposed to separate time-delays from the exponential nonlinearities, and relax the latter using a sector function. Then it is found that, for a system with single source and single link, these two parts can both have small gains under the condition

$$\alpha < \frac{\ln(\frac{x_{max}}{c})}{\frac{x_{max}}{c} - 1}. \tag{7}$$

However, this argument cannot be extended to multiple heterogeneous sources. For this case, a Lyapunov-Razuminhin approach was used in [16] to show that there exists a positive constant $\Theta$ such that for $\alpha_i < \Theta$, the system is globally asymptotically stable. This bound is based on an a priori bound that can be given for the system trajectories after a transient. However, an analytic value of $\Theta$ can only be found in the single-flow case to be

$$\alpha < \frac{c_0^2}{x_{max}^2}. \tag{8}$$

In this section, we will extend the bounding argument to show global stability under weaker parameter conditions.

We still focus on the case of a single link accessed by $N$ heterogeneous sources, i.e.,

$$x_i = x_{max,i} e^{(-\frac{\alpha_i p(t-\tau_i^b)}{\tau_i})}, \; i = 1,2,...,N$$
$$\dot{p} = \left( \frac{\sum_{i=1}^N x_i(t-\tau_i^f) - c_0}{c_0} \right)_p^+.$$

Then we have the closed-loop equation as

$$\dot{p}(t) = \frac{1}{c_0}\left\{\sum_{i=1}^{N} x_{\max,i} e^{\frac{-\alpha_i p(t-\tau_i)}{\tau_i}} - c_0\right\}^{+}_{p(t)}. \tag{9}$$

The initial values are assumed piecewise continuous, and

$$0 \le p(t) < \infty,\ t \in [-\tau_m, 0],$$

with $\tau_m = \max_i(\tau_i)$. Hence the price $p$ is always nonnegative, which is guaranteed by the positive projection. We assume that $\sum_{i=1}^{N} x_{max,i} > c_0$ so that the equilibrium $P_0$ is nonzero, i.e.,

$$\sum_{i=1}^{N} x_{max,i} e^{-\frac{\alpha_i P_0}{\tau_i}} = c_0. \tag{10}$$

### 3.1 Ultimately Bounded States

We first show some boundedness properties for the solution of the delay-differential equation (9), which will hold independently of any claims on global stability. In what follows, it will be convenient to introduce the function

$$F(u) := \frac{1}{c_0}\sum_{i=1}^{N} x_{\max,i} e^{-\frac{\alpha_i}{\tau_i}(P_0 - u)} - 1.$$

Note that $F(0) = 0$ and $F$ is strictly monotone increasing. Now, define

$$T_0 := -\tau_m,\ \delta P_0 := P_0;$$

the inequality

$$p(t) \ge P_0 - \delta P_0 \ \forall\ t \ge T_0 \tag{11}$$

is immediate from these definitions; we only state it because it is a first instance of a sequence of bounds to be derived. Based on this inequality, and the monotonicity of the exponentials in (9), we can make the following claims.

**Claim I:** For any $t \ge 0$, we have

$$\dot{p}(t) \le F(\delta P_0) = \sum_{i=1}^{N} \frac{x_{max,i}}{c_0} - 1.$$

In particular, it follows that from that time onwards, $p(t)$ cannot increase in any interval of length $\tau_m$ by more than

$$\Delta P_0 := \tau_m F(\delta P_0).$$

An immediate consequence is:
**Claim II:** If $p(t) > P_0 + \Delta P_0$ at some point $t \geq \tau_m$, then

$$p(t-\sigma) > P_0 \text{ for } \sigma \in [0, \tau_m],$$

and therefore $\dot{p}(t) < 0$ from (9).

The following propositions establish a bound on the system trajectories.

**Proposition 1.** *For the system (9), if $p(t^*) \leq P_0 + \Delta P_0$ for some $t^* \geq \tau_m$, then $p(t) \leq P_0 + \Delta P_0$ for all $t \geq t^*$.*

**Proof:** If the trajectory were to increase again beyond this bound, there should exist a time $t > \tau_m$ where $p(t) > P_0 + \Delta P_0$, $\dot{p}(t) > 0$; this contradicts Claim II. □

For the following proposition, the notation $[z]^+$ stands for $\max\{z, 0\}$.

**Proposition 2.** *For the system (9), define*

$$T_1 := 2\tau_m + \frac{[p(\tau_m) - (P_0 + \Delta P_0)]^+}{-F(-\Delta P_0)}. \tag{12}$$

*Then $p(t) \leq P_0 + \Delta P_0$ for all $t \geq T_1$.*

**Proof:** If $p(\tau_m) \leq P_0 + \Delta P_0$, the bound follows directly from Proposition 1.

For the case where $p(\tau_m) > P_0 + \Delta P_0$, first note from Claim II that $p(t)$ must decrease after $t = \tau_m$, at least while it remains above $P_0$. If it were to reach $P_0 + \Delta P_0$ during the interval $[\tau_m, 2\tau_m]$, we could again invoke the previous proposition to establish that the bound holds after that time, and therefore also after $T_1$ given in (12).

So it remains only to consider the case where $p(t) > P_0 + \Delta P_0$ for all $t \in [\tau_m, 2\tau_m]$. In that case, using monotonicity of the exponential we can establish the bound

$$\dot{p}(t) \leq F(-\Delta P_0) < 0$$

for all times $t \geq 2\tau_m$ in which $p(t)$ remains above $P_0 + \Delta P_0$. Integrating, we get

$$p(t) \leq p(2\tau_m) + F(-\Delta P_0)(t - 2\tau_m) \leq p(\tau_m) + F(-\Delta P_0)(t - 2\tau_m)$$

The right hand side decreases linearly, reaching the value $P_0 + \Delta P_0$ at the time

$$T_1 = 2\tau_m + \frac{p(\tau_m) - (P_0 + \Delta P_0)}{-F(-\Delta P_0)}.$$

Therefore, after that time $p(t)$ is guaranteed to have reached the desired bound, and cannot escape later invoking Proposition 1. □

We emphasize two points:

- The above bound holds independently on any claims about stability of the system (9).
- The time $T_1$ after which the bound can be claimed, depends on the system parameters, and on the initial conditions through the term $p(\tau_m)$; to make the latter point explicit, we could for instance write the bound

$$T_1 \leq 2\tau_m + \frac{[p(0) - P_0]^+}{-F(-\Delta P_0)}.$$

### 3.2 Global Exponential Stability

Our approach to the stability argument is based on developing a sequence of upper and lower bounds to the price trajectory. We already have the trivial lower bound (11), and the upper bound

$$p(t) \leq P_0 + \Delta P_0 \; \forall \, t \geq T_1, \tag{13}$$

developed in the previous section.

We now develop a more refined lower bound, using an analogous method as the one employed for the upper bound. The following claims are "dual" to Claims I-II, and follow directly from (13).

**Claim I':** For $t \geq T_1 + \tau_m$,

$$\dot{p}(t) \geq F(-\Delta P_0).$$

In particular, it follows that from this time onwards, $p(t)$ cannot *decrease* in any interval of length $\tau_m$, by more than

$$\delta P_1 := -\tau_m F(-\Delta P_0) > 0.$$

**Claim II':** If $p(t) < P_0 - \delta P_1$ at some point $t \geq T_1 + 2\tau_m$, then

$$p(t-\sigma) < P_0 \text{ for } \sigma \in [0, \tau_m],$$

and therefore $\dot{p}(t) > 0$.

From these facts, the following two propositions can be derived in an analogous way to Proposition 1 and 2. Details are omitted.

**Proposition 3.** *For the system (9), if $p(t^*) \geq P_0 - \delta P_1$ for some $t^* \geq T_1 + 2\tau_m$, then $p(t) \geq P_0 - \delta P_1$ for all $t \geq t^*$.*

**Proposition 4.** *For the system (9), define*

$$T_2 := T_1 + 3\tau_m + \frac{[P_0 - \delta P_1 - p(T_1 + 2\tau_m)]^+}{F(\delta P_1)}. \tag{14}$$

*Then $p(t) \geq P_0 - \delta P_1$ for all $t \geq T_2$.*

**Remarks:**

- It will be convenient for future analysis to write the bound

$$T_2 \leq T_1 + 3\tau_m + \frac{\delta P_0}{F(\delta P_1)}, \tag{15}$$

that follows from (14), and depends exclusively on system parameters, no longer on initial conditions.

- It could happen that $P_0 - \delta P_1 \leq 0$, in which case the situation of Claim II' could never happen, and the above propositions would be trivially true. We now give conditions under which we can show that $P_0 - \delta P_1 > 0$, or equivalently $\delta P_1 < \delta P_0$, so the new lower bound is indeed a refinement.

**Proposition 5.** *Assume that*

$$\frac{\alpha_i}{\tau_i} < \frac{1}{\tau_m}, \text{ for } i = 1, 2, ..., N. \tag{16}$$

*Then* $\delta P_1 < \delta P_0$.

**Proof:** Note that, by definition we have

$$\delta P_1 = -\tau_m F(-\Delta P_0) = -\tau_m F(-\tau_m F(\delta P_0)) =: G(\delta P_0);$$

so we must show the above-defined function $G$ is contractive. This follows by using the following bounds on $F$ (referring to (10))

$$F(u) := \frac{1}{c_0} \sum_{i=1}^{N} x_{\max,i} e^{-\frac{\alpha_i}{\tau_i} P_0} e^{\frac{\alpha_i}{\tau_i} u} - 1 \begin{cases} < e^{\frac{u}{\tau_m}} - 1 \text{ for } u > 0, \\ > e^{\frac{u}{\tau_m}} - 1 \text{ for } u < 0. \end{cases} \tag{17}$$

Applying the first case we conclude that

$$F(\delta P_0) < e^{\frac{\delta P_0}{\tau_m}} - 1, \Rightarrow -\tau_m F(\delta P_0) > -\tau_m (e^{\frac{\delta P_0}{\tau_m}} - 1),$$

so using the monotonicity of $F$, and the second case of (17), we have

$$F(-\tau_m F(\delta P_0)) > F\Big( -\tau_m (e^{\frac{\delta P_0}{\tau_m}} - 1) \Big) > e^{1 - e^{\frac{\delta P_0}{\tau_m}}} - 1.$$

From here we conclude that

$$G(\delta P_0) < \tau_m \left( 1 - e^{1 - e^{\frac{\delta P_0}{\tau_m}}} \right) =: H(\delta P_0)$$

Now, a study of the function $H(u)$ defined above shows that (see also Fig. 1)

$$H(0) = 0, \; H'(0) = 1, \text{ and } H''(u) < 0 \;\; \forall u > 0.$$

This implies $H(u) < u$ for $u > 0$, which implies the desired bound $G(\delta P_0) < \delta P_0$. □

**Remark:** By a slight refinement of the above argument, for instance, exploiting the strict inequality in (16) by

$$\frac{\alpha_i}{\tau_i} \leq (1 - \varepsilon) \frac{1}{\tau_m}, \text{ for } i = 1, 2, ..., N,$$

the following stronger condition holds:

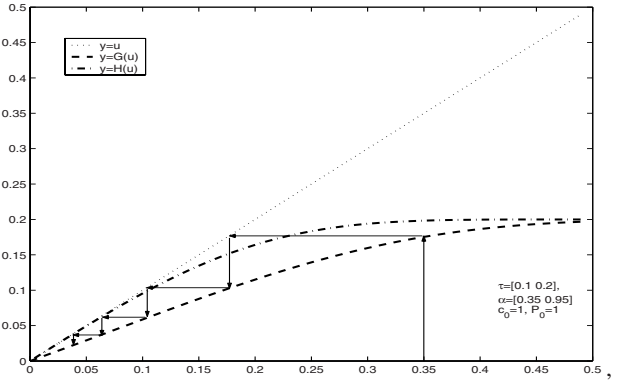



**Fig. 1.** Sketch of the function $G(u)$ and its bound $H(u)$.

$$G(u) \leq (1-\varepsilon)u \; \forall u > 0.$$

We have therefore showed that under the assumption (16), the initial lower bound from (11) can be refined to a condition

$$p(t) \geq P_0 - \delta P_1 \; \forall \; t \geq T_2$$

Now, recall that from (11) we were able to derive the *upper* bound (13) in the previous section. The fact that we have a finer lower bound means that we can now repeat the procedure to get a finer upper bound, then a finer lower bound, and so on. More precisely, we define the following recursion:

- Given a lower bound $p(t) \geq P_0 - \delta P_k$ that holds for $t \geq T_{2k}$, $k \geq 0$, define:

$$\begin{aligned} \Delta P_k &= \tau_m F(\delta P_k); \\ T_{2k+1} &= T_{2k} + 3\tau_m + \frac{[p(T_{2k}+2\tau_m) - P_0 - \Delta P_k]^+}{-F(-\Delta P_k)}. \end{aligned}$$

  An analogous argument to the one in Proposition 2 implies that $p(t) \leq P_0 + \Delta P_k$ for $t \geq T_{2k+1}$.
- Given an upper bound $p(t) \leq P_0 + \Delta P_k$ that holds for $t \geq T_{2k+1}$, define:

$$\begin{aligned} \delta P_{k+1} &= -\tau_m F(-\Delta P_k); \\ T_{2k+2} &= T_{2k+1} + 3\tau_m + \frac{[P_0 - \delta P_{k+1} - p(T_{2k+1}+2\tau_m)]^+}{F(\delta P_{k+1})} \end{aligned}$$

  An analogous argument to the one in Proposition 4 implies that $p(t) \geq P_0 - \delta P_{k+1}$ for $t \geq T_{2k+2}$.

  The progressive sequence of upper and lower bounds is illustrated in Fig. 2). Consider now the relation between two consecutive lower bounds,

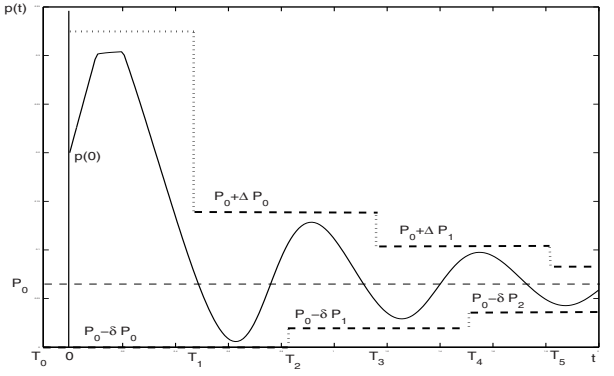



**Fig. 2.** Converging bounds for $p(t)$.

$$\delta P_{k+1} = -\tau_m F(-\Delta P_k) = -\tau_m F(-\tau_m F(\delta P_k)) = G(\delta P_k);$$

the behavior of this iteration is illustrated in Fig. 1, under assumption (16). Indeed, our previous analysis shows that the sequence $\delta P_k$ converges exponentially to zero. Also, since $F$ is monotonic with bounded derivative at zero, the same will happen with $\Delta P_k = \tau_m F(\delta P_k)$.

This leads to the main result on the stability of the "dual":

**Theorem 3.** *Under condition (16), the system (9) is globally exponentially stable.*

**Proof:** We have already established global *asymptotic* stability: after an initial time $T_1$ which depends on the initial conditions on the system, the trajectory will reach the upper bound $P_0 + \Delta P_0$; from here on, we have obtained an increasing sequence of times, and a narrowing set of upper and lower bounds, which imply that necessarily

$$\lim_{t\to\infty} p(t) = P_0.$$

We have also argued that upper and lower gaps $\Delta P_k$, $\delta P_k$ decrease exponentially to zero; now to establish exponential stability, we must show that the increments of time between these successive $T_k$ remain bounded as $k \to \infty$. This fact can be shown by generalizing the bound (15) to the rest of the iteration; indeed, it follows easily that, for $k \geq 1$, we have

$$T_{2k+1} \leq T_{2k} + 3\tau_m + \frac{\Delta P_{k-1}}{-F(-\Delta P_k)}, \tag{18}$$

$$T_{2k+2} \leq T_{2k+1} + 3\tau_m + \frac{\delta P_k}{F(\delta P_{k+1})}. \tag{19}$$

Since we know $\delta P_k, \Delta P_k \to 0$ as $k \to \infty$, we can write

$$\frac{\delta P_k}{F(\delta P_{k+1})} \approx \frac{\delta P_k}{F'(0)\delta P_{k+1}} \approx \frac{\delta P_k}{F'(0)G'(0)\delta P_k} = \frac{1}{F'(0)G'(0)},$$

and therefore that the last term of (19) remains bounded in $k$, as required. A similar argument applies to the last term of (18). □

How conservative is the above global stability result? For the case of a system with single flow (or many flows with homogeneous delays), with equations

$$x = x_m e^{\left(-\frac{\alpha q}{\tau}\right)}, \tag{20}$$

$$\dot{q} = \left(\frac{x(t-\tau)-c_0}{c_0}\right)_q^+, \tag{21}$$

the result is very tight:

**Corollary 1.** *The system (20-21) with nonnegative and integrable initial values, is globally exponentially stable when*

$$\alpha < 1.$$

This global stability result approximates the condition $\alpha < \frac{\pi}{2}$ for local stability, and is considerably tighter than the previously available bounds (7) and (8).

In this single-flow case, we can also reach similar conclusions by referring to work by [26], assuming we ignore the positive projection. For this purpose, define $y = (x - c_0)/c_0$, then we have

$$\dot{y}(t) = -\frac{\alpha}{\tau} y(t-\tau)(y(t)+1).$$

Let $t = \tau\tilde{t}$, we have a differential equation

$$\dot{y}(\tilde{t}) = -\alpha y(\tilde{t}-1)(y(\tilde{t})+1),$$

which shows that the dynamics of the system is time-scale invariant. This system is studied in [26], where it is argued that it's globally stable when $\alpha < \frac{3}{2}$ or even closer to $\frac{\pi}{2}$ if $y(0) > -1$. More references on the dynamics of this system can be found in [22].

For the case of heterogeneous flows, the result of Theorem 3 is conservative in that it would apply to the case where all flows run at the time-scale of the longest RTT. One could imagine less conservative conditions that would imply the contractiveness of the function $G(u)$, from which the result would follow. It is not easy , however, to make these conditions independent of the operating point.

## 4 Global Stability of the "Primal-Dual" Control

With one more state in the source control, the trajectory of the "Primal-Dual" control becomes more complicated. This confines us to consider only the simplest case: a system with single link and single flow, namely,

$$\tau\dot{\xi} = \beta(U'(x) - q), \tag{22}$$

$$x = x_m e^{(\xi - \frac{\alpha q}{\tau})}, \tag{23}$$

$$\dot{q} = \left(\frac{x(t-\tau) - c_0}{c_0}\right)_q^+. \tag{24}$$

By definition, $x$, $q$ and $\dot{q}$ are automatically lower bounded, that is,

$$q(t) \geq 0,\ x(t) > 0 \text{ and } \dot{q}(t) > -1 \text{ for } t > 0. \tag{25}$$

The equilibrium point is

$$x_0 = c_0,\ q_0 = U^{'}(c_0). \tag{26}$$

The utility function is strictly concave and increasing, which means

$$U^{'}(x) > 0,\ U''(x) < 0. \tag{27}$$

For many (not all) of the results below, we further restrict the utility function to have bounded derivative, as follows.

**Assumption 1** $U^{'}(0) < +\infty$.

For the following analysis, it is convenient to eliminate the state variable $\xi$, and write, from (22-23), the expression

$$\dot{x} = \frac{x\beta}{\tau}\left(U^{'}(x) - q - \frac{\alpha}{\beta}\dot{q}\right), \tag{28}$$

involving only price and rate; this will be applied a few times later.

### 4.1 Ultimately Bounded States

As in Section (3.1), the states $x$ and $q$ in (22-24) are not only trivially lower bounded, but are also upper bounded independent of any global stability condition. A priori bounds will be given in this section that must hold ultimately after some initial transient.

For given $\alpha$, $\beta$, we define a positive number $\overline{\gamma}$ such that

$$U^{'}((1+\overline{\gamma})c_0) - \frac{\alpha}{\beta}\overline{\gamma} = 0.$$

Note that $U'$ is a decreasing function, so $\overline{\gamma}$ always exists, and is an increasing function of $\beta$. We will use this value to derive upper bounds of $x(t)$, by directly analyzing its dynamics.

**Proposition 6.** *The rate $x(t)$ of the system (22-24) is bounded by $\overline{x}$ after a finite time, where*

$$\overline{x} = (1+\overline{\gamma})c_0 e^{\alpha(1+\overline{\gamma})}. \tag{29}$$

**Proof:** We argue first that, with any assumed initial values, $x(t)$ must reach the point $(1+\overline{\gamma})c_0$ by some time, say $T_0$.

Assume instead that $x(t) > (1+\overline{\gamma})c_0$ holds for $t \geq 0$, and analyze the dynamics in two phases.

For $t \in [0,\tau]$, we use the bound $U'(x) \leq U'((1+\overline{\gamma})c_0)$ and the trivial lower bounds (25) for $q$, $\dot{q}$, to get from (28) the bound

$$\dot{x} \leq x\frac{\beta}{\tau}\left(U'((1+\overline{\gamma})c_0) + \frac{\alpha}{\beta}\right).$$

Hence $x(t)$ is upper bounded by an exponential function, and by the time $t = \tau$, we have

$$x(\tau) \leq x(0)e^{\beta U'((1+\overline{\gamma})c_0)+\alpha}. \tag{30}$$

For $t > \tau$, from (24) , $\dot{q}(t) > \overline{\gamma}$ and therefore $q(t) > q(\tau) + \overline{\gamma}(t-\tau)$. Since $q(\tau) \geq 0$, $U'(x) < U'((1+\overline{\gamma})c_0)$, from (28) again, we have

$$\begin{aligned}\dot{x} &\leq x\frac{\beta}{\tau}(U'((1+\overline{\gamma})c_0) - \overline{\gamma}(t-\tau) - \frac{\alpha}{\beta}\overline{\gamma}) \\ &= -x\frac{\beta\overline{\gamma}}{\tau}(t-\tau) \\ &< 0.\end{aligned}$$

Therefore $x(t)$ is upper bounded by the function

$$x(t) \leq x(\tau)e^{-\frac{\beta\overline{\gamma}}{2\tau}(t-\tau)^2}$$

when $t \geq \tau$, and this implies $x(t)$ will reach $(1+\overline{\gamma})c_0$, as claimed.

Furthermore, the preceding analysis shows that if $x(t)$ were to go past this value again at some time $T_1$, then it must start to decrease before $t = T_1 + \tau$. Using a bound of the form (30) for the increase in that interval of length $\tau$, leads to the stated bound (29), which can never be exceeded after time $T_0$. □

We remark that the bound $\overline{x}$ depends on parameters such as $\alpha$ and $\beta$, but holds without any restriction on them. Moreover, it is an *increasing* function of $\beta$, since $\overline{\gamma}$ is increasing. When $\beta = 0$, we get exactly the upper bound of the rate in the "dual" control.

With this upper bound $\overline{x}$ and the trivial lower bound for $x(t)$, we can further deduce an upper bound for $q(t)$, under Assumption 1.

**Proposition 7.** *Under Assumption 1, pick any* $\varepsilon > 0$. *Then the price* $q(t)$ *of the system (22-24) is bounded by*

$$\overline{q} = U'(0) + \frac{\tau}{\alpha}\log\left(\frac{\overline{x}}{c_0}\right) + (\frac{\overline{x}}{c_0} - 1)\tau + \varepsilon, \tag{31}$$

*after a finite time.*

**Proof:** Consider the dynamics from $t = T_0$ when $x(t) \in [0,\overline{x}]$. Applying the two bounds to the dynamics (22-23) , we have

$$\dot{\xi} + \frac{\beta}{\tau} q = \frac{\beta}{\tau} U^{'}(x) \leq \frac{\beta}{\tau} U^{'}(0),$$
$$\xi - \frac{\alpha}{\tau} q = \log(\frac{x}{x_m}) \leq \log \frac{\overline{x}}{x_m}.$$

That is,

$$\frac{\beta}{\alpha}\xi + \dot{\xi} \leq \frac{\beta}{\tau} U^{'}(0) + \frac{\beta}{\alpha} \log(\frac{\overline{x}}{x_m}).$$

Multiply by $e^{\frac{\beta}{\alpha}(t-T_0)}$ on both sides and integrate from $T_0$ to $t$, then

$$\xi(t) \leq \frac{\alpha}{\tau} U^{'}(0) + \log(\frac{\overline{x}}{x_m}) + \left[\xi(T_0) - \left(\frac{\alpha}{\tau} U^{'}(0) + \log(\frac{\overline{x}}{x_m})\right)\right] e^{-\frac{\beta}{\alpha}(t-T_0)}.$$

Hence, $\xi(t)$ is bounded by

$$\overline{\xi} = \frac{\alpha}{\tau} U^{'}(0) + \log(\frac{\overline{x}}{x_m}) + \frac{\varepsilon}{2},$$

after a fixed time, say $\Delta t_1$, depending on the initial values and $\varepsilon$. Considering the dynamics from the point $T_0 + \Delta t_1$ onwards, we now derive a bound that must eventually hold for $q(t)$.

Define a reference price $\check{q}$ such that, with a small number $0 < \gamma_q < 1$,

$$x_m e^{(\frac{\overline{\xi}}{\alpha} - \frac{1}{\tau}\check{q})} = (1-\gamma_q) c_0.$$

Explicitly, the above equation gives

$$\check{q} = U'(0) + \frac{\tau}{\alpha}\left[\log\left(\frac{\overline{x}}{c_0}\right) + \frac{\varepsilon}{2} - \log(1-\gamma_q)\right]. \tag{32}$$

If, starting at time $T_1 > T_0 + \Delta t_1$, $q(t) \geq \check{q}$, then for $t > T_1 + \tau$ we would have $x(t-\tau) < (1-\gamma_q)c_0$, and therefore $\dot{q} < -\gamma_q$. So $q$ will strictly decrease, and must eventually go below $\check{q}$, at some time $T_2$. Furthermore, if it were ever to exceed $\check{q}$ again, it can only keep increasing for a maximum time of $\tau$; since also $\dot{q}$ is bounded through the bound $\overline{x}$, we obtain the overall bound

$$\overline{q} = \check{q} + (\tfrac{\overline{x}}{c_0} - 1)\tau$$

that $q$ cannot exceed after time $T_2$. From (32), this bound will take the form (31) if we choose small enough $\gamma_q$. □

In the next section, the bounds $\overline{q}$ and $\overline{x}$ are applied to show global stability. We will be using a fixed value of $\alpha < 1$, and derive conditions that hold for a small enough $\beta$. For this reason, it is important to describe the bounds' behavior as $\beta$ varies; the following remarks address this issue.

- As in the case of $\overline{x}$ and $\overline{\gamma}$, the bound $\overline{q}$ is monotonically increasing in $\beta$. In what follows, we will sometimes study the behavior of these bounds for the parameter $0 < \beta \leq \beta_0$; it is thus clear that there exist some absolute bounds

$$\overline{\gamma}^0, \quad \overline{x}^0, \quad \overline{q}^0$$

  that will hold throughout this region.
- It would be desirable to remove Assumption 1, that restricts the allowable utility functions; in this regard, we note that an upper bound for $q$ can still be given in the general case, for instance

$$\tilde{q} = U^{'}(c_0) + \frac{\alpha}{\beta} + \frac{\tau}{\alpha} \log\left(\frac{\overline{x}}{c_0}\right) + (\frac{\overline{x}}{c_0} - 1)\tau + \varepsilon.$$

  However, this bound does not have the desired monotonicity in $\beta$, and thus would not allow for the stability analysis that follows.

### 4.2 Global Asymptotic Stability

We will use a successive bounding technique similar to the one used in the case of the "dual" control, but considerably more involved given the additional state variable. In particular, we will exploit the upper bounds $\overline{x}$ and $\overline{q}$ to refine the trivial lower bounds for $x$ and $q$, after a sufficiently long time. Subsequently, these refined bounds will lead to refinements of the upper bounds, and so on. This iteration is made explicit in this section, and is shown to be globally convergent with appropriate values of the parameters.

Without loss of generality, we take

$$c_0 = 1$$

to simplify the notation.

We now derive new lower bounds from the upper bounds. For this purpose, consider the equation

$$U^{'}(1 - \underline{\gamma}) + \frac{\alpha}{\beta}\underline{\gamma} = \overline{q}; \tag{33}$$

in the unknown $\underline{\gamma} \in [0, 1]$. The left hand side is increasing in $\underline{\gamma}$, and $\overline{q} > U^{'}(1)$, so for large enough $\frac{\alpha}{\beta}$, the equation will have a (unique) solution, strictly less than 1. In the following, we will assume $\alpha < 1$ is fixed, and choose $\beta_0 > 0$ such that the above equation has a solution

$$\underline{\gamma}^0 < 1.$$

Then for any $\beta \leq \beta_0$, and the fixed $\alpha$, the equation (33) will have a root $\underline{\gamma} < \underline{\gamma}^0$; we will use the notation

$$\underline{\gamma} = \Gamma_2^{\beta}(\overline{q})$$

to denote this unique solution of (33), making explicit its dependence on $\beta$. Then we can obtain a refined (nontrivial !) lower bound for $x$, in a dual way to Proposition 6.

**Proposition 8.** *Assume* $\beta \leq \beta_0$ *as defined above, and* $\underline{\gamma} = \Gamma_2^{\beta}(\overline{q})$. *Then, after a finite time, x is lower bounded by*

$$\underline{x} = (1-\underline{\gamma})e^{\alpha(1-\underline{\gamma}-\overline{x})}. \tag{34}$$

**Proof:** Assume the bounds $\overline{q}$ and $\overline{x}$ are valid for $t \geq t_1$, and that $x(t_1) \leq 1-\underline{\gamma}$ (if the latter fact never ocurred, our bound will be automatically satisfied). Then for $t \in [t_1, t_1+\tau]$, we have

$$\begin{aligned} \dot{x} &\geq x\frac{\beta}{\tau}\Big(U'(1-\underline{\gamma}) - \overline{q} - \frac{\alpha}{\beta}(\overline{x}-1)\Big) \\ &= -x\frac{\alpha}{\tau}(\underline{\gamma}+\overline{x}-1). \end{aligned} \tag{35}$$

Also, for $t > t_1+\tau$, we have $\dot{q} \leq -\underline{\gamma}$ so $q(t) \leq \overline{q} - \underline{\gamma}(t-t_1-\tau)$, and therefore

$$\begin{aligned} \dot{x} &\geq x\frac{\beta}{\tau}\left(U'(1-\underline{\gamma}) - (\overline{q} - \underline{\gamma}(t-t_1-\tau)) + \frac{\alpha}{\beta}\underline{\gamma}\right) \\ &= x\frac{\beta\underline{\gamma}}{\tau}(t-(t_1+\tau)) \\ &> 0. \end{aligned}$$

We remark that if the "positive-projection" on $\dot{q}(t)$ were to become active during this period, it would give $q = \dot{q} = 0$ in some interval, and therefore $\dot{x} > 0$, so the same conclusion holds. This means $x$ will begin to increase before time $t_1+\tau$, and must do so until it reaches $1-\underline{\gamma}$.

If, later there is another time where $x$ decreases through this value, it again can only do so for a time no more than $\tau$; integrating the bound (35) over this length of time gives to the overall lower bound (34), that must hold henceforth. □

The next step is to obtain a refined lower bound for $q$.

**Proposition 9.** *For the system (22-24), suppose* $x(t) \in [\underline{x}, \overline{x}]$ *after a certain time. Then, for any* $\varepsilon > 0$, *q will be lower bounded by*

$$\underline{q} = [U'(\overline{x}) + \frac{\tau}{\alpha}\log(\underline{x}) + (\underline{x}-1)\tau - \varepsilon]^+ \tag{36}$$

*after a finite time.*

The proof is similar to that for Proposition 7; details are omitted. The only change is that the positive projection of $q$ might become active, as indicated above by the $[\,]^+$ operation. In that case, the lower bound will not be a refinement.

$\underline{x}$ and $\underline{q}$ are both decreasing functions of $\beta$. If we take $\beta \leq \beta_0$ as before, we can have absolute lower bounds for $\underline{\gamma}$, $x$ and $q$:

$$\underline{\gamma}^0, \quad \underline{x}^0, \quad \underline{q}^0.$$

The procedure is now turned into an iteration for the system (22-24), by means of the following claims, which are routine extensions of the above.

- If $x(t)$ and $q(t)$ were lower bounded by $\underline{q}_{k-1} \le U^{'}(1)$ and $\underline{x}_{k-1} \le 1$ respectively, $x$ is also upper bounded by

$$\overline{x}_k = (1+\overline{\gamma}_k)e^{\alpha(1+\overline{\gamma}_k-\underline{x}_{k-1})} =: R_1(\overline{\gamma}_k, \underline{x}_{k-1})$$

after a finite time, where $\overline{\gamma}_k = \Gamma_1^{\beta}(\underline{q}_{k-1})$ is the solution of the equation

$$U^{'}(1+\overline{\gamma}_k) - \frac{\alpha}{\beta}\overline{\gamma}_k = \underline{q}_{k-1}. \tag{37}$$

- For given bounds of $x(t)$ as $\underline{x}_{k-1} \in [0,1]$ and $\overline{x}_k \ge 1$, and any $\varepsilon_k > 0$, $q(t)$ is upper bounded by

$$\overline{q}_k = U^{'}(\underline{x}_{k-1}) + \frac{\tau}{\alpha}\log(\overline{x}_k) + (\overline{x}_k - 1)\tau + \varepsilon_k =: Q_1(\overline{x}_k, \underline{x}_{k-1}, \varepsilon_k),$$

after a finite time.

- For given upper bounds $\overline{q}_k \ge U'(1)$, $\overline{x}_k \ge 1$, $x$ is lower bounded by

$$\underline{x}_k = (1-\underline{\gamma}_k)e^{\alpha(1-\underline{\gamma}_k-\overline{x}_k)} =: R_2(\overline{x}_k, \underline{\gamma}_k),$$

after a finite time, provided that $\underline{\gamma}_k = \Gamma_2^{\beta}(\overline{q}_k)$ is the solution of the equation

$$U^{'}(1-\underline{\gamma}_k) + \frac{\alpha}{\beta}\underline{\gamma}_k = \overline{q}_k. \tag{38}$$

- For given bounds $\underline{x}_k \in (0,1]$ and $\overline{x}_k \ge 1$, and any $\varepsilon_k > 0$, $q(t)$ is upper bounded by

$$\underline{q}_k = [U^{'}(\overline{x}_k) + \frac{\tau}{\alpha}\log(\underline{x}_k) + (\underline{x}_k - 1)\tau - \varepsilon_k]^{+} := Q_2(\overline{x}_k, \underline{x}_k, \varepsilon_k),$$

after a finite time.

In summary, we have the iteration as follows, for $k \ge 1$,

$$\overline{\gamma}_k = \Gamma_1^{\beta}(\underline{q}_{k-1}); \tag{39}$$
$$\overline{x}_k = R_1(\underline{x}_{k-1}, \overline{\gamma}_k); \tag{40}$$
$$\overline{q}_k = Q_1(\overline{x}_k, \underline{x}_{k-1}, \varepsilon_k); \tag{41}$$
$$\underline{\gamma}_k = \Gamma_2^{\beta}(\overline{q}_k); \tag{42}$$
$$\underline{x}_k = R_2(\overline{x}_k, \underline{\gamma}_k); \tag{43}$$
$$\underline{q}_k = Q_2(\overline{x}_k, \underline{x}_k, \varepsilon_k). \tag{44}$$

Since there is no restriction on $\varepsilon_k$, we can take it as any sequence that monotonically decreases to zero. Moreover, we have the initial values

$$\underline{x}_0 = 0, \ \underline{q}_0 = 0. \tag{45}$$

Note that we are not making explicit the times after which each of these bounds is active, which would be quite involved. However it is clear that if we can prove that the iteration converges to the equilibrium point

$$\overline{\gamma}^* = \underline{\gamma}^* = 0,\ \overline{x}^* = \underline{x}^* = 1,\ \overline{q}^* = \underline{q}^* = U^{'}(1),\ \varepsilon^* = 0, \tag{46}$$

we will have shown global asymptotic stability. To prove the above, we will first argue that the iteration must converge to a fixed point, and then show that for $\alpha < 1$ and small enough $\beta$, the fixed point is unique.

Note that $\underline{x}_1$ and $\underline{q}_1$ are refined lower bounds. We first show in the following proposition that this is true for all $k \geq 1$ if $\beta \leq \beta_0$.

**Proposition 10.** *Assume* $\beta \leq \beta_0$, $\varepsilon_k \to 0$ *monotonically. Then the elements of the sequences (39-45) converge to certain values*

$$\overline{\gamma} \in [0, \overline{\gamma}^0],\ \underline{\gamma} \in [0, \underline{\gamma}^0], \tag{47}$$

$$\overline{x} \in [1, \overline{x}^0],\ \underline{x} \in [\underline{x}^0, 1] \tag{48}$$

$$\overline{q} \in [U'(1), \overline{q}^0],\ \underline{q} \in [\underline{q}^0, U'(1)]. \tag{49}$$

**Proof:** We will rely on monotonicity. Note first that since $\beta \leq \beta_0$, we have $\underline{\gamma}_1 < 1$ and thus

$$\underline{x}_1 > \underline{x}_0 = 0,\ \underline{q}_1 \geq \underline{q}_0.$$

This in turn gives

$$\overline{\gamma}_2 \leq \overline{\gamma}_1,\ \overline{x}_2 < \overline{x}_1,\ \overline{q}_2 < \overline{q}_1,\ \underline{\gamma}_2 < \underline{\gamma}_1.$$

We can now repeat the argument to show that $\overline{\gamma}_k$, $\underline{\gamma}_k$, $\overline{x}_k$, $\overline{q}_k$ are monotonically decreasing, and $\underline{x}_k$, $\underline{q}_k$ are monotonically increasing. Since the former are lower bounded (respectively by 0, 0 , 1 and $U'(1)$) and the latter are upper bounded (respectively by 1 and $U'(1)$), we find they must have limits as claimed. Also, the limits must be constrained by the absolute bounds we defined based on $\beta_0$. □

We must now show that, under appropriate parameter conditions, the above limit point is necessarily the one in (46). By continuity, it follows that our fixed point must satisfy the equations

$$\overline{x} = R_1(\underline{x}, \overline{\gamma}),\ \underline{x} = R_2(\overline{x}, \underline{\gamma}), \tag{50}$$

$$\overline{q} = Q_1(\overline{x}, \underline{x}, 0),\ \underline{q} = Q_2(\overline{x}, \underline{x}, 0), \tag{51}$$

$$\overline{\gamma} = \Gamma_1^{\beta}(\underline{q}),\ \underline{\gamma} = \Gamma_2^{\beta}(\overline{q}). \tag{52}$$

Note that (50) implicitly defines a pair $\overline{x}$, $\underline{x}$ for each $\overline{\gamma}$, $\underline{\gamma}$. The next result shows that for $\alpha < 1$, we can make this map explicit.

**Proposition 11.** *Assume that* $\alpha < 1$. *Then for each* $\overline{\gamma}$, $\underline{\gamma}$, *there is a unique*

$$(\overline{x}, \underline{x}) = X(\overline{\gamma}, \underline{\gamma})$$

*satisfying (50). Furthermore, the mapping X has bounded derivatives over the domain (47).*

The proof is given in the Appendix.

Now it is time to claim the main theorem on stability.

**Theorem 4.** *Suppose Assumption 1 holds. There exists* $\beta^* > 0$ *such that the system (22-24) is globally asymptotically stable if* $\alpha < 1$ *and* $\beta < \beta^*$.

**Proof:** For $\alpha < 1$ and $\beta \leq \beta_0$, the above analysis implies that our sequence of bounds converges to a point satisfying

$$\begin{aligned}(\overline{x}, \underline{x}) &= X(\overline{\gamma}, \underline{\gamma}),\\ (\overline{q}, \underline{q}) &= Q(\overline{x}, \underline{x}),\\ (\overline{\gamma}, \underline{\gamma}) &= \Gamma^{\beta}(\overline{q}, \underline{q}).\end{aligned}$$

Here the functions $Q$ and $\Gamma^{\beta}$ have been defined from their components in (51) and (52) in the natural way. Therefore the limits $(\overline{\gamma}, \underline{\gamma})$ must be a fixed point of the mapping

$$\Gamma^{\beta} \circ Q \circ X.$$

Of the three components of the above composition, note that the last two do not depend on the parameter $\beta$. Furthermore, we can give bounds of the form

$$\begin{aligned}\|X(\overline{\gamma}, \underline{\gamma}) - X(\overline{\gamma}', \underline{\gamma}')\| &\leq K_X \|(\overline{\gamma}, \underline{\gamma}) - (\overline{\gamma}', \underline{\gamma}')\|\\ \|Q(\overline{x}, \underline{x}) - Q(\overline{x}', \underline{x}')\| &\leq K_Q \|(\overline{x}, \underline{x}) - (\overline{x}', \underline{x}')\|\end{aligned}$$

on how much they can increase distances in their respective domains defined in (47) and (48), measured in some norm in $\mathbf{R}^2$. This follows directly through bounds in their derivatives; only aspects to note are:

- The derivatives of $X$ are bounded from Proposition 11.
- The domain of $\underline{x}$ keeps it away from zero, where the logarithm in $Q_2$ would have unbounded derivative.
- The positive projection $[\,]^+$ is a bounded operation.

We now bound the distance growth of the mapping $\Gamma^{\beta}$. Referring back to (37-38), we find that

$$\begin{aligned}\frac{\partial \Gamma_1^{\beta}}{\partial \underline{q}} &= \frac{1}{U''(1+\overline{\gamma}) - \frac{\alpha}{\beta}},\\ \frac{\partial \Gamma_2^{\beta}}{\partial \overline{q}} &= \frac{1}{-U''(1-\overline{\gamma}) + \frac{\alpha}{\beta}}.\end{aligned}$$

As $\beta \to 0$, both of the above derivatives converge to zero, uniformly in their domain of definition (49). So we can write, over this domain, the bound

$$\|\Gamma^{\beta}(\overline{q}, \underline{q}) - \Gamma^{\beta}(\overline{q}', \underline{q}')\| \leq K(\beta) \|(\overline{q}, \underline{q}) - (\overline{q}', \underline{q}')\|.$$

By choosing $0 < \beta^* \leq \beta_0$ such that

$$K(\beta^*)K_Q K_X < 1,$$

we find that the composition mapping $\Gamma^\beta \circ Q \circ X$ is a contraction. Therefore, it can have at most one fixed point. Since we already know that (46) is a fixed point, this must be the unique one. The convergence of our sequence of bounds to this point implies global asymptotic stability of the system. □

We remark that a similar result, for sufficiently small $\beta$, was obtained in [17] using singular perturbation methods; however the present one is considerably tighter in regard to conditions on $\alpha$.

## 5 Conclusion

The global performance of two nonlinear congestion control laws with time-delay is studied in this chapter, for the case of a single-bottleneck network. "Dual" laws were studied for the case of multiple heterogeneous sources, and "primal-dual" laws only for the single source case. We found absolute bounds for the state variables, valid after a sufficient amount of time, which itself can be bounded; these hold regardless of any stability restriction. We subsequently showed how these bounds could be successively refined, generating an iteration for each case. Under appropriate parameter conditions, we showed convergence of these iterations, which implies conditions on global exponential or asymptotic stability for the nonlinear time-delay systems. In the case of primal-dual laws, which have a source utility function as a degree of freedom, our stability result requires bounded derivative (marginal utility); this restriction might not be essential, but was required for our method of proof. An extension beyond this case remains open for future research.

## Appendix: Proof of Proposition 11

Consider the solution of the equations (50) for given $\overline{\gamma}$ and $\underline{\gamma}$ defined in (47).

We first show that the solution is unique for $\overline{x} \geq 1$ and $\underline{x} \leq 1$. This is equivalent to the uniqueness of the solution of the one-dimensional equation

$$\underline{x} = R_2(R_1(\underline{x}, \underline{\gamma}), \overline{\gamma}),$$

in the domain $\underline{x} \in [0,1]$. Denoting the composite function by $Z$, we rewrite the above as

$$\underline{x} = Z(\underline{x}) = (1-\underline{\gamma})\mathrm{e}^{\alpha[1-\underline{\gamma}-(1+\overline{\gamma})\mathrm{e}^{\alpha[1+\overline{\gamma}-\underline{x}]}]}. \tag{53}$$

For $\overline{\gamma} \geq 0$, $\underline{\gamma} \in [0,1)$, note that

$$Z(0) = (1-\underline{\gamma})e^{\alpha[1-\underline{\gamma}-(1+\overline{\gamma})e^{\alpha(1+\overline{\gamma}))}]} > 0, \tag{54}$$

$$Z(1) = (1-\underline{\gamma})e^{\alpha[1-\underline{\gamma}-(1+\overline{\gamma})e^{\alpha\overline{\gamma}}]} \leq 1, \tag{55}$$

$$\frac{\mathrm{d}Z}{\mathrm{d}\underline{x}} = Z(\underline{x})\alpha^2(1+\overline{\gamma})e^{\alpha(1+\overline{\gamma}-\underline{x})} > 0. \tag{56}$$

Moreover, we argue that $\frac{dZ}{d\underline{x}} < 1$ for any $\underline{x} \in [0,1]$ if $\alpha < 1$. Actually,

$$\begin{aligned}\frac{dZ}{d\underline{x}} &= (1-\underline{\gamma})e^{\alpha[1-\underline{\gamma}-(1+\overline{\gamma}_k)e^{\alpha(1+\overline{\gamma}-\underline{x})}]}\alpha^2(1+\overline{\gamma})e^{\alpha(1+\overline{\gamma}-\underline{x})} \\ &= \alpha(1-\underline{\gamma})e^{\alpha(1-\underline{\gamma})}\alpha(1+\overline{\gamma})e^{\alpha(1+\overline{\gamma}_k)}e^{-\alpha g(\underline{x})},\end{aligned}$$

where we have defined

$$g(\underline{x}) = \underline{x} + (1+\overline{\gamma})e^{\alpha(1+\overline{\gamma})}e^{-\alpha\underline{x}}.$$

Since

$$\frac{d^2 g}{d\underline{x}^2} = \alpha^2(1+\overline{\gamma})e^{\alpha(1+\overline{\gamma}-\underline{x})} > 0,$$

$g(\underline{x})$ is a strictly convex function which has a global minimum point $g(\underline{x}^*)$; to find this value we set $\frac{dg}{d\underline{x}} = 0$ and obtain

$$\alpha(1+\overline{\gamma})e^{\alpha(1+\overline{\gamma})}e^{-\alpha\underline{x}^*} = 1,$$

from where

$$g(\underline{x}^*) = \underline{x}^* + \alpha^{-1}.$$

We can now use this lower bound of $g(\underline{x})$ to find the *upper* bound

$$\begin{aligned}\frac{dZ}{d\underline{x}} &\le \alpha e^{\alpha}\alpha(1+\overline{\gamma})e^{\alpha(1+\overline{\gamma})}e^{-\alpha(\underline{x}^*+\alpha^{-1})} \\ &= \alpha e^{\alpha}e^{-1} \\ &< 1,\end{aligned}$$

for $\alpha < 1$. From this bound and (54-55) we conclude that, for each $\overline{\gamma}$ and $\underline{\gamma}$, the equation $Z(\underline{x}) = \underline{x}$ has a unique solution in $\underline{x} \in (0,1]$, which we denote by

$$\underline{x} = X_2(\overline{\gamma},\ \underline{\gamma}).$$

Now substitution into $\overline{x} = R_1(\underline{x},\ \underline{\gamma})$ gives a unique solution for $\overline{x}$, denoted by

$$\overline{x} = X_1(\overline{\gamma},\ \underline{\gamma}).$$

It remains to show that the mapping defined by $(X_1,\ X_2)$ has bounded derivatives over the domain (47). We obtain these derivatives by differentiating (50), and solving the implicit equations to get

$$\begin{bmatrix} d\overline{x} \\ d\underline{x} \end{bmatrix} = \frac{1}{1-\alpha^2\overline{x}\underline{x}} \begin{bmatrix} m_1 & \alpha m_2\overline{x} \\ -\alpha m_1\underline{x} & -m_2 \end{bmatrix} \cdot \begin{bmatrix} d\overline{\gamma} \\ d\underline{\gamma} \end{bmatrix},$$

where

$$m_1 = \overline{x}(\alpha + \frac{1}{1+\overline{\gamma}}),$$
$$m_2 = \underline{x}(\alpha + \frac{1}{1-\underline{\gamma}}).$$

Note that, referring to (56) we have

$$\begin{aligned}\alpha^2 \underline{x}\overline{x} &= \alpha^2 Z(\underline{x})(1+\overline{\gamma})e^{\alpha(1+\overline{\gamma}-\underline{x})} \\ &= \frac{dZ}{d\underline{x}} \\ &\leq \alpha^{-1}e^{\alpha-1} \\ &< 1,\end{aligned}$$

so $(1-\alpha^2\underline{x}\overline{x})^{-1}$ remains bounded. All the other elements are bounded as well over the domain (47). Therefore the partial derivatives are bounded as was to be proved. □

## References

[1] Kelly FP, Maulloo A, Tan D (1998) Rate control for communication networks: shadow prices, proportional fairness and stability. Journal of Operations Research Soceity, vol. 49, no.3, pp 237-252

[2] Low SH, Lapsley DE (1999) Optimization flow control, I: basic algorithm and convergence. IEEE/ACM Transactions on Networking, vol.7, no.6, pp861-874

[3] Johari R, Tan D (2001) End-to-end congestion control for the Internet: delays and stability. IEEE/ACM Transactions on Networking, 9, 818-832

[4] Paganini F, Doyle JC, Low SH (2001) Scalable laws for stable network congestion control. Proceedings of Conference on Decision and Control

[5] Hollot C, Misra V, Towsley D, Gong WB (2001) A control theoretic analysis of RED. Proceedings of IEEE Infocom

[6] Vinnicombe G (2002) On the stability of networks operating TCP-like congestion control. Presented at 15th IFAC World Congress, Barcelona, Spain

[7] Kunniyur S, Srikant R (2003) Stable, scalable, fair congestion control and AQM schemes that achieve high utilization in the Internet. IEEE Trans. on Automatic Control, Vol. 48, No11., pp. 2024-2029

[8] Jin C, Wei D, Low SH, Buhrmaster G, Bunn J, Choe DH, Cottrell RLA, Doyle JC, Newman H, Paganini F, Ravot S, Singh S (2003) FAST kernel: background theory and experimental results. Presented at the First International Workshop on Protocols for Fast Long-Distance Networks, February 3-4, CERN, Geneva, Switzerland

[9] Kelly T (2002) Scalable TCP: improving performance in highspeed wide area networks. Submitted for publication.

[10] Paganini F (2002) A Global Stability Result in Network Flow Control. Systems & Control Letters 46 (3) pp. 153-163

[11] Kunniyur S, Srikant R (2001) A time-scale decomposition approach to adaptive ECN marking. Proceedings of IEEE infocom
[12] Wen JT, Arcak M (2003) A unifying passivity framework for network flow control. Proceedings of IEEE Infocom
[13] Paganini F, Wang ZK, Low S, Doyle J (2003) A new TCP/AQM for stable operation in fast networks. Proceedings of IEEE Infocom
[14] Hollot CV, Chait Y (2001) Nonlinear stability analysis for a class of TCP/AQM schemes. Proceedings of IEEE Conference on Decision and Control
[15] Deb S, Srikant R (2003) Global stability of congestion controllers for the Internet. IEEE Transactions on Automatic Control, vol. 48, no. 6, June, pages 1055-1060
[16] Wang ZK, Paganini F (2002) Global stability with time delay in network congestion control. Proceedings of IEEE Conference on Decision and Control, Las Vegas, Nevada
[17] Wang ZK, Paganini F (2003) Global stability with time-delay of a primal-dual congestion control. Proceedings of IEEE Conference on Decision and Control, Maui, Hawii
[18] Krasovskii NN, Brenner JL (1963) Stability of motion. Stanford University Press
[19] Halanay A (1966) Differential equations: stability, oscillations, time lags. Academic Press
[20] Lakshmikantham V, Leela S (1969) Differential and integral inequalities: theory and applications, Vol. II. Academic Press
[21] Dugard L, Verriest EI (1997) Stability and Control of Time-delay Systems. Springer-Verlag, New York
[22] Hale JK, Lunel SMV (1993) Introduction to functional differential equations. Springer-Verlag
[23] Niculescu S (2001) Delay effects on stability: a rebust control approach. Springer
[24] Fan XZ, Arcak M, Wen JT (2003) L-p stability and delay robustness of network flow control. Proceeding of IEEE Conference on Decision and Control , Maui
[25] Paganini F, Wang ZK, Doyle J, Low S (2004) Congestion control for high performance, stability and fairness in general networks. To appear in IEEE/ACM Transactions on Networking
[26] Wright EM (1955) A nonlinear difference-differential equation. Journal fur die reine undangewandte Mathematik

# On the Optimization of Load Balancing in Distributed Networks in the Presence of Delay

Sagar Dhakal[1], Majeed M. Hayat[1], Jean Ghanem[1], Chaouki T. Abdallah[1], Henry Jerez[1], John Chiasson[2], and J. Douglas Birdwell[2]

[1] ECE Dept, University of New Mexico, Albuquerque NM 87131-1356, USA
{dhakal,hayat,jean,chaouki,hjerez}@ece.unm.edu
[2] ECE Dept, University of Tennessee, Knoxville TN 37996, USA
{chiasson,birdwell}@utk.edu

## 1 Introduction

Distributing the total computational load across available processors is referred to as *load balancing* in the literature. A typical distributed system consists of a cluster of physically or virtually distant and independent computational elements (CEs), which are linked to one another by some communication medium. The workload has to be distributed over all the available CEs in proportion to their processing speed and availability such that the overall work completion time is minimized. In most practical distributed systems, due to the unknown character of the incoming workload, the nodes exhibit non-deterministic run-time performance. Thus, it is advantageous to perform the load balancing periodically during a run-time so that the run-time variability is minimized. This is referred to in the literature as *dynamic load balancing* [1]. However, the frequent load balancing requires the periodic communication (and transfer of loads, of course) between the CEs so that the shared knowledge of the load state of the system can be used by individual CEs to judiciously assign an appropriate fraction of the incoming loads to less busy CEs according to some load-balancing policy.

Not surprisingly, the expected performance of this dynamic allocation of the workload among the CEs relies heavily on the time-delay factors of the physical medium. For example, in shared communication medium (such as the Internet or a wireless LAN), there is an inherent delay in the inter-node communications and transfer of loads. As a result, each node has dated knowledge of any other node in the system and each receives its share of the load after a delayed instant of time. It has been almost universally observed that the types of delay described above degrade the overall performance [1, 2, 3, 4, 5]. In particular, communication delays may lead to unnecessary transfer of loads and large load-transfer delays may lead to a situation where much time is wasted on transferring loads, back and forth, while certain CEs may be idle during the transfer. In other words, a situation may arise where a fraction of the work load remains trapped in transit.

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 223–244, 2004.

Many load-balancing policies have been proposed for distributed-system categories. These include local versus global, static versus dynamic, and centralized versus distributed scheduling [1, 2, 3]. Some of the existing approaches consider constant performance of the network while others consider deterministic communication and transfer delay. In [4] and [5], it is assumed that the communication channels have fixed delay times and the load balancing is completed within a finite interval. However, in actuality such delays vary according to the size of the loads to be transferred and also fluctuate due to the random condition of the communication medium that connects the CEs. This introduces the uncertainty in knowledge among the nodes and hence, degrades the performance of any load balancing policy designed for the deterministic delay case. In Fig. 1, a simple comparison is made between the deterministic and random communication delay effect on the "knowledge state" of the nodes. In the deterministic case, at any time node 2 receives a communication from node 1, it obtains new information about the queue length of node 1, delayed, however, by a fixed amount of time $t_c$. In contrast, when the delay is random, every time node 2 receives a communication from node 1, there is no guarantee that the received message carries the most recent state of node 1. Similarly, the randomness in the load-transfer delay leads to more uncertainty in the overall completion time. The limitations and the overheads involved with the implementation of the deterministic-delay load-balancing policy in a random delay environment has been discussed previously by the authors [6, 7, 8]. The degraded performance is evident from one of our earlier results, which is presented here in Fig. 2. This result was generated using Monte-Carlo simulation with the assumption that 12000 tasks were initially distributed unevenly among three CEs, each of which can process one task every 10 $\mu$s. The communication delay and the transfer delay per task were each set to be uniformly distributed in the intervals (0, 16 ms) and (0, 32ms), respectively. The deterministic-delay load-balancing policy calls for continuous execution of the scheduling algorithm. When this policy was applied, the randomness in delay led to an unnecessary exchange of tasks between CEs which results in an undesirable oscillatory behavior near the tail of the queue.

Recently, a Monte-Carlo technique [7] was used to investigate a dynamic load balancing scheme for distributed systems which incorporates the stochastic nature of the delay in both communication and load transfer. It was shown that indeed there is an interplay between the stochastic delay (e.g., its mean and its dependence on the load) and the strength and frequency of balancing. Moreover, it has been shown that limiting the number of load-balancing instants (in an effort to avoid the unnecessary exchange of loads between CEs and reduce communication overhead) while optimizing the strength of the load balancing and the actual load-balancing instants is a feasible solution to the problem of load balancing in a delay-limited environment. It was also observed that by fixing the number of load balancing instants, the performance of the balancing policy becomes very sensitive to the balancing instant (measured relative to the time when load arrives at the system). This is primarily due to the observation that it is more advantageous to balance at a delayed time, up to a point, simply because the CEs will have more time to exchange their respective load states, thereby allowing for "better-informed" load balancing.

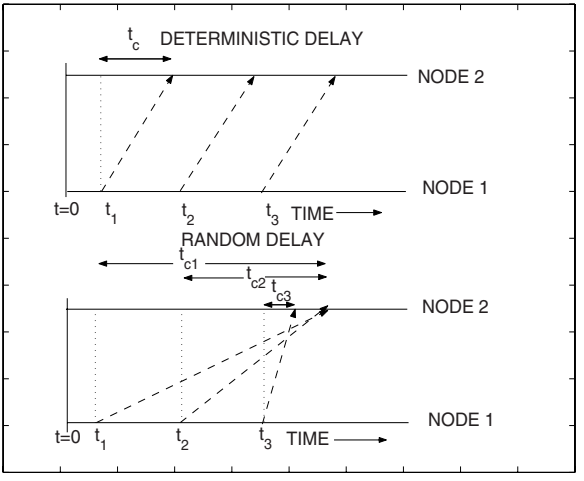



**Fig. 1.** A schematic comparison of the random-delay case with the deterministic-delay case. The dashed lines represent the communication sent from node 1 to node 2.

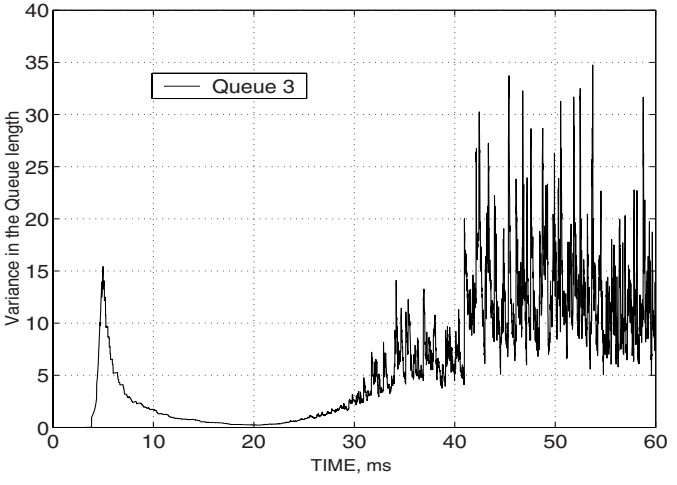



**Fig. 2.** The variance of the queue size in node 3 (in a three-node system) as a function of time. High uncertainty in the queue size is evident near the tail. The random delay for transferring small amounts of tasks back and forth causes this oscillation.

One strategy which we have explored is the single-instant load-balancing policy, where upon the arrival of a workload, the load-balancing policy decides on when and how to execute a single load-balancing attempt. This policy has been implemented on a real distributed system consisting of a wireless LAN as well as PlanetLab. In this Chapter, we evaluate the performance of this policy with respect to the balancing instant and the load-balancing gain parameter. We show that the choice of the

balancing strategy is an optimization problem with respect to the balancing instant and the load-balancing gain parameter. We then utilize the delay statistics, obtained from our experiments, to generate the simulations using custom-made Monte Carlo software. The experimental results are compared to the results from the simulation to verify the predictive capability of our model. Finally, an approach for modeling the distributed system dynamics, as a queuing system, is introduced using the concept of regeneration in stochastic processes. We define the events (called regeneration events) whose occurrence will regenerate the queues with similar statistical properties and dynamics. More specifically, on every arrival or departure of a task at any of the nodes, the system of queues is said to be regenerated because they preserve the same stochastic dynamics of the queues prior to this arrival/departure event but will start from different set of initial conditions. The initial conditions consist of the initial load of queues as well as their knowledge state of each other. To this end, a novel concept of "information states" of the queues is introduced. We obtain four difference-differential equations which characterize the mean of overall completion time of a two-node system. For brevity, we consider only the zero-input response of the queues, and therefore the arrival of tasks at a node beyond the initial time is solely a result of load transfer from other nodes. Based on this analytical model, we present some initial results showing the optimization of the mean of the overall completion time with respect to the balancing gain. The system model, which is developed here for the special case of two CEs, also maintains the gist of the solution for the multiple-CE problem and conveys the underlying principles of our analytical solution while keeping the algebra at a minimum. The approach, nonetheless, can be extended to the multi-CE case in a straightforward fashion.

This chapter is organized as follows. Section 2 contains the description of the load-balancing policy. The experimental results, on a physical wireless 3-node LAN, are given in Section 3. The simulation results are presented in Section 4 and the stochastic analysis of the distributed system is detailed in Section 5. Conclusions and future extensions are discussed in Section 6.

## 2 Description of the Load Balancing Policy

We begin by briefly describing the queuing model that characterizes the stochastic dynamics of the load balancing problem described in [6]. Suppose that we have a cluster of $n$ nodes. Let $Q_i(t)$ denote the number of tasks awaiting processing at the $i$th node at time $t$. Assume that the $i$th node completes tasks according to a Poisson process and at a constant rate $\mu_i$. Let the counting process $J_i(t_1, t_2)$ denote the number of external tasks (requests) arriving at node $i$ in the interval $[t_1, t_2)$. (For example, to accommodate the bursty nature of workload arrivals, we can think of $J_i(t_1, t_2)$ as a compound Poisson process [9].) The load transfer between nodes is accomplished as follows. The $i$th node, at the $\ell$th specific load-balancing instant $t_l^i$, looks at its own load $Q_i(t_l^i)$ and the loads of other nodes (at randomly delayed instants due to communication delays), and decides whether it should allocate a fraction $K$ of its excess load to the other nodes according to a deterministic policy. On the other hand,

at the time when it is not attempting to load balance, it may receive loads from the neighboring nodes subject to random delays (due to the load-transfer delays).

We now write the dynamics of the $i$th queue in a differential form as follows:

$$Q_i(t+\Delta t) = Q_i(t) - C_i(t,t+\Delta t) - \sum_{j\neq i}\sum_{\ell} L_{ji}(t_l^i) I_{[t_l^i, t_l^i+\Delta t)}(t) + \sum_{j\neq i}\sum_{k} L_{ij}(t_k^j) I_{[t_k^j-\tau_{ij,k}, t_k^j-\tau_{ij,k}+\Delta t)}(t) + J_i(t,t+\Delta t), \tag{1}$$

where $I_A$ is an indicator function for a set $A$, $C_i(t,t+\Delta t)$ is a Poisson process (with rate $\mu_i$) describing the random number of tasks completed in the interval $[t,t+\Delta t)$, and $\tau_{ij,k}$ is the delay in transferring load $L_{ij}(t_k^j)$ from node $j$ to node $i$ at the $k$th load balancing instant of node $j$. More precisely, for $k \neq \ell$, the random load $L_{k\ell}(t)$ diverted from node $\ell$ to node $k$ has the form $L_{k\ell}(t) \stackrel{\triangle}{=} g_{k\ell}(Q_\ell(t), Q_k(t-\eta_{\ell k}), \ldots, Q_j(t-\eta_{\ell j}), \ldots)$, where for any $j \neq k$, $\eta_{kj} = \eta_{jk}$ is the communication delay between the $k$th and $j$th nodes (with the obvious convention $\eta_{ii} = 0$). The function $g_{k\ell}$ dictates the load-balancing policy between the $k$th and $\ell$th nodes. In this chapter, we will use the special form:

$$g_{k\ell}(Q_\ell(t), Q_k(t-\eta_{\ell k}), \ldots, Q_j(t-\eta_{\ell j}), \ldots) = K p_{k\ell} \left( Q_\ell(t) - n^{-1} \sum_{j=1}^{n} Q_j(t-\eta_{\ell j}) u(t-\eta_{\ell j}) \right) \cdot u\left( Q_\ell - n^{-1} \sum_{j=1}^{n} Q_j(t-\eta_{\ell j}) u(t-\eta_{\ell j}) \right), \tag{2}$$

where $u(\cdot)$ is the unit step function, $K$ is a gain parameter that controls the overall strength of load balancing, and $p_{k\ell}$ is the fraction of the excess load at node $i$ to be sent to $k$th node ($\sum_{k\neq\ell} p_{k\ell} = 1$). Narratively, in this policy the $\ell$th node simply compares its load to the average over all load and sends out a fraction $p_{k\ell}$ of its excess load to the $k$th node. Finally, the fractions $p_{k\ell}$ are defined as (assuming $n \geq 3$):

$$p_{k\ell} = \begin{cases} \frac{1}{n-2}\left(1 - \frac{Q_k(t-\eta_{\ell k})}{\sum_{i\neq\ell} Q_i(t-\eta_{\ell i})}\right), & k \neq \ell, \\ \frac{1}{n-1}, & \text{otherwise} \end{cases} \tag{3}$$

In this definition, a node sends a larger fraction of its excess load to a node with a small load relative to all other candidate recipient nodes. For the special case when $n = 2$, $p_{k\ell} = 1$, where $k \neq \ell$. In all the examples considered later in this chapter, only one load-balancing execution is permitted. That is, $t_l^i = \infty$, for all $l \geq 2$ and all $i \geq 1$.

## 3 Experimental Results

We have developed an in-house wireless testbed to study the effects of the gain parameter $K$ as well as the selection of the load-balancing instant. We would like to

highlight the importance of these experiments since, to our knowledge, no previous work has been done in the optimization with respect to the load-balancing instant $t_b$ especially over a wireless network. The details of the system are described below.

### 3.1 Description of the experiments

The experiments were conducted over a 802.11b wireless network. The testing was completed on three computers: a 1.6 GHz Pentium IV processor machine (node 1) and two 1 GHz Transmeta Processor machines (nodes 2 & 3). To see the level and variability of the delays over our testbed, we computed the empirical probability density function (pdf) for the wireless-LAN system. The testing was performed by letting the nodes exchange a fixed size frame several times. The empirical pdf was computed and it is shown in Fig. 3. Note the initial almost-zero range, which is dictated by the physical lower bound for the delay, and the approximately exponential decay thereafter.

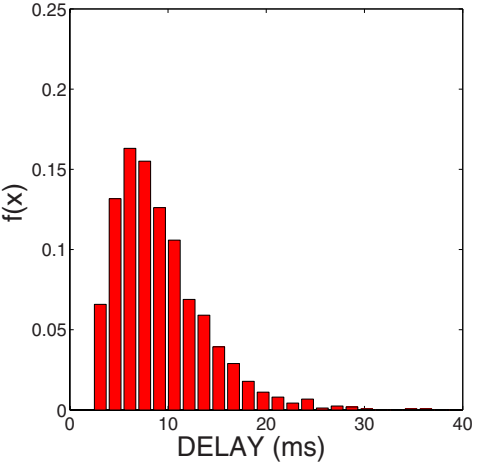



**Fig. 3.** Empirical estimate of the probability density function of the delay between two nodes (wireless LAN with an access point).

To scale the system to higher delay values proportional to the task execution time, the access point was kept busy by third-party machines, which continuously downloaded files. We consider the case where all nodes execute a single load balancing at a common balancing time $t_b$.

The application used to illustrate the load-balancing process was matrix multiplication, where one task has been defined as the multiplication of one row by a static matrix duplicated on all nodes (3 nodes in our experiment). The size of the elements in each row was generated randomly from a specified range, which made the execution time of a task variable. On average, the completion time of a task was 525 ms

on node 1, and 650 ms on the other two nodes. As for the communication part of the program, UDP was used to exchange queue size information among the nodes and TCP was used to transfer the data or tasks from one machine to another. The load-balancing policy used in the experiments is governed by Eqn. (3), for which the node decides whether or not to utilize either one of the other nodes according to its knowledge state of other nodes.

The aim of the first experiment is to optimize the overall completion time with respect to the balancing instant $t_b$ while fixing the gain value $K$ to 1. Each node was assigned a certain number of tasks according to the following distribution: Node 1 was assigned 60 tasks, node 2 was assigned 30 tasks, and node 3 was assigned 120 tasks. The information exchange delay (viz., communication delay) was on average 850 ms. Several experiments were conducted for each case of the load-balancing instant and the average was calculated using five independent realizations for each selected value of the load-balancing instant. In the second set of experiments, the load-balancing instant was fixed at 1.4 s and the experiments sought to find the optimal gain $K$ that minimized the overall completion time. The initial distribution of tasks was as follows: 60 tasks were assigned to node 1, 150 tasks were assigned to node 2, and 10 tasks were assigned to node 3. The average information exchange delay was 322 ms and the average data transfer delay per task was 485 ms.

### 3.2 Discussion of results

The results of the first set of experiments show that if the load balancing is performed blindly, as in the onset of receiving the initial load, the performance is poorest. This is demonstrated by the relatively large average completion time (namely 45 s $\sim$ 50 s) when the balancing instant is prior to the time when all the communication between the CEs have arrived (namely when $t_b$ is approximately below 1 s), as shown in Fig. 4. Note that the completion time drops significantly (down to 40 s) as $t_b$ begins to approximately exceed the time when all inter-CE communications have arrived (e.g., $t_b > 1.5$ s). In this scenario of $t_b$, the load balancing is done in an informative fashion, that is, the nodes have knowledge of the initial load of every CE. Thus, it is not surprising that the load balancing is more effective than the case the load balancing is performed on the onset of the initial load arrival for which the CEs have not yet received the state of the other CEs. The explanation for the sudden rise in the completion time for balancing instants between 0.5 s and 1 s is that the knowledge states in the system are "hybrid," that is, some nodes are aware of the queue sizes of the others while others aren't. When this hybrid knowledge state is used in the load-balancing policy (Eqn. (3)), the resulting load distribution turns out to be severely uneven across the nodes, which in turn, has an adverse effect on the completion time. Finally, we observe that as $t_b$ increases farther beyond the time all the inter-CE communications arrive (i.e., $t_b > 5$ s), then the average completion time begins to increase. This occurs precisely because any delay in executing the load balancing beyond the arrival of the inter-CE communications time would enhance the possibility that some CEs will run out of tasks in the period before any transferred load arrives to them.

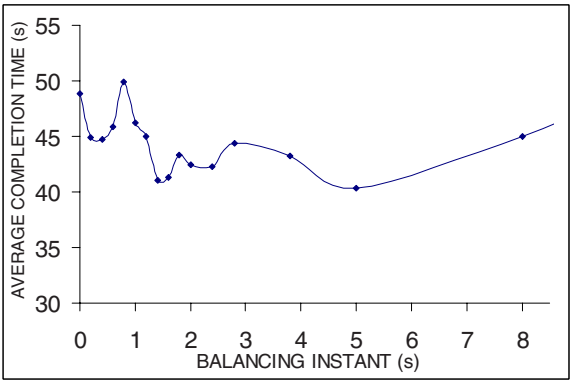



**Fig. 4.** Average total task-completion time as a function of the load-balancing instant. The load-balancing gain parameter is set at $K = 1$. The dots represent the actual experimental values and the solid curve is a best polynomial fit. This convention is used throughout Fig. 7.

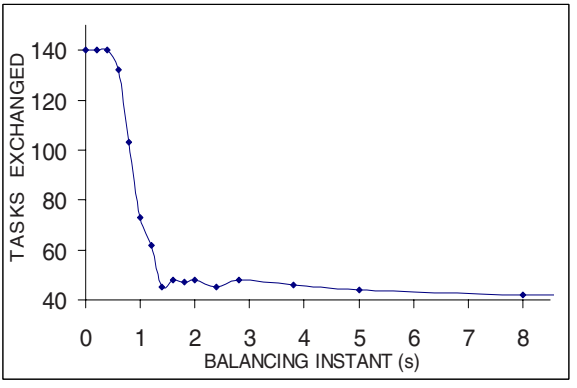



**Fig. 5.** Average total excess load decided by the load-balancing policy to be transferred (at the load-balancing instant) as a function of the balancing instant. The load-balancing gain parameter is set at $K = 1$.

Next, we examine the size of the loads transferred as a function of the instant at which the load balancing is executed, as shown in Fig. 5. This behavior shows the dependence of the size of the total load transferred on the "knowledge state" of the CEs. It is clear from the figure that for load-balancing instants up to approximately the time when all CEs have accurate knowledge of each other's load states, the average size of the load assigned for transfer is unduly large. Clearly, this seemingly "uninformed" load balancing leads to the waste of bandwidth on the interconnected network.

The results of the second set of experiments indeed confirm our earlier prediction (as reported in [7]) that when communication and load-transfer delays are prevalent, the load-balancing gain must be reduced to prevent "overreaction" (sending unnec-

essary excess load). This behavior is shown in Fig. 6, which demonstrates that the optimal performance is achieved not at the maximal gain ($K = 1$) but when $K$ is approximately 0.8. This is a significant result as it is contrary to what we would expect in a situation when the delay is insignificant (as in a fast Ethernet case), where $K = 1$ yields the optimal performance. Figure 7 shows the dependence of the total load to be transferred as a function of the gain. A large gain (near unity) results in a large load to be transferred, which, in turn, leads to a large load-transfer delay. Thus, large gains increase the likelihood of a node (that may not have been overloaded initially) to complete all its load and remain idle until the transferred load arrives. This would clearly increase the total average task completion time, as confirmed earlier by Fig. 6.

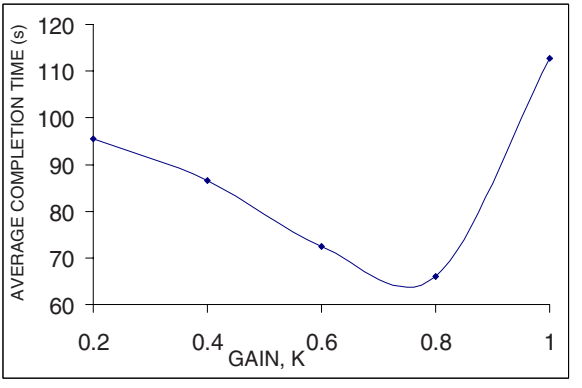



**Fig. 6.** Average total task-completion time as a function of the balancing gain. The load-balancing instant is fixed at 1.4 s.

## 4 Simulation Results

We developed generated a Monte-Carlo simulation tool that allows the simulation of the queues described in Section 2 [Eqns. (1) to (3)]. The network parameters (i.e., the statistics of the communication delays $\eta_{kj}$ and the load-transfer delays $\tau_{ij}$) and the task execution time in the simulation were set according to the respective average values obtained from the experiments described in Section 3.1. Further, we modeled the load-dependent nature of random transfer delay $\tau_{ij}$ by requiring that its average value, $\theta_{ij} \triangleq \mathsf{E}[\tau_{ij}]$, be a function of $L_{ij}$ according to the following transformation

$$\theta_{ij} = d_{\min} - \frac{1 + \exp([(L_{ij} d\beta)]^{-1})}{1 - \exp([(L_{ij} d\beta)]^{-1})}, \tag{4}$$

where $d_{\min}$ is the minimum possible transfer delay, and $d$ and $\beta$ are fitting parameters. This simple, ad-hoc delay model assumes that up to some threshold load, the

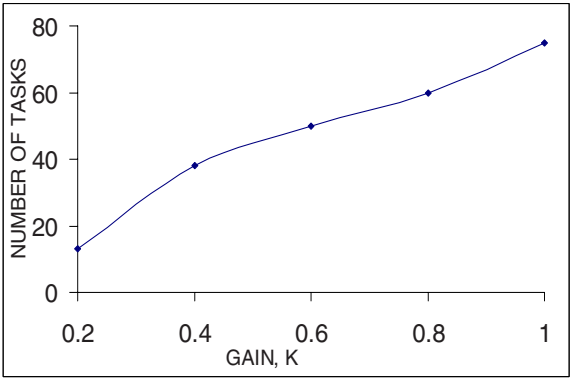



**Fig. 7.** Average total excess load to be transferred by the load-balancing policy to be transferred (at the load-balancing instant) as a function of the balancing gain. The load-balancing instant is fixed at 1.4 s.

average delay is $d_{\min}$, which is independent of the load size. (This minimum delay is dependent, however, on the architecture of the communication medium and we will not be concerned with this dependence in this work.) Beyond this threshold, however, the average delay is expected to increase monotonically with the load size. A typical example of the mean of load-dependent transfer delay versus the load is shown in Fig. 8.

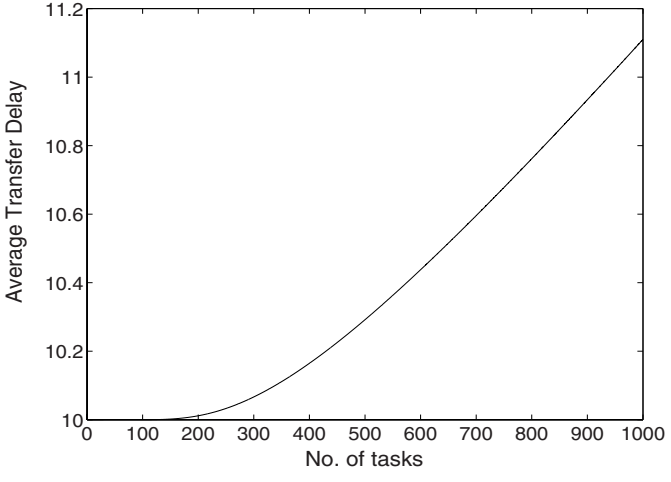



**Fig. 8.** Average load-transfer delay as a function of the load size according to (4). In this example, $d = 0.082618$ and $b = 0.04955$.

For our simulations, $d_{\min}$ was set to be equal to the average value of the transfer delay per task as obtained from the experiments. The parameters $d = 0.082618$ and $b = 0.04955$ were selected so that the delay model is in close approximation with the overall average delay for all the actual transfers that occurred in the experiments.

We used the simulation tool to validate the correspondence between the stochastic queuing model and the experimental setup. In particular, we have generated the simulated versions of Figs. 4, 5 and 6, which are shown below. It is observed that the general characteristics of the simulated curves are very similar to the experiment, but they are not exactly identical, due to the unpredictable behavior and complexity of the wireless environment. Nevertheless, the results of the first simulation, shown in Fig. 9, were consistent with the experimental result (shown in Fig. 4) where we can clearly identify the sudden rise in the completion time around the balancing instant corresponding to the communication delay (850 ms). The reason was described in the experimental section. As for the excess transferred load plotted in Fig. 10, the simulation resulted in the same curve and transition shape obtained from the experiment (Fig. 5).

The curve characteristics of the second simulation, shown in Fig. 11, are analogous to the ones obtained in the experiment (Fig. 6). Indeed, the gain values found are almost the same: $K = 0.8$ from the experiment and $K = 0.87$ from the simulation. As indicated before, the small difference is due to the unstable delay values and other factors present in the wireless environment which has been approximated both by the model and the simulator.

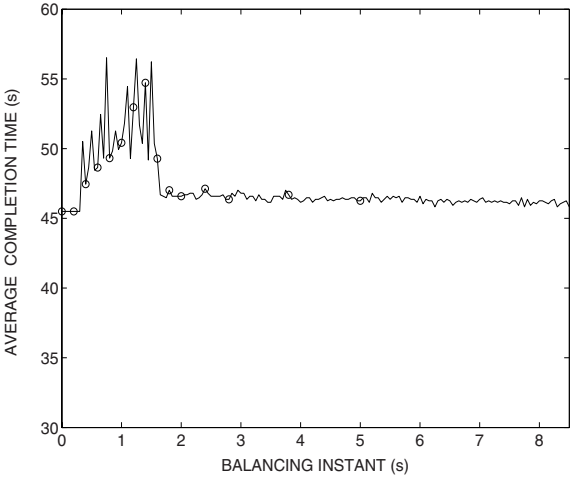



**Fig. 9.** Simulation results for the average total task-completion time as a function of the load-balancing instant. The load-balancing gain parameter is set at $K = 1$. The circles represent the actual experimental values.

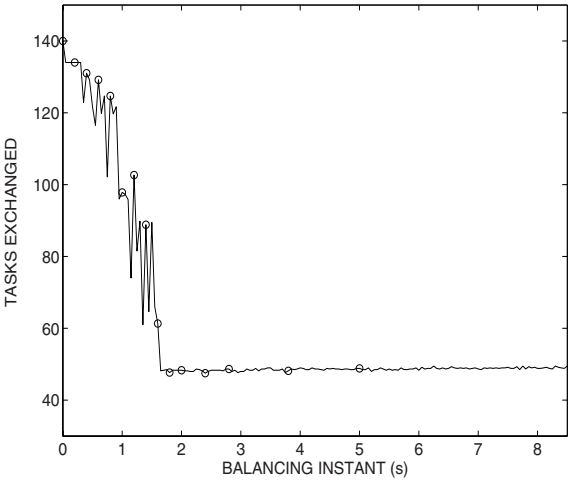



**Fig. 10.** Simulation results for the average total excess load decided by the load-balancing policy to be transferred (at the load-balancing instant) as a function of the balancing instant. The load-balancing gain parameter is set at $K = 1$. The circles represent the actual experimental values.

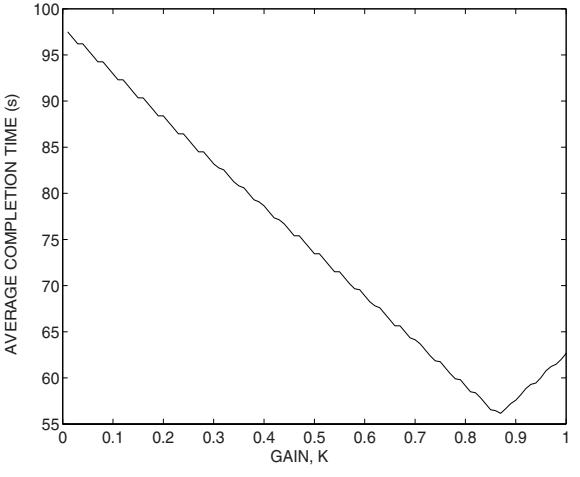



**Fig. 11.** Simulation results for the average total task-completion time as a function of the balancing gain. The load-balancing instant is fixed at 1.4 s.

## 5 Stochastic Analysis of the Queuing Model: A Regeneration Approach

Motivated by the fact that we are dealing with a computationally intesive optimization problem, in which we wish to optimize the load-balancing gain and the balancing instants to minimize the average completion time, we have developed a novel

regenerative approach that will facilitate the analysis of the queuing model described in Section 2. The concept of regeneration has proven to be a powerful tool in the analysis of complex stochastic systems [9, 10, 11].

### 5.1 Rationale

The idea of our approach is to define an *initial event*, defined as the completion of a task by any node or the arrival of a communication by any node, and analyzing the queues that emerge immediately after the occurrence of the initial event. We assume that initially all queues have zero knowledge about the state of the other queues. The point here is that immediately after the occurrence of the initial event, we will have a set of new queues whose stochastic dynamics are identical to the original queues. However, there will be a different set of initial conditions (i.e., different initial load distribution if the initial event is a task completion) or different knowledge state (if the initial event happens to be a communication arrival rather than a task completion). Thus, in addition to having an initial load state, we introduce the novel concept of knowledge states (which we informally referred to in previous sections) to be defined next.

In a system of $n$ nodes, any node will receive $n-1$ number of communications, one from each of the other nodes. Depending upon the choice of the balancing instant, a node may receive all of those communication or may receive none by the time balancing is done. We assign a vector of size $n-1$ to each of the nodes and initially set all its elements to 0 (corresponding to the null knowledge state). If a communication arrives from any of the node, the bit position corresponding to that particular node is set to 1. There will be a total of $2^{n(n-1)}$ number of knowledge states. In the case when two nodes are present, the knowledge states are: 1) state $(0,0)$, corresponding to the case when the nodes do not know about each others initial load; 2) state $(1,1)$, when both nodes know about each other's initial load states; 3) $(1,0)$, corresponding to the case when only node 1 knows about node 2; and 4) state $(0,1)$, which is the opposite of the $(1,0)$ case.

### 5.2 Regenerative equations

To simplify the description, we consider the case where only two nodes are present. We will assume that each node has an exponential service time with parameters $\lambda_{D1}$ and $\lambda_{D2}$, respectively. Let $m$ and $n$ be the initial number of tasks present at nodes 1 and 2, respectively. The communication delays from node 1 to node 2 and from node 2 to node 1 are also assumed to follow an exponential distribution with rates $\lambda_{21}$ and $\lambda_{12}$, respectively. Let $W$, $X$, $Y$ and $Z$ be the waiting times for the departure of the first task at node 1, departure of the first task at node 2, the arrival of the communication sent from node 1 to node 2 and the arrival of the communication sent from 2 to 1, respectively. Let $T$=min($W$,$X$,$Y$,$Z$). A straight-forward calculation shows that the pdf of $T$ can be characterized as $f_T(t) = \lambda e^{-\lambda t} u(t)$, where $\lambda = \lambda_{D1} + \lambda_{D2} + \lambda_{21} + \lambda_{12}$.

Let $\mu_{m,n}^{k_1,k_2}(t_b)$ be the expected value of the overall completion time given that the balancing is executed at time $t_b$, where nodes 1 and 2 are assumed to have $m$ and $n$

tasks at time $t = 0$, and the system knowledge state is $(k_1, k_2)$ at time $t = 0$. Suppose that the initial event happens to be the departure of a task at node 1 at time $t = \tau$, $0 \leq \tau \leq t_b$. At this instant, the system dynamics remain the same except that node 1 will now have $m-1$ tasks. Thus, the queue has re-emerged (with a different initial load, nonetheless) and the average of the overall completion time is now $\tau + \mu_{m-1,n}^{k_1,k_2}(t_b - \tau)$. The effect of other possibilities for the initial event are taken into account similarly. However, to calculate this we need to define the completion time for all cases, i.e., the system initially being in any of the four knowledge states. Based on this discussion, we can characterize the average of the completion times for all four cases below, namely, $\mu_{m,n}^{0,0}(t_b)$, $\mu_{m,n}^{0,1}(t_b)$, $\mu_{m,n}^{1,0}(t_b)$ and $\mu_{m,n}^{1,1}(t_b)$. For example, in the case of $\mu_{m,n}^{0,0}(t_b)$ we obtain the following integral equation

$$\begin{aligned}
\mu_{m,n}^{0,0}(t_b) = & \int_{t_b}^{\infty} f_T(s)[\mu_{m,n}^{0,0}(0) + t_b]ds \\
& + \int_0^{t_b} f_T(s)[\mu_{m-1,n}^{0,0}(t_b - s) + s]\mathsf{P}\{T = W\}ds \\
& + \int_0^{t_b} f_T(s)[\mu_{m,n-1}^{0,0}(t_b - s) + s]\mathsf{P}\{T = X\}ds \\
& + \int_0^{t_b} f_T(s)[\mu_{m,n}^{0,1}(t_b - s) + s]\mathsf{P}\{T = Y\}ds \\
& + \int_0^{t_b} f_T(s)[\mu_{m,n}^{1,0}(t_b - s) + s]\mathsf{P}\{T = Z\}ds. \qquad (5)
\end{aligned}$$

In a similar way, recursive equations can be obtained for the queues corresponding to other knowledge states. This would yield (not all shown here for space constraints) similar equations for $\mu_{m,n}^{0,1}(t_b)$, etc. For example, for $\mu_{m,n}^{1,1}(t_b)$, we have

$$\begin{aligned}
\mu_{m,n}^{1,1}(t_b) = & \int_{t_b}^{\infty} f_T(s)[\mu_{m,n}^{1,1}(0) + t_b]ds \\
& + \int_0^{t_b} f_T(s)[\mu_{m-1,n}^{1,1}(t_b - s) + s]\mathsf{P}\{T = W\}ds \\
& + \int_0^{t_b} f_T(s)[\mu_{m,n-1}^{1,1}(t_b - s) + s]\mathsf{P}\{T = X\}ds \\
& + \int_0^{t_b} f_T(s)[\mu_{m,n}^{1,1}(t_b - s) + s]\mathsf{P}\{T = Y\}ds \\
& + \int_0^{t_b} f_T(s)[\mu_{m,n}^{1,1}(t_b - s) + s]\mathsf{P}\{T = Z\}ds. \qquad (6)
\end{aligned}$$

Moreover, simple calculations yield

$$\begin{aligned}
\mathsf{P}\{T = W\} = \frac{\lambda_{D1}}{\lambda}, \quad \mathsf{P}\{T = X\} = \frac{\lambda_{D2}}{\lambda}, \\
\mathsf{P}\{T = Y\} = \frac{\lambda_{21}}{\lambda}, \quad \mathsf{P}\{T = Z\} = \frac{\lambda_{12}}{\lambda}. \qquad (7)
\end{aligned}$$

The above integral equations can be further simplified by converting them into differential equations of standard form. For example, by differentiating Eq. (5) with

respect to $t_b$, we obtain

$$\frac{\partial \mu_{m,n}^{0,0}(t_b)}{\partial t_b} = \lambda_{D1}\mu_{m-1,n}^{0,0}(t_b) + \lambda_{D2}\mu_{m,n-1}^{0,0}(t_b) + \lambda_{21}\mu_{m,n}^{0,1}(t_b) + \lambda_{12}\mu_{m,n}^{1,0}(t_b) - \lambda\mu_{m,n}^{0,0}(t_b) + 1 \tag{8}$$

and from Eq. (6) we obtain

$$\frac{\partial \mu_{m,n}^{1,1}(t_b)}{\partial t_b} = \lambda_{D1}\mu_{m-1,n}^{1,1}(t_b) + \lambda_{D2}\mu_{m,n-1}^{1,1}(t_b) + \lambda_{21}\mu_{m,n}^{1,1}(t_b) + \lambda_{12}\mu_{m,n}^{1,1}(t_b) - \lambda\mu_{m,n}^{1,1}(t_b) + 1. \tag{9}$$

Therefore, we arrive at a set of four difference-differential equations which completely defines the queuing dynamics of our distributed system. These equations are coupled with each other in the sense that solving Eqn. (8) requires $\mu_{m,n}^{0,1}(t_b)$ and $\mu_{m,n}^{1,0}(t_b)$, each of which in turn requires $\mu_{m,n}^{1,1}(t_b)$. Clearly, Eqn. (9) should be solved initially. The recursion involved in this computation is shown in Fig. (12), which is drawn for the case $m = 6$ and $n = 5$. The bottom of this structure corresponds to the completion time for $(m = 0, n = 1)$ and $(m = 1, n = 0)$ which depend only on $\lambda_{D2}$ and $\lambda_{D1}$, respectively. Therefore, we start at the bottom of the structure and move one level upwards to compute completion times for all cases corresponding to that level, until we finally compute $\mu_{6,5}^{1,1}(t_b)$ in this case. It is also intuitively clear that while solving each of these equations, we need to solve for their corresponding initial conditions, i.e., $\mu_{m,n}^{0,0}(0)$, $\mu_{m,n}^{0,1}(0)$, $\mu_{m,n}^{1,0}(0)$ and $\mu_{m,n}^{1,1}(0)$, which are determined according to the load-balancing algorithm as defined in Section 2. The details of computing the initial conditions are discussed next.

### 5.3 Initial conditions

Based on the load-balancing policy described in Section 2, we develop a methodology to find $\mu_{m,n}^{1,1}(0)$, where $m \geq n$, using the regeneration principle discussed above. According to Eq. (2), and with $p_{21} = 1$, $L_{21}(0)$ becomes

$$L_{21}(0) = \text{floor}\left(\frac{K(m-n)}{2}\right) \triangleq L. \tag{10}$$

*Case I:* $L > 0$

This case corresponds to the scenario for which $L$ tasks are transferred from node 1 to node 2 at time $t = 0$, and hence, node 1 has $m - L$ tasks. If $T_1$ is the waiting time before all of them are served, then the pdf of $T_1$ can be characterized by the Erlang distribution (since the inter-departure times are independent). More precisely, the distribution function of $T_1$ is

$$F_{T_1}(t_1) = \left(1 - \sum_{x=0}^{m-L-1} e^{-\lambda_{D1}t_1}\frac{(\lambda_{D1}t_1)^x}{x!}\right)u(t_1). \tag{11}$$

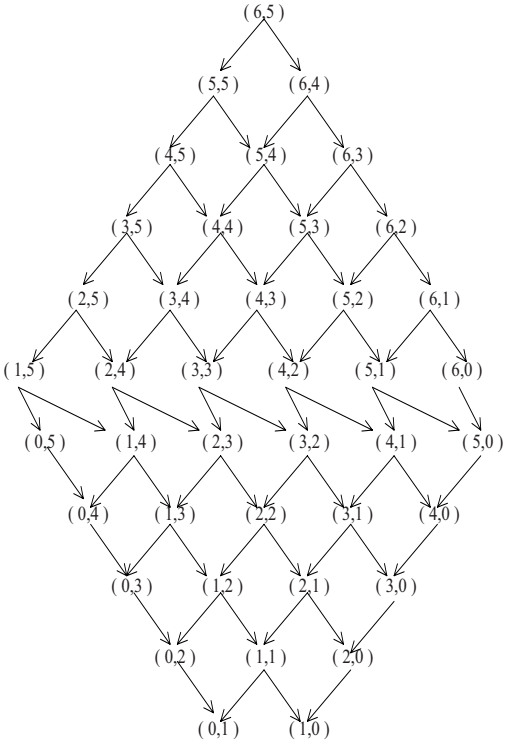



**Fig. 12.** Tree for solving Eqn. (9) recursively

Justified by the experimental results described earlier, the load-dependent random transfer delay $\tau_{21}$ is assumed to follow exponential distribution whose rate, $\lambda_t$, is required to be a function of $L$ according to Eqn. (4), where $\theta_{21} = 1/\lambda_t$. Let $R$ denote the number of tasks that are served at node 2 by the time $L$ tasks arrive, and let $T_R$ denote the waiting time before remaining tasks at node 2 are served. Thus, the total completion time for node 2 is $T_2 = \tau_{21} + T_R$. With this decomposition of $T_2$, we can calculate its probability distribution function as follows:

$$\begin{aligned} F_{T_2}(t_2) &= \mathsf{P}\{T_2 \le t_2\} \\ &= \int_{-\infty}^{\infty} \mathsf{P}\{\tau_{21} + T_R \le t_2 | \tau_{21} = t\} f_{\tau_{21}}(t) dt \\ &= \int_{-\infty}^{\infty} \sum_{r=0}^{n} \mathsf{P}\{T_R \le t_2 - t | R = r, \tau_{21} = t\} \mathsf{P}\{R = r | \tau_{21} = t\} f_{\tau_{21}}(t) dt. \end{aligned} \tag{12}$$

Note that the dependence of $T_R$ on $\tau_{21}$ is through $R$. Therefore, $\mathsf{P}\{T_R \le t_2 - t | R = r, \tau_{21} = t\} = \mathsf{P}\{T_R \le t_2 - t | R = r\}$, which is given as

$$\mathsf{P}\{T_R \le t_2 - t | R = r\} = \left(1 - \sum_{x=0}^{L+n-r-1} e^{-\lambda_{D2}(t_2-t)} \frac{(\lambda_{D2}(t_2-t))^x}{x!}\right) u(t_2-t). \quad (13)$$

We also maintain that

$$\mathsf{P}\{R = r | \tau_{21} = t\} = \frac{(\lambda_{D2}t)^r}{r!} e^{-\lambda_{D2}t} u(t),\ 0 \le r \le n-1\,. \quad (14)$$

Using Eqns. (12), (13), and (14), we obtain

$$F_{T_2}(t_2) = 1 - e^{-\lambda_t t_2} - \lambda_t e^{-\lambda_{D2} t_2} \left[\sum_{r=0}^{n-1} \sum_{x=L}^{L+n-r-1} \frac{(\lambda_{D2})^r}{r!} \frac{(\lambda_{D2})^x}{x!} g(t_2; r; x)\right] - \lambda_t e^{-\lambda_{D2} t_2} \left[\sum_{x=0}^{L-1} \frac{(\lambda_{D2})^x}{x!} g_1(t_2; 0; x)\right], \quad (15)$$

where

$$g(t_2; r; x) = \int_0^{t_2} t^r (t_2 - t)^x e^{-\lambda_t t} dt = \sum_{k=0}^{x} (-1)^k \frac{x!}{(x-k)!k!} (t_2)^{x-k} \frac{(r+k)!}{\lambda_t^{r+k+1}} \left[1 - e^{\lambda_t t_2} \sum_{j=0}^{r+k} \frac{(\lambda_t t_2)^j}{j!}\right]$$

$$g_1(t_2; 0; x) = \int_0^{t_2} (t_2 - t)^x e^{-(\lambda_t - \lambda_{D2})t} dt$$

The overall completion time is given as $T_C = \max(T_1, T_2)$, and its average $E[T_C]$ is, according to our definition, $\mu_{m,n}^{1,1}(0)$. Now, by exploiting the independence of $T_1$ and $T_2$, we obtain

$$\mu_{m,n}^{1,1}(0) = \int_0^{\infty} t\,[f_{T_1}(t) F_{T_2}(t) + F_{T_1}(t) f_{T_2}(t)]\,dt \quad (16)$$

*Case II:* $L = 0$

In this case, no load transfer occurs at all and $\mu_{m,n}^{1,1}(0)$ can be calculated in a straightforward manner to yield

$$\mu_{m,n}^{1,1}(0) = \begin{cases} \frac{m}{\lambda_{D1}} + \frac{n}{\lambda_{D2}} - \frac{(\lambda_{D1})^m}{(m-1)!} \sum_{x=0}^{n-1} \frac{(m+x)!}{(\lambda_{D1}+\lambda_{D2})^{m+x+1}} \frac{(\lambda_{D2})^x}{x!} \\ -\frac{(\lambda_{D2})^n}{(n-1)!} \sum_{x=0}^{m-1} \frac{(n+x)!}{(\lambda_{D1}+\lambda_{D2})^{n+x+1}} \frac{(\lambda_{D1})^x}{x!}, & n > 0 \\ \frac{m}{\lambda_{D1}}, & n = 0 \end{cases}$$

### 5.4 Discussion of results

We present an example of the analytical results described earlier for a typical case for which one of the nodes does not have any initial tasks. This not only simplifies our calculations, but also signifies the interplay between the balancing gain $K$

and the balancing instant $t_b$, and their effect on the overall completion time. Since Eqn. (9) corresponds to the case where $m > 0$ and $n > 0$, we will need to develop a new version of it which would handle the special case considered here. Using the principle of regeneration, as described before, $\mu_{m,0}^{k_1,k_2}(t_b)$ can be characterized as a set of four coupled difference-differential equations, each corresponding to a particular initial knowledge state. In the special case considered here, Eq. (9) takes the modified following form

$$\frac{\partial \mu_{m,0}^{1,1}(t_b)}{\partial t_b} = \lambda_{D1}\mu_{m-1,0}^{1,1}(t_b) + \lambda_{21}\mu_{m,0}^{1,1}(t_b) + \lambda_{12}\mu_{m,0}^{1,1}(t_b) - \lambda\mu_{m,0}^{1,1}(t_b) + 1, \tag{17}$$

which can be reduced to

$$\frac{\partial \mu_{m,0}^{1,1}(t_b)}{\partial t_b} = -\lambda_{D1}\mu_{m,0}^{1,1}(t_b) + \lambda_{D1}\mu_{m-1,0}^{1,1}(t_b) + 1. \tag{18}$$

Clearly, the communication rate does not play any role here as the initial knowledge state is already $(1,1)$. The solution to Eqn. (18) for $m \geq 2$ is

$$\mu_{m,0}^{1,1}(t_b) = \frac{m}{\lambda_{D1}} + \sum_{p=2}^{m} \frac{(\lambda_{D1})^{m-p-1}}{(m-p)!}[\lambda_{D1}\mu_{p,0}^{1,1}(0) - p]t_b^{m-p}e^{-\lambda_{D1}t_b}, \tag{19}$$

with the obvious fact that $\mu_{0,0}^{1,1}(t_b) = 0$ and $\mu_{1,0}^{1,1}(t_b) = \frac{1}{\lambda_{D1}}$.

We now present a numerical example. Let node 1 and node 2 each have a service rate given by $\lambda_{D1} = 1$ task per second. The average load-transfer delay parameters (as defined in Eqn. (4)) are set to $d_{\min} = 20$ s, $d = 0.082618$ and $\beta = 0.04955$. The initial load distribution is 200 tasks, which are assigned to node 1; node 2 has no tasks assigned to it. It is observed that for any given value of balancing gain $K$, the optimal balancing instant is found to be at $t_b = 0$. Therefore, with $t_b = 0$, optimization over $K$ is performed as depicted in Fig. 13. The optimal average completion time is 120 s and the optimal gain is 0.9. When the service rate of node 2 is made twice as fast as that for node 1, the optimal balancing instant remains same as before but the optimal $K$ turns out to be equal to 1 as shown in Fig. 14. When the service rate of both the processors is increased to 4 tasks per second. The optimal balancing gain turns out to be 0.7 and the optimal average completion time is 43 s, which is shown in Fig. 15. Using the same values for all the parameters, the Monte-Carlo simulation tool is used to generate Fig. 16, which is the empirical version of the analytical results shown in Fig. 15. The strong resemblance between the analytical and simulations is evident. Finally, in Fig. 17, we set $d_{\min}$ to be 30 s and repeat the calculations. The optimal gain in this case is 0, and therefore, it is better not to send any task to node 2 in this case.

With these results, we clearly see that the choice of the optimal $K$ is dependent on a number of factors like the processing speed, transfer delay, and number of initial tasks. Further, the optimal balancing instant in all these cases is found to be at

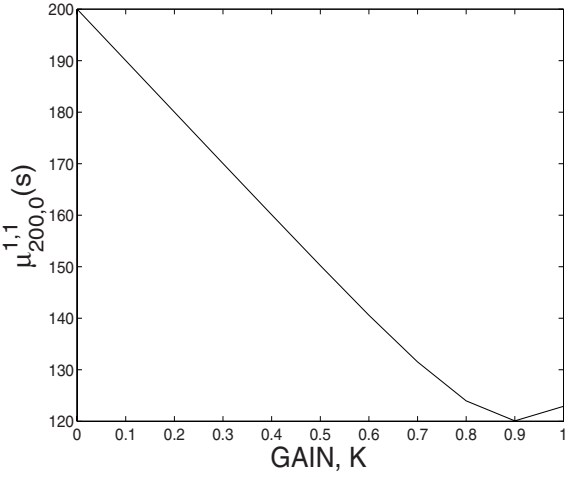



**Fig. 13.** Average total task-completion time as a function of the balancing gain. The load-balancing instant $t_b = 0$.

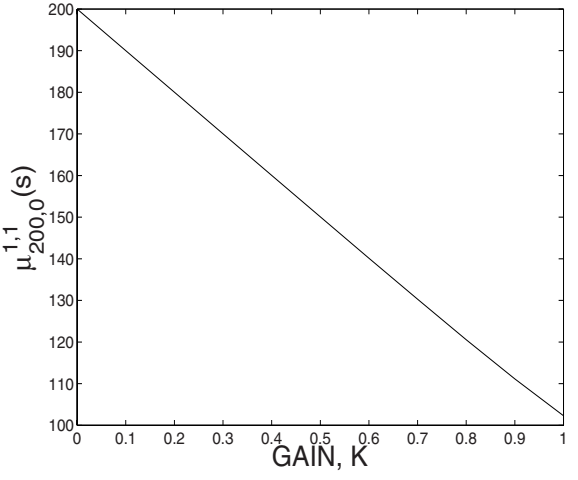



**Fig. 14.** Same as Fig. 13 but here node 2 is twice as fast as node 1.

$t_b = 0$ s, which is also predicted by our earlier simulation model. Intuitively, when the initial knowledge state is (1,1), delaying the scheduling instant is not going to bring any new information. Therefore, it is more motivating to optimize the average completion time over $t_b$ when the initial knowledge state is not (1,1). Nonetheless, the optimization over $K$ in itself verifies our notion of presenting the load balancing as an optimization problem.

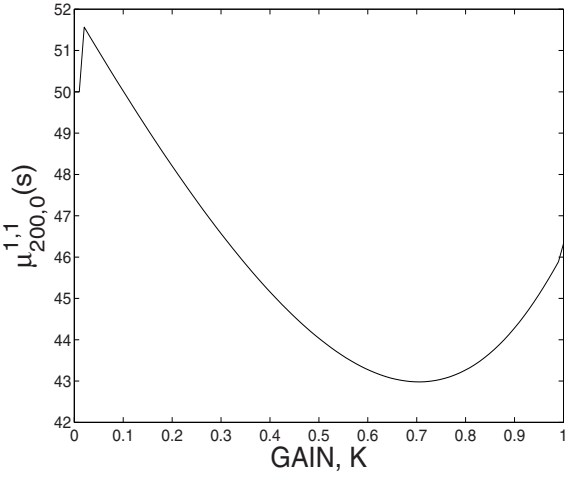



**Fig. 15.** Average total task-completion time as a function of the balancing gain. Each processor has a service rate of 4 tasks per second.

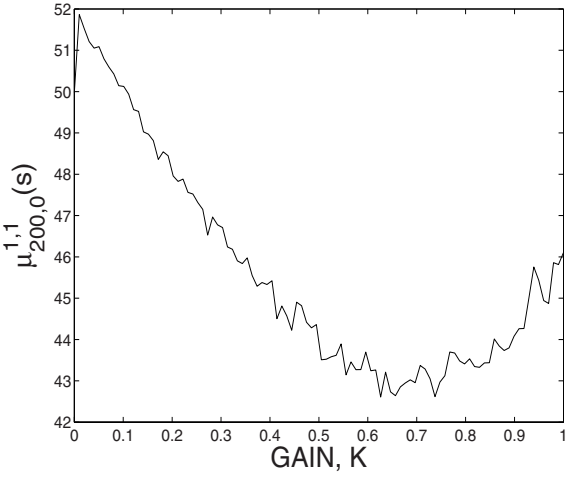



**Fig. 16.** Same as Fig. 15 but generated by MC simulation.

## 6 Conclusions

We have performed experiments and simulations to investigate the performance of load-balancing policy that involves redistributing the load of the nodes only once after a large load arrives at the distributed system. Our experimental results (using a wireless LAN) and simulations both indicate that in distributed systems, for which communication and load-transfer delays are tangible, it is best to execute the load balancing after each node receives communications from other nodes regarding their

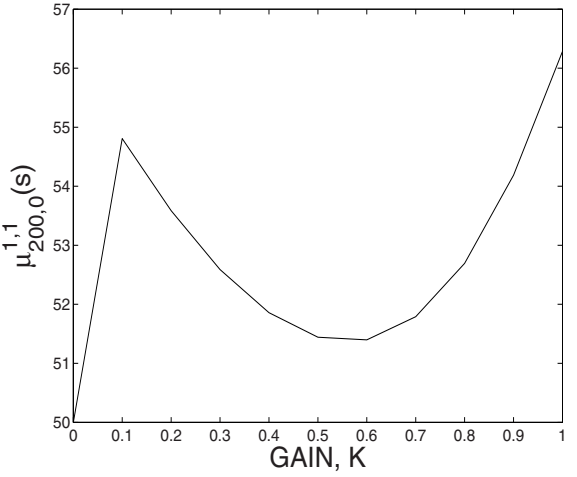



**Fig. 17.** Same as Fig. 15 but with larger transfer delay.

load states. In particular, our results indicate that the loss of time in waiting for the inter-node communications to arrive is overcompensated by the informed nature of the load balancing. Moreover, the optimal load-balancing gain turns out to be less than unity, contrary to systems that do not exhibit significant latency. In delay infested systems, a moderate balancing gain has the benefit of reduced load-transfer delays, as the fraction of the load to be transferred is reduced. This, in turn, will result in a reduced likelihood of certain nodes becoming idle as soon as they are depleted of their initial load. We have also developed the analytical model to characterize the expected value of the total completion time for a distributed system when a single scheduling is performed. Our preliminary results signify the interplay between delay and load-balancing gain and verifies our notion that the load balancing is an optimization problem.

In our future work, we will use our optimal single-time load-balancing strategy to develop an autonomous on-demand (sender initiated) load-balancing scheme. Every node would have its look-up table for the optimal instant of load-balancing and the optimal gain. The look up-table would be generated up-front using our analytical solution to the queuing model which was described earlier. As each external request arrives at a node, the node will autonomously decide whether or not, when, and how to execute a single-instant load-balancing action. There would be no need to synchronize the balancing instants between all the nodes, and therefore, load balancing can be done dynamically. Although the proposed autonomous on-demand load-balancing dynamic balancing strategy may not lead us to the globally optimal solution, we believe that it will offer an effective solution at a reduced implementation complexity.

**Acknowledgements.**

This work is supported by the National Science Foundation under Information Technology Research (ITR) grants No. ANI-0312611 and ANI-0312182. Additional support was received from the National Science Foundation through grant No. INT-9818312.

## References

[1] Z. Lan, V. E. Taylor, and G. Bryan, "Dynamic load balancing for adaptive mesh refinement application," in *Proc. ICPP'2001*, Valencia, Spain, 2001.

[2] T. L. Casavant and J. G. Kuhl, "A taxonomy of scheduling in general-purpose distributed computing systems," *IEEE Trans. Software Eng.*, vol. 14, pp. 141–154, Feb. 1988.

[3] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *IEEE Trans. Parallel and Distributed Computing*, vol. 7, pp. 279–301, Oct. 1989.

[4] J. D. Birdwell, J. Chisson, Z. Tang, T. Wang, C. T. Abdallah, and M. M. Hayat, "Dynamic time delay models for load balancing. Part I: Deterministic models," *CNRS-NSF workshop: Advances in Control of Time-Delay Systems*, Paris, France, pp. 106–114, January 2003.

[5] C-C. Hui and S. T. Chanson,"Hydrodynamic load balancing," *IEEE Trans. Parallel and Distributed Systems*, vol. 10, Issue 11, Nov. 1999.

[6] M. M. Hayat, S. Dhakal, C. T. Abdallah, J. Chisson, and J. D. Birdwell " Dynamic time delay models for load balancing. Part II: Stochastic analysis of the effect of delay uncertainty, *CNRS-NSF Workshop: Advances in Control of Time-Delay Systems*, Paris, France, pp. 115–122, January 2003.

[7] S. Dhakal, B. S. Paskaleva, M. M. Hayat, E. Schamiloglu, C. T. Abdallah "Dynamical discrete-time load balancing in distributed systems in the presence of time delays," Proceedings of the *IEEE CDC 2003*, pp. 5128–5134, Maui, Hawaii.

[8] J. Ghanem, S. Dhakal, M. M. Hayat, H. Jérez, C.T. Abdallah, and J. Chiasson "Load balancing in distributed systems with large time delays: Theory and experiment," *submitted to the IEEE 12th Miditerranean Conference on Control and Automation, MED'04*, Izmir, Turkey.

[9] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes*. Springer-Verlag, 1988.

[10] C. Knessly and C. Tiery,"Two tandem queues with general renewal input I: Diffusion approximation and integral representation,"*SIAM J. Appl. Math.*,vol.59, pp1917-1959, 1999.

[11] F. Bacelli and P. Bremaud, "Elements of queuing theory: Palm-martingale calculus and stochastic recurrence", New York: Springer-Verlag, 1994.

# Closed Loop Control of a Load Balancing Network with Time Delays and Processor Resource Constraints

Zhong Tang[1], J. Douglas Birdwell[1], John Chiasson[1], Chaouki T. Abdallah[2] , and Majeed M. Hayat[2]

[1] ECE Dept, The University of Tennessee, Knoxville, TN 37996-2100, USA
{ztang,birdwell,chiasson}@utk.edu
[2] ECE Dept, University of New Mexico, Alburquerque, NM 87131-1356, USA
{chaouki,hayat}@ece.unm.edu

## 1 Introduction

The objective of parallel processing is to reduce wall-clock time and increase the size of solvable problems by dividing the code into multiple fragments that can be executed simultaneously on each of a set of computational elements (CE) interconnected via a high bandwidth network. A common parallel computer architecture is the cluster of otherwise independent computers communicating through a shared network. To make use of parallel computing resources, problems must be broken down into smaller units that can be solved individually by each CE while exchanging information with CEs solving other problems. For example, the Federal Bureau of Investigation (FBI) National DNA Index System (NDIS) and Combined DNA Index System (CODIS) software are candidates for parallelization. New methods developed by Wang et al. [1][2][3][4] lead naturally to a parallel decomposition of the DNA database search problem while providing orders of magnitude improvements in performance over the current release of the CODIS software.

To effectively utilize parallel computer architecture, the computational loads throughout all parallel nodes need to be distributed more or less evenly over the available CEs. The qualifier "more or less" is used because the communications required to distribute the load consume both computational resources and network bandwidth. A point of diminishing returns exists. To distribute the load evenly is a key activity in producing efficient implementations of applications on parallel architectures.

Distribution of computational load across available resources is referred to as the *load balancing* problem in the literature. Various taxonomies of load balancing algorithms exist. Direct methods examine the global distribution of computational load and assign portions of the workload to resources before processing begins. Iterative methods examine the progress of the computation and the expected utilization of resources, and adjust the workload assignments periodically as computation progresses. Assignment may be either deterministic, as with the dimension exchange/diffusion [5] and gradient methods, stochastic, or optimization based. A

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 245–268, 2004.

comparison of several deterministic methods is provided by Willebeek-LeMair and Reeves [6]. Approaches to modeling and static load balancing are given in [7][8][9].

To adequately model load balancing problems, several features of the parallel computation environment should be captured: (1) The workload awaiting processing at each CE; (2) the relative performances of the CEs; (3) the computational requirements of each workload component; (4) the delays and bandwidth constraints of CEs and network components involved in the exchange of workloads and, (5) the delays imposed by CEs and the network on the exchange of measurements. A queuing theory [10] approach is well-suited to the modeling requirements and has been used in the literature by Spies [11] and others. However, whereas Spies assumes a homogeneous network of CEs and models the queues in detail, the present work generalizes queue length to an expected waiting time, normalizing to account for differences among CEs, and aggregates the behavior of each queue using a continuous state model.

The present work focuses upon the effects of delays in the exchange of information among CEs, and the constraints these effects impose on the design of a load balancing strategy. Previous results by the authors appear in [12][13][14][15][16][17][18]. An issue that was not considered in this previous work is the fact that the load balancing operation involves processor time which is not being used to process tasks. Consequently, there is a trade-off between using processor time/network bandwidth and the advantage of distributing the load evenly between the nodes to reduce overall processing time. The fact that the simulations of the model in [18] showed the load balancing to be carried out faster than the corresponding experimental results motivated the authors to refine the model to account for processor constraints. A new deterministic dynamic time-delay system model is developed here to capture these constraints. The model is shown to be self consistent in that the queue lengths are always nonnegative and the total number of tasks in all the queues and the network are conserved (i.e., load balancing can neither create nor lose tasks). In contrast to the results in [18] where it was analytically shown the system was always asymptotically stable, the new model is now only (Lyapunov) stable, and asymptotic stability must be insured by judicious choice of the feedback.

A controller on any node which uses delayed information from other nodes can cause unnecessary data transfers among the nodes. Here a control law is proposed that uses not only its estimate of the queue size of the other nodes, but also measurements of the number of tasks in transit to it. The communication of the number of tasks being sent to a node is much faster than the actual transfer of the tasks. In each processing loop, each node broadcasts its current queue size as well as the number of tasks it is sending to each of the other nodes (rather than have the other nodes wait for the actual arrival of the tasks to obtain this information). This way each node's estimate of the whole network status will be more up to date when making its control decision. Experimental results on a parallel computer network are presented to demonstrate the efficacy of the load balancing strategy. In particular, load balancing experiments using the controller based on the local queue size only are compared with those using the controller based on the anticipated local queue size, demonstrating a substantial improvement.

Section 2 presents our approach to modeling the computer network and load balancing algorithms to incorporate the presence of delay in communicating between nodes and transferring tasks. Section 3 shows that the proposed model correctly predicts that the queue lengths are nonnegative and that the total number of tasks in all the queues are conserved by the load balancing algorithm. This section ends with a proof of (Lyapunov) stability of the system model. Section 4 addresses the feedback control law on a local node and how a node decides to portions out its tasks to the other nodes. Feedback controllers based on the actual queue size and on the *anticipated* queue size are discussed in this section. Section 5 presents how the model parameters are obtained and the experimental setup. Section 6 presents a comparison of the feedback controller based on the actual queue size with the feedback controller based on the anticipated queue size. This comparison is done with the actual experimental data. Finally, Section 7 is a summary and conclusion of the present work.

## 2 Mathematical Model of Load Balancing

In this section, a nonlinear continuous time model is developed to model load balancing among a network of computers. Consider a computing network consisting of $n$ computers (nodes) all of which can communicate with each other. At start up, the computers are assigned an equal number of tasks. However, when a node executes a particular task it can in turn generate more tasks so that very quickly the loads on various nodes become unequal. To balance the loads, each computer in the network sends its queue size $q_j(t)$ to all other computers in the network. A node $i$ receives this information from node $j$ *delayed* by a finite amount of time $\tau_{ij}$; that is, it receives $q_j(t-\tau_{ij})$. Each node $i$ then uses this information to compute its local estimate[3] of the average number of tasks in the queues of the $n$ computers in the network. The simple estimator $\left(\sum_{j=1}^{n} q_j(t-\tau_{ij})\right)/n$, $(\tau_{ii}=0)$ which is based on the most recent observations is used. Node $i$ then compares its queue size $q_i(t)$ with its estimate of the network average as $\left(q_i(t)-\left(\sum_{j=1}^{n} q_j(t-\tau_{ij})\right)/n\right)$ and, if this is greater than zero or some positive threshold, the node sends some of its tasks to the other nodes. If it is less than zero, no tasks are sent (see Figure 1). Further, the tasks sent by node $i$ are received by node $j$ with a delay $h_{ij}$. The task transfer delay $h_{ij}$ depends on the number of tasks to be transferred and is much greater than the communication delay $\tau_{ij}$. The controller (load balancing algorithm) decides how often and fast to do load balancing (transfer tasks among the nodes) and how many tasks are to be sent to each node. As just explained, each node controller (load balancing algorithm) has only *delayed* values of the queue lengths of the other nodes, and each transfer of data from one node to another is received only after a finite time delay. An important issue considered here is the effect of these delays on system performance. Specifically, the model developed here represents our effort to capture the effect of the delays in load

[3] It is an estimate because at any time, each node only has the delayed value of the number of tasks in the other nodes.

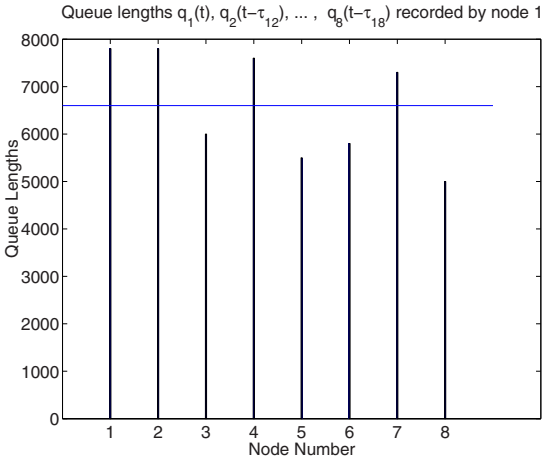



**Fig. 1.** Graphical description of load balancing. This bar graph shows the load for each computer vs. node of the network. The thin horizontal line is the average load as estimated by node 1. Node 1 will transfer (part of) its load only if it is above its estimate of the network average. Also, it will only transfer to nodes that it estimates are below the network average.

balancing techniques as well as the processor constraints so that system theoretic methods could be used to analyze them.

### 2.1 Basic Model

The basic mathematical model of a given computing node for load balancing is given by

$$\begin{aligned}\frac{dx_i(t)}{dt} &= \lambda_i - \mu_i(1-\eta_i(t)) - U_m(x_i)\eta_i(t) \\ &\quad + \sum_{j=1}^{n} p_{ij}\frac{t_{p_i}}{t_{p_j}} U_m(x_j(t-h_{ij}))\eta_j(t-h_{ij}) \qquad (1)\\ p_{ij} &\geqslant 0, p_{jj} = 0, \sum_{i=1}^{n} p_{ij} = 1\end{aligned}$$

where

$$\begin{aligned}U_m(x_i) &= U_{m0} > 0 \text{ if } x_i > 0 \\ &= 0 \qquad\quad \text{if } x_i \leq 0.\end{aligned}$$

In this model we have

- $n$ is the number of nodes.
- $x_i(t)$ is the *expected waiting time* experienced by a task inserted into the queue of the $i^{th}$ node. With $q_i(t)$ the number of *tasks* in the $i^{th}$ node and $t_{p_i}$ the average time needed to process a task on the $i^{th}$ node, the expected (average) waiting time is then given by $x_i(t) = q_i(t)t_{p_i}$. Note that $x_j/t_{p_j} = q_j$ is the number of tasks in the node $j$ queue. If these tasks were transferred to node $i$, then the waiting time transferred is $q_j t_{p_i} = x_j t_{p_i}/t_{p_j}$, so that the fraction $t_{p_i}/t_{p_j}$ converts waiting time on node $j$ to waiting time on node $i$.
- $\lambda_i \geq 0$ is the rate of generation of waiting time on the $i^{th}$ node caused by the addition of tasks (rate of increase in $x_i$).
- $\mu_i \geq 0$ is the rate of reduction in waiting time caused by the service of tasks at the $i^{th}$ node and is given by $\mu_i \equiv (1 \times t_{p_i})/t_{p_i} = 1$ for all $i$ if $x_i(t) > 0$, while if $x_i(t) = 0$ then $\mu_i \triangleq 0$, that is, if there are no tasks in the queue, then the queue cannot possibly decrease.
- $\eta_i = 1$ or 0 is the *control input* which specifies whether tasks (waiting time) are processed on a node or tasks (waiting time) are transferred to other nodes.
- $U_{m0}$ is the limit on the rate at which data can be transmitted from one node to another and is basically a bandwidth constraint.
- $p_{ij}U_m(x_j)\eta_j(t)$ is the rate at which node $j$ sends waiting time (tasks) to node $i$ at time $t$ where $p_{ij} \geqslant 0, \sum_{i=1}^{n} p_{ij} = 1$ and $p_{jj} = 0$. That is, the transfer from node $j$ of expected waiting time (tasks) $\int_{t_1}^{t_2} U_m(x_j)\eta_j(t)dt$ in the interval of time $[t_1, t_2]$ to the other nodes is carried out with the $i^{th}$ node being sent the fraction $p_{ij}\int_{t_1}^{t_2} U_m(x_j)\eta_j(t)dt$ of this waiting time. As $\sum_{i=1}^{n}\left(p_{ij}\int_{t_1}^{t_2} U_m(x_j)\eta_j(t)dt\right) = \int_{t_1}^{t_2} U_m(x_j)\eta_j(t)dt$, this results in removing *all* of the waiting time $\int_{t_1}^{t_2} U_m(x_j)\eta_j(t)dt$ from node $j$.
- The quantity $-p_{ij}U_m(x_j(t-h_{ij}))\eta_j(t-h_{ij})$ is the rate of transfer of the expected waiting time (tasks) at time $t$ from node $j$ by (to) node $i$ where $h_{ij}$ $(h_{ii} = 0)$ is the time delay for the task transfer from node $j$ to node $i$.
- The factor $t_{p_i}/t_{p_j}$ converts the waiting time from node $j$ to waiting time on node $i$.

In this model, all rates are in units of the *rate of change of expected waiting time*, or *time/time* which is dimensionless. As $\eta_i = 1$ or 0, node $i$ can only send tasks to other nodes and cannot initiate transfers from another node to itself. A delay is experienced by transmitted tasks before they are received at the other node. Model (1) is the basic model, but one important detail remains unspecified, namely the exact form $p_{ji}$ for each sending node $i$. One approach is to choose them as constant and equal

$$p_{ji} = 1/(n-1) \text{ for } j \neq i \text{ and } p_{ii} = 0 \tag{2}$$

where it is clear that $p_{ji} \geqslant 0, \sum_{j=1}^{n} p_{ji} = 1$. Another approach is to base them on the estimated state of the network; this approach is given in section 5.

## 3 Model Consistency and Stability

It is now shown that the open loop model is consistent with actual working systems in that the queue lengths cannot go negative and the load balancing algorithm cannot create or lose tasks; it can only move then between nodes.

### 3.1 Non Negativity of the Queue Lengths

To show the non negativity of the queue lengths, recall that the queue length of each node is given by $q_i(t) = x_i(t)/t_{p_i}$. The model is rewritten in terms of these quantities as

$$\frac{d}{dt}\left(x_i(t)/t_{p_i}\right) = \frac{\lambda_i - \mu_i(1-\eta_i(t))}{t_{p_i}} - \frac{1}{t_{p_i}} U_m(x_i)\eta_i(t) + \sum_{j=1}^{n} \frac{p_{ij}}{t_{p_j}} U_m(x_j(t-h_{ij}))\eta_j(t-h_{ij}). \tag{3}$$

Given that $x_i(0) > 0$ for all $i$, it follows from the right-hand side of (3) that $q_i(t) = x_i(t)/t_{p_i} \geq 0$ for all $t \geq 0$ and all $i$. To see this, suppose without loss of generality that $q_i(t) = x_i(t)/t_{p_i}$ is the first queue to go to zero, and let $t_1$ be the time when $x_i(t_1) = 0$. At the time $t_1$, $\lambda_i - \mu_i(1-\eta_i(t)) = \lambda_i \geq 0$ as $\mu_i(x_i) = 0$ if $x_i = 0$. Also, $\sum_{j=1}^{n} \frac{p_{ij}}{t_{p_j}} U_m(x_j(t-h_{ij}))\eta_j(t-h_{ij}) \geq 0$ as $\eta_j \geq 0$. Further, the term $U_m(x_i) = 0$ for $x_i \leq 0$. Consequently

$$\frac{d}{dt}\left(x_i(t)/t_{p_i}\right) \geq 0 \text{ for } x_i = 0$$

and thus the queues cannot go negative.

### 3.2 Conservation of Queue Lengths

It is now shown that the total number of tasks in all the queues and the network are conserved. To do so, sum up equations (3) from $i = 1, ..., n$ to get

$$\frac{d}{dt}\left(\sum_{i=1}^{n} q_i(t)\right) = \sum_{i=1}^{n}\left(\frac{\lambda_i - \mu_i(x_i)(1-\eta_i)}{t_{p_i}}\right) - \sum_{i=1}^{n} \frac{U_m(x_i(t))}{t_{p_i}}\eta_i + \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{p_{ij}}{t_{p_j}} U_m(x_j(t-h_{ij}))\eta_j(t-h_{ij}) \tag{4}$$

which is the rate of change of the total queue lengths on all the nodes. However, the network itself also contains tasks. The dynamic model of the queue lengths in the network is given by

$$\frac{d}{dt} q_{net_i}(t) = -\sum_{j=1}^{n} \frac{p_{ij}}{t_{p_j}} U_m(x_j(t-h_{ij}))\eta_j(t-h_{ij}) + \sum_{j=1}^{n} \frac{p_{ij}}{t_{p_j}} U_m(x_j(t))\eta_j(t). \tag{5}$$

Here $q_{net_i} \triangleq \sum_{j\neq i}^{n} q_{net_{ij}}$ is the number of tasks put on the network that are being sent to node $i$ from the other nodes. This equation simply says that the $j^{th}$ node is putting the number of tasks $q_{net_{ij}}$ on the network to be sent to node $i$ at the rate $dq_{net_{ij}}/dt = \left(p_{ij}/t_{p_j}\right) U_m(x_j(t))\eta_j(t)$ while the $i^{th}$ node is taking these tasks from node $j$ off the network at the rate $-p_{ij}/t_{p_j} U_m(x_j(t-h_{ij}))\eta_j(t-h_{ij})$. Summing (5) over all the nodes, one obtains

$$\begin{aligned}
&\frac{d}{dt}\left(\sum_{i=1}^{n} q_{net_i}(t)\right) \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{p_{ij}}{t_{p_j}} U_m(x_j(t-h_{ij}))\eta_j(t-h_{ij}) + \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{p_{ij}}{t_{p_j}} U_m(x_j(t))\eta_j(t) \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{p_{ij}}{t_{p_j}} U_m(x_j(t-h_{ij}))\eta_j(t-h_{ij}) + \sum_{j=1}^{n} \frac{U_m(x_j(t))\eta_j(t)}{t_{p_j}}. \qquad (6)
\end{aligned}$$

Adding (4) and (6), one obtains

$$\frac{d}{dt}\sum_{i=1}^{n}\left(q_i(t) + q_{net_i}(t)\right) = \sum_{i=1}^{n}\left(\frac{\lambda_i - \mu_i(1-\eta_i)}{t_{p_i}}\right). \qquad (7)$$

In words, the total number of tasks which are in the system (i.e., in the nodes and/or in the network) can increase only by the rate of arrival of tasks $\sum_{i=1}^{n} \lambda_i/t_{p_i}$ at all the nodes, or similarly, decrease by the rate of processing of tasks $\sum_{i=1}^{n} \mu_i(1-\eta_i)/t_{p_i}$ at all the nodes. The load balancing itself cannot increase or decrease the total number of tasks in all the queues.

### 3.3 Stability of the Model

Combining the results of the previous two subsections, one can show Lyapunov stability of the model. Specifically, we have

**Theorem:** Given the system described by (1) and (5) with $\lambda_i = 0$ for $i = 1,...,n$ and initial conditions $x_i(0) \geq 0$, then the system is Lyapunov stable for any choice of the switching times of the control input functions $\eta_i(t)$.

**Proof:** First note that the $q_{net_i}$ are non negative as

$$q_{net_i}(t) = \sum_{j=1}^{n} \frac{p_{ij}}{t_{p_j}}\left(\int_{t-h_{ij}}^{t} U_m(x_j(\tau))\eta_j(\tau)d\tau\right) \geq 0. \qquad (8)$$

By the non-negativity property of the $q_i$, the linear function

$$V\left(q_i(t), q_{net_i}(t)\right) \triangleq \sum_{i=1}^{n}\left(q_i(t) + q_{net_i}(t)\right)$$

is a positive definite function. Under the conditions of the theorem, equation (7) becomes

$$\frac{d}{dt}\sum_{i=1}^{n}\left(q_i(t)+q_{net_i}(t)\right)=-\sum_{i=1}^{n}\frac{\mu_i(q_i/t_{p_i})}{t_{p_i}}(1-\eta_i) \tag{9}$$

which is negative semi-definite. By standard Lyapunov theory (e.g., see [19]), the system is Lyapunov stable.

## 4 Feedback Control

In the work [18], the feedback law at each node $i$ was based on the value of $x_i(t)$ and the *delayed* values $x_j(t-\tau_{ij})$ $(j\neq i)$ from the other nodes. Here $\tau_{ij}$ $(\tau_{ii}=0)$ denote the time delays for communicating the expected waiting time $x_j$ from node $j$ to node $i$. These communication delays $\tau_{ij}$ are much smaller than the corresponding data transfer delays $h_{ij}$ of the actual tasks. Define

$$x_{i\_avg}\triangleq\left(\sum_{j=1}^{n}x_j(t-\tau_{ij})\right)/n \tag{10}$$

to be the *local average* which is the $i^{th}$ node's *estimate* of the average of all the nodes. (This is only an estimate due to the delays). Further, define

$$y_i(t)\triangleq x_i(t)-x_{i\_avg}(t)=x_i(t)-\frac{\sum_{j=1}^{n}x_j(t-\tau_{ij})}{n}$$

to be the expected waiting time relative to the estimate of the network average by the $i^{th}$ node.

The control law considered here is

$$\eta_i(t)=h\left(y_i(t)\right) \tag{11}$$

where $h(\cdot)$ is a hysteresis function given by

$$h(y)=\begin{cases}1 \text{ if } y\geqslant y_0\\ 1 \text{ if } 0<y<y_0 \text{ and } \dot{y}>0\\ 0 \text{ if } 0<y<y_0 \text{ and } \dot{y}<0\\ 0 \text{ if } y<0.\end{cases}$$

The control law basically states that if the $i^{th}$ node waiting time $x_i(t)$ is above the estimate of the network average $\left(\sum_{j=1}^{n}x_j(t-\tau_{ij})\right)/n$ by the threshold amount $y_0$, then it sends data to the other nodes; while if it is less than its estimate of the network average, nothing is sent. The hysteresis loop is put in to prevent chattering. (In the time interval $[t_1,t_2]$, the $j^{th}$ node receives the fraction $\int_{t_1}^{t_2}p_{ji}\left(t_{p_i}/t_{p_j}\right)U_m(x_i)\eta_i(t)dt$ of transferred waiting time $\int_{t_1}^{t_2}U_m(x_i)\eta_i(t)dt$ delayed by the time $h_{ij}$.)

However, there is additional information that can be made available to the nodes. Specifically, the information of the tasks that are in the network being sent to the $i^{th}$ node $q_{net_i}$ or equivalently, the waiting time $x_{net_i}\triangleq t_{p_i}q_{net_i}$. Here it is proposed to base

the controller not only on the local queue size $q_i$, but also use information about the number of tasks $q_{net_i}$ being sent to node $i$. The node $j$ sends to each node $i$ in the network information on the number of tasks $q_{net_{ij}}$ it has decided to send to each of the other nodes in the network. This way the other nodes can take into account this information (without having to wait for the actual arrival of the tasks) in making their control decision. The communication of the number of tasks $q_{net_{ij}}$ being sent from node $j$ to node $i$ is much faster than the actual transfer of the tasks. Furthermore, each node $i$ also broadcasts its total (anticipated) amount of tasks, i.e., $q_i + q_{net_i}$ to the other nodes so that they have a more current estimate of the tasks on each node (rather than have to wait for the actual transfer of the tasks). The information that each node has will be a more up to date estimate of the state of network using this scheme.

Define

$$z_i \triangleq x_i + x_{net_i} = t_{p_i}(q_i + q_{net_i}) \tag{12}$$

which is the *anticipated* waiting time at node $i$. Further, define

$$z_{i\_avg} \triangleq \left( \sum_{j=1}^{n} z_j(t - \tau_{ij}) \right) / n \tag{13}$$

to be the $i^{th}$ node's estimate of the average anticipated waiting time of all the nodes in the network. This is still an estimate due to the communication delays. Therefore,

$$w_i(t) \triangleq x_i(t) - z_{i\_avg}(t) = x_i(t) - \frac{\sum_{j=1}^{n} z_j(t - \tau_{ij})}{n}$$

to be the anticipated waiting time relative to the estimate of average (anticipated) waiting time in the network by the $i^{th}$ node. A control law based on the anticipated waiting time is chosen as

$$\eta_i(t) = h(w_i(t)). \tag{14}$$

The difference between (14) and (11) will be illustrated in Section 6.

### 4.1 Nonlinear Model with Non Constant $p_{ij}$

The model (1) did not have the $p_{ij}$ specified explicitly. For example, they can be considered constant as specified by (2). However, it is useful to use the local information of the waiting times $x_i(t), i = 1, .., n$ or the anticipated waiting time $z_i(t), i = 1, .., n$ to set their values.

#### $p_{ij}$ based on the $x_i$

Recall that $p_{ij}$ is the fraction of $\int_{t_1}^{t_2} U_m(x_j)\eta_j(t)dt$ in the interval of time $[t_1, t_2]$ that node $j$ allocates (transfers) to node $i$ and conservation of the tasks requires $p_{ij} \geqslant 0, \sum_{i=1}^{n} p_{ij} = 1$ and $p_{jj} = 0$. The quantity $x_i(t - \tau_{ji}) - x_{j\_avg}$ represents what node $j$ estimates the waiting time in the queue of node $i$ to be relative to node $j$'s estimate of

the network average. If the queue of node $i$ is above the network average as estimated by node $j$, then node $j$ does not send any tasks to it. Define a saturation function by

$$\text{sat}(x) = \begin{cases} x \text{ if } x > 0 \\ 0 \text{ if } x \leq 0. \end{cases}$$

Then $\text{sat}(x_{j\_avg} - x_i(t - \tau_{ji}))$ is node $j$'s estimate of how much node $i$ is *below* the network average. Node $j$ then repeats this computation for all the other nodes and portions out its tasks among the other nodes according to the amounts they are *below* its estimate of the network average, that is,

$$p_{ij} = \frac{\text{sat}\left(x_{j\_avg} - x_i(t - \tau_{ji})\right)}{\sum_{i \ni i \neq j} \text{sat}\left(x_{j\_avg} - x_i(t - \tau_{ji})\right)}. \tag{15}$$

This is illustrated in Figure 2.

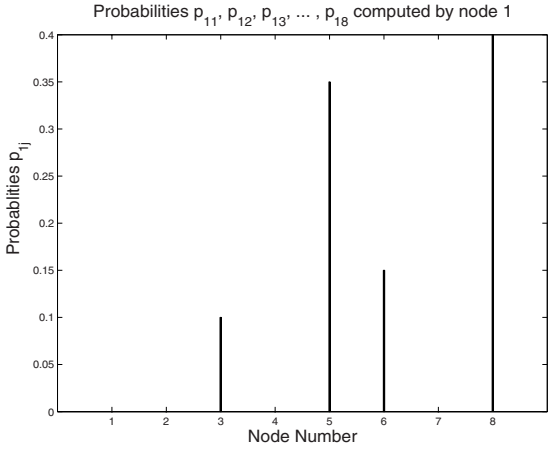



**Fig. 2.** Illustration of a hypothetical distribution $p_{i1}$ of the load at some time $t$ from node 1's point of view. Node 1 will send data out to node $i$ in the proportion $p_{i1}$ that it estimates node $i$ is below the average where $\sum_{i=1}^{n} p_{i1} = 1$ and $p_{11} = 0$

**Remark** If the denominator

$$\sum_{i \ni i \neq j} sat\left(x_{j\_avg} - x_i(t - \tau_{ji})\right)$$

is zero, then $x_{j\_avg} - x_i(t - \tau_{ji}) \leq 0$ for all $i \neq j$. However, by definition of the average,

$$\sum_{i \ni i \neq j} \left(x_{j\_avg} - x_i(t - \tau_{ji})\right) + x_{j\_avg} - x_j(t) = \sum_{i} \left(x_{j\_avg} - x_i(t - \tau_{ji})\right) = 0$$

which implies

$$x_{j\_avg} - x_j(t) = -\sum_{i \ni i \neq j} \Big( x_{j\_avg} - x_i(t - \tau_{ji}) \Big) \geq 0.$$

That is, if the denominator is zero, the expected waiting time on node $j$ is not greater than its estimate of the network average and therefore it is not sending out any tasks.

### $p_{ij}$ based on the $z_i$

Similarly, the $p_{ij}$ can be specified using the anticipated waiting time $z_j$ of the other nodes. The quantity $z_{j\_avg} - z_i(t - \tau_{ji})$ represents what node $j$ estimates the network's average anticipated waiting time is relative to its estimate of the anticipated waiting time in the queue of node $i$. If the estimate of the queue of node $i$ (i.e., $z_i(t - \tau_{ji})$) is above what node $j$ estimates the network's average (i.e., $z_{j\_avg}$) is, then node $j$ sends tasks to node $i$. Otherwise, node $j$ sends no tasks to node $i$. Therefore $\text{sat}(z_{j\_avg} - z_i(t - \tau_{ji}))$ is a measure by node $j$ as to how much node $i$ is *below* the local average. Node $j$ then repeats this computation for all the other nodes and then portions out its tasks among the other nodes according to the amounts they are below its estimate of the network average, that is,

$$p_{ij} = \frac{\text{sat}\,(z_{j\_avg} - z_i(t - \tau_{ji}))}{\sum_{i \ni i \neq j} \text{sat}\,(z_{j\_avg} - z_i(t - \tau_{ji}))}. \tag{16}$$

It is obvious that $p_{ij} \geqslant 0, \sum_{i=1}^{n} p_{ij} = 1$ and $p_{jj} = 0$. All $p_{ij}$ are defined to be zero, and no load is transferred if the denominator is zero, i.e., if

$$\sum_{i \ni i \neq j} sat \Big( z_{j\_avg} - z_i(t - \tau_{ji}) \Big) = 0.$$

As in the above remark, it is easy to see that in this case node $j$ would be below (its estimate of) the network average.

## 5 Model Parameters and Experimental Setup

### 5.1 Model Parameters

In this section, the determination of the model parameters is discussed. Experiments were performed to determine the value of $U_{m0}$ and the threshold $y_0$. The top plot in Figure 3 is the experimentally determined time to transfer data from one node to another in microseconds as function of message size in bytes. Each host measures the average, minimum and maximum times required to exchange data between itself and every other node in the parallel virtual machine (PVM) environment. The node starts the timer when initiating a transfer and stops the timer when it receives the

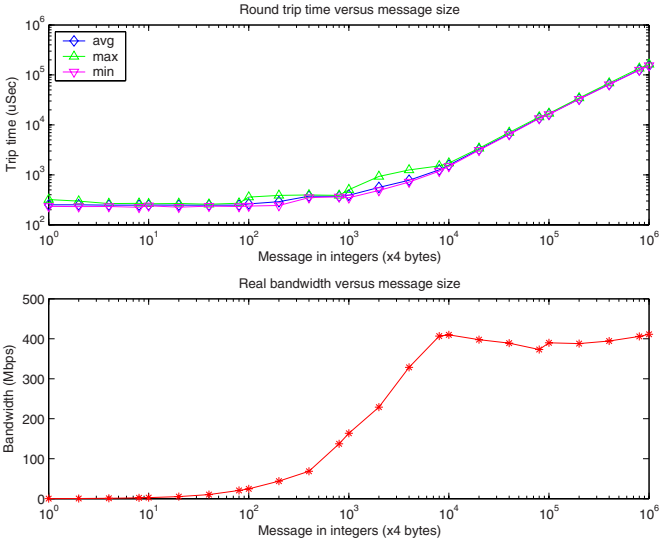



**Fig. 3.** Top: Round trip time vs. amount of data transfered in bytes. Bottom: Measured bandwidth vs. amount of data transfered in bytes.

data back, so the round-trip transfer time is measured. This also avoids the problem of synchronizing clocks on two different machines. The data sizes vary from 4 bytes to 4 Mbytes. In order to ensure that anomalies in message timings are minimized the tests are repeated 20 times for each message size. The bottom plot in Figure 3 is the experimentally determined bandwidth in Mbps versus the message size in bytes. Based on this data, the threshold for the size of the data transfer could be chosen to be less than $4 \times 10^4$ bytes so that these data are transferred at a bandwidth of about 400 Mbps. Messages of larger sizes will not improve the bandwidth and result in higher task transfer delays. That means the system must yield more computing time for communication time. Here the threshold is chosen to be $4 \times 10^3$ bytes since, as the top of Figure 3 shows, this is a trade-off between bandwidth and transfer time. With a typical task to be transferred of size 400 bytes/task (3200 bits/task), this means that the threshold is 10 tasks. The hysteresis threshold $y_0$ is then

$$y_0 = 10 \times t_{pi},$$

while the bandwidth constraint $U_{m0}$ is given by

$$\frac{U_{m0}}{t_{pi}} = \frac{400 \times 10^6 \text{ bps}}{3200 \text{ bits/task}} = 12.5 \times 10^4 \text{ tasks/second}$$

$$U_{m0} = 12.5 \times 10^4 \times t_{pi} \frac{\text{(waiting-time) seconds}}{\text{second}}$$

where the average processing time $t_{pi} = 400\mu$sec is used.

### 5.2 Experimental Setup of the Parallel Machine

A parallel machine has been built and used as an experimental facility for evaluation of load balancing strategies. A root node communicates with $k$ groups of networked computers. Each of these groups is composed of $n$ nodes (hosts) holding identical copies of a portion of the database. (Any pair of groups correspond to different databases, which are not necessarily disjoint. A specific record is in general stored in two groups for redundancy to protect against failure of a node.) Within each node, there are either one or two processors. In the experimental facility, the dual processor machines use 1.6 GHz Athlon MP processors, and the single processor machines use 1.33 GHz Athlon processors. All run the Linux operating system. Our interest here is in the load balancing in any one group of $n$ nodes/hosts.

The database is implemented as a set of queues with associated search engine threads, typically assigned one per node of the parallel machine. The search engine threads access tree-structured inidices to locate database records that match search or store requests. The structure of the network is shown in Figure 4. Due to the structure

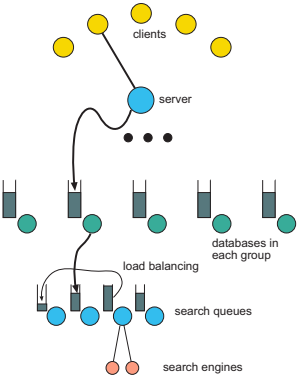



**Fig. 4.** A depiction of multiple search threads in the database index tree. To even out the search queues, load balancing is done between the nodes (hosts) of a group. If a node has a dual processor, then it can be considered to have two search engines for its queue.

of the search process, search requests can be formulated for any target profile and associated with any node of the index tree. These search requests are created not only by the database clients; the search process itself also creates search requests as the index tree is descended by any search thread. This creates the opportunity for parallelism; search requests that await processing may be placed in any queue associated with a search engine, and the contents of these queues may be moved arbitrarily among the processing nodes of a group to achieve a balance of the load.

## 6 Experimental Results

Experiments are presented to indicate the effects of time delays in load balancing. In the first set of experiments, the load balancing is performed *once at a fixed time* and is referred to as an "open loop" run. This open loop experiment is done in order to facilitate an explanation of the system dynamics with the effects of the delays. The open loop experiments are done for two cases. The first case uses the $p_{ij}$ specified by (15) which are based on the $x_i$ and the second case uses the $p_{ij}$ specified by (16) which are based on the $z_i$.

In the second set of experiments, the closed loop controller (11) with the $p_{ij}$ specified by (15) is compared with the closed loop controller (14) with the $p_{ij}$ specified by (16).

### 6.1 Open Loop Experiments

**Case 1:** $p_{ij}$ **based on** $x_i$

Figure 5 is the experimental response of the queues versus time with an initial queue distribution of $q_1(0) = 600$ tasks, $q_2(0) = 200$ tasks and $q_3(0) = 100$ tasks. The average time to do a search task is $t_{p_i} = 400\,\mu$sec. In this open loop experiment, the software was written to execute the load balancing algorithm at $t_0 = 1$ millisec using the $p_{ij}$ as specified by (15). Figure 5 shows that the data transfer delay from node 1 to node 2 is $h_{21} = 1.8$ millisec while the data transfer delay from node 1 to node 3 is $h_{31} = 2.8$ millisec. In this experiment the inputs were set as $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0$. All time symbols for this experimental run are shown in Figure 7.

Figure 6 is a plot of each nodes' estimate of the network average, i.e.,

$$q_{i\_avg}(t) \triangleq \left( \sum_{j=1}^{n} q_j(t - \tau_{ij}) \right) / n$$

where $q_{i\_avg}(t) = x_{i\_avg}(t)/t_{p_i}$ (see (10)). Figure 7 is a plot of the queue size relative to its estimate of the network average, i.e.,

$$q_{i\_diff}(t) \triangleq q_i(t) - q_{i\_avg}(t) = q_i(t) - \left( \sum_{j=1}^{n} q_j(t - \tau_{ij}) \right) / n$$

for each of the nodes. Note the effect of the delay in terms of what each local node *estimates* as the queue average and therefore whether it computes itself to be above or below it. Figure 6 is now discussed in detail as follows.

At the time of load balancing $t_0 = 1$ millisec, node 1 computes its queue size *relative* to its estimate of the network average $q_{1\_diff}$ to be 300, node 2 computes its queue size *relative* to its estimate of the network average $q_{2\_diff}$ to be $-100$ and node 3 computes its queue size *relative* to its estimate of the network average $q_{3\_diff}$ to be $-200$. Node 1 sends tasks out according to (15), which transfers about 100 and 200 tasks to node 2 and node 3 respectively.

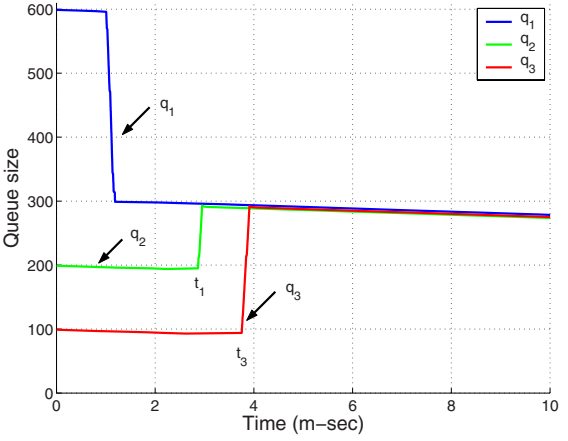



**Fig. 5.** Results of the load balancing algorithm executed at $t_0 = 1$ msec.

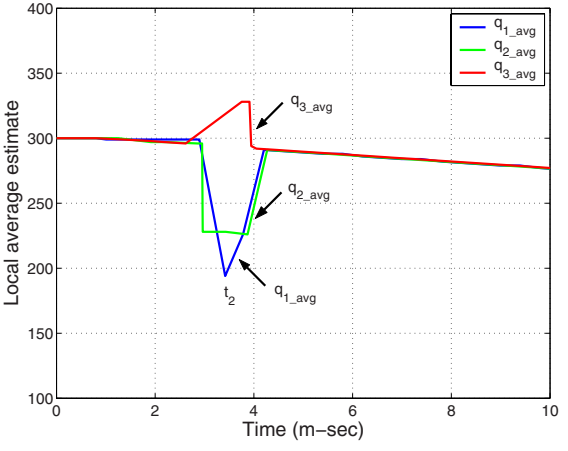



**Fig. 6.** Plot of $q_{i\_avg}(t)$ for $i = 1, 2, 3$.

At time $t_1$ node 2 receives 100 tasks from node 1 and updates its local queue size to be 300 as well as its estimate of node 1's queue size to be (about) 300, so that node 2's estimate of the network average is now $(300 + 300 + 100)/3 \approx 233$ so that its estimate of its queue size relative to the estimate of the network average is now $q_{2\_diff} \approx 300 - 233 = 67$. That is, node 2 is not aware of the 200 tasks being sent from node 1 to node 3.

At time $t_2$ node 1 has already sent out the tasks to node 2 and 3 and updated to its local queue size to 300. Node 1 receives the broadcast of two queue sizes of node 2

and 3 which are still 200 and 100 respectively, as the task transfers from node 1 are still on the way to node 2 and 3. Thus, node 1's estimate of the network average is about $(300+200+100)/3=200$ making the computation of its queue size relative to the estimate of the network average now $q_{1\_diff} \approx 300-200=100$.

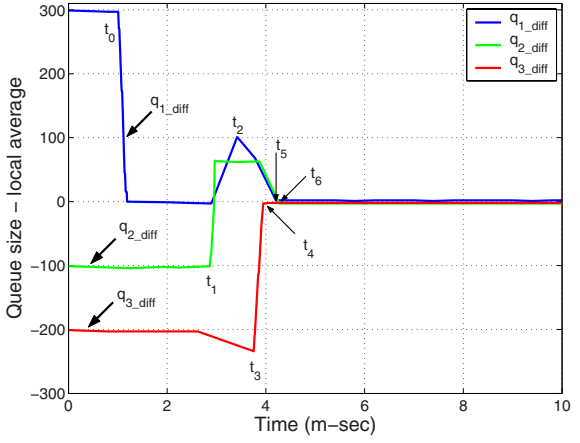



**Fig. 7.** Plot of $q_{i\_diff}(t) = q_i(t) - \left(\sum_{j=1}^{n} q_j(t-\tau_{ij})\right)/n$ for $i=1,2,3$ using $p_{ij}$ specified in (15).

At time $t_3$, node 3 receives the queue size of node 2 (which has already increased to about 300 as shown in Figure 5). Node 3 now computes $q_{3\_diff}$ to be about $(100-(600+300+100)/3) \approx -233$.

At time $t_4$, node 3 finally receives the 200 tasks from node 1 and updates its queue size and its estimate of node 1's queue size to be about 300. The network average computed by node 3 is now about $(300+300+300)/3=300$ so that $q_{3\_diff} \approx (300-300)=0$.

At time $t_5$, node 1 receives the communications for the queue sizes of node 2 and node 3 (which is now about 300 - see Figure 5). Then node 1 computes $q_{1\_diff} \approx (300-(300+300+300)/3)=0$.

Finally, at time $t_6$, node 2 receives the queue size of node 3 (which is now about 300 - see Figure 5). Node 2 now computes its queue size relative to local average $q_{1\_diff} \approx (300-(300+300+300)/3)=0$.

The transfer of 200 tasks to node 3 takes more time than that of 100 tasks to node 2. It is the task transfer delay that delays node 2's receipt of the new queue size of node 3 at time $t_1$, and that delays node 3's receipt of the tasks until $t_4 > t_3$. It is the communication delay that delays node 1's receipt of the new queue size of node 2 at time $t_2$. The effect of both delays could cause the unnecessary transfers for node 2 at time $t_1$ and node 1 at time $t_2$ if the load balancing is done in closed loop.

**Case 2:** $p_{ij}$ **based on** $z_i$

The node that transfers tasks computes the amounts to be sent to other nodes according to how far below the network average its estimate of each recipient node's queue is. Therefore, it is feasible to send the amounts of its next task transfers to each of other nodes before actually transferring tasks. Such communications are efficient; the communication delay of each transferred measurement is much smaller than the actual task transfer delays. A key finding of this research is that knowledge of the anticipated queue sizes can be used to compensate the effect of delays. Figure 8 is a plot of the local average of queue sizes on all nodes using measurements of anticipated transfers

$$q^{est}_{i\_avg}(t) \triangleq \left(\sum_{j=1}^{n} q^{est}_{j}(t-\tau_{ij})\right)/n = \left(\sum_{j=1}^{n} q_j(t-\tau_{ij}) + q_{net_j}(t)\right)/n$$

where $q_{net_j}(t) = \sum_{i=1}^{n} q_{net_{ji}}(t-\tau_{ij})$. The nodes have an initial queue distribution of $q_1(0) = 600$ tasks, $q_2(0) = 200$ tasks and $q_3(0) = 100$ tasks. The average time to do a search task is $t_{p_i} = 400\ \mu$sec. Note that $q^{est}_{i\_avg}(t) = z_{i\_avg}(t)/t_{p_i}$ (see (13)). In this experiment, the load balancing algorithm was executed at $t_0 = 1$ millisec using the $p_{ij}$ as specified by (16) with $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0$. Compared with Figure 6, the estimates by the nodes of the network average are substantially improved. Figure 9

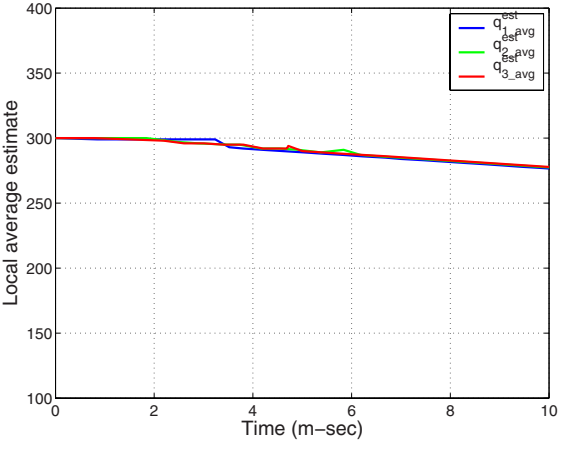



**Fig. 8.** Plot of $q^{est}_{i\_avg}(t) = \left(\sum_{j=1}^{n} q^{est}_{j}(t-\tau_{ij})\right)/n$ for $i = 1,2,3$ using $p_{ij}$ by (16).

is a plot of the queue size relative to the local average, i.e.,

$$q^{est}_{i\_diff}(t) \triangleq q_i(t) - q^{est}_{i\_avg}(t) = q_i(t) - \left(\sum_{j=1}^{n} q^{est}_{j}(t-\tau_{ij})\right)/n$$

for each of the nodes. Note the effect of the delay in terms of what each local node *estimates* as the queue average and therefore it determines that whether it is above or below the network average is greatly diminished. Compared with Figure 7, the tracking differences are near or below zero and no further unnecessary transfers are initiated. Figures 10(a) and 10(b) show the estimates of the queue sizes by node 2

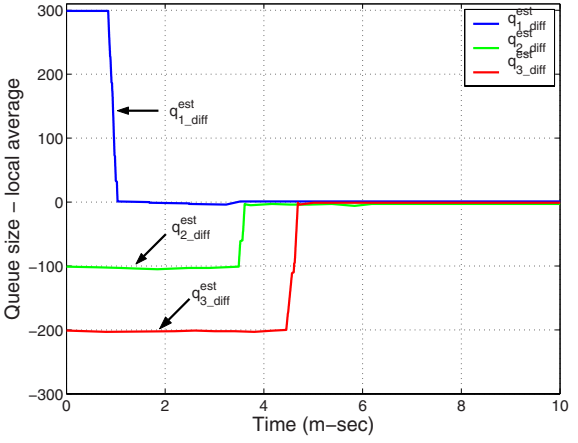



**Fig. 9.** Plot of $q^{est}_{i\_diff}(t) = q_i(t) - \left(\sum_{j=1}^{n} q^{est}_j(t-\tau_{ij})\right)/n$ for $i = 1,2,3$ using $p_{ij}$ by (16).

when using $x_i$ and when using $z_i$, respectively. In Figure 10(a), after node 2 received the tasks from node 1, node 2 computed its queue size relative to its local average to be about 67. In Figure 10(b), node 1 sends the numbers of tasks to be transferred before actually transferring tasks to the other nodes; node 2 receives the estimate of node 3's queue size and updates its local queue size relative to its estimate of the network average to be zero.

Figures 11(a) and 11(b) show the estimates of the queue sizes by node 3 when using $x_i$ and when using $z_i$, respectively. In Figure 11(a), when node 2 broadcasts its updated queue size to node 3, node 3 has not received the tasks sent from node 1 due to transfer delays. Node 3 updates its queue size relative to its estimate of the network average to be $-233$. In Figure 11(b), node 1 sends the numbers of tasks to be transferred before transferring actual tasks to the other nodes; node 3 receives the anticipated estimate of the queue sizes on other nodes, and computes its queue size relative to its estimate of the network average to $-200$. When node 3 receives the tasks transferred from node 1, it updates its local queue and computes the amount relative to the network average to be zero. Compared with Figures 10 and 11 shown above, these anticipated estimates significantly compensate the effect of delays of task transfers. The comparison of local queue sizes and anticipated estimates on the two recipient nodes 2 and 3 are shown in Figures 12 (a) and (b) respectively.

### 6.2 Closed Loop Experiments

The following experiments show the responses using closed loop load balancing with an initial queue distribution of $q_1(0) = 600$ tasks, $q_2(0) = 200$ tasks and $q_3(0) = 100$ tasks. The average time to do a search task is 400 $\mu$sec. After initial communications, the closed loop load balancing algorithm (see Section 4) was initiated using the $p_{ij}$ as specified by (15) and by (16), respectively. In these experiments the inputs were set as $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0$. Figures 13(a) and (b) show the responses of the queues versus time using $p_{ij}$ specified in (15) and $p_{ij}$ specified in (16) respectively. Note the substantial difference in the load balancing performance between these two schemes.

The tracking differences between local queue sizes and estimated network averages (from delayed values of other nodes) using $p_{ij}$ by (15) are shown in Figure 14(a). There are unnecessary exchanges of tasks back and forth among nodes. Although the system reaches a balanced condition at around $t = 14$ millisec, those additional transfers cost processing time and prolong the completion time. Figure 14(b) shows the tracking differences between local queue sizes and anticipated estimates of the network averages using $p_{ij}$ by (16) based on $z_i$.

Figure 15(a) shows node 2's estimates of the queue sizes in the network of closed loop load balancing using $p_{ij}$ based on the $x_i$. Node 2 estimates the network average using the delayed information from other nodes only, and its controller based on its queue size relative to the estimated average causes those unnecessary exchanges of tasks back and forth, as shown in Figure 15(a). Figure 15(b) shows node 2's estimates of the anticipated queue sizes in the network of closed loop load balancing using $p_{ij}$ based on the $z_i$. Node 1 sends the numbers of tasks before actually transferring tasks to the other nodes. Node 2 receives the anticipated estimate of node 3's queue size and updates its queue size relative to its estimate of the network average as zero. From the Figure 15, we can see that the anticipated estimates are used to compensate the effect of delays of task transfers so that there are no unnecessary task transfers initiated.

## 7 Summary and Conclusions

In this work, a load balancing algorithm was modeled as a nonlinear time-delay system. It was shown that the model was consistent in that the total number of tasks was conserved and the queues were always non negative. It was also shown the system was always stable, but not necessarily asymptotically stable. Furthermore, a closed loop controller was proposed based on the local queue size and estimate of the tasks being sent to the queue from other nodes. Experimental results showed a dramatic increase in performance obtained in using this information in the control law.

**Acknowledgements.**
The work of J. D. Birdwell, J. Chiasson and Z. Tang was supported by U.S. Department of Justice, Federal Bureau of Investigation under contract J-FBI-98-083 and by the National Science Foundation grant number ANI-0312182. Drs. Birdwell and Chiasson were also partially supported by a Challenge Grant Award from the Center for Information Technology Research at the University of Tennessee. Drs. C. T. Abdallah and M. M. Hayat were supported by the National Science Foundation under grant number ANI-0312611. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Government.

## References

[1] J. D. Birdwell, R. D. Horn, D. J. Icove, T. W. Wang, P. Yadav, and S. Niezgoda, "A hierarchical database design and search method for CODIS," in *Tenth International Symposium on Human Identification*, September 1999. Orlando, FL.

[2] J. D. Birdwell, T. W. Wang, R. D. Horn, P. Yadav, and D. J. Icove, "Method of indexed storage and retrieval of multidimensional information," in *Tenth SIAM Conference on Parallel Processing for Scientific Computation*, September 2000. U. S. Patent Application 09/671,304.

[3] J. D. Birdwell, T.-W. Wang, and M. Rader, "The university of Tennessee's new search engine for CODIS," in *6th CODIS Users Conference*, February 2001. Arlington, VA.

[4] T. W. Wang, J. D. Birdwell, P. Yadav, D. J. Icove, S. Niezgoda, and S. Jones, "Natural clustering of DNA/STR profiles," in *Tenth International Symposium on Human Identification*, September 1999. Orlando, FL.

[5] A. Corradi, L. Leonardi, and F. Zambonelli, "Diffusive load-balancing polices for dynamic applications," *IEEE Concurrency*, vol. 22, pp. 979–993, Jan-Feb 1999.

[6] M. H. Willebeek-LeMair and A. P. Reeves, "Strategies for dynamic load balancing on highly parallel computers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 9, pp. 979–993, 1993.

[7] C. K. Hisao Kameda, Jie Li and Y. Zhang, *Optimal Load Balancing in Distributed Computer Systems*. Springer, 1997. Great Britain.

[8] H. Kameda, I. R. El-Zoghdy Said Fathy, and J. Li, "A performance comparison of dynamic versus static load balancing policies in a mainframe," in *Proceedings of the 39th IEEE Conference on Decision and Control*, pp. 1415–1420, December 2000. Sydney, Australia.

[9] E. Altman and H. Kameda, "Equilibria for multiclass routing in multi-agent networks," in *Proceedings of the 40th IEEE Conference on Decision and Control*, pp. 604–609, December 2001. Orlando, FL USA.

[10] L. Kleinrock, *Queuing Systems Vol I : Theory*. John Wiley & Sons, 1975. New York.

[11] F. Spies, "Modeling of optimal load balancing strategy using queuing theory," *Microprocessors and Microprogramming*, vol. 41, pp. 555–570, 1996.

[12] C. T. Abdallah, N. Alluri, J. D. Birdwell, J. Chiasson, V. Chupryna, Z. Tang, and T. Wang, "A linear time delay model for studying load balancing instabilities in parallel computations," *The International Journal of System Science*, vol. 34, pp. 563–573, August-September 2003.

[13] M. Hayat, C. T. Abdallah, J. D. Birdwell, and J. Chiasson, "Dynamic time delay models for load balancing, Part II: A stochastic analysis of the effect of delay uncertainty," in *CNRS-NSF Workshop: Advances in Control of Time-Delay Systems, Paris France*, January 2003.

[14] C. T. Abdallah, J. D. Birdwell, J. Chiasson, V. Chupryna, Z. Tang, and T. W. Wang, "Load balancing instabilities due to time delays in parallel computation," in *Proceedings of the 3rd IFAC Conference on Time Delay Systems*, December 2001. Sante Fe NM.

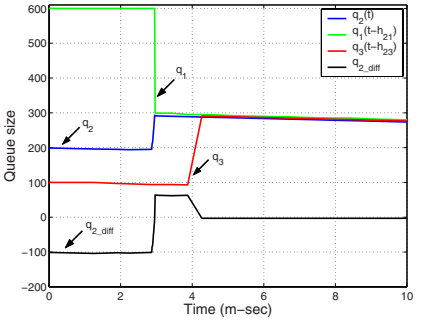
[15] J. D. Birdwell, J. Chiasson, Z. Tang, C. T. Abdallah, M. Hayat, and T. Wang, "Dynamic time delay models for load balancing Part I: Deterministic models," in *CNRS-NSF Workshop: Advances in Control of Time-Delay Systems, Paris France*, January 2003.

[16] J. D. Birdwell, J. Chiasson, Z. Tang, C. T. Abdallah, M. Hayat, and T. Wang, *Dynamic Time Delay Models for Load Balancing Part I: Deterministic Models*, pp. 355–368. CNRS-NSF Workshop: Advances in Control of Time-Delay Systems, Keqin Gu and Silviu-Iulian Niculescu, Editors, Springer-Verlag, 2003.

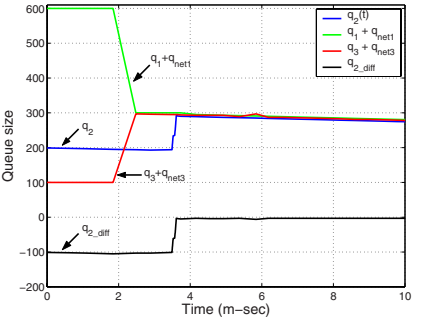
[17] J. D. Birdwell, J. Chiasson, Z. Tang, C. T. Abdallah, and M. M. Hayat, "The effect of feedback gains on the performance of a load balancing network with time delays," in *IFAC Workshop on Time-Delay Systems TDS03*, September 2003. Rocquencourt, France.

[18] J. D. Birdwell, J. Chiasson, C. T. Abdallah, Z. Tang, N. Alluri, and T. Wang, "The effect of time delays in the stability of load balancing algorithms for parallel computations," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, December 2003. Maui, Hi.

[19] H. K. Khalil, *Nonlinear Systems Third Edition*. Prentice-Hall, 2002.

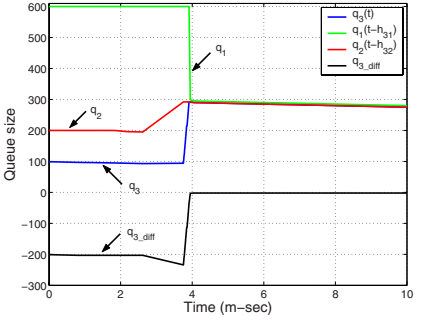



*(a)* node 2: using $x_i$

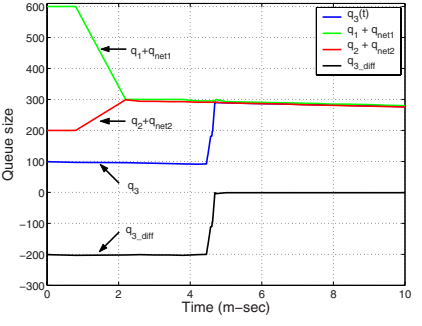



*(b)* node 2: using $z_i$

**Fig. 10.** Plot of the estimates by node 2 of the queue sizes (a) using $x_i$ and (b) using $z_i$.

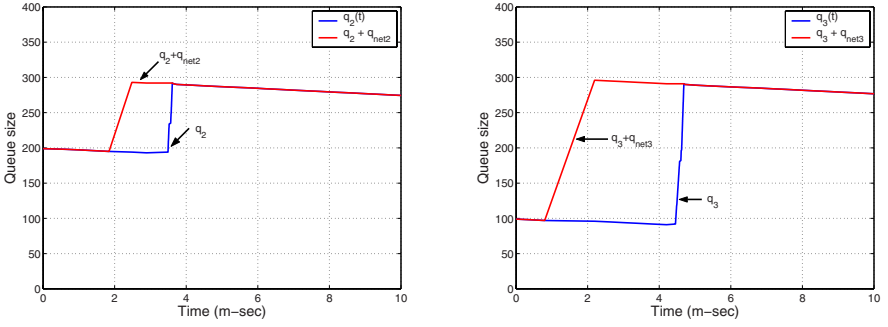



*(a)* node 3: using $x_i$



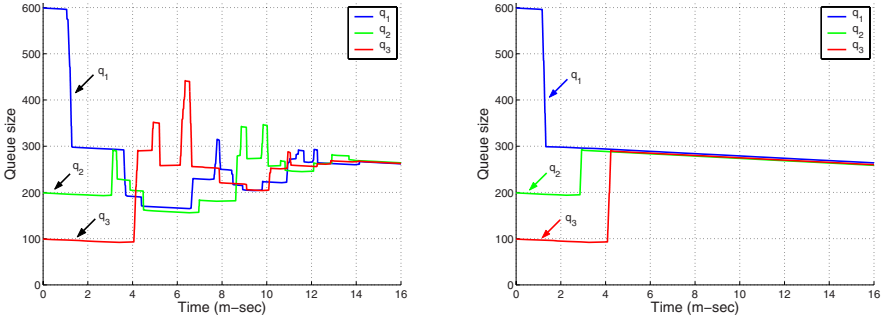

*(b)* node 3: using $z_i$

**Fig. 11.** Plot of the estimates by node 3 of the queue sizes (a) using $x_i$ and (b) using $z_i$.



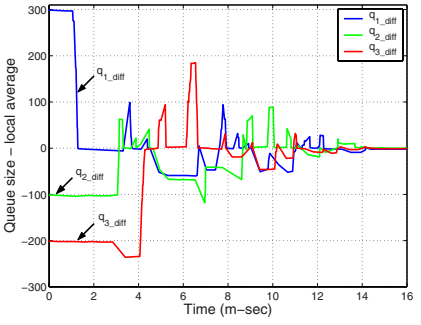

*(a)* node 2 *(b)* node 3

**Fig. 12.** Comparison of the actual queue size and the anticipated estimate of queue size on (a) node 2 and (b) node 3.

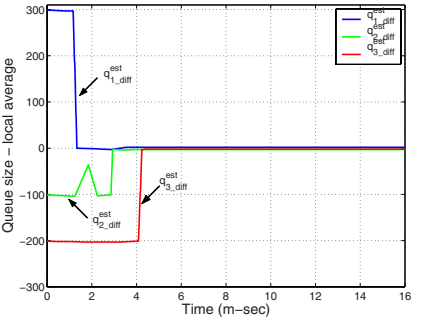



*(a)* queue sizes: using $x_i$ *(b)* queue sizes: using $z_i$

**Fig. 13.** Results of closed loop load balancing. (a) Plot of queue size using $x_i$ and (15) and (b) queue sizes using $z_i$ and (16).

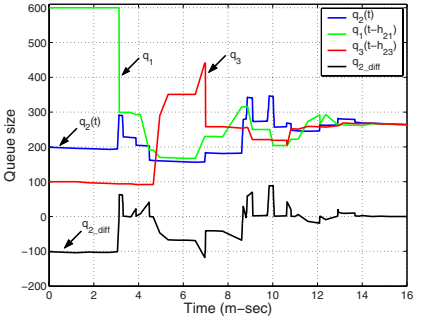



*(a)* $q_{i\_diff}(t)$ by (15) based on $x_i$

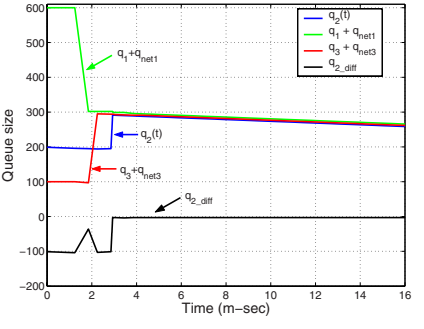



*(b)* $q^{est}_{i\_diff}(t)$ by (16) based on $z_i$

**Fig. 14.** Plot of tracking difference between the queue size and the estimated network average on each of the nodes in closed loop load balancing. (a) $q_{i\_diff}(t) = q_i(t) - \left(\sum_{j=1}^{n} q_j(t-\tau_{ij})\right)/n$ and (b) $q^{est}_{i\_diff}(t) = q_i(t) - \left(\sum_{j=1}^{n} q^{est}_j(t-\tau_{ij})\right)/n$ for $i = 1,2,3$.




*(a)* node 2: using $x_i$




*(b)* node 2: using $z_i$

**Fig. 15.** Plot of estimated queue sizes by node 2 in closed loop load balancing (a) using $x_i$ and (15), and (b) using $z_i$ and (16).

# Position and Force Tracking in Bilateral Teleoperation

Nikhil Chopra[1], Mark W. Spong[1], Romeo Ortega[2], and Nikita E. Barabanov[3]

[1] Coordinated Science Laboratory
University of Illinois
1308 W. Main St.
Urbana IL 61801 USA (nchopra,mspong)@uiuc.edu
[2] LSS/CNRS/Supélec
Plateau de Moulon
91192 Gif Sur Yvette Cedex, France Romeo.Ortega@lss.supelec.fr
[3] Department of Software Engineering
Electrotechnical University
St. Petersburg, Russia nikita@freya.etu.ru

## 1 Introduction

A teleoperator is a dual robot system in which a remote slave robot tracks the motion of a master robot, which is, in turn, commanded by a human operator. To improve the task performance, information about the remote environment is needed. Feedback can be provided to the human operator by many different forms, including audio, visual displays, or tactile. However, force feedback from the slave to the master, representing contact information, provides a more extensive sense of telepresence. When this is done the teleoperator is said to be controlled bilaterally.
In bilateral teleoperation, the master and the slave manipulators are coupled via a communication network and time delay is incurred in transmission of data between the master and slave site. It is well known that the delays in a closed loop system can destabilize an otherwise stable system. Time delay instability in force reflecting teleoperation was a long standing impediment to bilateral teleoperation with force feedback. The breakthrough to the bilateral teleoperation problem was achieved in [1] where concepts from Network Theory, Passivity and Scattering Theory were used to analyze mechanisms responsible for loss of stability and derive a time delay compensation scheme to guarantee stability independent of the (constant) delay. These results were then extended in [7], where the notion of wave-variables was introduced to define a new configuration for force-reflecting teleoperators.

In a bilateral teleoperator, apart from the basic necessity of a stable system, there are primarily two design goals which ensure a close coupling between the human operator and the remote environment. The first goal is that the slave manipulator should track the position of the master manipulator and the second goal is that the environmental force acting on the slave, when it contacts a remote environment, be accurately transmitted to the master. It this paper we primarily address the position

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 269–280, 2004.

tracking problem in a bilateral teleoperation system. The standard approaches to control of bilateral teleoperators over constant delay networks, based on the scattering approach [1] or the equivalent wave variable formulation [7], guarantee robust stability of the teleoperator but lead to sluggish response for high transmission delays, as observed in [5]. In [6], the fundamental limits of performance and design tradeoffs of bilateral teleoperation, without addressing a particular architecture, were analyzed. Several architectures were quantitatively compared in terms of transparency and stability, and the results demonstrated that although the passivity based approach was stable compared to other schemes such as [4], the passive architecture does not guarantee good transparency. Transparency, as defined in [6], is quantified in terms of a match between the environmental impedance and the impedance transmitted to the human operator. This definition of transparency is more suited for the case where the master and the slave have different workspace, but in this paper we work under the assumption that the master/slave robots have the same workspace, and therefore aim for correspondence between the master/slave position and force responses to be the measure of transparency [10, 12]. In [8] two methods to solve the position tracking problem were presented. The first one involved sending the integral of the wave variables to communicate direct position data to the slave side while the second method added a correction to the wave variables, based on the position error, to recover good tracking. The integral of the wave-variables contains position and momentum information and thus a passive coupling between the master/slave robots (which are passive from force to velocity) to the communication block transmitting position/momentum information is non-obvious. It was demonstrated in [9] that such a coupling is possible in a single-degree-of-freedom manipulator and a configuration to solve the position tracking problem was presented. Recently in [3], a feedforward position control was advocated to improve the position tracking performance. In this paper we propose a new architecture for bilateral teleoperation, which builds upon the model of [3], to solve the position tracking problem.

## 2 Bilateral Teleoperation: The Traditional Architecture

A teleoperator consists of the following subsystems: the human operator, the master, the communication block, the slave and the environment. The human operator commands the master with force $F_h$ to move it with velocity $\dot{x}_m$, which is sent to the slave through the communication block and a local control($F_s$) on the slave side drives the slave velocity $\dot{x}_s$ towards the master velocity. If the slave contacts a remote environment, the remote force $F_e$ is communicated back from the slave side and received at the slave side as the force $F_m$.

The passivity based standard bilateral teleoperation system of [1] with the scattering transformation is shown in Figure 1. Passivity is perhaps one of the most physically appealing concepts of system theory and it was used as a fundamental tool in the development of the bilateral teleoperator of [1]. A system is said to be passive if and only if the energy flowing in exceeds the flowing out for all time. A more practical and important feature of the passivity formulation is the fact that the interconnection

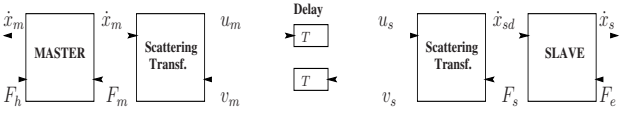



**Fig. 1.** A standard bilateral teleoperation setup

of two passive systems connected in either a feedback or a parallel configuration is again passive. A complementary concept to passivity is the idea of two-port system from circuit theory [11] which was used extensively in the control design in [1]. In the two-port network of Figure 2, we see four variables identified-two inputs ($x_l$, $y_r$) and two outputs ($y_l$, $x_r$). Passivity is then equivalent to the relationship

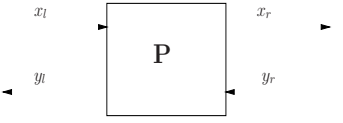



**Fig. 2.** A Two-Port Network

$$\int_0^t \left( x_l^T(\tau) y_l(\tau) - y_r^T(\tau) x_r(\tau) \right) d\tau \geq 0$$

The teleoperator can then be modelled as an electrical network, where the master, communication block can be represented and interconnected as two-ports and the operator and the environment are represented by one-ports. By designing control laws which impose the passivity property described earlier on each of the elements of the teleoperation system, claims can be made about the stability of the teleoperation system.

The scattering transformation approach in [1] or the equivalent wave variable transformation proposed in [7] guarantees passivity of the network block in the face of constant delay in the network. This transformation is given, using the notation of [7], as

$$\begin{aligned} u_m &= \tfrac{1}{\sqrt{2b}}(F_m + b\dot{x}_m) \; ; \; v_m = \tfrac{1}{\sqrt{2b}}(F_m - b\dot{x}_m) \\ u_s &= \tfrac{1}{\sqrt{2b}}(F_s + b\dot{x}_{sd}) \; ; \; v_s = \tfrac{1}{\sqrt{2b}}(F_s - b\dot{x}_{sd}) \end{aligned} \tag{1}$$

where $\dot{x}_m$ is the master velocity and $\dot{x}_{sd}$ is the velocity derived from the scattering transformation at the slave side. $F_m$ is the force that is reflected back to the master from the slave robot and $F_s$ is given as

$$F_s(t) = K_s \int_0^t (\dot{x}_{sd} - \dot{x}_s) dt + B_{s2}(\dot{x}_{sd} - \dot{x}_s)$$

The master and the slave are passive from force to velocity and the network block is passified by the scattering transformation. This system, when interconnected with a passive human operator and remote environment, is passive. However, this configuration places an inherent limitation on the transparency (measure of position and force tracking) of the system because instead of transmitting position signals, linear combination of velocity and force signals are transmitted from the master to the slave and vice versa.

Position tracking in bilateral teleoperation has emerged to be a two-faceted problem. Consider a task where the slave intermittently contacts the remote environment. During this task, the master and the slave might not have the same initial position after an environmental contact and as only the master velocity is transmitted across the channel codified in the wave variables, and is then integrated to recover the master position, is not possible for the slave to track the master position. This results in a drift between the master and the slave robot which might increase with time due to successive environmental contacts. Also, teleoperating devices over the Internet requires packets to be transmitted via an unreliable packet switched network. The network may induce packet drops which will lead to an unrecoverable position drift between the master and the slave robot.

The other case is the degradation of position tracking in the event of high network delays of the order of 0.5s or more. It is well known that position tracking in a teleoperation architecture, as described in this section, deteriorates with increase in the network delay. We demonstrate below that loss of tracking is a resulting pitfall of the scattering transformation or the wave-variable approach and in the next section we propose a new configuration to counter this problem. Using (1), the transmission equations can we rewritten as

$$\begin{aligned} b\dot{x}_{sd}(t) &= b\dot{x}_m(t-T) + F_m(t-T) - F_s(t) \\ F_m(t) &= F_s(t-T) + b\dot{x}_m(t) - b\dot{x}_{sd}(t-T) \end{aligned}$$

Using the above equations it follows that

$$b\dot{x}_{sd} = 2b\dot{x}_m(t-T) + F_s(t-2T) - F_s(t) - b\dot{x}_{sd}(t-2T) \tag{2}$$

Substituting the value of $F_s$, the above reduces to

$$\begin{aligned} \dot{x}_{sd} = {} & \frac{2b}{b+B_{s2}}\dot{x}_m(t-T) + \frac{B_{s2}-b}{b+B_{s2}}\dot{x}_{sd}(t-2T) + \frac{B_{s2}}{b+B_{s2}}(\dot{x}_s(t) - \dot{x}_s(t-2T)) \\ & + \frac{K_s}{b+B_{s2}}(\triangle x(t-2T) - \triangle x(t)) \end{aligned}$$

where $\triangle x(t) = \int_0^t (\dot{x}_{sd} - \dot{x}_s)dt$. Thus, the reference position for the slave is given as

$$\begin{aligned} x_{sd}(t) - x_{sd}(0) = {} & \frac{2b}{b+B_{s2}}(x_m(t-T) - x_m(0)) + \frac{B_{s2}-b}{b+B_{s2}}(x_{sd}(t-2T) - x_{sd}(0)) \\ & + \frac{1}{b+B_{s2}}\int_{t-2T}^{t}(B_{s2}\dot{x}_s(\tau) - K_s\triangle x(\tau))d\tau \end{aligned}$$

To improve the transient performance, impedance matching has been advocated in [7], which entails choosing $B_{s2} = b$. Using this in the above, and assuming that the $x_{sd}(0) = x_m(0)$, the above equation simplifies as

$$x_{sd}(t) = x_m(t-T) + \frac{1}{2b}\int_{t-2T}^{t}(B_{s2}\dot{x}_s(\tau) - K_s\triangle x(\tau))d\tau$$

It is easily seen from the above equation that even in the best-possible scenario, i.e. with matched impedance and no initial offsets, the reference position signal for the slave $x_{sd}$ is a function of the delay and transient position drift between the master and the slave robot increases with increase in delay. Thus use of the integral of the scattering variables is not a judicious choice in obtaining the reference signal (master position signal) for the controllers on the slave side. In the next section we propose a new architecture which solves the position tracking problem in bilateral teleoperation.

## 3 A New Architecture for Bilateral Teleoperation

The proposed architecture is shown in Figure 3, where the master and the slave posi-

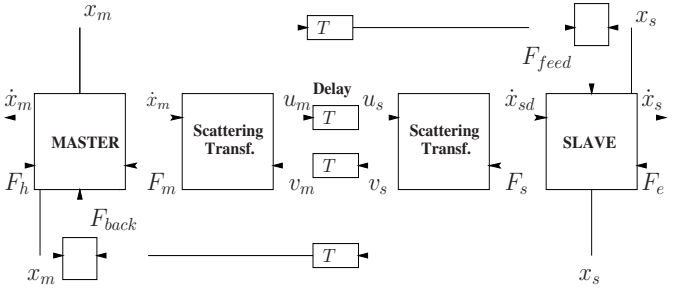



**Fig. 3.** A New Configuration for Bilateral Teleoperation

tion data are explicitly sent across the communication channel. This configuration is similar to the standard bilateral teleoperation setup, but has additional proportional controllers on the master and the slave side which use the delayed position data (from both master and slave) as the reference signal. For simplicity, the master and the slave have been modelled as mass-damper systems. The system dynamics are given by

$$\begin{aligned} M_m\ddot{x}_m + B_m\dot{x}_m &= F_h + F_{back} - F_m \\ M_s\ddot{x}_s + B_{s1}\dot{x}_s &= F_s + F_{feed} - F_e \end{aligned} \tag{3}$$

where $M_m$ and $M_s$ are the respective inertias and $B_m$, $B_{s1}$ represent the master and the slave damping respectively. $F_h$ is the operator torque, $F_e$ is the environment torque and other torques are defined as

$$\begin{aligned} F_s &= B_{s2}(\dot{x}_{sd} - \dot{x}_s) \\ F_{back} &= K(x_s(t-T) - x_m) \\ F_{feed} &= K(x_m(t-T) - x_s) \end{aligned} \tag{4}$$

In the analysis that follows we assume that

- The human operator and the environment can be modelled as passive systems.
- The operator and the environmental force are bounded by known functions of the master and the slave velocities respectively.
- All signals belong to $\mathcal{L}_{2e}$, the extended $\mathcal{L}_2$ space.
- The velocities $\dot{x}_m$ and $\dot{x}_s$ equal zero for t $< 0$.

Define the position tracking error as

$$e = x_m(t-T) - x_s(t) \tag{5}$$

where $x_m(t-T)$ is the delayed master position received on the slave side.

**Proposition 1.** *Consider the system described by* (1), (3), (4) *and Figure 3. Then for a range of the gain* ($0 < K < K^*$), *the master and slave velocities asymptotically converge to the origin and the position tracking error defined by* (5) *remains bounded.*

*Proof.* Define a positive definite function for the system as

$$\begin{aligned} V = \ & \frac{1}{2}\{M_m\dot{x}_m^2 + M_s\dot{x}_s^2 + K(x_m - x_s)^2\} + \int_0^t (F_e\dot{x}_s - F_h\dot{x}_m)d\tau \\ & + \int_0^t (F_m\dot{x}_m - F_s\dot{x}_{sd})d\tau \end{aligned} \tag{6}$$

The human operator and the remote environment are passive (by assumption). Hence

$$\int_0^t F_e\dot{x}_s d\tau \geq 0 \ ; \ -\int_0^t F_h\dot{x}_m d\tau \geq 0$$

Using the scattering transformation (1), we have

$$\int_0^t (F_m\dot{x}_m - F_s\dot{x}_{sd})d\tau = \frac{1}{2}\int_{t-T}^t (u_m^2 + v_m^2)d\tau \geq 0$$

which shows that the communication block is passive. Thus the function $V$ is positive-definite. The derivative of (6) along trajectories of the system is given by

$$\begin{aligned} \dot{V} &= M_m\dot{x}_m\ddot{x}_m + M_s\dot{x}_s\ddot{x}_s + K(x_m - x_s)(\dot{x}_m - \dot{x}_s) + F_m\dot{x}_m - F_s\dot{x}_{sd} \\ &\quad + F_e\dot{x}_s - F_h\dot{x}_m \\ &= \dot{x}_m(-B_m\dot{x}_m + K(x_s(t-T) - x_m) + F_h - F_m) + \dot{x}_s(-B_{s1}\dot{x}_s \\ &\quad + K(x_m(t-T) - x_s) + F_s - F_e) + K(x_m - x_s)(\dot{x}_m - \dot{x}_s) + (F_m - F_h)\dot{x}_m \\ &\quad - F_s\dot{x}_{sd} + F_e\dot{x}_s \\ &= -B_m\dot{x}_m^2 - B_{s1}\dot{x}_s^2 + (\dot{x}_{sd} - \triangle v)F_s - F_s\dot{x}_{sd} + K(x_s(t-T) - x_s)\dot{x}_m \\ &\quad + K(x_m(t-T) - x_m)\dot{x}_s \end{aligned}$$

where $\triangle v = \dot{x}_{sd} - \dot{x}_s$. Using the fact that

$$x_i(t-T) - x_i = -\int_0^T \dot{x}_i(t-\tau)d\tau \;\; ; \;\; i = m,s$$

and integrating the above equation we get

$$\int_0^{t_f} \dot{V}dt \leq -B_m||\dot{x}_m||_2^2 - B_{s1}||\dot{x}_s||_2^2 - B_{s2}||\triangle v||_2^2 - K\int_0^{t_f}\dot{x}_m\int_0^T \dot{x}_s(t-\tau)d\tau dt$$
$$-K\int_0^{t_f}\dot{x}_s\int_0^T \dot{x}_m(t-\tau)d\tau dt$$

where the notation $||\cdot||_2$ denotes the $\mathcal{L}_2$ norm of a signal on the interval $[0,t_f]$. Using Schwartz inequality and the fact that Arithmetic Mean(A.M.)$\geq$ Geometric Mean(G.M.), it is easily seen that, for any $\alpha_1, \alpha_2 > 0$

$$\begin{aligned} 2\int_0^{t_f}\dot{x}_m\int_0^T \dot{x}_s(t-\tau)d\tau dt \leq \;& \alpha_1\int_0^{t_f}\dot{x}_m^2 dt + \frac{1}{\alpha_1}\int_0^{t_f}\Big(\int_0^T \dot{x}_s(t-\tau)d\tau\Big)^2 dt \\ &\leq \alpha_1||\dot{x}_m||_2^2 + \frac{T}{\alpha_1}\int_0^{t_f}\int_0^T \dot{x}_s^2(t-\tau)d\tau dt \\ &\leq \alpha_1||\dot{x}_m||_2^2 + \frac{T^2}{\alpha_1}||\dot{x}_s||_2^2 \end{aligned}$$

Similarly, it can be shown that

$$2\int_0^{t_f}\dot{x}_s\int_0^T \dot{x}_m(t-\tau)d\tau dt \leq \; \alpha_2||\dot{x}_s||_2^2 + \frac{T^2}{\alpha_2}||\dot{x}_m||_2^2$$

Therefore the integral inequality reduces to

$$\int_0^t \dot{V}dt \leq -B_m||\dot{x}_m||_2^2 - B_{s1}||\dot{x}_s||_2^2 - B_{s2}||\triangle v||_2^2 + K\{(\frac{\alpha_1}{2} + \frac{T^2}{2\alpha_2})||\dot{x}_m||_2^2$$
$$+(\frac{\alpha_2}{2} + \frac{T^2}{2\alpha_1})||\dot{x}_s||_2^2\}$$

So in order for $\dot{x}_m, \dot{x}_s \in \mathcal{L}_2$, the following inequalities are sufficient to be satisfied

$$K(\frac{\alpha_1}{2} + \frac{T^2}{2\alpha_2}) < B_m$$
$$K(\frac{\alpha_2}{2} + \frac{T^2}{2\alpha_1}) < B_{s1}$$

The above inequalities have a positive solution $\alpha_1, \alpha_2$ if

$$K^2T^2 < B_mB_{s1}$$

In principle the damping gains $B_m$, $B_{s1}$ can be arbitrarily assigned and the above inequality has a solution for any constant value of the delay. Thus we conclude that

the signals $\dot{x}_m$, $\dot{x}_s$, $x_m - x_s$ are bounded and it can also be seen that $(\dot{x}_m, \dot{x}_s, \triangle v) \in \mathcal{L}_2$. As the operator and the environmental force is bounded by known functions of the master and the slave velocities respectively, the forces $F_e$ and $F_h$ are bounded. To demonstrate that these signals converge to zero, we need to show that their derivatives are bounded. Using (2) and (4) we get that

$$\dot{x}_{sd} = \frac{B_{s2} - b}{B_{s2} + b}\dot{x}_{sd}(t - 2T) + \frac{2b}{b + B_{s2}}\dot{x}_m(t - T) + \frac{B_{s2}}{b + B_{s2}}(\dot{x}_s(t) - \dot{x}_s(t - 2T)) \tag{7}$$

As $\frac{B_{s2}-b}{B_{s2}+b} \leq 1$, the above equation is a stable difference equation with a bounded input and thus $\dot{x}_{sd}$ is bounded which in turn ensures that $F_s$ is bounded. Hence the slave acceleration is bounded and similarly it can be show that master acceleration is also bounded. Differentiating (7), it can be shown that $\ddot{x}_{sd}$ is also bounded which guarantees asymptotic convergence of the signals $\dot{x}_m$, $\dot{x}_s$, $\triangle v$ to the origin (using Barbalat's Lemma). The tracking error defined in (5) can be rewritten as

$$e = x_m(t) - x_s(t) - \int_{t-T}^{t} \dot{x}_m(\tau)d\tau$$

Thus the position tracking error is bounded.

The above result only guarantees boundedness of the tracking error and not the convergence of the tracking error to the origin. In the next result, we discuss the position tracking abilities of the controller in free space.

**Corollary 1.** *In the steady state, i.e. when*

$$\dot{x}_i, \ddot{x}_i = 0 \quad i = m, s \tag{8}$$

*and $F_e$ equals zero, the tracking error defined by* (5) *goes to zero.*

*Proof.* Under the above conditions, the slave dynamics reduce to

$$F_s + F_{feed} - F_e = 0$$

As $\triangle v$ asymptotically approaches the origin, $F_s \to 0$. Thus it can be observed from the above equation that $F_{feed}$ equals zero and hence the result follows.

*Remark 1.* Under the condition that (8) holds, the proposed architecture ensures that when the slave contacts a remote environment, the contact force is accurately transmitted to the master. As before, $F_s \to 0$ and the slave dynamics reduce to

$$F_{feed} = F_e$$

In steady state

$$\begin{aligned} F_{back} + F_{feed} &= K(x_s(t - T) - x_m(t)) + K(x_m(t - T) - x_s(t)) \\ &= -K\int_{t-T}^{t} \dot{x}_m(\tau) + \dot{x}_s(\tau)d\tau = 0 \end{aligned}$$

or we have $F_{back} = -F_e$ which guarantees good force tracking on the master side.

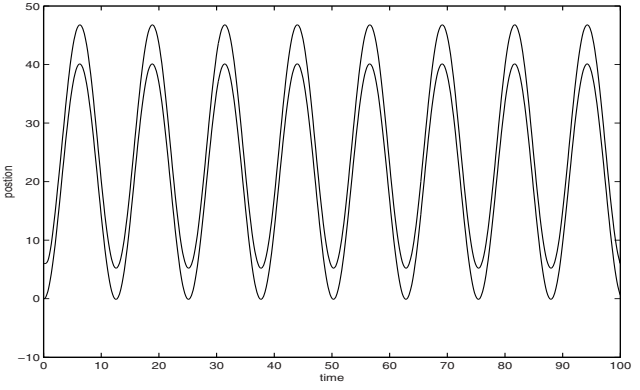



**Fig. 4.** Reference and slave position for system with an initial offset: Traditional architecture cannot ensure good tracking after an initial offset between the master and the slave robot

## 4 Simulations

In this section we verify the efficacy of the proposed architecture. The simulations were carried out on a single-degree of freedom master/slave robots whose dynamics are given by (3). The master was moved sinusoidally and in the scenario, where there is an initial offset between the master and the slave, there is a constant drift between the two robots as shown in Figure 4. However, the proposed architecture ensures that the slave is able to recover from the initial position offset, as shown in Figure 5. We now compare the effect of delays on the transient tracking errors in the traditional and the proposed configuration. In this simulation we assumed that there is no initial position offset between the two robots. The position tracking error (5) in the traditional and the new architecture, with a network delay of 0.8s, is shown in Figure 6. As

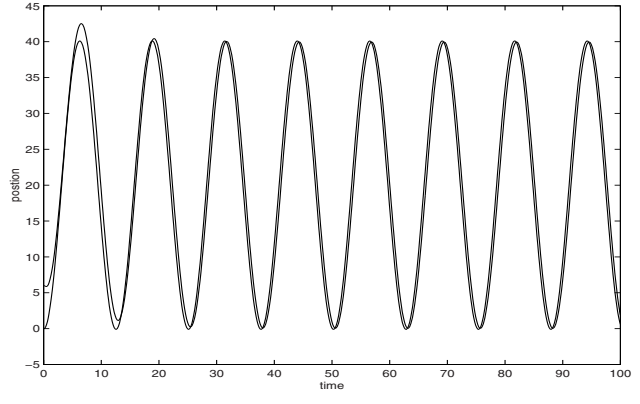



**Fig. 5.** Reference and slave position for system with an initial offset: The new architecture ensures good position tracking

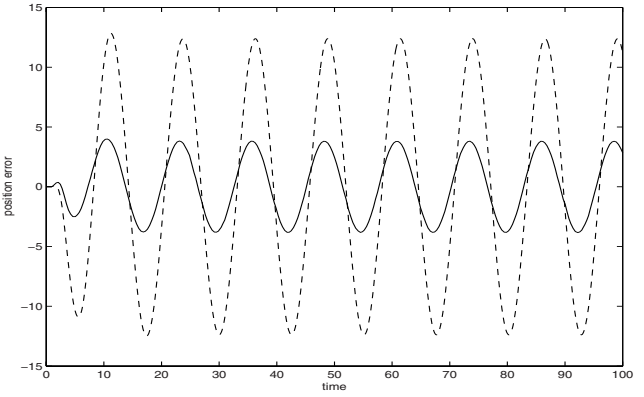



**Fig. 6.** Tracking error with a delay of 0.8s in the traditional architecture(dashed) and the new architecture

evident from the figure, the new architecture reduces the tracking error considerably. In the last set of simulations, we show that new architecture has force tracking abilities which are comparable to those of the traditional bilateral setup. The environment was assumed to be a passive mass-damper system and the master was commanded with a constant velocity. In the new architecture, the slave position faithfully tracks the master until it impacts the environment at t=20s, as seen in Figure 7, and on contact, the environmental force is accurately reflected (using the signals $F_m$ and $F_{back}$) back to the master as seen in Figure 8. The two force ($F_m + F_{back}$, $F_e$) signals, after the impact time of t=20s, literally coincide as seen in Figure 8. The tracking performance is comparable to the tracking performance using the traditional setup as seen in Figure 9. Due to different controllers on the slave side, the interaction of the slave with the environment is different in both cases which leads to different contact force profiles but the point to be taken away from here is that the contact force is faithfully

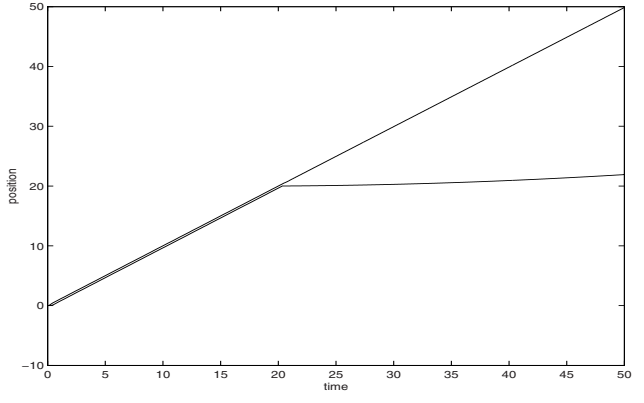



**Fig. 7.** Reference and slave position on contact with the environment

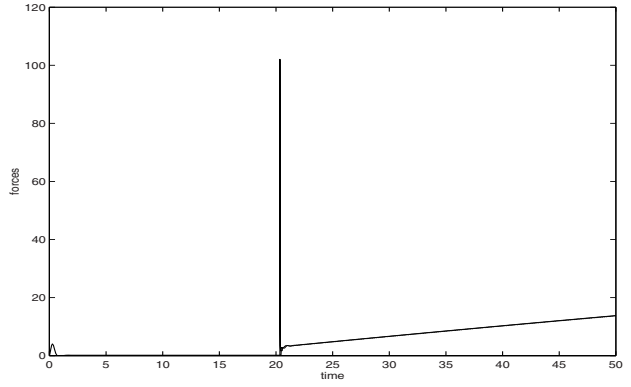



**Fig. 8.** Forces at the master and the slave side on contact with the environment: The new architecture

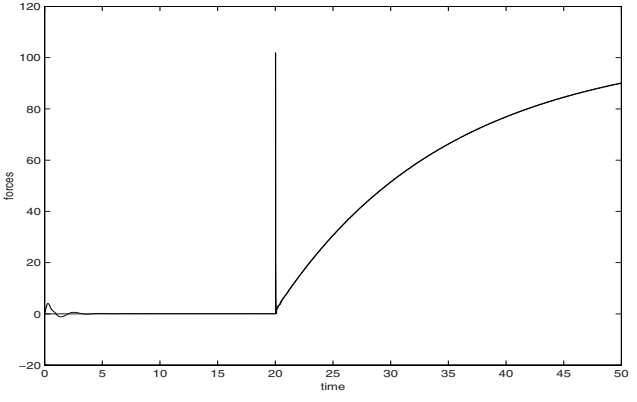



**Fig. 9.** Forces at the master and the slave side on contact with the environment: The traditional architecture

reproduced on the master side with the new configuration. Thus the new architecture has the benefits of better position tracking and retains the force tracking abilities of the bilateral teleoperation setup as proposed in [1].

## 5 Conclusions

In this paper a new architecture for bilateral teleoperation was proposed which has better position tracking and comparable force tracking abilities than the traditional teleoperator model of [1, 7]. The new configuration builds upon the traditional bilateral teleoperator model but has additional proportional controllers on the master and the slave side which use the delayed position data (from both master and slave) as the reference signal. The range of the corresponding proportional gains was established

using Lyapunov analysis has been found to be inversely proportional to the network delay. Future work entails extending these results for time-varying delay networks.

## References

[1] Anderson, R.J., and Spong, M.W., "Bilateral Control of Teleoperators with Time Delay," *IEEE Transactions on Automatic Control*, AC-34, No. 5, pp. 494–501, May, 1989.

[2] Anderson, R.J., and Spong, M.W., "Asymptotic stability for force reflecting teleoperators with time delay," *The International Journal of Robotics Research*, Vol. 11, pp. 135-149, April, 1992.

[3] Chopra, N., Spong M.W., Hirche, S., Buss, M., "Bilateral Teleoperation over the Internet: the Time Varying Delay Problem," *Proceedings American Control Conference*, pp. 155-160, Denver, CO, June 4-6, 2003.

[4] Kim, W.S., "Developments of New Force Reflecting Control Schemes and Application to a Teleoperation Training Simulator," *Proceedings IEEE International Conference on Robotics and Automation*, Nice, France, pp. 1412-1419, May, 1992.

[5] Lawn, C.A., Hannaford, B., " Performance Testing of Passive Communication and Control in Teleoperation with Time Delay," *Proceedings IEEE International Conference on Robotics and Automation*, pp. 776-783, 1993.

[6] Lawrence, D.A., "Stability and Transparency in Bilateral Teleoperation," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 5, pp. 624-637, 1993.

[7] Niemeyer, G. and Slotine, J., "Stable Adaptive Teleoperation," *IEEE Journal of Oceanographic Engineering*, Vol 16, No. 1, pp 152-162, 1991.

[8] Niemeyer, G. and Slotine, J., "Designing Force Reflecting Teleoperators with Large Time Delays to Appear as Virtual Tools," *Proceedings IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, 1997, pp.2212-2218.

[9] Ortega, R., Chopra, N. and Spong, M.W., A New Passivity Formulation for Bilateral Teleoperation with Time Delay, *In Advances in Time-Delay Systems*, Workshop CNRS-NSF, Paris France, January 22-24, 2003.

[10] Salcudean, S.E., Zhu, M., Zhu, W.H., and Hashtrudi-Zaad, K., "Transparent Bilateral Teleoperation under Position and Rate Control", *The International Journal of Robotics Research*, Vol. 19, No. 12, pp. 1185-1202, Dec, 2000.

[11] Valkenburg, M.E., *Network Analysis*, Eaglewood Cliffs, NJ: Prentice Hall, 1964

[12] Yokokohji, Y. and Yoshikawa, T., "Bilateral Control of Master-Slave Manipulators for Ideal Kinesthetic Coupling," *IEEE Transactions on Robotics and Automation*, Vol.10, pp. 605-620, Oct, 1994.

# Suboptimal Control Techniques for Networked Hybrid Systems

Sorin C. Bengea[1], Peter F. Hokayem[2], Ray A. DeCarlo[3], and Chaouki T. Abdallah[4]

[1] Purdue University bengea@purdue.edu
[2] University of Illinois at Urbana - Champaign hal@uiuc.edu
[3] Purdue University decarlo@ecn.purdue.edu
[4] University of New Mexico chaouki@ece.unm.edu

## 1 Introduction

Over the past decade, major advancements in the areas of communications and computer networks [20] have made it possible for control engineers to include them in feedback systems in order to achieve real-time requirements. This gave rise to a new paradigm in control systems analysis and design, namely Networked Control Systems. *Networked Control Systems (NCSs), are control systems whose control loop is closed around a communications network*. In such systems, the feedback is no longer instantaneous and lossless as in classical control systems. Many systems fall under such classification and several examples of NCSs can be found in various areas such as: Automotive industry [3, 4, 6], teleautonomy [27, 28], teleoperation of robots [24, 25, 26, 30], and automated manufacturing systems [29]. Including the network into the design of systems has made it possible to increase mobility, reduce the cost of dedicated cabling, ease upgrading of systems, and render maintenance easier and cheaper. The drawback, however, is that the complexity of analysis and design of such systems increases manyfold (see [21] for an overview).

In this chapter we explore another setting for networked systems, specifically a hybrid system operating in a suboptimal fashion via a network. Our framework is general enough to encompass several applications that have drawn special attention over the past decades some of which, to cite a few, are teleautonomous robots, hybrid electric vehicles (HEV) [1] and distributed fuel cell systems. From a control perspective, the HEV consists of the interconnection of a variety of subsystems among which are the internal combustion engine, a battery, an electric motor/generator, a transmission, and a drive train. Each subsystem has a local controller whose operation is coordinated with the other subsystems and is actuated by a supervisory controller through a communication network. This network, like all communication networks, is subject to delays, interference, disturbances, lost data packets and so forth. In such situations, the supervisory controller must try to maintain adequate performance while simultaneously optimizing fuel economy and minimizing emissions. The problem is a hybrid (optimal) control problem in which the future must

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 281–301, 2004.

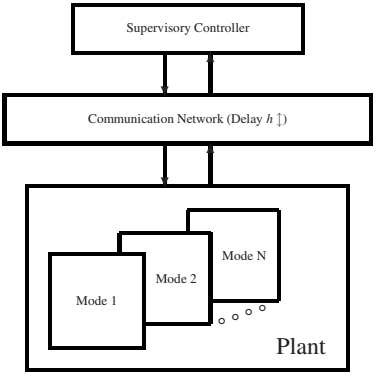



**Fig. 1.** Supervisory control of a multi-mode plant over a communication network with delays.

be estimated even with global positioning availability: an "optimal" local (decentralized) and supervisory control strategy must be determined and implemented in real time over a communication network.

In a quite different context there is the projected use of distributed fuel cells (DFC) for residential and industrial use to complement delivery of electrical energy from the power grid so as to reduce world dependence on fossil-fuel based electrical power generation. Distributed fuel cells can also help guarantee minimal power levels in the presence of power grid failures, black outs, etc. DFC's have local controllers to maintain the proper operation of the fuel cell (FC). A supervisory controller coordinates power flows to and from the power grid for each FC through a communication network.

Finally, Teleautonomy of robots refers to a new paradigm where a group of robots, rather than individual ones are tele-controlled by an operator. As such the operator may remotely control a master unit, which then will act as a supervisory controller to the individual robots in the group. While teleoperation poses its own challenges [24, 25, 26], we are here concerned with the supervisory controller actions over a communication network.

In all three cases, a set of plants interact and must be individually controlled (decentralized) and coordinated by a supervisory controller through a communication link to achieve some optimal performance. Assuming no disturbances, in the simplest case, sensor measurements to the supervisor are inherently delayed, and the control signals issued by the supervisor undergo a similar delay. Figure 1 illustrates the basic idea.

The problem then is to robustly solve a hybrid optimal control problem in the presence of delayed measurements and delayed control inputs. The approach here is to combine recent advances in hybrid optimal control with the emerging work on model predictive control (MPC) to achieve a viable solution to this very important networked control system (NCS) problem.

## 2 Problem Statement and Proposed Approach

We consider a plant that operates in one of a set of possible structures or modes at any one time. This necessitates two control inputs: a continuous-valued input and a discrete-valued switching input that selects the proper structure at each instant of time. In the presence of state and input constraints the objective is to minimize a cost functional by appropriately selecting the structure and generating its controls.

In the absence of delays associated with the communication network, the problem can be solved by embedding the switched system into a larger family of systems and then solving a more general hybrid optimal control problem [8]. Implementation of a solution requires instantaneous and continuous availability of the state measurements, which is not possible due to the delays inherent in a real communication network. We assume that the supervisory controller receives periodic updates of the plant state. These periodic updates of the delayed state initialize a possible uncertain prediction of the state trajectory for use in a model predictive control algorithm that also accounts for the communication delay to the plant.

For the rest of this chapter we consider, for notational tractability, a switching plant model that consists of two possible modes of operation. However, the theoretical results can be scaled to encompass $n$-modes. In our subsequent analysis, we will use the notation pertaining to model predictive control: (i) for any control signal $u$, $u(t_1|t_2)$ is the value generated at (based on the information up to) time $t_2$ and that is predicted to be applied to the system at time $t_1 (\geq t_2)$; (ii) for any state signal $x$, $x(t_1|t_2)$ is the value generated at time $t_2$ and is estimated for time $t_1$ (in this case $t_1$ and $t_2$ can be in any ordering relationship).

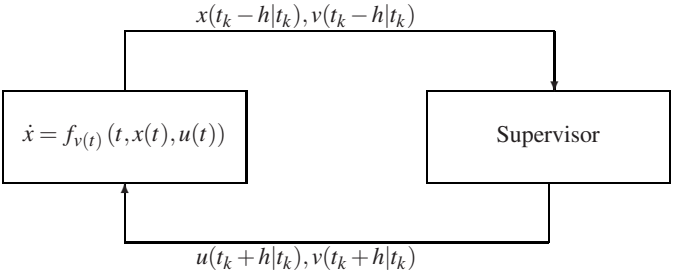



**Fig. 2.**

### 2.1 Problem Formulation

Within the afore mentioned framework, we assume that the communication network imposes a constant delay $h$ on the communicated signals as shown in Figure 2. Moreover, for minimizing the communication bandwidth, we assume that the supervisory

controller receives the plant state, over the total optimization problem horizon $[t_0, t_f]$, only at times $t_k \in \{t_0+h, t_0+2h, ..., t_f-2h\}$; therefore, the supervisory controller receives an exact state measurement, $x(t_k-h)$. Using $x(t_k-h)$ and the known control inputs $u(t)$ and $v(t)$, on $[t_k, t_k+h]$, the supervisor predicts the state trajectory (denoted as $\hat{x}$) forward in time and simultaneously solves an optimization problem to generate the optimal input $u^\star$ and switching signal $v^\star$ over the interval $[t_k+h, t_k+h+N_f]$, where $N_f$ is a suitable multiple of $h$ and represents a variable horizon. The interval of optimization begins with $(t_k+h)$ to account for the communication delay from the supervisor to the plant. Thus the control inputs for $t < t_k+h$ are known, having been previously computed. In accordance with model predictive control strategies [2, 5], only the "optimal" inputs $u^\star$ and $v^\star$ over the interval $[t_k+h, t_k+2h]$ are sent. The optimal control problem is then resolved at $(t_k+h)$ for the interval $[t_k+2h, t_k+2h+N_f']$ where $N_f'$ may be different from $N_f$. And the process repeats.

Of course, the time interval elapsed between successive plant state transmissions may be different than the communication delay. We assume however that both are equal in order to avoid excessive notation. Without loss of generality, we also assume that the computational time required to solve the optimization problems at each step is equal to zero, otherwise, the communication delay can be increased to include this time interval. Furthermore, the proposed method can be appropriately modified for handling situations where the communication delays on the receiving and transmitting channels are unequal.

The two-switched system model adopted in this chapter has system state $x(t) \in \mathbb{R}^n$ at time $t$ with dynamics

$$\dot{x}(t) = f_{v(t)}(t, x(t), u(t)), \qquad x(t_0) = x_0 \tag{1}$$

where at each $t \geq t_0$, $v(t) \in \{0,1\}$ is the switching control, $u(t) \in \Omega \subset \mathbb{R}^m$ is the control input constrained to the convex and compact set $\Omega$, and $f_0$ and $f_1$ are real vector-valued functions, $f_0, f_1 : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, of class $\mathcal{C}^1$. The control inputs, $v(t)$ and $u(t)$, are both measurable functions.

The optimization functional over the interval $[t_0, t_f]$ is:

$$J(x,u,v) = \int_{t_0}^{t_f} f^0_{v(t)}(t, x(t), u(t))dt \tag{2}$$

where $f_0^0, f_1^0 : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ are of class $\mathcal{C}^1$. The objective is:

$$\min_{v,u} J(x,u,v)$$

subject to the constraints

(a) $x(\cdot)$ satisfies the (uncertain) model equation given by (1), with a given $x_0$;
(b) for each $t \in [t_0, t_f]$, $v(t) \in \{0,1\}$ and $u(t) \in \Omega$,

in the presence communication delays, and exact knowledge of the plant state $x$ only at discrete instants, as described above. Relaxation of the above exact knowledge assumption is addressed later.

A similar use of an uncertain model to predict the continuous state of the plant within a networked control systems context has appeared in [22, 23], however the target there was to prove stability for the overall system in a deterministic [22] and stochastic [23] sense.

### 2.2 Proposed Suboptimal Solution

Our goal is to offer, in a deterministic context, a suboptimal solution to this problem. Admittedly, the optimization problem is not rigorously stated as one has limited information on the model uncertainties. Therefore, for a given control input, the supervisor cannot accurately predict the plant state trajectory. Assuming certain structure on the model uncertainties, a rigorous formulation is possible in the context of stochastic optimal control or with norm bounded possibly structured uncertainties. Remaining in a deterministic context, we make use of model predictive control techniques, whose advantages in dealing with uncertainties have been underlined in the literature (for example [2]). However, due to the delays in transmitting the control signal to the plant and in receiving the state measurement from the plant, direct application of MPC results is not possible. The adapted strategy is described in the following.

Since the plant is subject to parameter uncertainties, structural perturbations, and possible unknown disturbances, the supervisory controller uses a nominal/estimation model of (1) described by

$$\begin{aligned}\dot{\hat{x}}(t) &= \hat{f}_{v(t)}(t,\hat{x}(t),u(t)) \\ &= [1-v(t)]\cdot\hat{f}_0(t,\hat{x}(t),u(t)) + v(t)\cdot\hat{f}_1(t,\hat{x}(t),u(t))\end{aligned} \tag{3}$$

$v(t)\in\{0,1\}$. At each time instant $t_k\in\{t_0+h,t_0+2h,...,t_f-2h\}$:

(i) The supervisory controller receives the exact plant state $x(t_k-h)$, at $t_k$, and updates its state estimate accordingly: $\hat{x}(t_k-h|t_k)=x(t_k-h)$;

(ii) Using the update $\hat{x}(t_k-h|t_k)$, and the control inputs $u([t_k-h,t_k]|t_k-2h), v([t_k-h,t_k]|t_k-2h)$ – generated at time $t_k-2h$ for the interval $[t_k-h,t_k]$ – and $u([t_k,t_k+h]|t_k-h), v([t_k,t_k+h]|t_k-h)$ – generated at time $t_k-h$ for the interval $[t_k,t_k+h]$, the supervisory controller estimates the plant state up to the time instant $t_k+h$. Having the initial state estimate at $t_k+h$, the supervisor solves the **Switched Optimal Control Problem** (SOCP), over the horizon $N_f(>h)$:

$$\min_{v,u} J_S(u,v) \stackrel{\triangle}{=} \min_{v,u} \int_{t_k+h}^{t_k+h+N_f} f^0_{v(\tau|t_k)}(\tau,\hat{x}(\tau|t_k),u(\tau|t_k))d\tau \tag{4}$$

subject to the constraints

(a) $\dot{\hat{x}}(\tau|t_k)=\hat{f}_{v(\tau|t_k)}(\tau,\hat{x}(\tau|t_k),u(\tau|t_k))$, for $\tau\in[t_k+h,t_k+h+N_f]$

(b) for each $\tau\in[t_k+h,t_k+h+N_f]$, $v(\tau|t_k)\in\{0,1\}$ and $u(\tau|t_k)\in\Omega$.

(iii) Solving the optimization problem in (ii) produces the control signals $u(t|t_k)$ and $v(t|t_k)$ on the interval $[t_k+h, t_k+h+N_f]$. Due to the delay $h$, the controls are applied to the plant on the desired interval $[t_k+h, t_k+2h]$.

Although the controls generated at step (ii) are optimal for the SOCP associated with the nominal/estimation model, due to plant uncertainties, these controls are suboptimal for the control problem associated with the plant. For related switching optimal control problem formulations and approaches see Branicky *et al.* [10], Giua *et al.* [12], Riedinger *et al.* [15], [16], Sussman [18], Xu and Antsaklis [19].

At this point, we have the plant model (1) with state $x$, and a nominal/estimation model (3) with state $\hat{x}$ used by the supervisor to solve the optimization problem of step (ii). Details of the solution are given in Section 3.

## 3 Piecewise Estimated Optimal Solution

In this section, we solve the subproblem of step (ii) in subsection 2.2. This is a standard switched optimal control problem. Our approach is to embed the switched (estimation model) system (3) into a larger family of systems and to reformulate the problem for the latter. The viability of the approach is given by a certain relationship between the set of trajectories of the switched and embedded systems. In subsection 3.2, necessary conditions for optimality are formulated.

### 3.1 Reformulation of the SOCP

The difficulty in solving the SOCP stems from the presence of both continuous-time dyanmics ($\hat{f}_0$ and $\hat{f}_1$) and discrete events (the switching instants). To overcome this difficulty we embed the switching system (3) into a larger family of systems as follows: Let $v(t)$ now take values in the interval $[0,1]$, i.e. $v(t) \in [0,1]$, and let $u_i(t)$ define the control input corresponding to vector field $\hat{f}_i$, $i = 0,1$; then define the family of systems parametrized by $v(t) \in [0,1]$ as

$$\dot{\hat{x}}(t) = [1-v(t)] \cdot \hat{f}_0(t, \hat{x}(t), u_0(t)) + v(t) \cdot \hat{f}_1(t, \hat{x}(t), u_1(t)) \tag{5}$$

with associated cost functional

$$J_E(v, u_0, u_1) = \int_{t_k+h}^{t_k+h+N_f} \left\{ [1-v(t)] \cdot f_0^0(t, \hat{x}(t), u_0(t)) + v(t) \cdot f_1^0(t, \hat{x}(t), u_1(t)) \right\} \tag{6}$$

The optimal control problem of interest now becomes:

**Embedded optimal control problem**(EOCP): Minimize the functional (6) over all functions $v$, $u_0$, and $u_1$, subject to the following constraints:

(i) $\hat{x}(\cdot)$ satisfies equation (5), with initial condition $\hat{x}(t_k+h|t_k)$;
(ii) for each $t \in [t_k, t_k+h+N_f]$, $v(t) \in [0,1]$ and $u_0(t), u_1(t) \in \Omega$ (convex and compact).

The EOCP is a classical optimal control problem for which we can study sufficient and necessary conditions for optimality via classical tools [9]. The viability of our approach stems from the fact that the set of trajectories of the system (3) is dense in the set of trajectories of system (5). Each trajectory, $\hat{x}_E$, of (5) can be approximated within any desired accuracy, $\varepsilon$, by a trajectory, $\hat{x}_S$, of the (estimation model) switched system (3) corresponding to a proper choice of the switching input $v_\varepsilon(\cdot)$, with values in the set $\{0,1\}$, and of the control input $u_\varepsilon(\cdot)$.

**Theorem 1.** *Let $u_0(t) \in \Omega$, $u_1(t) \in \Omega$, and $v(t) \in [0,1]$ be a control triplet for the embedded system (5). Let both systems, (3) and (5), have initial point $\hat{x}(t_k+h|t_k)$. Let $\hat{x}_E(\cdot)$ be a trajectory for the system (5) corresponding to $u_0(\cdot), u_1(\cdot), v(\cdot)$ on $[t_k+h, t_k+h+N_f]$. Then, for each $\varepsilon > 0$, there are controls $v_\varepsilon(t) \in \{0,1\}$ and $u_\varepsilon(\cdot)$ defined on $[t_k+h, t_k+h+N_f]$, with the following properties: For almost all $t \in [t_k+h, t_k+h+N_f]$, $u_\varepsilon(t) \in \Omega$, and the trajectory $\hat{x}_{S,\varepsilon}(\cdot)$ (of the switched system (3) corresponding to the controls $v_\varepsilon(\cdot)$ and $u_\varepsilon(\cdot)$) satisfies: for all $t \in [t_k+h, t_k+h+N_f]$, $|\hat{x}_{S,\varepsilon}(t) - \hat{x}_E(t)| < \varepsilon$.*

■

For a proof and a construction of the controls $v_\varepsilon$ and $u_\varepsilon$ see [7, 8]. The crucial role in the justification of this result is played by the Chattering Lemma as in [9, 11]. Since the integrands $f_0^0$ and $f_1^0$ of the cost functional $J_E$ are of class $\mathcal{C}^1$, the approximating trajectory $\hat{x}_{S,\varepsilon}$ (of the switched system (3)) can be constructed such that $|J_S(u_\varepsilon, v_\varepsilon) - J_E(u_0, u_1, v)| < \varepsilon$. Therefore, if $\hat{x}_E$ is an optimal trajectory for the EOCP, then the trajectories $\hat{x}_{S,\varepsilon}$ are suboptimal solutions for the SOCP.

### 3.2 Necessary Conditions for Optimality

In the general context of the EOCP, we are interested in sufficient and necessary conditions for optimality. It can be shown (see [7] and [8]) that both the assumptions made in Section 2.1 and the satisfaction of conditions S1–S4, specified below, guarantee that the EOCP has a solution.

(S1) $\hat{f}_0(t,\hat{x},z_0) = A_0(t,\hat{x}) + B_0(t,\hat{x}) \cdot z_0$;
(S2) $\hat{f}_1(t,\hat{x},z_1) = A_1(t,\hat{x}) + B_1(t,\hat{x}) \cdot z_1$;
(S3) for each $(t,\hat{x})$, the function $f_0^0(t,\hat{x},z_0)$ is a convex function of $z_0$;
(S4) for each $(t,\hat{x})$, the function $f_1^0(t,\hat{x},z_1)$ is a convex function of $z_1$.

For the remainder of this section, we presume that EOCP has a solution, $(\hat{x}_E^\star(t), u_0^\star(t), u_1^\star(t), v^\star(t))$. With the system given in (5) and the cost functional given by (6) we associate the Hamiltonian function
$H : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m \times [0,1] \times \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ defined as

$$\begin{aligned} H(t,\hat{x},z_0,z_1,\mu,p^0,p) &= p^0[(1-\mu)f_0^0(t,\hat{x},z_0) + \mu \cdot f_1^0(t,\hat{x},z_1)] \\ &\quad + p^T[(1-\mu)\hat{f}_0(t,\hat{x},z_0) + \mu\hat{f}_1(t,\hat{x},z_1)] \\ &\stackrel{\triangle}{=} \mu E_1(t,\hat{x},z_0,z_1,p^0,p) + E_2(t,\hat{x},z_0,z_1,p^0,p) \end{aligned} \tag{7}$$

with the obvious definition of the functions $E_1$ and $E_2$.

The assumptions made in Section 2.1 make it possible to apply the Maximum Principle [9]. Hence, there exist a constant $\lambda^0 \leq 0$ and an absolutely continuous function $\lambda$ defined on $[t_k+h, t_k+h+N_f]$ such that:

(i) the vector $(\lambda^0, \lambda(t)) \neq 0$ on $[t_k+h, t_k+h+N_f]$;
(ii) for almost all $t \in [t_k+h, t_k+h+N_f]$,

$$\begin{aligned} \dot{\hat{x}}_E^\star(t) &= H_p(t, \hat{x}_E^\star(t), u_0^\star(t), u_1^\star(t), v^\star(t), \lambda^0, \lambda(t)) \\ \dot{\lambda}(t) &= -H_x(t, \hat{x}_E^\star(t), u_0^\star(t), u_1^\star(t), v^\star(t), \lambda^0, \lambda(t)) \end{aligned} \tag{8}$$

where $H_p$ and $H_x$ denote appropriate partial derivatives;
(iii) for almost all $t \in [t_k+h, t_k+h+N_f]$,

$$H(t, \hat{x}_E^\star(t), u_0^\star(t), u_1^\star(t), v^\star(t), \lambda^0, \lambda(t)) \geq H(t, \hat{x}_E^\star(t), z_0, z_1, \mu, \lambda^0, \lambda(t)) \tag{9}$$

for all $z_0, z_1 \in \Omega$ and all $\mu \in [0,1]$.
It can be shown ([8]) that for almost all $t \in [t_k+h, t_k+h+N_f]$,

$$\begin{aligned} &H(t, \hat{x}_E^\star(t), u_0^\star(t), u_1^\star(t), v^\star(t), \lambda^0, \lambda(t)) \\ &\quad = \max \left\{ \max_{(z_0,z_1)\in\Omega\times\Omega} H(t, \hat{x}_E^\star(t), z_0, z_1, 0, \lambda^0, \lambda(t)), \right. \\ &\qquad\qquad \left. \max_{(z_0,z_1)\in\Omega\times\Omega} H(t, \hat{x}_E^\star(t), z_0, z_1, 1, \lambda^0, \lambda(t)) \right\} \end{aligned} \tag{10}$$

Let $T_k \stackrel{\triangle}{=} \{t \in [t_k+h, t_k+h+N_f] | E_1(t, \hat{x}_E^\star(t), u_0^\star(t), u_1^\star(t), \lambda^0, \lambda(t)) = 0\}$. It can be shown ([8]) that for almost all $t \in [t_k+h, t_k+h+N_f] \setminus T_k$, $v^\star(t) \in \{0,1\}$.

**Theorem 2.** *For almost all $t \in [t_k+h, t_k+h+N_f] \setminus T_k$,*

*(i) (MODE 0) If*

$$\max_{z_0\in\Omega} H(t, \hat{x}_E^\star(t), z_0, z_1, 0, \lambda^0, \lambda(t)) > \max_{z_1\in\Omega} H(t, \hat{x}_E^\star(t), z_0, z_1, 1, \lambda^0, \lambda(t)), \tag{11}$$

*then $v^\star(t) = 0$ and equations (8) become:*

$$\begin{aligned} \dot{\hat{x}}_E^\star(t) &= \hat{f}_0(t, \hat{x}_E^\star(t), u_0^\star(t)) \\ \dot{\lambda}(t) &= -p^0 \left[ \underline{\frac{\partial f_0^0}{\partial \hat{x}}} \right] \Bigg|_{(t, \hat{x}_E^\star(t), u_0^\star(t))} - \left[ \underline{\frac{\partial \hat{f}_0}{\partial \hat{x}}} \right]^T \Bigg|_{(t, \hat{x}_E^\star(t), u_0^\star(t))} \lambda(t) \end{aligned} \tag{12}$$

*The optimal control $u_1^\star(t)$ is indeterminate in $\Omega$, and*

$$u_0^\star(t) = \arg\max_{z_0\in\Omega} [\lambda^0 f_0^0(t, \hat{x}_E^\star(t), z_0) + \lambda^T(t) \hat{f}_0(t, \hat{x}_E^\star(t), z_0)]$$

*(ii) (MODE 1) If*

$$\max_{z_1\in\Omega} H(t, \hat{x}_E^\star(t), z_0, z_1, 1, \lambda^0, \lambda(t)) > \max_{z_0\in\Omega} H(t, \hat{x}_E^\star(t), z_0, z_1, 0, \lambda^0, \lambda(t)), \tag{13}$$

*then* $v^\star(t) = 1$ *and equations (8) become:*

$$\begin{aligned}\dot{\hat{x}}_E^\star(t) &= \hat{f}_1(t,\hat{x}_E^\star(t),u_1^\star(t))\\ \dot{\lambda}(t) &= -p^0\left[\frac{\partial f_1^0}{\partial \hat{x}}\right]\Big|_{(t,\hat{x}_E^\star(t),u_1^\star(t))} - \left[\frac{\partial \hat{f}_1}{\partial \hat{x}}\right]^T\Big|_{(t,\hat{x}_E^\star(t),u_1^\star(t))}\lambda(t)\end{aligned} \tag{14}$$

*The optimal control* $u_0^\star(t)$ *is indeterminate in* $\Omega$*, and*

$$u_1^\star(t) = \arg\max_{z_1\in\Omega}[\lambda^0 f_1^0(t,\hat{x}_E^\star(t),z_1) + \lambda^T(t)\hat{f}_1(t,\hat{x}_E^\star(t),z_1)]$$

*(iii) If*

$$\max_{z_1\in\Omega} H(t,\hat{x}_E^\star(t),z_0,z_1,1,\lambda^0,\lambda(t)) = \max_{z_0\in\Omega} H(t,\hat{x}_E^\star(t),z_0,z_1,0,\lambda^0,\lambda(t)), \tag{15}$$

*then either* $v^\star(t) = 0$ *or* $v^\star(t) = 1$*, and the above corresponding equations hold.*

■

**Corollary 1.** *If the set* $T_k$ *has measure zero for an optimal trajectory of the EOCP, then this trajectory is also optimal for the SOCP and the optimal control input for the SOCP is* $u^\star(t) = [1-v^\star(t)]u_0^\star(t) + v^\star(t)u_1^\star(t)$*, for almost all* $t \in [t_k + h, t_k + h + N_f]$*.*

■

**Theorem 3.** *For almost all* $t \in T_k$*,*

$$\max_{z_0\in\Omega} H(t,\hat{x}_E^\star(t),z_0,z_1,0,\lambda^0,\lambda(t)) = \max_{z_1\in\Omega} H(t,\hat{x}_E^\star(t),z_0,z_1,1,\lambda^0,\lambda(t)) \tag{16}$$

*and*

$$\begin{aligned}u_0(t) &= \arg\max_{z_0\in\Omega}[\lambda^0 f_0^0(t,\hat{x}_E^\star(t),z_0) + \lambda^T(t)\hat{f}_0(t,\hat{x}_E^\star(t),z_0)],\\ u_1(t) &= \arg\max_{z_1\in\Omega}[\lambda^0 f_1^0(t,\hat{x}_E^\star(t),z_1) + \lambda^T(t)\hat{f}_1(t,\hat{x}_E^\star(t),z_1)].\end{aligned} \tag{17}$$

■

The above theorems are proved in [8].

From (16), it follows that in the singular cases the optimal switching control $v^\star(t)$ (with $t \in T_k$), may take values in the interval $(0,1)$. However, from the constraints on the system – input constraints, state and costate equations (8) – it still may be possible to take $v^\star(t) \in \{0,1\}$ for almost all $t \in T_k$, and once again obtain a solution to the SOCP. Recall that, in the singular cases, if no solutions of the EOCP are of the bang-bang type, then the SOCP may or may not have a solution. In such cases, suboptimal solutions of the SOCP may be constructed as in [8].

## 4 Error Analysis

In this section, we present some bounds on the system performance when a discrepancy exists between the actual plant model and the estimation model used by the supervisor. To obtain these bounds we utilize a slight generalization of Gronwall's Inequality. For easier referencing, we first state Gronwall's Inequality as in [9], page 139.

**Lemma 1.** *Let* $\rho$ *and* $\beta$ *be non-negative real valued functions continuous on* $[0,\infty]$ *such that*

$$\rho(t) \leq \alpha + \int_{t_0}^{t} \beta(s)\rho(s)ds \qquad \alpha \geq 0 \tag{18}$$

*for all* $t_0$, $t$ *in* $[0,\infty)$. *Then*

$$\rho(t) \leq \alpha \cdot e^{\int_{t_0}^{t} \beta(s)ds}. \tag{19}$$

■

The following lemma is a slight generalization of Gronwall's Inequality and is useful in our development on the error bounds.

**Lemma 2.** *Let* $\rho$, $\alpha$, *and* $\beta$ *be non-negative real valued functions continuous on* $[0,\infty]$, *such that*

$$\rho(t) \leq \alpha(t) + \int_{t_0}^{t} \beta(s)\rho(s)ds \qquad \alpha \geq 0 \tag{20}$$

*for all* $t_0$, $t$ *in* $[0,\infty)$. *Let* $\alpha$, *be monotonically increasing. Then*

$$\rho(t) \leq \alpha(t) \cdot e^{\int_{t_0}^{t} \beta(s)ds}. \tag{21}$$

**Proof:** For each $t_0 \leq t \leq t_1 = \sup\{s > t_0 : \alpha(s) = 0\}$, $\rho(t) = 0$, by direct application of the Gronwall's Inequality with $\alpha(t) = 0$. For each $t > t_1$, upon division by $\alpha(t)$ (which is nonzero for $t > t_1$) we have

$$\frac{\rho(t)}{\alpha(t)} \leq 1 + \int_{t_1}^{t} \beta(s)\frac{\rho(s)}{\alpha(t)}ds \tag{22}$$

Since $\alpha$ is monotonically increasing,

$$\frac{\rho(t)}{\alpha(t)} \leq 1 + \int_{t_1}^{t} \beta(s)\frac{\rho(s)}{\alpha(s)}ds \tag{23}$$

We can now directly apply the Gronwall's Inequality and we have that

$$\frac{\rho(t)}{\alpha(t)} \leq e^{\int_{t_1}^{t} \beta(s)ds} \tag{24}$$

Since $\rho(t) = 0$ for $t_0 \leq t \leq t_1$, the lemma is proved.

■

Consider the plant given in (1) and the estimation model (3). We assume that the difference between the two models is upper bounded for a given state $x$ (see similar assumption in [14] Chapter 9), that is for each $t \in [t_0, t_f]$

$$\left\| f_0(t,x,u) - \hat{f}_0(t,x,u) \right\|_2 \leq g(t)b(x), \tag{25}$$

and similarly for $f_1, \hat{f}_1$, with $g$ integrable, and $b$ positive and Lipschitz

$$|b(x_1) - b(x_2)| \leq M_b \|x_1 - x_2\|_2. \tag{26}$$

Note that the input for both dynamics is exactly the same since the supervisor knows the exact input that it generated and transmitted to the plant. We make the assumption that for each control input $u(t) \in \Omega$ the solution of equation (3) is bounded. Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^n$ such that $x(t) \in \mathcal{X}$ for all $t \in [t_0, t_f]$. Without loss of generality, we assume that the set $\mathcal{X}$ is convex (otherwise we can consider it as a subset of a convex (compact) set). Since, for each $t \in [t_0, t_f]$ the function $\hat{f}_0(t,\cdot,\cdot)$ is $\mathcal{C}^1$ on the compact and convex set $\mathcal{X} \times \Omega$, it follows that the norm of its Jacobian is bounded, and, upon application of Theorem 9.19 in [17], it follows that there is a function $m(t)$ such that the Lipschitz condition

$$\left\| \hat{f}_0(t,x_1,u_1) - \hat{f}_0(t,x_2,u_2) \right\|_2 \leq m(t) \left\| \begin{bmatrix} x_1 \\ u_1 \end{bmatrix} - \begin{bmatrix} x_2 \\ u_2 \end{bmatrix} \right\|_2 \tag{27}$$

holds for all $x_1, x_2 \in \mathcal{X}$, and all $u_1, u_2 \in \Omega$. We make the assumption that the function $m(\cdot)$ is integrable. Without loss of generality, we also assume that the function $\hat{f}_1$ satisfies equation (27).

There are two scenarios that arise in such a setting. The perfect scenario where $f_i(.,.,.) \equiv \hat{f}_i(.,.,.)$ for $i = 0, 1$, which results in a perfect reconstruction of the state vector and hence both functions will have the same value in a time period of $2h$ after the system is turned on. The more challenging scenario is when the system equations do not match exactly, which results in a discrepancy between the state of the plant and that of the estimation model. In this case, we need to obtain a bound on the estimation error. Since the supervisory controller receives updates on the plant states every $h$ seconds, it is sufficient to study the estimation error on an interval $[t, t+h]$ assuming that the supervisory controller receives $x(t-h)$ at $t$; that is, $\hat{x}(t-h|t) = x(t-h)$. Utilizing the control inputs on the interval $[t-h, t]$, $u([t-h,t]|t-2h)$ and $v([t-h,t]|t-2h)$, and those on the interval $[t, t+h]$, $u([t,t+h]|t-h)$ and $v([t,t+h]|t-h)$, we have that for each $\tau \in [t, t+h]$

$$\|x(\tau)-\hat{x}(\tau|t)\|_2 \leq$$
$$\leq \left\| \left[ \int_{t-h}^{t} f_{v(s|t-2h)}(s,x(s),u(s|t-2h))ds + \int_{t}^{\tau} f_{v(s|t-h)}(s,x(s),u(s|t-h))ds \right] \right.$$
$$\left. - \left[ \int_{t-h}^{t} \hat{f}_{v(s|t-2h)}(s,\hat{x}(s|t),u(s|t-2h))ds + \int_{t}^{\tau} \hat{f}_{v(s|t-h)}(s,\hat{x}(s|t),u(s|t-h))ds \right] \right\|_2$$

$$= \left\| \int_{t-h}^{t} \left[ f_{v(s|t-2h)}(s,x(s),u(s|t-2h)) - \hat{f}_{v(s|t-2h)}(s,x(s),u(s|t-2h)) \right] ds \right.$$
$$+ \int_{t}^{\tau} \left[ f_{v(s|t-h)}(s,x(s),u(s|t-h)) - \hat{f}_{v(s|t-h)}(s,x(s),u(s|t-h)) \right] ds$$
$$+ \int_{t-h}^{t} \left[ \hat{f}_{v(s|t-2h)}(s,x(s),u(s|t-2h)) - \hat{f}_{v(s|t-2h)}(s,\hat{x}(s|t),u(s|t-2h)) \right] ds$$
$$\left. + \int_{t}^{\tau} \left[ \hat{f}_{v(s|t-h)}(s,x(s),u(s|t-h)) - \hat{f}_{v(s|t-h)}(s,\hat{x}(s|t),u(s|t-h)) \right] ds \right\|_2$$

Notice that we have added and subtracted integrals of the estimation model with the exact (plant) state so that we can apply the bounds of equation (25). Utilizing the triangle inequality we further have

$$\|x(\tau)-\hat{x}(\tau|t)\|_2 \leq \tag{28}$$
$$\leq \int_{t-h}^{t} \left\| f_{v(s|t-2h)}(s,x(s),u(s|t-2h)) - \hat{f}_{v(s|t-2h)}(s,x(s),u(s|t-2h)) \right\|_2 ds$$
$$+ \int_{t}^{\tau} \left\| f_{v(s|t-h)}(s,x(s),u(s|t-h)) - \hat{f}_{v(s|t-h)}(s,x(s),u(s|t-h)) \right\|_2 ds$$
$$+ \int_{t-h}^{t} \left\| \hat{f}_{v(s|t-2h)}(s,x(s),u(s|t-2h)) - \hat{f}_{v(s|t-2h)}(s,\hat{x}(s|t),u(s|t-2h)) \right\|_2 ds$$
$$+ \int_{t}^{\tau} \left\| \hat{f}_{v(s|t-h)}(s,x(s),u(s|t-h)) - \hat{f}_{v(s|t-h)}(s,\hat{x}(s|t),u(s|t-h)) \right\|_2 ds$$

Combining the bounds given by (25) and (27) in (28), we obtain that, for each $\tau \in [t,t+h]$,

$$\|x(\tau)-\hat{x}(\tau|t)\|_2 \leq \int_{t-h}^{\tau} g(s)b(x(s))ds + \int_{t-h}^{\tau} m(s)\|x(s)-\hat{x}(s|t)\|_2 ds \tag{29}$$

Note that in order to obtain (29), we have used equation (27) with $u_1 = u_2$.

Upon utilization of the triangle inequality and of relationship (26), and recombination of the integral terms, we have

$$\|x(\tau)-\hat{x}(\tau|t)\|_2 \leq \int_{t-h}^{\tau} g(s)b(\hat{x}(s|t))ds + \int_{t-h}^{\tau} [m(s)+g(s)M_b]\|x(s)-\hat{x}(s|t)\|_2 ds$$

Observe that the above inequality holds in fact for every $\tau \in [t-h, t+h]$ (although our interest only is in what happens on the interval $[t, t+h]$). The function $\int_{t-h}^{\tau} g(s)b(\hat{x}(s|t))ds$ is continuous, non-negative and monotonically increasing (because the functions $g$ and $b$ are non-negative). Using Lemma 2, it follows that

$$\|x(\tau)-\hat{x}(\tau|t)\|_2 \leq \left(\int_{t-h}^{\tau} g(s)b(\hat{x}(s|t))ds\right) \cdot \left(e^{\int_{t-h}^{\tau}[m(s)+g(s)M_b]ds}\right) \tag{30}$$

for all $\tau \in [t, t+h]$. Notice that in (30), if $g = 0$ then the error is zero: expectedly, since in this case the estimator is an exact replica of the plant model (see 25). If the delay $h$ equals zero, the error for any $\tau$ between successive samples of the state of the plant is bounded by $\left(\int_{t}^{\tau} g(s)b(\hat{x}(s|t))ds\right) \cdot \left(e^{\int_{t}^{\tau}[m(s)+g(s)M_b]ds}\right)$.

## 5 Example

In this section we apply the MPC/EOCP method developed in this chapter to a version of Example 1 [13], which is a crude two-dimensional model of a car with two gears, controlled through a communication network. The state $x$ is the velocity of the car (according to some reference coordinate system), with some gear efficiencies $g_0(\zeta)$ and $g_1(\zeta)$, as plotted in Figure 3. The (plant) vehicle dynamics corresponding to each mode are

$$\begin{aligned} \dot{x}(t) &= f_0(t,x,u) = g_0(x)\cdot u + r(t)\cdot x^2 \qquad \text{and} \\ \dot{x}(t) &= f_1(t,x,u) = g_1(x)\cdot u + r(t)\cdot x^2 \end{aligned} \tag{31}$$

where $r(t)$ is uncertain with values in the interval $[-0.06, +0.03]$, which roughly models the road imperfections, drag forces, etc., and the input-throttle command, $u$, is constrained to the set $\Omega = [-1, 1]$. Initially, the car is at rest, i.e. $x(0) = 0$, and is being controlled through a communication network with delay $h = 0.1$. The goal is to track with minimum control energy the following velocity profile: $x^{des}(t) = 0.5t$ on $[0,2]$, $x^{des}(t) = 1$ on $[2,4]$, and $x^{des}(t) = 0.5(4-t)+1$ on $[4,6]$. Accordingly, the performance measure is chosen to be

$$J_S(v,u) = \int_0^6 \left\{10\cdot[x(t)-x^{des}(t)]^2 + u^2(t)\right\} dt. \tag{32}$$

On the other side of the network, the supervisor controlling the car has only access to the actual state value periodically and uses for the optimization problem the following dynamics of the estimate model:

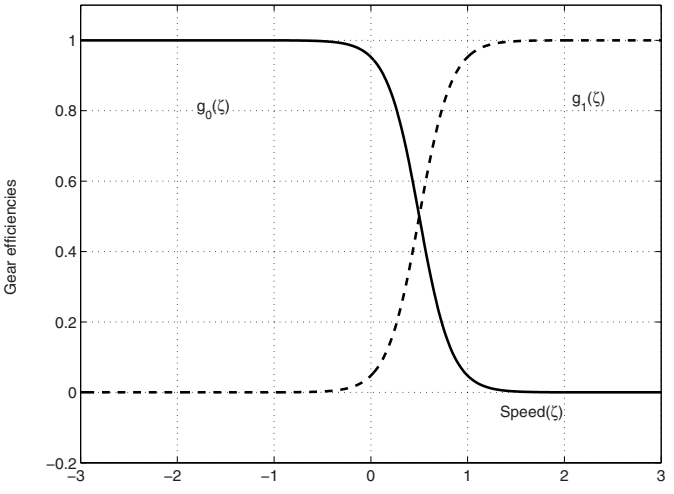



**Fig. 3.** Gear efficiencies

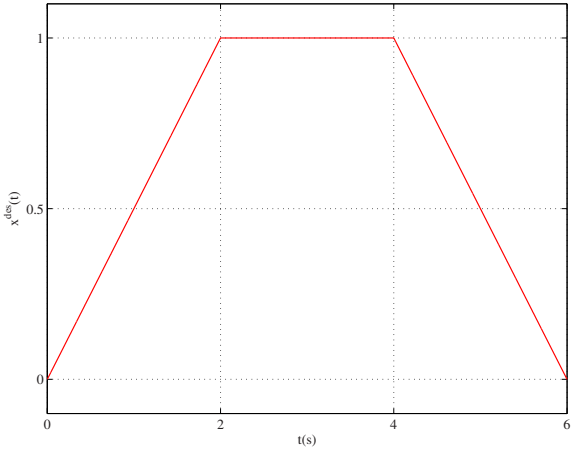



**Fig. 4.** Desired velocity profile

$$\begin{aligned} \dot{\hat{x}}(t) &= \hat{f}_0(t,\hat{x},u) = g_0(\hat{x})\cdot u \qquad \text{and} \\ \dot{\hat{x}}(t) &= \hat{f}_1(t,\hat{x},u) = g_1(\hat{x})\cdot u \end{aligned} \tag{33}$$

and differ from the actual vehicle dynamics (31) by the absence of the term $r(t)x^2$. Applying the embedding technique discussed earlier to the estimation model results with the following state equation:

$$\dot{\hat{x}}(t) = [1-v(t)]\cdot g_0(\hat{x}(t))\cdot u_0(t) + g_1(\hat{x}(t))\cdot u_1(t) \tag{34}$$

and the controls $u_0(t)$ and $u_1(t)$ are constrained to $[-1,1]$. The cost functional associated with the embedded system is

$$J_E(v,u) = \int_0^6 \left\{ 10 \cdot [\hat{x}(t) - x^{des}(t)]^2 + [1 - v(t)]u_0^2(t) + v(t)u_1^2(t) \right\} dt. \tag{35}$$

For illustrating the suboptimal control technique, we consider that the supervisory controller solves the model predictive control problem over the following intervals (of type $[t_k + h, t_k + h + N_f]$, as mentioned in Section 2.1): $[0.1, 2.1]$, $[2.1, 4.1]$, and $[4.1, 6]$. In this setting, for simplifying the illustration of the technique, the supervisory controller does not uses all the received plant state information, as would happen in a real situation.

In the following, on each of the afore mentioned subintervals, an optimization is solved as indicated in step (ii) of Section 2.2, using the embedding technique of Section 3. On each of these subintervals, $[t_k + h, t_k + h + N_f]$, all the conditions S1–S4 mentioned in Section 3.2 hold and an optimal solution exists; denote it by $(\hat{x}^\star(t), u_0^\star(t), u_1^\star(t), v^\star(t))$. The optimal state estimate satisfies

$$\dot{\hat{x}}^\star(t) = [1 - v^\star(t)]g_0(\hat{x}^\star(t))u_0^\star(t) + v^\star(t)g_1(\hat{x}^\star(t))u_1^\star(t) \tag{36}$$

with costate equation

$$\begin{aligned} \dot{\lambda}(t) = & -20\lambda^0[\hat{x}^\star(t) - x^{des}(t)] \\ & -\lambda(t)[1 - v^\star(t)]\tfrac{dg_0(\hat{x})}{d\hat{x}}\big|_{\hat{x}^\star(t)} + v^\star(t)\tfrac{dg_1(\hat{x})}{d\hat{x}}\big|_{\hat{x}^\star(t)} \end{aligned} \tag{37}$$

The Hamiltonian associated with system (34) is $H : \mathbb{R}^7 \to \mathbb{R}$

$$\begin{aligned} H(t,\hat{x},z_0,z_1,\mu,p^0,p) &= \mu\left\{p^0[z_1^2 - z_0^2] + p[g_1(\hat{x})z_1 - g_0(\hat{x})z_0]\right\} \\ &\quad + p^0\left\{[(\hat{x} - x^{des}(t)]^2 + z_0^2\right\} + pg_0(\hat{x})z_0 \\ &\stackrel{\triangle}{=} \mu E_1(t,\hat{x},z_0,z_1,p^0,p) + E_2(t,\hat{x},z_0,z_1,p^0,p) \end{aligned} \tag{38}$$

Except on the last subinterval, the final state, $\hat{x}(t_k + h + N_f)$ is free on each subinterval $[t_k + h, t_k + h + N_f]$; upon using the transversality conditions as in [9], page 190, it follows that $\lambda(t_f + h + N_f) = 0$, which in turns implies, from the Maximum Principle, that $\lambda^0 = -1$. On each subinterval, the solution $(\hat{x}^\star(t), u_0^\star(t), u_1^\star(t), v^\star(t))$ of the EOCP is not guaranteed, in general, to be a solution of the SOCP.

**Proposition 1.** *In the context of this example, there is an optimal solution of the EOCP with* $v^\star(t) \in \{0,1\}$ *and hence a solution of the SOCP.*

*Proof*: The following arguments hold for any subinterval $[t_k + h, t_k + h + N_f]$. Motivated by Theorems 2 and 3, we study the possibility of the expression $E_1$ being identically zero on an interval $[t', t''] \subset [t_k + h, t_k + h + N_f]$. If such were the case, Theorem 3 would imply that for each $t \in [t', t'']$

$$-u_1^\star(t) + \lambda(t)g_1(\hat{x}^\star(t))u_1^\star = -u_0^\star(t) + \lambda(t)g_0(\hat{x}^\star(t))u_0^\star \tag{39}$$

and

$$u_0^\star(t) = \begin{cases} \frac{\lambda(t)g_0(\hat{x}^\star(t))}{2} & , \text{ if } -1 < \frac{\lambda(t)g_0(\hat{x}^\star(t))}{2} < 1 \\ -1 & , \text{ if } \frac{\lambda(t)g_0(\hat{x}^\star(t))}{2} \leq -1 \\ 1 & , \text{ if } 1 \leq \frac{\lambda(t)g_0(\hat{x}^\star(t))}{2} \end{cases} \tag{40}$$

$$u_1^\star(t) = \begin{cases} \frac{\lambda(t)g_1(\hat{x}^\star(t))}{2} & , \text{ if } -1 < \frac{\lambda(t)g_1(\hat{x}^\star(t))}{2} < 1 \\ -1 & , \text{ if } \frac{\lambda(t)g_1(\hat{x}^\star(t))}{2} \leq -1 \\ 1 & , \text{ if } 1 \leq \frac{\lambda(t)g_1(\hat{x}^\star(t))}{2} \end{cases} \tag{41}$$

Using equations (36), (37), (40), and (41), it can be shown that the equality (39) can hold only if for all $t \in [t', t'']$ $\lambda(t) = 0$, $\hat{x}^\star(t) = x^{des}(t)$, $u_0^\star(t) = u_1^\star(t) = 0$, and $v^\star(t)$ can be chosen arbitrarily in the interval $[0,1]$. Therefore, the proposition is proved.

■

Upon applying Theorem 2, it follows that the mode of operation is given by the sign of the expression $E_1(t, \hat{x}^\star(t), u_0^\star(t), u_1(t), \lambda^0 = -1, \lambda(t))$. Utilizing the expression of $E_1$ from (38), and Theorem 2, it follows that the optimal solution and controls satisfy, for almost all $t \in [t_k + h, t_k + h + N_f]$, the following:

(i) (MODE 0) If

$$\max_{z_0 \in [-1,1]} \left[ -z_0^2 + \lambda(t) g_0(\hat{x}^\star(t)) z_0 \right] > \max_{z_1 \in [-1,1]} \left[ -z_1^2 + \lambda(t) g_1(\hat{x}^\star(t)) z_1 \right] \tag{42}$$

then $v^\star(t) = 0$, equation (40) holds, and

$$\begin{aligned} \dot{\hat{x}}^\star(t) &= g_0(\hat{x}^\star(t)) u_0^\star(t) \\ \dot{\lambda}(t) &= 20[\hat{x}^\star(t) - x^{des}(t)] - \lambda(t) \left. \frac{dg_0(\hat{x})}{d\hat{x}} \right|_{\hat{x}^\star(t)} u_0^\star(t) \end{aligned} \tag{43}$$

(ii) (MODE 1) If

$$\max_{z_0 \in [-1,1]} \left[ -z_0^2 + \lambda(t) g_0(\hat{x}^\star(t)) z_0 \right] < \max_{z_1 \in [-1,1]} \left[ -z_1^2 + \lambda(t) g_1(\hat{x}^\star(t)) z_1 \right] \tag{44}$$

then $v^\star(t) = 1$, equation (41) holds, and:

$$\begin{aligned} \dot{\hat{x}}^\star(t) &= g_1(\hat{x}^\star(t)) u_1^\star(t)) \\ \dot{\lambda}(t) &= 20[\hat{x}^\star(t) - x^{des}(t)] - \lambda(t) \left. \frac{dg_1(\hat{x})}{d\hat{x}} \right|_{\hat{x}^\star(t)} u_1^\star(t) \end{aligned} \tag{45}$$

A particular aspect of this example is that the optimal controls, $u_0^\star$ and $u_1^\star$, can be explicitly computed, using (40) and (41), as functions of the co-state, $\lambda(t)$. Therefore, if the initial value of lambda, $\lambda(t_k + h)$, and the initial state, $\hat{x}^\star(t_k + h)$, are known, the optimal controls and estimated optimal trajectories can be determined using the above equations for the two modes of operation. But, due to model uncertainties, it is not possible to exactly know the state of the system and therefore it is not possible to know beforehand the values $\lambda(t_k + h)$. However, for this example, the on-line computational burden can be reduced as described below.

The following computations are performed off-line.

(a) Since the delay $h = 0.1$, on the interval $[0, 0.1]$ the control is zero; hence the velocity at $t = 0.1$ is zero.

(b) Using $\hat{x}^\star(0.1) = 0$, the costate, $\lambda(0.1)$ can be determined numerically by comparing the costs associated with the trajectories and controls for values of $\lambda$ in an appropriately chosen interval. This is achieved by generating, using the equations (40), (41), (42), and (44), the trajectories and controls over the interval $[0.1, 2.1]$ for each $\lambda$ in an appropriately chosen set (via simulations in this case).

(c) Using $\lambda(0.1)$, the corresponding estimated optimal trajectory, $\hat{x}^\star$, on $[0.1, 2.1]$, and the bounds on the uncertainties, an interval is determined for the state $x(2.1)$ of system (31) as follows.
Firstly, observe that the equation (25) can be written as

$$|f_0(t,x,u) - \hat{f}_0(t,x,u)| \leq |r(t)| \cdot x^2$$

and therefore the functions $g$ and $b$ can be taken to be $g(t) = |r(t)|$ and $b(x) = x^2$. It is reasonable to assume (and the simulations confirm) that $x(t) \leq 1.5$ on the interval $[0, 6]$. Hence, we can take $M_b = 3 (= 2 \cdot 1.5)$.
Secondly, the function $m$ of equation (27) (with $u_1 = u_2$, see the remark on page 13) can be obtained as follows

$$|\hat{f}_0(t,x_1,u) - \hat{f}_0(t,x_2,u)| \leq |g_0(x_1) - g_1(x_2)||u|.$$

Since, the control is bounded, $|u| \leq 1$, and the gear efficiencies are bounded as well, we can take $m(t) = 1$.
Thirdly, using $\lambda(0.1)$ computed in part (b), one can determine (off-line) $\hat{x}^\star(t)$ on $[0.1, 2.1]$, and therefore compute $\int_{0.1}^{2} [\hat{x}^\star(t)]^2 dt \approx 0.5$.
Utilizing the aforementioned functions, $g$, $b$ and $m$, and bounds in (30) it follows that

$$|x(2) - \hat{x}(2|0)| \leq \left( 0.06 \int_{0.1}^{2} [\hat{x}^\star(s)]^2 ds \right) \cdot e^{\left( \int_{0.1}^{2} (1 + 0.06 \cdot 3) ds \right)} \approx 0.28$$

Therefore, $x(2) \in [0.76 - 0.28, 0.76 + 0.28] = [0.48, 1.04]$. For a sufficiently large discrete set of values of $x$ in this interval, the optimal value of $\lambda$ is generated as in described in (b), but now the problem is considered on the interval $[2.1, 4.1]$. Therefore, we numerically obtain a correspondence between the possible values of $x(2.1)$ and the corresponding $\lambda(2.1)$ which generate the optimal controls on the interval $[2.1, 4.1]$.

(d) This part is similar to part (c), but it is performed for the interval $[4.1, 6]$. Another difference is that the correspondence between $x$ and $\lambda$, at $t = 4.1$, is determined such that $x(6) = 0$. The equations (40), (41), (42), and (44) are used for this interval, as well.

The on-line procedure for the suboptimal control of this networked system is described in the following. On the interval $[0, 2.1]$, the optimal controls are computed

off-line, since the initial state of (31) is known exactly. At $t = 2$, the supervisory controller receives the system state $x(1.9)$. Therefore, as explained at step (ii) in Section 2.2, the state of the system (31) can be estimated. At $t = 2$, using the estimated state $\hat{x}(2.1)$, the value of $\lambda(2.1)$ is determined based on the off-line correspondence calculated as aforementioned at step (b). Using this value of $\lambda(2.1)$ the optimal controls on the interval $[2.1, 4.1]$ are generated using the relationships (40) and (41). These computations require a small amount of time and, hence, are suitable for the on-line control scheme. At $t = 4$, the supervisory controller receives the state $x(3.9)$ (of the system (31)), and the procedure for generating the optimal controls on $[4.1, 6.1]$ follows the same steps as for the the interval $[2.1, 4.1]$.

The estimated trajectory (of the system (33)) and the trajectory of the system (31) are plotted in Figure 5. The discontinuities present at $t = 2$ and $t = 4$ in the estimated velocity are due to the updates based on the information received at these instances. The controls are illustrated in Figure 6. As expected, the gear switchings occur when the *estimated* velocity equals 0.5, where the two gear efficiencies are equal. Observe that the car velocity at $t = 6$ is slightly smaller than zero. This is due to the disturbances (uniformly distributed in $[-0.06, 0.03]$). For meeting the final constraint, the control $u(t)$ can be set to zero as soon as the velocity equals zero. Finally, for this problem the overall estimated cost is 2.9702 and the overall cost associated with the trajectory of system (31) is 3.0790. Clearly, if we had solved the optimization problem after each reception of the updated state, we would have obtained a lower cost and a better approximation to $x^{des}(t)$.

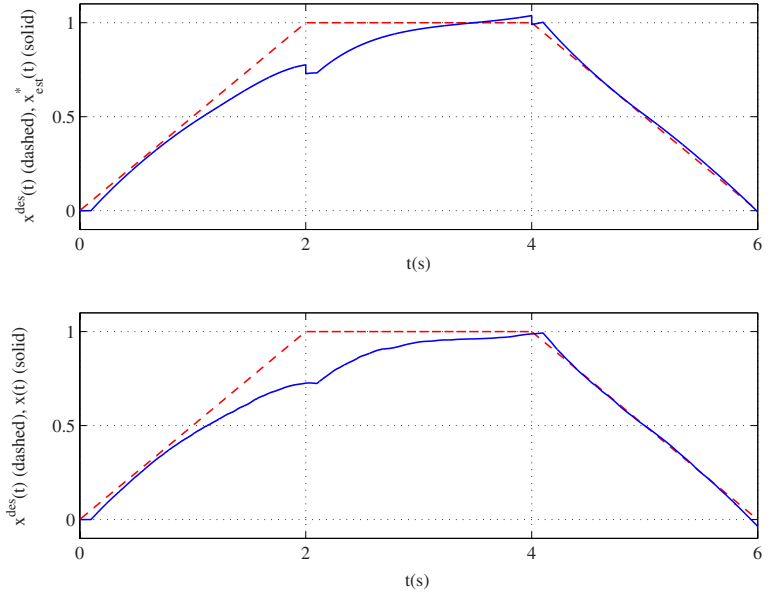



**Fig. 5.** Desired (dashed), Estimated (upper figure, solid), and Vehicle (lower figure, solid) Trajectories

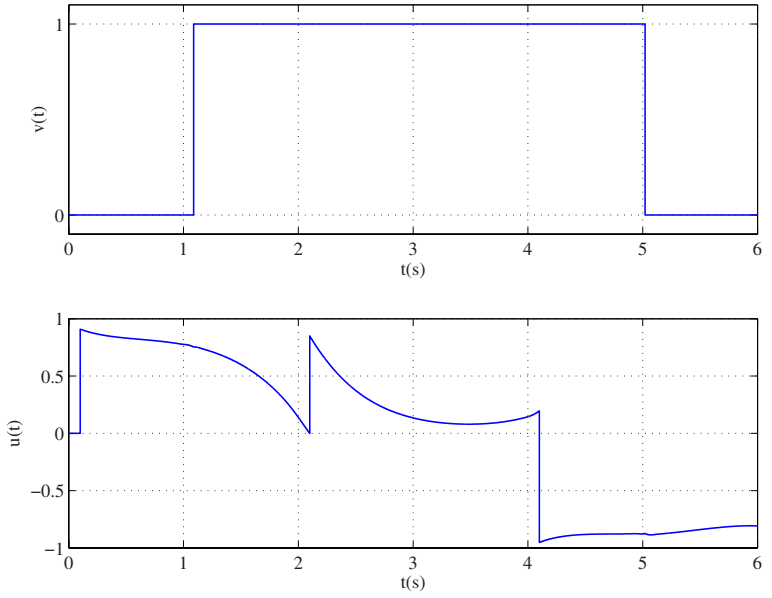



**Fig. 6.** Suboptimal Controls

## 6 Conclusions

In this chapter we solve the hybrid optimal switching control problem with delays introduced by a communication network that resides between the multi-mode plant and the supervisor. Due to the presence of delays, plant uncertainties, and disturbances, only a suboptimal solution can be obtained via utilizing a model-predictive approach and solving the optimal switching problem over subintervals. We illustrated our approach through a gear-switching example with input constraints. For future work we propose expanding our approach to include information losses between the plant and the supervisor, and studying the robustness of our method in that context.

## References

[1] Brahma, A., Guezennec, Y., Rizzoni, G., "Optimal Energy Management in Series Hybrid Electric Vehicles," in Proceedings of the American Control Conference, Chicago, Illinois, June 2000, pp. 60-64.

[2] Camacho, E.F. and C. Bordons, "Model Predictive Control," Springer-Verlag, London, 1998.

[3] Jurgen, R.K., "Coming from Detroit: networks on wheels", *IEEE Spectrum*, vol. 23, no. 6, 1986, pp. 53-59.

[4] Leen, G., and Heffernan, D., "Expanding Automotive Electronic Systems," *IEEE Computer*, vol. 35, no. 1, January 2002, pp. 88-93.
[5] Maciejowski, J.M., "Predictive Control with Constraints," Prentice Hall, 2002.
[6] Özgüner, Ü., Göktaş, H., Chan, H., Winkelman, J., Liubakka, M., Krtolica, R., "Automotive Suspension Control Through a Computer Communication Network," *IEEE Conference on Control Applications*, vol. 2, 1992, pp. 895-900.
[7] Bengea, S. C., DeCarlo, R. A., "Optimal and Suboptimal Control of Switching Systems," In *Proceedings of the IEEE Conference on Control and Decision*, Maui, Hawaii, Dec. 9-12, 2003, pp. 5295-5300.
[8] Bengea, S. C., DeCarlo, R. A., "Optimal Control of Switching Systems," Provisionally accepted for publication in *Automatica*, Dec. 2003.
[9] Berkovitz, L. D., *Optimal Control Theory*, Springer-Verlag, New York, 1974.
[10] Branicky, M.S., Borkar, V.S., Mitter, S.K., "A Unified Framework for Hybrid Control: Model and Optimal Control Theory," in *IEEE Transactions on Automatic Control*, Vol. 43, No. 1, January 1998, pp. 31-45.
[11] Gamkrelidze, R. V., *Principles of optimal control theory*, Plenum Press, New York, 1978.
[12] Giua, A., Seatzu, C., Van Der Mee, C., "Optimal control of switched autonomous linear systems," in *Proceedings of the* $40^{th}$ *IEEE Conference on Decision and Control*, Orlando, Florida, USA, December 2001, pp. 2472-2477.
[13] Hedlund, S., Rantzer, A., "Optimal Control for Hybrid Systems," in *Proceedings of the* $38^{th}$ *IEEE Conference on Decision and Control*, Phoenix, Arizona, USA, December 1999, pp. 3972-3977.
[14] Khalil, H.K., *Nonlinear Systems,* $3^{rd}$ *Ed.*, Prentice Hall, NJ, 2002.
[15] Riedinger, P., Zanne, C., Kratz, F., "Time Optimal Control of Hybrid Systems," in *Proceedings of the American Control Conference*, San Diego, California, June 1999, pp. 2466-2470.
[16] Riedinger, P., Kratz, F., Iung, C., Zanne, C., "Linear Quadratic Optimization for Hybrid Systems," in *Proceedings of the* $38^{th}$ *Conference on Decision and Control*, Pheonix, Arizona, USA, December 1999, pp. 3059-3064.
[17] Rudin, W., "Principles of Mathematical Analysis," McGraw-Hill, New York, 1976.
[18] Sussmann, H., "A maximum principle for hybrid optimal control problems, " in *Proceedings of the* $38^{th}$ *IEEE Conference on Decision and Control*, Pheonix, Arizona, USA, December 1999, pp. 425-430.
[19] Xu, X., Antsaklis, P.J., "Optimal Control of Switched Systems: New Results and Open Problems," in *Proceedings of the American Control Conference*, Chicago, Illinois, USA, June 2000, pp. 2683-2687.
[20] Raji, R.S., "Smart networks for control," *IEEE Spectrum*, June 1994, pp. 49-55.
[21] Hokayem, P.F. and Abdallah, C.T, "Inherent Issues in Networked Control Systems: A Survey," Accepted to the 2004 American Control Conference, Boston, MA, June 30-July 2, 2004.
[22] Montestruque, L.A. and Antsaklis, P.J., "Model-Based Networked Control System - Stability," ISIS Technical Report ISIS-2002-001, Notre Dame, IN, January 2001.

[23] Montestruque, L.A. and Antsaklis, P.J., "Stochastic Stability for Model-Based Networked Control Systems," *Proceedings of the American Control Conference,* Denver, CO, June 2003, pp. 4119-4124.

[24] Niemeyer, G. and Slotine, J.-J. E., "Towards Force-Reflecting Teleoperation Over the Internet," *Proceedings of the 1998 International Conference on Robotics & Automation*, Leuven, Belgium, May 1998, pp. 1909-1915.

[25] Niculescu, S.-I., Abdallah, C.T. and Hokayem, P.F., "Effects of Channel Dynamics on the Stability of Teleoperation," IFAC Workshop on Time-Delay Systems INRIA, Rocquencourt, France, September 8-10, 2003.

[26] Anderson, R.J. and Spong, M.W., "Bilateral Control of Teleoperators with Time Delay," *IEEE Transactions on Automatic Control*, AC-34, no.5, May 1989, pp. 494-501.

[27] Oda, M., Doi, T., and Wakata, K.,"Tele-Manipulation of a satellite mounted robot by an on-ground astronaut", *Proceedings of the IEEE International Conference on Robotics & Automation*, Seoul, Korea, May 2001, pp. 1891-1896.

[28] Trivedi, M., Hall, B., Kogut, G., and Roche, S., "Web-based Teleautonomy and Telepresence," $45^{th}$ *SPIE Optical Science and Technology Conference*, Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation III, vol. 4120, San Diego, August 2000.

[29] Lian, F.L., Moyne, J.R., and Tilbury, D.M., "Implementation of Networked Machine Tools in Reconfigurable Manufacturing Systems," *Proceedings of the 2000 Japan-USA Symposium on Flexible Automation*, Ann Arbor, MI, July 2000.

[30] Göktas, F., mith, J.M., and Bajcsy, R., "Telerobotics Over Communication Networks," *Proceedings of the IEEE Conference on Decision and Control*, San Diego, Calif. USA, vol. 3, Dec 1997, pp. 2399-2404.

# Communication Requirements for Networked Control

Sekhar Tatikonda[1] and Nicola Elia[2]

[1] Yale University sekhar.tatikonda@yale.edu
[2] Iowa State University nelia@iastate.edu

## 1 Introduction

There is an increasing interest in studying control systems employing multiple sensors and actuators that are geographically distributed. Communication is an important component of these distributed and networked control systems. Hence there is a need to understand the interactions between the control components and the communication components of the distributed system. A recent report on future research directions in control listed understanding control over communication networks as a major challenge for the controls field [22]. Here we present a general framework for examining the communication requirements for networked control. In particular we are interested in the relationship between control performance and communication rate.

This framework allows one to determine upper and lower bounds on the channel capacity needed to achieve given control objectives like observability, stabilizability, and performance. We discuss specific channels, e.g. bit rate channels, Gaussian channels, erasure channels, where these bounds are tight. In doing so we characterize the "information complexity" of different control objectives.

A complete understanding of the interaction between control and communication will need to use tools from both control theory and information theory. Traditional information theory, though, does not take into account delays and often the coding system is non-causal. These are undesirable properties in control situations. Herein we present a sequential notion rate distortion theory to deal with delays and causality.

Our task involves designing an encoder, decoder, and controller to achieve some control objective. This entails specifying what the encoder, decoder, and controller, know and when they know it. This specification is called the information pattern [36]. We discuss the important role different information patterns can have on the control design. In particular we examine two information patterns in detail. One where the encoder has access to the control signals being applied to the plant and one where the encoder is required to be time-invariant.

This chapter is divided into two parts. Section 2 treats noiseless channels and section 3 treats noisy channels. The former is easier to treat because the encoder

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 303–326, 2004.

knows what information is being received by the decoder. The latter, though, is more realistic in practice.

*Noiseless Channels*

We first compute a lower bound on the rate required to achieve the different control objectives. This lower bound is independent of the information pattern in place and depends only on the plant. Specifically we show that a necessary condition on the rate for asymptotic observability and asymptotic stabilizability in a linear, discrete time, system is $R > \sum_{\lambda(A)} \max\{0, \log|\lambda(A)|\}$ where the sum is over the eigenvalues of the $A$ matrix (see equation (1).) This result relates the speed of the dynamics of the plant to the information rate of the channel. In the case where the encoder observes the control signals, the co-location case, we show that for any rate larger than this lower bound there exists an encoder, decoder, and controller that achieves the control objective. Hence this bound is also sufficient. We then discuss time-invariant encoders. These involve a different set of bounds. We extend these results to time-sampled continuous time systems and more general network settings.

Our problem formulation was inspired by the work of Borkar and Mitter [6]. There they formulated an LGQ control problem under a rate-constrained channel. They discussed the interactions between coding, delay, and performance. The papers of Wong and Brockett were also influential [37, 38]. They provided conditions connecting the channel rate to the dynamics of the system to insure stabilizability of the system. Tatikonda and Mitter [29] generalized these results. Elia and Mitter examined the stabilizability problem in the case where the encoder is time-invariant [12]. They use a Lyapunov based synthesis scheme to design the underlying quantizers. Liberzon and Brockett have also examined a Lyapunov based design with time-varying encoders [9]. Nair and Evans give rate conditions for stabilizability of an ARMA model [23, 24]. See also Baillieul [2].

*Noisy Channels*

For noisy channels we first provide a general necessary condition on the channel to insure observability and stabilizability. We then show that these conditions are sufficient for Internet-like channels that suffer erasures. Then we examine performance issues for the LQG problem over a Gaussian channel. To this end we examine two classical notions of separation. The first notion is the control-theoretic separation between state estimation and control. We present conditions that insure the optimality of the certainty equivalent control law. The second notion is the information-theoretic separation between source encoder and channel encoder. In particular, in the limit of long delays, it is known that one can, without loss of generality, design the source encoder and the channel encoder separately [16]. This separation is known to hold quite broadly [35] but in general fails for both short delays and for unstable processes. Since delay is an important issue in control applications we cannot apply the information-theoretic separation results to our problem. To deal with this delay issue we present the sequential rate distortion framework [31].

Bansal and Basar examine the simultaneous design of the measurement and control strategies for a class of stochastic systems [3]. Borkar and Mitter introduced the present problem of LQG control under communication constraints [6]. Nair and Evans [23] examined mean-square stabilizability over a noiseless channel. The paper [7] examines the role of nonclassical information patterns for Markov decision problems. The design of optimal sequential quantization schemes for uncontrolled Markov processes was examined in [8].

We conclude this introduction with some comments on the content of the paper. First, in general we provide intuition and only sketch the proofs. References to complete proofs are given. Second, the field of control with communication constraints is rapidly growing. We have not made any attempt to be exhaustive in our references. Third, the content that follows represents the work of the authors. The interested reader is encouraged to explore the referenced papers.

## 2 Control Over a Noiseless Channels

In this section we discuss results for control over noiseless communication channels. In section 3 we discuss noisy channels.

Consider the following linear, discrete-time, time-invariant system:

$$X_0 \in \Lambda_0, \quad X_{t+1} = AX_t + BU_t, \quad Y_t = CX_t, \quad \forall t \geq 0 \tag{1}$$

where $\{X_t\}$ is a $\mathbb{R}^d$-valued state process, $\{U_t\}$ is a $\mathbb{R}^m$-valued control process, and $\{Y_t\}$ is a $\mathbb{R}^l$-valued observation process. We have $A \in \mathbb{R}^{d\times d}$, $B \in \mathbb{R}^{d\times m}$, and $C \in \mathbb{R}^{l\times d}$. The initial position $X_0 \in \Lambda_0$ where $\Lambda_0 \subseteq \mathbb{R}^d$ is assumed to be an open set. If $C = I$, where $I$ is the identity matrix, then we have full state observation at the encoder.

We envision a communication channel connecting the plant observation at the encoder to the decoder at the controller. This channel is modelled as a noiseless digital channel that can transmit at each time step one of $2^R$ symbols denoted $\sigma \in \Sigma$, $|\Sigma| = 2^R$. Specifically at each time step the channel can transmit without error $R$ bits of information. Throughout this chapter log refers to logarithm base 2.

The control problems we look at involve the design of an encoder, decoder, and controller. We need to specify the *information pattern* [36] of each component. Let $X^t \doteq (X_0, ..., X_t)$. The encoder at time $t$ is a map $\mathcal{E}_t : \mathbb{R}^{l(t+1)} \times \Sigma^t \times \mathbb{R}^{mt} \to \Sigma$ taking $(Y^t, \sigma^{t-1}, U^{t-1}) \mapsto \sigma_t$. In the next section we will discuss different restrictions on the available inputs to the encoder. The decoder at time $t$ is a map $\mathcal{D}_t : \Sigma^{t+1} \times \mathbb{R}^{mt} \to \mathbb{R}^d$ taking $(\sigma^t, U^{t-1}) \mapsto \hat{X}_t$. The output of the decoder is an estimate of the state of the plant. We will discuss how this state estimate is computed in the next section. The controller at time $t$ is a map $\mathcal{C}_t : \mathbb{R}^d \to \mathbb{R}^m$ taking $\hat{X}_t \mapsto U_t$. Note that we are assuming that the controller takes as input only the decoder's state estimate. Hence we are assuming a separation structure between the decoder and the controller. We will show that for the specific information pattern considered in section 2.2 there is no loss of generality in making this separation assumption.

### 2.1 Necessary Conditions

Here we examine necessary conditions on the channel rate to insure observability and stabilizability. Note that the usual algebraic conditions for observability, e.g. certain Grammians having full rank, are still necessary but no longer sufficient. The additional necessary conditions take the form of lower bounds on the channel rate. These lower bounds will be universal in the sense that they hold independently of the actual encoder, decoder, and controller used. They hold independently of the information pattern in place.

The purpose of any good observer is to distinguish points in the state space. In a time horizon of $T$ we have at most $2^{TR}$ possible symbols arriving at the decoder. Thus at time $T$ we must be able to approximate the state by one of at most $2^{TR}$ points. Let the *error* be $e_t = X_t - \hat{X}_t$ where $\hat{X}_t$ is the state estimate.

We say system (1) is *asymptotically observable* if there exists an encoder and decoder such that the following holds for any control sequence $\{U_t\}$: $\forall\, \varepsilon > 0$, $\forall\, \delta > 0$, $\exists\, T(\varepsilon,\delta)$ such that $\|X_0\|_2 \le \delta$ implies $\|e_t\|_2 \le \varepsilon$, $\forall t \ge T(\varepsilon,\delta)$. Let $\sum_{\lambda(A)}$ represent the sum over the eigenvalues of $A$. For a given set $\Omega \subseteq \mathbb{R}^d$ define the $\mathrm{diam}(\Omega) \doteq \sup_{x,y\in\Omega} \|x-y\|_2$.

**Proposition 1.** *A necessary condition for system (1) to be asymptotically observable is* $R > \sum_{\lambda(A)} \max\{0, \log|\lambda(A)|\}$.

*Proof.* Assume without loss of generality that the initial uncertainty contains the bounded set $\Omega_0 = \{X : \|X\|_2 \le L\} \subseteq \Lambda_0$. Note, possibly after a coordinate transformation, that the matrix $A$ can be written in the form $\begin{bmatrix} A_s & \\ & A_u \end{bmatrix}$ where the $A_s$ block corresponds to the stable subspace and the $A_u$ block corresponds to the marginally stable and unstable subspace.

Let $\Pi_s$ represent the projection onto the stable subspace. Fix an arbitrary control sequence $\{U_t\}$. Then $X_t = A^t X_0 + \alpha_t$ where $\alpha_t = \sum_{j=0}^{t-1} A^{t-1-j} B U_j$. For any control sequence we have $\lim_{t\to\infty} \Pi_s(X_t - \alpha_t) = 0$. Thus knowledge of the control signals alone is enough to estimate the projection of the state onto the stable subspace. Hence, without loss of generality, we can restrict our attention to $A$ matrices that contain only marginally stable and unstable eigenvalues.

The set of points that $X_t$ can take contains the following set $\Omega_t \doteq \{X \,:\, X = A^t X_0 + \alpha_t \text{ for some } X_0 \in \Omega_0\}$. If the system is asymptotically observable then it must be the case that $\forall \varepsilon > 0$ and $\forall X_0 \in \Omega_0$ there is a $T(\varepsilon, L)$ such that for $t \ge T(\varepsilon, L)$ we have $\|e_t\|_2 \le \varepsilon$. In particular this must hold for $\varepsilon < L$. A lower bound on the rate can be computed by counting the number of regions of diameter less than $2\varepsilon$ it takes to cover $\Omega_t$ for $t \ge T(\varepsilon, L)$.

If the diameter of a set $\Omega \subseteq \mathbb{R}^d$ is less than $2\varepsilon$ then the volume of that set must be less than $K_d \varepsilon^d$ (where $K_d$ is the constant in the formula for the volume of a sphere in $\mathbb{R}^d$.) Thus to cover $\Omega_t$ by regions of diameter $2\varepsilon$ we require at least $R \ge \frac{1}{t} \log \frac{\mathrm{volume}(\Omega_t)}{K_d \varepsilon^d} \overset{(a)}{=} \frac{1}{t} \log \frac{|\det(A^t)|\ \mathrm{volume}(\Omega_0)}{K_d \varepsilon^d} = \sum_{\lambda(A)} \log|\lambda(A)| + \frac{d}{t} \log \frac{L}{\varepsilon}$ where (a) follows from theorem 10.38 in [1]. Since $\frac{d}{t} \log \frac{L}{\varepsilon}$ is positive the proposition holds. □

We now discuss stabilizability under a rate constraint. We say system (1) is *asymptotically stabilizable* if there exists an encoder, decoder, and controller such that $\forall\, \varepsilon > 0,\ \delta > 0,\ \exists\, T(\varepsilon,\delta)$ such that $\|X_0\|_2 \leq \delta$ implies $\|X_t\|_2 \leq \varepsilon, \forall t \geq T$.

**Proposition 2.** *Assume $(A,B)$ is a stabilizable pair. A necessary condition for system (1) to be asymptotically stabilizable is $R > \sum_{\lambda(A)} \max\{0, \log|\lambda(A)|\}$.*

The proof uses a counting argument similar to that given in proposition 1. Full details can be found in [26, 29].

### 2.2 Sufficient Conditions

We now provide sufficient conditions. This is accomplished by explicitly constructing an encoder, a decoder, and a controller. Recall that the encoder at time $t$ is a map, $\mathcal{E}_t$, that takes $(Y^t, \sigma^{t-1}, U^{t-1}) \mapsto \sigma_t$. In this case the encoder knows the past outputs, past channel symbols, and past controls.

One may envision other scenarios where the information pattern at the encoder is different. The information pattern described above is at one extreme. At the other extreme we would have the empty information pattern. There are many scenarios inbetween these two extremes. For example it, when the encoder is geographically separated from the plant it may not be reasonable to allow the encoder access to the past controls. This scenario is discussed in [29]. In section 2.4 of this chapter we discuss the scenario where the encoder is required to be time-invariant. As you might expect the choice of information pattern greatly effects the communication requirements need to achieve different control objectives.

To get some intuition we first discuss the scalar case: $X_{t+1} = aX_t + bU_t$. Assume that $e_0 = x_0 \in \Lambda_0 \subseteq [-L_0, L_0]$. At time $t$ let $L_t$ represent the box that the error lives in: $e_t \in [-L_t, L_t]$. The encoder knows the decoder's state estimation error $e_t$. At each time step the encoder partitions $[-L_t, L_t]$ into $2^R$ equal regions and sends the partition label information to the decoder. Taking into account the dynamics of the system we see that $L_{t+1} = |a|2^{-R}L_t$ is a bounding interval for $e_{t+1}$. Clearly if $R > \max\{0, \log|a|\}$ then $L_t \to 0$ and we have observability.

For the vector case we will employ the following *primitive quantizer:*

**Definition 1.** *A* primitive quantizer *is a four-tuple $(c, \underline{R}, \underline{L}, \Phi)$ where $c \in \mathbb{R}^d$ represents the centroid, $\underline{R} = (R_1, \ldots, R_d)' \in \mathbb{R}^{d,+}$ represents the rate vector, $\underline{L} = (L_1, \ldots, L_d)' \in \mathbb{R}^{d,+}$ represents the dynamic range, and $\Phi$ is an invertible matrix that represents a coordinate transformation. This quantizer partitions the region $\Lambda = \{X \in \mathbb{R}^d : \Phi(X-c) \in \{[-L_1, L_1] \times \ldots \times [-L_d, L_d]\}\}$ into boxes with side lengths $\frac{2L_i}{2^{R_i}}$. Let $R = \sum_{i=1}^d R_i$. Each of the $2^R$ boxes is represented by an element $\sigma \in \Sigma$.*

Note that this quantizer is essentially a set of boxes. In words, the encoder upon observing $X$ applies the $(c, \underline{R}, \underline{L}, \Phi)$-quantizer. It first subtracts off $c$, it then applies the coordinate transformation $\Phi$, it then determines which box it falls into (points that land on the boundary of more than one box are given the label of one of those boxes according to some fixed priority rule), and finally it then transmits the $\sigma$ representing

that box. If $X$ falls outside the region $\Lambda$ then the quantizer transmits a special symbol representing an overflow. Thus we have $2^R + 1$ symbols. The set $\Lambda$ is called the *support* of the quantizer.

The sufficiency proof is based on the following idea. At time $t$ the encoder first computes the decoder's estimate of the state. It can compute this estimate because it knows both the policy of the decoder and the control and channel signals available to the decoder. The encoder then computes the difference, the *innovation*, between the true value of the state and the decoder's estimate of the state. Next the encoder chooses a primitive quantizer and quantizes this innovation. The $c$ in the primitive quantizer corresponds to the decoder's estimate. The $\Phi$ corresponds to the coordinate transformation transforming $A$ into real Jordan form. The encoder then transmits to the decoder the appropriate channel symbol corresponding to this quantization value. The decoder knows which primitive quantizer the encoder used and hence can decode this channel symbol and update its state estimate. The encoder then updates the primitive quantizer by updating the centroid $c$ and the dynamic range $\underline{L}$ appropriately [29].

There are many situations where the rate bounds presented in the previous section will be non-integer. One can time-share between two primitive quantizers such that on average one sends a non-integer number of bits [29]. For the rest of this chapter the statement "a sufficient condition for property X to hold is $R > \sum_{\lambda(A)} \max\{0, \log|\lambda(A)|\}$" will be taken to mean we are using the fixed rate scheme for integer rates and the average rate scheme for non-integer rates.

**Proposition 3.** *For system (1), $(A,C)$ is observable, and bounded initial set $\Lambda_0$ a sufficient condition on the rate for asymptotic observability is $R > \sum_{\lambda(A)} \max\{0, \lceil \log|\lambda(A)| \rceil\}$.*

The proof is based on the intuitive description of the encoder above [29]. The rate of convergence of the scheme described above is exponential and is given by $\|e_t\|_2 \leq \kappa 2^{-t(\min_i(R_i - \log|\lambda_i(A)|))}$ where $\kappa$ is a constant independent of $t$ [29].

We can treat the case when $\Lambda_0$ is bounded but the encoder does not *a priori* know that bound. In particular the condition $R > \sum_{\lambda(A)} \max\{0, \log|\lambda(A)|\}$ continues to be sufficient. The idea is to first grow the quantizer's dynamic range at a rate faster than the dynamics of the system. The encoder will continue to send the overflow symbol until the state is captured. Then one can apply proposition 3. The rate of convergence result no longer holds here for we do not know *a priori* how long we have to wait before we capture the state. This idea of growing the dynamic range first appeared in [9].

The methodology above can be applied to systems with bounded additive process disturbances.

$$X_{t+1} = AX_t + BU_t + W_t, \quad Y_t = X_t, \quad t \geq 0 \tag{2}$$

where $\|W_t\|_2 \leq D$. Due to the noise term $W_t$ there will always be a nonzero state estimation error. The following proposition shows that the rate condition is sufficient to insure bounded error [29].

**Proposition 4.** *For system (2) and bounded initial set* $\Lambda_0$ *the rate* $R > \sum_{\lambda(A)} \max\{0, \log|\lambda(A)|\}$ *is sufficient to insure* $\overline{\lim}_{t\to\infty} \|e_t\|_2$ *is bounded.*

We can combine the properties of asymptotic observability and full state feedback stabilizability to get output feedback stabilizability.

**Proposition 5.** *For system (1),* $(A,B)$ *stabilizable,* $(A,C)$ *is observable, and initial set* $\Lambda_0$ *bounded a sufficient condition for asymptotic stabilizability is* $R > \sum_{\lambda(A)} \max\{0, \log|\lambda(A)|\}$.

*Proof.* Let $K$ be a stabilizing controller, i.e. $A+BK$ is stable. Apply the certainty equivalent controller $U_t = K\hat{X}_t$ where $\hat{X}_t$ is the decoder's state estimate. We need to show that under this control law the system is stable. As before let $e_t = X_t - \hat{X}_t$. Then $X_t = (A+BK)^t X_0 - \sum_{j=0}^{t-1}(A+BK)^{t-1-j}BKe_j$. By proposition 1 we can asymptotically observe the state. Thus the $\lim_{t\to\infty}\|e_t\|_2 = 0$. Since $A+BK$ is stable and $\Lambda_0$ is bounded the first addend in the above equation goes to zero. One can show that second addend also goes to zero [29]. Hence the proposition follows. □

### 2.3 Continuous Time Sampled Systems

Here we consider the following continuous time linear system:

$$\dot{X}(t) = AX(t) + BU(t) \tag{3}$$

where $X(t)$ is a $\mathbb{R}^d$-valued state process and $U(t)$ is a $\mathbb{R}^m$-valued control process. The initial position $X_0 \in \Lambda_0$ where $\Lambda_0 \subseteq \mathbb{R}^d$.

Let the sampling period be $T$ and let the control input be held in the intersample interval with zero-order hold. Define $X_n = X(nT)$. Let $U(t) = U_n, \;\; nT \le t < (n+1)T$ for some control sequence $\{U_n\}$. Then for $t = nT + \delta$ we have $X(t) = e^{\delta A}X_n + \int_0^\delta e^{(\delta-\tau)A}BU_n d\tau$. We can rewrite this as $X_{n+1} = \bar{A}(T)X_n + \bar{B}(T)U_n$ where $\bar{A}(T) = e^{TA}$ and $\bar{B}(T) = \int_0^T e^{(T-\tau)A}Bd\tau$.

We would like to know what the best sampling period is. By "best" we mean the minimum number of bits per time unit. Thus if we have a scheme that samples the system every $T$ time units and transmits $R$ bits then the number of bits per time unit is $\frac{R}{T}$.

As before let $e(t) = X(t) - \hat{X}(t)$. We can generalize definition (1) of asymptotic observability to continuous time in the natural way.

**Proposition 6.** *Given system (3), bounded* $\Lambda_0$*, and sampling period is* $T$*. A sufficient and necessary condition on the rate for asymptotic observability is* $R > \sum_{\lambda(\bar{A})}\{0, \log|\lambda(\bar{A})|\}$.

*Proof.* By proposition 3 we know that under the rate condition $\lim_{n\to\infty} e(nT) = 0$. We need to show that the error in any intersample period stays bounded. Let $t = nT + \delta$ then $e(t) = e^{\delta A}X_n + \int_0^\delta e^{(\delta-\tau)A}BU_n d\tau - (e^{\delta A}\hat{X}_n + \int_0^\delta e^{(\delta-\tau)A}BU_n d\tau) = e^{\delta A}e(nT)$. For finite $T$ we have $\sup_{0\le\delta\le T}\|e^{\delta A}\|$ is bounded.

By proposition 1 the rate condition is necessary for $e(nT)$ to go to zero. Clearly if $e(nT)$ does not go to zero then $e(t)$, for general intersample times $t$, will not go to zero. $\square$

Let $A$ have eigenvalues $\lambda_1, ..., \lambda_d$. Then $\bar{A}$ has eigenvalues $\bar{\lambda_i} = e^{T\lambda_i}$. The rate is $\frac{R(T)}{T} > \frac{1}{T}\sum_{\lambda(\bar{A})} \max\left\{0, \log|\lambda(\bar{A})|\right\} = \sum_{\lambda(A)} \max\left\{0, \frac{\text{Real}(\lambda(A))}{\ln 2}\right\}$. Thus the optimal number of bits per time unit is independent of the sampling period $T$. We are assuming, though, that at the start of every epoch we are instantaneously transmitting $R(T)$ bits. So though the average rate $\frac{R(T)}{T}$ is independent of $T$ the peak value is very much dependent on $T$.

Next we treat the problem of stabilizability under sampling. Assume that $(A,B)$ are controllable. Recall that if $(A,B)$ is controllable and $T$ is such that $T(\lambda - \mu) \neq 2k\pi i \;\; k = \pm 1, \pm 2, ...$ for every two eigenvalues of $A$ then $(\bar{A}, \bar{B})$ is controllable [19]. The proof of the following proposition follows from propositions 2, 5, and 6 [26].

**Proposition 7.** *Given system (3), (A, B) controllable, a sampling period $T$ insuring that $(\bar{A}, \bar{B})$ is controllable, and bounded $\Lambda_0$. The rate $R > \sum_{\lambda(\bar{A})} \max\{0, \log|\lambda(\bar{A})|\}$ is a necessary and sufficient condition for asymptotic stabilizability.*

### 2.4 Time-Invariant Quantizers

Here we examine time-invariant encoders. This should be contrasted with the encoders in section 2.2. We begin our study with discrete-time systems and quantizers with countable number of levels. In particular, assuming that the system is quadratically stabilizable, we show that the quantizer is logarithmic (the fixed levels follow a logarithmic law). Further, we characterize the coarsest, or least dense, (largest spacing between levels) logarithmic quantizer over all quadratic control Lyapunov functions in terms of the solution of a special linear quadratic regulator (LQR) problem. We next extend the definition of density of quantization to the density of sampling and quantization in a natural way, and search for the coarsest (least dense) sampling and quantization scheme. We show that the resulting optimal sampling time is only a function of the unstable eigenvalues of the continuous time system, and provide a formula for it. The associated optimal quantizer is logarithmic with the logarithmic base being a universal constant independent of the system.

The system is assumed unstable, single input, stabilizable, and governed by the following equation:

$$X_{t+1} = AX_t + BU_t \tag{4}$$

where here $B \in \mathbb{R}^{n\times 1}$. Since the system is stabilizable and linear, it is quadratically stabilizable, i.e., there is a control input $u$, function of $x$, that makes a quadratic function of the state a Lyapunov function for the closed loop system. Such Lyapunov functions are called Control Lyapunov Functions (CLF).

Given a quadratic CLF $V(x) = x'Px$ with $P > 0$, $P$ is assumed to be symmetric, we propose to select a set of fixed control values such that $V(x)$ is still a Lyapunov function for the system, i.e., it decreases along the system's trajectories. Specifically we want to find a set $\mathcal{U} = \{u_i \in \mathbb{R} : i \in \mathbb{Z}\}$ and a function $f : X \to \mathcal{U}$ such that $f(x) = -f(-x)$, and such that for any $x \in X$, $x \neq 0$ $\Delta V(x) = V(Ax + Bf(x)) - V(x) < 0$

With a slight abuse of terminology $f$ is called the quantizer. Note that unlike in section 2.3 the quantizer is not restricted to consist of boxes. The range of $f$ induces a partition in the state-space of the system, where equivalence classes of states correspond to the same adopted control value, i.e., $\Omega_i = \{x \in X \mid f(x) = u_i\}$. Notice that we consider only quantizers that are symmetric with respect to the origin and with a countable number of levels. The first is not a restriction given the natural symmetry of the system and the Lyapunov function. The second is also not a restriction since, as will see in the development, such quantizers are required to solve problem.

Our objective is to find the coarsest quantizer that still stabilizes the system. We measure the coarseness of a quantizer by measuring its density. Given $V(x) = x'Px$, $P > 0$, a CLF for system (4), let $\mathcal{Q}(V)$ denote the set of all quantizers for which $\Delta V(x) < 0, x \neq 0$. For $g \in \mathcal{Q}(V)$ and $0 < \varepsilon \leq 1$, let $\#g[\varepsilon]$ denote the number of levels that $g$ has in the interval $\left[\varepsilon, \frac{1}{\varepsilon}\right]$. Define the *quantization density* of $g$ to be $\eta_g = \overline{\lim}_{\varepsilon \to 0} \frac{\#g[\varepsilon]}{-\ln \varepsilon}$. A quantizer $f$ is said to be *coarsest* for $V(x)$ if it has the smallest density of quantization: $f = \arg\inf_{g \in \mathcal{Q}(V)} \eta_g$.

A quantizer which is *coarsest* for $V(x)$ need not be unique since, different sets $\mathcal{U}$ may satisfy the asymptotic property, and for the same $\mathcal{U}$, there may be different ways to define the function $f$ mapping $X$ into $\mathcal{U}$. Moreover, a quantizer which is *coarsest* for $V(x)$ may not be an element of $\mathcal{Q}(V)$. At any rate, since the quantizer induces a partition on $X$, the density of quantization induces a measure of coarseness on the partitions in the state-space.

The main idea in the derivation of $\mathcal{U}$ and $f$ is to consider the CLF as a robust Lyapunov function where, for a given fixed control value, we are interested in finding the set of all states for which $\Delta V(x)$ is negative enough. We want to emphasize that the idea of using robust Lyapunov functions to design nonlinear control strategies, in particular quantizers, for linear systems is new and has great advantages with respect to traditional approaches based on optimal control which even in the case of a given fixed quantizer would often lead to intractable integer programming problems.

We now show that, for a given CLF, there is a natural quantization of the control values and a partition of state-space which follows a logarithmic law. This result captures in a precise way the intuitive notion that, the farther from the origin the state is, the less precise the control action and knowledge about the location of the state in the state-space need to be.

Given a CLF, $V(x) = x'Px$ ($P > 0$), for system (4), the control that makes $\Delta V(x)$ the most negative is given by $u = -\frac{B'PA}{B'PB}x \doteq K_{GD}x$. Where $GD$ stands for gradient descent. Let $A_c = A + BK_{GD}$ be the resulting closed loop state-transition matrix. Also let $Q = P - A_c'PA_c = P - A'PA + \frac{A'PBB'PA}{B'PB}$. Note that $Q > 0$ since $x'Px$ is a CLF.

**Proposition 8.** *Let $V(x) = x'Px$, $P > 0$ be a CLF for system (4). A quantizer $f : X \to \mathcal{U}$, $f(x) = u$, which is coarsest for $V(x)$, has fixed control values that follow a logarithmic law, and it is characterized as follows.*

$$\mathcal{U} = \{\pm u_i, : u_{i+1} = \rho u_i,\ u_0 = \beta_u,\ i \in Z\} \cup \{0\},$$

*the constant $0 \leq \rho < 1$ is given by*

$$\rho = \left( \sqrt{\frac{B'PAQ^{-1}A'PB}{B'PB}} - 1 \right) \Bigg/ \left( \sqrt{\frac{B'PAQ^{-1}A'PB}{B'PB}} + 1 \right).$$

*Where* $\Omega_i^+ = \{x \in X \,|\, \alpha_{i+1} < K_{GD}x \leq \alpha_i,\ \alpha_{i+1} = \rho\alpha_i,\ \alpha_0 = \beta_\alpha,\ i \in Z\}$, $\Omega_{zero} = \{x \in X \,|\, K_{GD}x = 0\}$, *and* $\beta_\alpha, \beta_u$ *are constants.*

*Proof.* We sketch the proof here. Full details can be found in [12]. For all $x \in X$, $x \neq 0$, let $U(x) = \{u \in \mathbb{R} \,|\, \Delta V(x) < 0\}$. Note that $U(x)$ can be equivalently characterized by the following open interval $U(x) = \left\{ u \in \mathbb{R} \,|\, u^{(1)} < u < u^{(2)} \right\}$ where $u^{(1)}$ and $u^{(2)}$ are the roots of the second order equation in $u$: $x'(A'PA - P)x + 2x'A'PBu + u'B'PBu = 0$. These are given by the following expression $u^{(1),(2)} = K_{GD}x \pm \sqrt{\frac{x'Qx}{B'PB}}$. The set $U(x)$ is nothing but the set of control values $u$ that can be selected (for the given $x$) to ensure that the Lyapunov function is still decreasing along trajectories. $U(x)$ has the following two important properties: *scaling:* $U(\alpha x) = \alpha U(x)$ for $\alpha > 0$ and *symmetry:* $u^{(1)} = -u^{(2)}$ for all $x$ such that $K_{GD}x = -\dfrac{B'PAx}{B'PB} = 0$.

From the symmetry property we see that $u = 0$ can be used for all $x \perp K'_{GD}$ to ensure that the Lyapunov function decreases along trajectories. This also implies that, in searching for a quantizer which is coarsest, we can restrict our attention to the partition induced by the quantizer in the direction $K'_{GD}$.

From the scaling property one can show that the coarsest covering in the direction $\overline{K}'_{GD}$ follows a logarithmic law. Let $\rho$ be the maximum range of states along the direction $\overline{K}'_{GD}$ for which there is a common control value that still decreases the Lyapunov function in the next step. The value of $\rho$ can be computed and is stated above. □

We now characterize the coarsest quantizer (smallest $\rho$) by searching over all quadratic CLF's. The optimal quantizer is related to a special LQR problem: $\rho^* = \inf_{\{V = x'Px,\ P>0, V \text{ CLF}\}} \rho$.

**Proposition 9.** *Assume that system (4) is unstable, and let* $\lambda_i^u$ $i = 1, \ldots, k \leq n$ *denote the eigenvalues of the matrix A with magnitude greater or equal than* 1. *Then, the optimal quadratic Lyapunov function corresponding to* $\rho^*$ *is given by the positive semi-definite solution of the following Riccati equation:* $P^* = A'P^*A - \frac{A'P^*BB'P^*A}{B'P^*B+1}$ *which is also the solution to the special LQR problem* $\min \sum_{k=0}^{\infty} u_k^2$ *over stabilizing control laws. This corresponds to the minimum energy control that stabilizes the system.*

*Furthermore* $K_{GD}^* = -\frac{B'P^*A}{B'P^*B}$ *and it is parallel to* $K_{LQR} = -\frac{B'P^*A}{B'P^*B+1}$ *the LQR optimal controller. Finally*

$$\rho^* = \left( \prod_{1 \leq i \leq k} |\lambda_i^u| - 1 \right) \Bigg/ \left( \prod_{1 \leq i \leq k} |\lambda_i^u| + 1 \right) \tag{5}$$

Note that this theorem [12] is the first to capture in very precise terms the intuitive argument that when the system is far from the equilibrium we do not need precise

knowledge of the state and therefore we can use imprecise controls to steer the system in the right direction.

We now consider both sampling and quantization of the LTI system. In particular, we extend the previous quantization results by studying the case of uniform sampling and derive a criterion for optimal sampling and quantization. These results should be contrasted with those in section 2.3.

We assume that the state of the linear system $\dot{X} = FX + GU$ is sampled with sampling time $T$, and that the control input is held in the intersample interval with zero-order hold. Furthermore, let $\lambda_i^u(F)$, $i = 1,\ldots,k \leq n$ denote the eigenvalues of the matrix $F$ with positive real part. Let the resulting discrete LTI system be $X_{t+1}^d = A_T X_t^d + B_T u_t^d$ where $A_T = e^{FT}$ and $B_T = \int_0^T e^{A(T-\tau)} B d\tau$ are functions of $T$. It is implicitly assumed that the discretized system is stabilizable. As stated before this is true for all but at most a countable set of sampling times [19]. The critical sampling times where there is a loss of stabilizability are not considered in what follows.

From before we know there is a coarsest quantizer that is logarithmic with base $\rho^*(T)$. Since $\lambda_i(A) = e^{\lambda_i(F)T}$, we can compute $\rho^*(T)$ from equation (5). This formula states that we need to quantize more finely if we sample slower in order to maintain stability of a given system. Since sampling is nothing but quantization in time we can generalize the concept of density to measure the coarseness of quantizers in space$\times$time, and derive a criterion for optimality of sampling and quantization.

Given a CLF $V(x) = x'Px$, $P > 0$, let $\mathcal{Q}_T(V)$ denote the set of all quantizers that stabilize the system under uniform sampling $T$. Let the *density of sampling and quantization* of (T,$g$) to be: $\eta_{g,T} = \overline{\lim}_{t\to\infty} \overline{\lim}_{\varepsilon\to 0} \frac{(\#of samples \in [0,t])(\#g[\varepsilon])}{-t\ln\varepsilon} = \frac{1}{T} \overline{\lim}_{\varepsilon\to 0} \frac{\#g[\varepsilon]}{-\ln\varepsilon}$. We define the pair $(T^*, f^*)$ of sampling time and quantizer to be coarsest for quadratic stability if they minimize the density of sampling and quantization over all quadratic CLF.

**Proposition 10.** *The optimal sampling time $T^*$ satisfies the following equation $T^* \sum_{i=1}^k \lambda_i^u(F) = \ln(1+\sqrt{2})$ and the corresponding optimal quantizer is logarithmic with base $\rho^*(T^*) = \sqrt{2}-1$ which is independent of the system.*

Note that the above theorem [12] states that the product of the optimal sampling time, $T^*$, and the sum of the unstable poles of the system is a constant independent of the system, and in this sense, *universal*. Also note that at $T^*$, $\rho^*(T^*) = \sqrt{2}-1$ is independent of the system. This indicates that $\rho^*(T^*)$ is the base of the *universal* logarithmic quantizer for a single input continuous time linear system for which the density of sampling and quantization is minimized.

In practice one would use a quantizer with a finite number of levels. One can replace logarithmic quantizers with countable levels as described above by a logarithmic quantizer with finite number of levels and still maintain *practical* stability [12].

### 2.5 Multiple Sensors

Here we examine the case of multiple sensors. We first formulate the problem and then relate its solution to the Slepian-Wolf coding theorem [10]. Consider a linear system with $M$ distributed observations and one controller:

$$X_{t+1} = AX_t + BU_t, \quad Y(m)_t = C_m X_t, \quad m = 1,...,M \tag{6}$$

where $A \in \mathbb{R}^{d\times d}, B \in \mathbb{R}^{d\times m}, C_m \in \mathbb{R}^{l_m \times d}$, and $X_0 \in \Lambda_0$. Each encoder is connected to the decoder via a noiseless digital channel.

For each $m$ the *encoder* $\mathcal{E}_m$ is a map that takes $(Y(m)^t, \sigma(m)^{t-1}, U^{t-1}) \mapsto \sigma(m)_t$. Encoder $\mathcal{E}_m$ is allowed to depend only on its own past and current observations, its own past channel signals, and the control signals. The *decoder* $\mathcal{D}$ is a map that takes $(\sigma(1)^t, ..., \sigma(M)^t, U^{t-1}) \mapsto \hat{X}_t$. This is a centralized decoder. It is allowed to observe all the different channel signals produced by the $M$ encoders. The *controller* $\mathcal{C}$ is a map that takes $\hat{X}_t \mapsto U_t$. We will assume a certainty equivalent controller.

Let $C = [C_1', ..., C_M']'$. We assume that the pair $(A,C)$ is observable and the pair $(A,B)$ is stabilizable. Though $(A,C)$ is observable it is not generally the case that $(A,C_m)$ will be observable. Define $\mathcal{N}_m$ to be the unobservable subspace associated with $(A,C_m)$. Define $\mathcal{O}_m$ to be the subspace orthogonal to $\mathcal{N}_m$. It can be shown that $\mathcal{N}_m$ is indeed a subspace and furthermore that it is $A$-invariant [19]. By assumption $\cap_{m=1}^{M} \mathcal{N}_m = \emptyset$. Also any $x \in \mathcal{O}_m$ can be observed by encoder $m$.

Associate with each $m$ the set $\Lambda_m = \{\lambda(A) : \text{those eigenvalues of } A \text{ corresponding to the subspace } \mathcal{O}_m\}$. Thus $\cup_{m=1}^{M} \Lambda_m = \{\lambda(A)\}$. In general the $\Lambda_m$ will not be disjoint. Because they are not disjoint we have freedom in determining what each encoder sends to the decoder. We will show that this freedom can be captured by the following rate region. For each encoder $\mathcal{E}_m$ define its rate vector to be $\underline{R}_m = (R_{m,1}, ..., R_{m,\dim \mathcal{O}_m})$. (Where $\underline{R}_m$ is the rate vector of the primitive quantizer employed by the encoder $\mathcal{E}_m$. See [26, 29] for more details.)

For a given matrix $A$ and observation matrices $C_1, ..., C_M$ define $\mathcal{R} = \{(\underline{R}_1, ..., \underline{R}_M)$ such that $\sum_{m:\lambda(A)\in\Lambda_m} R_{m,j_{\lambda(A)}} \geq \max\{0, \log|\lambda(A)|\}, \quad \forall\, \lambda(A)\}$ where $j_{\lambda(A)}$ represents the index of the rate component associated with that eigenvalue.

**Proposition 11.** *For system (6) a necessary and sufficient condition on the rate so that the system is asymptotically observable is* $(\underline{R}_1, ..., \underline{R}_M) \in \mathcal{R}$.

*Proof.* From proposition 1 we know that the total rate coming from all the encoders to the decoder must satisfy $R \geq \sum_{\lambda(A)} \max\{0, \log|\lambda(A)|\}$. Furthermore, for each unstable eigenvalue $\lambda$, all the encoders $\mathcal{E}_m$ such that $\lambda \in \Lambda_m$ are required to transmit in total at least $\log|\lambda|$ bits. The region $\mathcal{R}$ contains all rate vectors that satisfy these requirements.

We prove necessity for the case when $M = 2$. The proof of necessity for the more general case is straightforward. In this case there are three sets of interest: $\Lambda_1 \cap \Lambda_2$, $\Lambda_1 \setminus (\Lambda_1 \cap \Lambda_2)$, and $\Lambda_2 \setminus (\Lambda_1 \cap \Lambda_2)$. By proposition 3 we know that the decoder can asymptotically observe the subspace associated with $\Lambda_1 \setminus (\Lambda \cap \Lambda_2)$ with $\sum_{\lambda\in\Lambda_1\setminus(\Lambda_1\cap\Lambda_2)} \max\{0, \log|\lambda|\}$ bits transmitted only by encoder 1. Similarly the decoder can asymptotically observe the subspace associated with $\Lambda_2 \setminus (\Lambda_1 \cap \Lambda_2)$ with $\sum_{\lambda\in\Lambda_2\setminus(\Lambda_1\cap\Lambda_2)} \max\{0, \log|\lambda|\}$ bits transmitted only by encoder two.

This leaves us with $\Lambda_1 \cap \Lambda_2$. Clearly we don't need both encoder one and encoder two sending the same information. Thus any splitting of the rate between encoder 1 and encoder 2 needed to describe $\Lambda_1 \cap \Lambda_2$ is sufficient for the decoder to asymptotically observe the subspace associated with $\Lambda_1 \cap \Lambda_2$. We only require that the

combined rate used to describe $\Lambda_1 \cap \Lambda_2$ be greater than $\sum_{\lambda \in \Lambda_1 \cap \Lambda_2} \max\{0, \log|\lambda|\}$. For example encoder one can send coarse bits while encoder two sends fine bits (i.e. most significant digits and least significant digits.) □

Now we provide lower and upper bounds for asymptotic stabilizability.

**Proposition 12.** *For system (6) a necessary and sufficient condition on the rate so that the system is asymptotically stabilizable is* $(\underline{R}_1, ..., \underline{R}_M) \in \mathcal{R}$.

The proof builds on propositions 2, 5, and 11. See [26, 27].

The distributed sensor problem we have formulated is related to the problem of coding of correlated sources. The Slepian-Wolf coding problem concerns itself with transmitting the discrete random variable $(Z_1, ..., Z_M)$ losslessly. Encoder $m$ observes $Z_m$ and transmits information at rate $R_m$. For the case where $M = 2$ the rates must satisfy the following three inequalities: (1) $R_1 + R_2 \geq H(Z_1, Z_2)$, (2) $R_1 \geq H(Z_1|Z_2)$, and (3) $R_2 \geq H(Z_2|Z_1)$ where $H(\cdot)$ and $H(\cdot|\cdot)$ are the discrete entropy and conditional discrete entropy respectively. Under the correspondence: $H(Z_m) \Longleftrightarrow \sum_{\lambda \in \Lambda_m} \max\{0, \log|\lambda|\}$ and $H(Z_m|Z_{3-m}) \Longleftrightarrow \sum_{\lambda \in \Lambda_m \setminus (\Lambda_m \cap \Lambda_{3-m})} \max\{0, \log|\lambda|\}$ for $m = 1, 2$, we see that $\mathcal{R}$ is indeed the region specified by the Slepian-Wolf region. Extensions to more general wired networks can be found in [27].

## 3 Control Over Noisy Channels

In section 2 we examined control over noiseless channels. Here we examine noisy channels. We show that ideas from information theory are important to the analysis of such control systems. There are two main reasons why noisy channels are more difficult to treat than noiseless channels. First, because of the noise, the encoder may not know what information is being received by the decoder. Second, because of the noise, one needs to design both a source encoder and a channel encoder.

We first present a general necessary condition on the channel, based on the Shannon capacity, that insures observability and stabilizability. We then give two examples for which this condition is tight: erasure channel with acknowledgement and the LQG problem over a Gaussian channel.

### 3.1 Noisy Channels

Consider the system described earlier, system (1),

$$X_{t+1} = AX_t + BU_t \quad \forall t \geq 0 \tag{7}$$

excepting in this case the initial position, $X_0$, is distributed according to the probability density $p(X_0)$ with support on the open set $\Lambda_0 \subseteq \mathbb{R}^d$ and finite differential entropy $h(X_0)$.

A noisy channel is a sequence of stochastic kernels $\{P(B_t \mid a^t, b^{t-1})\}$. Specifically for each realization of $(a^t, b^{t-1})$ the conditional probability of $B_t$ given $(a^t, b^{t-1})$ is given by $P(B_t \mid a^t, b^{t-1})$. At time $t$ the encoder produces a channel input symbol

$A_t = a_t$ and the channel outputs the channel output symbol $B_t$ according to the probability $P(B_t \mid a^t, b^{t-1})$. Some examples include: noiseless digital channel with rate $R$, delayed noiseless digital channel with delay $\Delta$, erasure channel with erasure probability $\alpha$, and the Gaussian channel with power $P$. The latter two channels will be defined explicitly in the following sections.

Here we examine observability and stabilizability over general communication channels. In this case the *error* is $E_t = X_t - \hat{X}_t$, where $\hat{X}_t$ is the state estimate, is a random variable. System (7) is *almost surely asymptotically observable* if there exists a control sequence $\{U_t\}$ and an encoder and decoder such that the state estimation error $\|E_t\|_2 \to 0$ almost surely. System (7) is *almost surely asymptotically stabilizable* if there exists an encoder, decoder, and controller such that $\|X_t\|_2 \to 0$ almost surely. We discuss convergence in mean in section 4.

Our goal is to determine properties of the channel that insure almost sure asymptotic observability and stabilizability for general channels. To that end we need a measure of channel quality. Shannon's channel capacity turns out to be the correct measure. Given a channel $\{P(B_t \mid a^t, b^{t-1})\}$, the *Shannon capacity* over a time horizon of length $T$ is defined as the supremum of the mutual information over all channel input distributions $P(A^T)$. Specifically $C_T^{\text{cap}} = \sup_{P(A^T)} I(A^T; B^T)$ where $I(\cdot;\cdot)$ is the mutual information [10]. For any given channel define $C^{\text{cap}} \doteq \underline{\lim}_{T\to\infty} \frac{1}{T} C_T^{\text{cap}}$.

We will also need the rate-distortion function [10]. Let $X$ be a source with distribution $P(X)$ and $d(x,\hat{x})$ a distortion measure. Here a distortion measure is any nonnegative function that measures the relative fidelity in reconstructing $x$ by $\hat{x}$. Given a source $P(X)$ the *rate distortion* function is defined as the infimum of the mutual information over all channels, $P(\hat{X}|x)$, that satisfy the distortion condition: $R(D) = \inf_{P(\hat{X} \mid x)} \{I(X;\hat{X}) \text{ such that } E(d(X,\hat{X})) \le D\}$. The rate distortion function $R(D)$ represents a lower bound on the capacity of a channel needed to insure an end-to-end distortion of $D$. Under suitable conditions this bound is tight [10].

We will find the following parameterized family of distortion measures useful: $d^{\varepsilon}(x,\hat{x}) \doteq \begin{cases} 0 & \text{if } \|x-\hat{x}\|_2 \le \varepsilon \\ 1 & \text{if } \|x-\hat{x}\|_2 > \varepsilon \end{cases}$ where $\varepsilon > 0$. This choice of distortion measure will allow us to compute the probability that $X$ and $\hat{X}$ are farther than $\varepsilon$ apart. Specifically $E\left(d^{\varepsilon}(X,\hat{X})\right) = \Pr(\|X - \hat{X}\|_2 > \varepsilon)$.

We now present our necessary conditions for almost sure observability and stabilizability for general channels [30].

**Proposition 13.** *For system (7) a necessary condition on the channel capacity for almost sure asymptotic observability is* $C^{cap} > \sum_{\lambda(A)} \max\{0, \log|\lambda(A)|\}$.

*Proof.* We sketch the proof here. Assume that there exists an encoder and decoder such that system (7) is almost surely asymptotically observable. As in proposition 3 we see that, without loss of generality, we can restrict our attention to $A$ matrices that contain only unstable eigenvalues.

By almost sure asymptotic observability we know that for any $\varepsilon > 0$ there exists a $T(\varepsilon)$ such that the error $E_t = X_t - \hat{X}_t$ satisfies $\Pr(\sup_{t \ge T(\varepsilon)} \|E_t\|_2 > \varepsilon) \le \varepsilon$. Thus for $t \ge T(\varepsilon)$ we have $E\left(d^{\varepsilon}(X_t,\hat{X}_t)\right) = \Pr(\|X_t - \hat{X}_t\|_2 > \varepsilon) \le \varepsilon$.

It can be shown that the rate distortion function, $R_t^{\varepsilon}(\varepsilon)$, for $X_t$ and the distortion measure $d^{\varepsilon}$ can be bounded below by $R_t^{\varepsilon}(\varepsilon) \geq t(1-\varepsilon)\sum_{\lambda(A)} \log|\lambda(A)| + \left((1-\varepsilon)h(X_0) - \log(K_d \varepsilon^d) - \frac{1}{2}\right)$ [30]. Then by the data processing inequality [10] we have $\frac{1}{t}C_t^{\text{cap}} \geq \frac{1}{t}R_t^{\varepsilon}(\varepsilon)$. Hence $C^{\text{cap}} = \underline{\lim}_{t\to\infty} \frac{1}{t}C_t^{\text{cap}} = (1-\varepsilon)\sum_{\lambda(A)} \log|\lambda(A)|$. Since $\varepsilon$ can be chosen arbitrarily small we see $C^{\text{cap}} > \sum_{\lambda(A)} \log|\lambda(A)|$. □

**Proposition 14.** *For system (7) with $(A,B)$ a stabilizable pair a necessary condition on the channel capacity for almost sure asymptotic stabilizability is $C^{cap} > \sum_{\lambda(A)} \max\{0, \log|\lambda(A)|\}$.*

*Proof.* Assume there exists an encoder, decoder, and controller such that the system (7) is almost surely asymptotically stabilizable. For a given control sequence $U_0, U_1, ..., U_{t-1}$ we have $X_t = A^t X_0 - \alpha_t(U_0, ..., U_{t-1})$ where $\alpha_t(U_0, ..., U_{t-1}) \doteq -\sum_{i=0}^{t-1} A^{t-1-i} B U_i$. Almost sure asymptotic stabilizability implies that for any $\varepsilon$ there exists a $T(\varepsilon)$ such that $\Pr(\sup_{t\geq T(\varepsilon)} \|X_t\|_2 > \varepsilon) \leq \varepsilon$. We can view $\alpha_t$ as a reconstruction of $A^t X_0$ with distortion $E\left(d^{\varepsilon}(A^t X_0, \alpha_t)\right) \leq \varepsilon$. Using an argument similar to that given in proposition 13 a necessary condition to achieve this distortion is $C^{\text{cap}} > \sum_{\lambda(A)} \max\{0, \log|\lambda(A)|\}$. □

In the previous proposition we interpreted the following function of the control signals, $\alpha_t(U_0, ..., U_{t-1}) = -\sum_{i=0}^{t-1} A^{t-1-i} B U_i$, as a reconstruction of $A^t X_0$. We can view a particular sequence of control signals as a "codeword" in a reconstruction codebook [10, 30].

### 3.2 Erasure Channel

The erasure channel with packet size $R$ and erasure probability $\alpha$ is a channel that delivers a packet of size $R$ with probability $1-\alpha$ and drops the packet with probability $\alpha$. The Shannon capacity over $T$ channel uses is $C_T^{\text{cap}} = (1-\alpha)TR$ [10]. By proposition 13 a necessary condition for almost sure asymptotic observability is $C^{\text{cap}} = (1-\alpha)R > \sum_{\lambda(A)} \max\{0, |\log\lambda(A)|\}$. Hence we require a packet size of at least $R > \frac{1}{1-\alpha}\sum_{\lambda(A)} \max\{0, |\log\lambda(A)|\}$.

Now we examine sufficiency. To that end we will extend the erasure channel model to include acknowledgements. Specifically the decoder will feed back to the encoder an acknowledgment whether the packet was erased or not. This acknowledgment feature is common in the TCP network protocol. The encoder then knows what information has been delivered to the decoder. Hence, in the language of [30], we say that the encoder and decoder are equi-memory. Because the erasure channel is memoryless, acknowledgement feedback cannot increase the channel capacity [10]. Hence the necessity condition above continues to hold for erasure channels with acknowledgement feedback.

For simplicity we consider the system (7), $X_{t+1} = AX_t + BU_t$, with full state observation, $C = I$, at the encoder. The partially observed case can be treated in the manner described in [30].

**Proposition 15.** *Given system (7), a bound on $\Lambda_0$, and an erasure channel with erasure probability $\alpha$ and feedback acknowledgements the packet size $R > \frac{1}{1-\alpha}\sum_{\lambda(A)} \max\{0, |\log\lambda(A)\}$ is sufficient to ensure almost sure asymptotic observability.*

*Proof.* We first treat the scalar case: $X_{t+1} = aX_t + bU_t$. Let $E_t = X_t - \hat{X}_t$ and $E_0 = X_0 \in \Lambda_0 \subseteq [-L_0, L_0]$. At time $t$ let $L_t$ represent the box that the error lives in: $E_t \in [-L_t, L_t]$. We will construct a scheme such that $L_t \to 0$ almost surely and hence $E_t \to 0$ almost surely.

The decoder feeds back acknowledgments to the encoder. Hence the encoder can compute the decoder's uncertainty set $[-L_t, L_t]$. At time $t+1$ the encoder partitions the interval $[-|a|L_t,\ |a|L_t]$ into $2^R$ equal sized regions and sends the index of that region across the channel. If the erasure channel does not drop the packet then $L_{t+1} = \frac{|a|}{2^R}L_t$. If the packet is dropped then $L_{t+1} = |a|L_t$. This can be described by the stochastic difference equation: $L_{t+1} = |a|F_tL_t$ where the random variables $F_t$ are IID with common distribution: $\Pr(F_t = 1) = \alpha$ and $\Pr(F_t = 2^{-R}) = 1-\alpha$.

Since $L_t = L_0 \prod_{j=0}^{t-1}|a|F_j$ we need only show that $\prod_{j=0}^{t-1}|a|F_j \to 0$ almost surely. By the strong law of large numbers: $\frac{1}{t}\sum_{j=0}^{t-1}\log|a|F_j \to E(\log|a|F)$ almost surely. If $E(\log|a|F) < 0$ then $\prod_{j=0}^{t-1}|a|F_j = 2^{t(\frac{1}{t}\sum_{j=0}^{t-1}\log|a|F_j)} \to 0$ a.s. This result can be found in any standard text on large deviations. See for example [11]. Now $E(\log|a|F) = \alpha\log|a| + (1-\alpha)\log\frac{|a|}{2^R} = \log|a| - (1-\alpha)R$. Thus $E(\log|a|F)$ is negative if and only if $R > \frac{\log|a|}{1-\alpha}$. The vector case can be treated similarly. See [30] for full details. □

The proof of the following proposition follows analogously [30].

**Proposition 16.** *Given an erasure channel with erasure probability $\alpha$ and feedback acknowledgments the packet size $R > \frac{1}{1-\alpha}\sum_{\lambda_A} \max\{0, |\log\lambda(A)\}$ is sufficient to ensure almost sure asymptotic stabilizability.*

### 3.3 LQG

Here we examine the role communication has on the classical linear quadratic Gaussian (LQG) problem. To this end we present the sequential rate distortion (SRD) framework. We derive bounds on the achievable performance and show the inherent tradeoffs between control and communication costs. In particular we show that the optimal quadratic cost decomposes into two terms: a full knowledge cost and a sequential rate distortion cost. The size of the latter cost depends on how well the decoder can reconstruct the state of the plant.

There are two classical notions of separation that we examine in this section. The first notion is the control-theoretic separation between state estimation and control. We present conditions that insure the optimality of the certainty equivalent control law. These build on the work of Bar-Shalom and Tse [4]. The second notion is the information-theoretic separation between source encoder and channel encoder. In particular, in the limit of long delays, it is known that one can, without loss of generality, design the source encoder and the channel encoder separately [16]. This

separation is known to hold quite broadly, [35], but in general fails for both short delays and for unstable processes. Since delay is an important issue in control applications we cannot apply the information-theoretic separation results to our problem. To deal with this delay issue we present the sequential rate distortion framework first introduced in [18] and further developed in [31].

We consider the following discrete time, stochastic, linear system:

$$X_{t+1} = AX_t + BU_t + W_t, \quad t \geq 0 \tag{8}$$

where $\{X_t\}$ is a $\mathbb{R}^d$-valued state process, $\{U_t\}$ is a $\mathbb{R}^m$-valued control process, and $\{W_t\}$ is an IID sequence of Gaussian random variables with zero mean and covariance $K_W$. The initial position $X_0$ is Gaussian with zero mean and covariance $K_{X_0}$. Let $A \in \mathbb{R}^{d\times d}$, $B \in \mathbb{R}^{d\times m}$ and assume $(A,B)$ is a controllable pair. The decoder output process $\{\hat{X}_t\}$ is a $\mathbb{R}^d$-valued process. The output $\hat{X}_t$ represents the decoder's estimate of the state of the system at time $t$. Note that the control $U_t$ becomes a well-defined random variable only after a control law has been defined.

We examine the case where the channel is a memoryless vector Gaussian channel with power constraint $P$. Specifically the channel is: $B_t = a_t + Z_t$ where $\{Z_t\}$ is an IID sequence of Gaussian random variables with zero mean and covariance $K_Z$. The input symbols $A_t \in \mathbb{R}^d$ satisfy the power constraint: $E(\|A_t\|_2^2) \leq P, \ \forall t$. This channel is often used as a simplified model of a wireless channel. The supremizing input distribution $P(dA_t)$ can be shown to be a zero mean, vector-valued, Gaussian random variable. Hence $E(\|A_t\|_2^2) = \mathrm{tr}(K_A)$. See [10] for more details. The capacity is given by $C^{\mathrm{cap}} = \max_{\mathrm{tr}(K_A)\leq P} \frac{1}{2}\log\frac{|K_A+K_Z|}{|K_Z|}$.

Our performance objective is the LQG cost:

$$\overline{\lim_{T\to\infty}} \frac{1}{T} E\left[\sum_{t=0}^{T-1} X_t' Q X_t + U_t' S U_t\right] \tag{9}$$

where $Q$ and S are positive definite.

Under full state observation it is well known that the optimal steady state control law is a linear gain of the form $U_t = LX_t$ where $L = -(B'\Sigma B + S)^{-1}B'\Sigma A$ and $\Sigma$ satisfies the Riccati equation $\Sigma = A'(\Sigma - \Sigma B(B'\Sigma B + S)^{-1}B'\Sigma)A + Q$. The optimal cost is given by $\overline{\lim}_{T\to\infty}\frac{1}{T}E[\sum_{t=0}^{T-1} X_t'QX_t + U_t'SU_t] = E(W'\Sigma W) = \mathrm{tr}(\Sigma K_W)$.

Furthermore these results continue to hold for the LQ problem where the process disturbances $\{W_t\}$ in system (1) are no longer Gaussian but are uncorrelated and have zero mean and common covariance matrix $K_W$. That is, for separation to hold, only second order statistics are needed. These results are standard and can be found in [5].

The addition of a Gaussian communication channel converts the fully observed LQG control problem above into a partially observed LQ control problem. The structure of the partial observation is determined by both the communication channel and the choice of encoder and decoder.

We will present a condition, called the *no dual effect* [4] property, that insures the optimality of the certainty equivalent controller law: $U_t = L\hat{X}_t$. Here $\hat{X}_t = E(X_t \mid B^t, U^{t-1})$ is the decoder's estimate of the state of the plant. It will be

shown that the no dual effect property implies that we can, with loss of generality, separate the design of the controller from the encoder/decoder.

Let the state estimation error be $\Delta_t = X_t - E(X_t \mid B^t, U^{t-1})$ and let $\Lambda_t = \mathrm{cov}(\Delta_t)$ be the error covariance of the state estimate. These are well defined once an encoder, decoder, and controller have been chosen.

We know that for any control sequence $\{u_t\}$ we have $X_t = A^t X_0 + \sum_{i=0}^{t-1} A^{t-1-i}(Bu_i + W_i)$. Define $\overline{X}_t = X_t - \sum_{i=0}^{t-1} A^{t-1-i} B u_i$ to be the uncontrolled state. Similarly let $\overline{B}_t$ denote the channel output when all the controls are set to zero. Specifically for each particular realization of the primitive random variables $X_0$, $\{W_t\}$, and $\{Z_t\}$ the value $B_t$ represents the channel output under the control sequence $\{u_t\}$ and $\overline{B}_t$ represents the channel output when all the controls are set to zero. The control is said to have *no dual effect* if for all $\{u_t\}$ and all $t$ we have $E[\Delta_t \Delta_t^{'} \mid B^t, u^{t-1}] = E[\Delta_t \Delta_t^{'} \mid \overline{B}^t] \;\; P-a.s.$ The condition of no dual effect essentially states that the error covariance is independent of the control signals applied. The term "dual" comes from the control's dual role: effecting state evolution and probing the system to reduce state uncertainty. If the control has no dual effect the latter probing property will not be in effect.

**Proposition 17.** *The optimal control law for system (7) is a certainty equivalent control law if and only if the control has no dual effect.*

The proof is an extension of the work by Bar-Shalom and Tse [4]. Full details can be found in [34].

For the vector Gaussian channel we restrict ourselves to encoders that are deterministic and linear: $\{A_t = \gamma_{1t} x^t + \gamma_{2t} u^{t-1} + \gamma_{3t} a^{t-1} + \gamma_{4t} b^{t-1}\}$ where the $\gamma$'s are matrices of appropriate dimension. Some of the $\gamma$'s may be set to zero depending on the information pattern of the encoder. It can be shown that there is no loss of generality by choosing linear encoders [34].

Since the plant dynamics, the encoder, and the channel are linear we can rewrite $B_t$ in terms of the primitive random variables and the controls: $B_t = \eta_{1t} X_0 + \eta_{2t} W^{t-1} + \eta_{3t} Z^t + \eta_{4t} U^{t-1}$ where the $\eta$'s are matrices of appropriate dimension. Similarly the "uncontrolled" channel output can be written as $\overline{B}_t = \eta_{1t} X_0 + \eta_{2t} W^{t-1} + \eta_{3t} Z^t$ with the same $\eta$ matrices. Hence we can write $\overline{B}^t = B^t - \Gamma_t U^{t-1}$ for appropriate matrices $\Gamma_t$. The information in $B^t, U^{t-1}$ relevant to estimating $\overline{X}_t$ is summarized by $\overline{B}^t$. Thus for linear encoders the control has no dual effect. See lemma 5.2.1 of [5] for more details.

We now reduce the partially observed control problem to a fully observed LQ problem with no dual effect. The running cost at time $t$ can be written for every $(b^t, u^{t-1})$: $E[X_t^{'} Q X_t + U_t^{'} S U_t \mid b^t, u^{t-1}] = E[\hat{X}_t^{'} \;\; Q \;\; \hat{X}_t + U_t^{'} S U_t \mid b^t, u^{t-1}] + E[\Delta_t^{'} Q \Delta_t \mid b^t, u^{t-1}]$. Note that if the control has no dual effect the last term of the running cost, $E[\Delta_t^{'} Q \Delta_t \mid b^t, u^{t-1}]$, does not depend on $u^{t-1}$.

We define a new "fully observed" process with the decoder's estimate of the state, $\hat{X}_t$, as the new state: $\hat{X}_{t+1} = E[X_{t+1} \mid B^{t+1}, U^t] = A\hat{X}_t + BU_t + \overline{W}_t$ with "process disturbance:" $\overline{W}_t = E[A\Delta_t + W_t \mid B^{t+1}, U^t]$. Our new system has the dynamics $\hat{X}_{t+1} = A\hat{X}_t + BU_t + \overline{W}_t$. The term $\overline{W}_t$ represents the new information being transmitted from the encoder to the decoder. One can show that the $\overline{W}_t$ are uncorrelated [34]. Note that $K_{\overline{W}_t} = A\Lambda_t A' + K_W - \Lambda_{t+1}$.

Now the cost upto time $T$ can be written as $E[\sum_{t=1}^{T} X_t'QX_t + U_t'SU_t] = \sum_{t=1}^{T} E(\hat{X}_t'Q\hat{X}_t + U_t'SU_t) + \sum_{t=1}^{T} E(\Delta_t'Q\Delta_t)$. If we further assume that $\Lambda_t = \Lambda$ for all $t$ then

$$\begin{aligned}\overline{\lim_{T\to\infty}} \frac{1}{T} E\left[\sum_{t=1}^{T} X_t'QX_t + U_t'SU_t\right] &= \mathrm{tr}(\Sigma K_{\overline{W}}) + \mathrm{tr}(Q\Lambda) \\ &= \mathrm{tr}\left(\Sigma(A\Lambda A' + K_W - \Lambda)\right) + \mathrm{tr}(Q\Lambda) \\ &= \mathrm{tr}(\Sigma K_W) + \mathrm{tr}\left((A'\Sigma A - \Sigma + Q)\Lambda\right)\end{aligned}$$

We see that the optimal cost decomposes into two terms. The first term is the cost under full state observation and the second term is a cost that depends only on $\Lambda$ the steady state estimation error covariance. Thus we have reduced the problem of computing the optimal cost to that of minimizing $\mathrm{tr}((A'\Sigma A - \Sigma + Q)\Lambda)$ for a given information pattern and channel. The $\Lambda$ term can be viewed as the steady state estimation error covariance and the matrix $(A'\Sigma A - \Sigma + Q)$ can be viewed as a weighting matrix. This matrix depends on the parameters in the cost in equation (9) as well as the dynamics of the plant in system (8).

This latter problem is related to the sequential rate distortion problem. We briefly highlight the main results here. The source-channel separation principle in information theory states that we can separately design the source encoder and the channel encoder. Unfortunately traditional information theory is not causal. When delay is an issue the separation principle fails. Thus we need to design the source and channel encoders together. The *sequential rate distortion function* [18], [31] was developed to treat causality.

**Definition 2.** *The* sequential rate distortion function *is given by* $R_T^{SRD}(D) = \inf_{P\in\mathcal{F}_T^{SRD}} \frac{1}{T} I(X^T;\hat{X}^T)$ *where* $\mathcal{F}_T^{SRD} = \{\{P(d\hat{X}_t \mid x^t, \hat{x}^{t-1})\}_{t=1}^{T} \ : \ E[\|X_t - \hat{X}_t\|_M^2] \le D, \ \forall t\}$.

Here $M = A'\Sigma A - \Sigma + Q$ is a weighting matrix. The sequential rate distortion function gives a lower bound on the channel capacity needed to causally reconstruct a given source upto a given distortion. In the case when the infimizing channel in the above definition can be realized over a given channel then the lower bound is tight [31]. This occurs when the channel is *matched* [17] to the source [26].

For the source $X_{t+1} = AX_t + W_t$ (i.e. no controls) and the weight matrix $M$ the sequential rate distortion function can be computed. Specifically, in the scalar case we have $\lim_{T\to\infty} R_T^{SRD}(D) = \max\left\{0, \frac{1}{2}\log\left(A^2 + \frac{MK_W}{D}\right)\right\}$. In the vector case, for $D$ sufficiently small we have $\lim_{T\to\infty} R_T^{SRD}(D) = \frac{1}{2}\log(|M| \ |AM^{-1}A' + \frac{d}{D}K_W|)$ [31, 34].

Note that $\frac{1}{2}\log(|M| \ |AM^{-1}A' + \frac{d}{D}K_W|) \ge \sum_{\lambda(A)} \log|\lambda(A)|$. The bound becomes tight as $D \to \infty$. Hence for unstable systems there is a nonzero rate needed just to insure the distortion is bounded. This result should be compared to those in section 2. The added rate above and beyond $\sum_{\lambda(A)} \log|\lambda(A)|$ is required because the LQG performance objective (9) is more demanding than that of observability or stabilizability.

The following proposition gives a lower bound on the channel capacity needed to achieve a given LQG cost. The proof can be found in [34].

**Proposition 18.** *Given system (8), Gaussian channel, and linear encoder. To achieve an LQG cost of $tr(\Sigma K_W) + D$ a necessary condition on the channel capacity is $C^{cap} > \lim_{T\to\infty} R_T^{SRD}(D)$.*

For the scalar case, to achieve an LQG cost of $\Sigma K_W + D$ we would need a channel with capacity at least $\max\left\{0, \frac{1}{2}\log\left(A^2 + \frac{(A^2\Sigma-\Sigma+Q)K_W}{D}\right)\right\}$. Alternatively if we had access to a channel of rate $R$ then we could not achieve an LQG cost less than $\Sigma K_W + \frac{K_W\left(A^2\Sigma-\Sigma+Q\right)}{2^{2R}-A^2}$. Note that if $R < \log|A|$ then the LQG cost equals infinity. On the other hand as the channel rate $R \to \infty$ the cost converges to the full knowledge LQG cost. In the vector case, for $D$ sufficiently small, to achieve an LQG cost of $\mathrm{tr}(\Sigma K_w) + D$ we would need a channel with capacity at least $\frac{1}{2}\log(|M|\ |AM^{-1}A' + \frac{d}{D}K_W|)$.

**Proposition 19.** *Given system (8) and a Gaussian channel matched to the sequential rate distortion infimizing channel. To achieve an LQG cost of $tr(\Sigma K_W) + D$ a sufficient condition on the channel capacity is $C^{cap} > \lim_{T\to\infty} R_T^{SRD}(D)$.*

As stated before the sequential rate distortion function can be achieved over Gaussian channels with linear encoders. This occurs when the channel is *matched* [17] to the source. Specifically the sequential rate distortion infimizing channel in definition 2 can be realized over a Gaussian channel. Full details can be found in [26, 31].

## 4 Extensions

Here we present some extensions and future work.

*Equi-Memory*

In propositions 15 and 16 we assumed that there exists acknowledgement feedback from the decoder to the encoder. This allow the encoder to know what information the decoder is receiving. This property is sometimes called *equi-memory* [29]. Relaxing this assumption is in general difficult. There are, though, a few scenarios where we do not need an explicit feedback acknowledgement. We discuss two here. Both require signaling the occurrence of an erasure via the control signal $U_t$. In this way the control takes on a "dual effect:" that of satisfying the control objective and of helping the encoder/decoder estimate the state. This signalling will ensure that the encoder continues to track the decoder's estimate of the state.

Scenario 1: Here we assume that the encoder knows the control policy $K$ where $U_t = K\hat{X}_t$. We will prove using induction that the encoder can compute the decoder's estimate at each time step. At time zero the encoder knows the decoder's state estimate. Assume that at time $t-1$ the encoder knows the decoder's state estimate: $\hat{X}_{t-1}$. At time $t$ the decoder's state estimate, based on $\hat{X}_{t-1}$ and the channel message, can take one of two values depending on whether there was an erasure or not. Hence the control $U_t$ can take one of two values. The encoder, by observing $U_t$ and using its knowledge of the control law $K$, can determine whether an erasure has occurred or

not and hence can determine the decoder's estimate $\hat{X}_t$. Thus the encoder can compute the decoder's estimate at each time step.

Scenario 2: Here we assume that the controller adds signalling information, $\beta_t$, to the control signal: $U_t = K\hat{X}_t + \beta_t$. Then $X_t = (A+BK)^t X_0 - \sum_{j=0}^{t-1}(A+BK)^{t-i-j}B(Ke_j - \beta_j)$. If $\lim_{t\to\infty}\beta_t = 0$ then the sum $\lim_{t\to\infty}\sum_{j=0}^{t-1}(A+BK)^{t-i-j}B\beta_j = 0$ and hence does not effect the long term behavior of the state. Fix an integer $M$ and assume that the controller knows if an erasure has occurred or not. Let

$$\beta_t = \begin{cases} -2^{-Mt}\lfloor K\hat{X}_t \rfloor_{2^{-Mt}} & \text{if erasure} \\ -2^{-Mt}\lfloor K\hat{X}_t \rfloor_{2^{-Mt}} + 2^{-Mt}\text{ones(m)} & \text{if no erasure} \end{cases}$$

where $\lfloor K\hat{X}_t \rfloor_{2^{-Mt}}$ is a $\{0,1\}^m$-valued vector that contains the coefficient of $2^{-Mt}$ in the component-wise binary expansion of the vector $K\hat{X}_t$. Note that $\beta_t \to 0$. The controller applies the control $U_t = K\hat{X}_t + \beta_t$ to the plant. In words $U_t$ replaces the coefficient of $2^{-Mt}$ in the binary expansion of $K\hat{X}_t$ by a vector of all zeroes or all ones depending on whether there was an erasure or not. The encoder observes the control applied. Thus it can determine the coefficient of $2^{-Mt}$ in the binary expansion of $U_t$. Hence the encoder will know if an erasure has occurred or not. Thus the encoder can compute the decoder's estimate at each time step.

Neither scenario is completely satisfactory. The first case assumes the encoder knows the control policy. The second case is not robust if there is noise on the channel connecting the controller to the plant. But both cases show that the necessary conditions presented in propositions 13 and 14 are tight even for scenarios without explicit acknowledgement feedback.

*Convergence in Mean*

In this chapter we have been primarily concerned with almost sure asymptotic observability and stabilizability. Sahai, in [25] treats the case of mean-square observability. In his *any-time capacity* framework he presents channel capacity results that ensure mean-square observability. In general the capacity conditions are different under the almost sure and the mean-square convergence criteria.

Depending on the control application, one may prefer an almost sure convergence criteria or a mean-square convergence criteria. In the former one is interested in finding a channel capacity so that almost all realizations of the system's trajectories are typical. And in fact with probability one all realizations will satisfy the control objective. Atypical realizations, also called large deviations excursions, can occur but with probability approaching zero. If in addition, one wants to penalize the atypical trajectories by the size of their large deviation excursion then the mean-square formulation is appropriate. The fact that one gets different results under the almost sure convergence criteria and the mean-square convergence criteria is a generic property of the multiplicative law of large numbers [11].

*Equivalence between Feedback Stability and Feedback Capacity*

Here we describe the equivalence between feedback stabilization and feedback communication over an additive Gaussian channel with memory [13, 14, 15]. In particu-

lar a feedback control system which stabilizes an unstable plant over a communication channel, can be transformed into a communication system for the same channel, with the encoder that has access to noiseless feedback from the decoder, and viceversa. Moreover, the achievable transmission rate of the communication system with feedback is directly related to the degree of instability of the plant, and to the fundamental performance limitation of causal feedback systems, known as the Bode's integral formula.

In particular the stable feedback interconnection of an LTI SISO discrete-time unstable system $A$, with an ISI additive Gaussian channel is equivalent to a feedback communication system, which reliably (in the sense of Shannon) transmits the initial state, $x^0$, of the unstable part of $K$ at a rate $R = \sum_i \ln|\lambda_i^u(A)| - \varepsilon$, for any $\varepsilon > 0$. Moreover, the achievable transmission rate of the communication scheme is fundamentally limited by the Bode's integral formula for the Sensitivity transfer function: $\mathcal{S}$:

$$\lim_{T\to\infty} \frac{1}{T} \overrightarrow{I}(U^T \to Y^T) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \ln|\mathcal{S}(e^{j2\pi\theta})| d\theta = \sum_{i=1}^{d} \ln|\lambda_i^u(A)|.$$

where $I(U^T \to Y^T)$ is the *directed information* [32] between the control and the output. The directed information is the appropriate measure of mutual infirmation for channels with feedback.

This equivalence should enable the use of the rich tool set of design and optimization methods developed in control theory to the design of efficient communication schemes and improve (when possible) existing information theory results. One example of the use of these ideas is in the computation of the capacity of the Markov fading channel with output feedback [20, 21].

*General Networks*

This chapter has focused on control problems with one communication channel. An important challenge is to extend these ideas to more general distributed settings with many encoders and decoders communicating over a shared medium.

## References

[1] S. Axler, *Linear Algebra Done Right*, Springer-Verlag, New York, 1997.
[2] J. Baillieul, "Feedback Designs in Information-Based Control." Proceedings of the Workshop on Stochastic Theory and Control, University of Kansas, October 2001.
[3] R. Bansal and T. Basar, "Simultaneous Design of Measurement and Control Strategies for Stochastic Systems with Feedback." Automatica, Volume 25, No. 5, pp 679-694, 1989.
[4] Y. Bar-Shalom and E. Tse, "Dual effect, Certainty Equivalence, and Separation in Stochastic Control." IEEE Trans. on Automatic Control, Vol. AC-19, No. 5, October 1974, pp. 494-500.

[5] D. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 2000.

[6] V. Borkar and S. Mitter, "LQG Control with Communication Constraints." In *Communications, Computation, Control, and Signal Processing: a Tribute to Thomas Kailath*. Norwell, MA, Kluwer Academic Publishers: 1997.

[7] V. Borkar, S. Mitter, and S. Tatikonda, "Markov Control Problems Under Communication Constraints." Communications in Information and Systems (CIS), Volume 1, Number 1, pp. 15-32, January 2001.

[8] V. Borkar, S. Mitter, and S. Tatikonda, "Optimal Sequential Vector Quantization of Markov Sources." SIAM Journal on Control and Optimization, Volume 40, Number 1, pp. 135-148, January 2001.

[9] R. Brockett and D. Liberzon, "Quantized Feedback Stabilization of Linear Systems." IEEE Trans. on Automatic Control, Vol. 45, No. 7, pp. 1279-1289, July 2000.

[10] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley, New York, 1991.

[11] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*. Springer-Verlag, 1998.

[12] N. Elia and S. Mitter, "Stabilization of Linear Systems with Limited Information." IEEE Trans. on Automatic Control, Vol. 46, No. 9, pp. 1384-1400, September 2001.

[13] N. Elia, "Design and Analysis of Communication Systems with access to Feedback: A Control Theory Approach." Proceedings of the 41$^{st}$ Allerton Conference on Communication, Control, and Computing, 2003.

[14] N. Elia, "Control oriented Feedback Communication Schemes." Proceedings of 42$^{nd}$ IEEE Conference on Decision and Control, Maui 2003.

[15] N. Elia, "When Bode meets Shannon: Control Oriented Feedback Communication Schemes." Accepted to IEEE Transactions on Automatic Control Special Issue on Networked Control, 2004.

[16] R. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[17] M. Gastpar, B. Rimoldi, and M. Vetterli. "To Code, or not to Code: Lossy Source-Channel Communication Revisited." IEEE Trans. on Information Theory, Vol. 49, No. 5, pp. 1147-1158, May 2003.

[18] A. Gorbunov and M. Pinsker, "Prognostic Epsilon Entropy of a Gaussian Message and a Gaussian Source." Translation from Problemy Peredachi Informatsii, Vol. 10, No. 2, pp. 5-25, April-June 1973.

[19] T. Kailath, *Linear Systems*. Prentice Hall, Information and Systems Science Series, Englewood Cliffs N.J. 1980.

[20] J. Liu, N. Elia, and S. Tatikonda, "Capacity Achieving Feedback Scheme for Markov Channels with Channel State Information." Submitted to IEEE Transactions on Information Theory, 2003.

[21] J. Liu, N. Elia, and S. Tatikonda, "Capacity Achieving Feedback Scheme for Flat Fading Channels with Channel State Information." To appear in IEEE American Control Conference, 2004.

[22] R. Murray, editor., "Control in an Information Rich World: Report of the Panel on Future Directions in Control, Dynamics, and Systems." SIAM, June 30, 2002. Also available at: http://www.cds.caltech.edu/%7Emurray/cdspanel/
[23] G. Nair and R. Evans, "Stabilization with Data-Rate-Limited Feedback: Tightest Attainable Bounds." Systems and Control Letters, Vol. 41, pp. 49-76, 2000.
[24] G. Nair and R. Evans, "Exponential Stabilisability of Finite-Dimensional Linear Systems with Limited Data Rates." Automatica, Vol. 39, pp. 585-593, 2003.
[25] A. Sahai, "Anytime Information Theory." MIT Ph.D. thesis, February 2001.
[26] S. Tatikonda, "Control Under Communication Constraints." MIT Ph.D. thesis, August 2000.
[27] S. Tatikonda, "Some Scaling Properties of Large Distributed Control Systems." Proceedings of the 42nd IEEE Conference on Decision and Control, December 2003.
[28] S. Tatikonda and S. Mitter, "Control Under Communication Constraints." Proceedings of the 38th Annual Allerton Conference on Communication, Control, and Computing, October 2000.
[29] S. Tatikonda and S. Mitter, "Control Under Communication Constraints." Accepted to the IEEE Trans. on Automatic Control, February 2004.
[30] S. Tatikonda and S. Mitter, "Control Over Noisy Channels." Accepted to the IEEE Trans. on Automatic Control, February 2004.
[31] S. Tatikonda and S. Mitter, "The Sequential Rate Distortion Function." Submitted to IEEE Transactions on Information Theory, April 2004.
[32] S. Tatikonda and S. Mitter, "The Capacity of Channels with Feedback." Submitted to IEEE Transactions on Information Theory, December 2001.
[33] S. Tatikonda, A. Sahai, and S. Mitter, "Control of LQG Systems Under Communication Constraints." Proceedings of the 37th IEEE Conference on Decision and Control, Tampa, Florida, December 1998.
[34] S. Tatikonda, A. Sahai, and S. Mitter, "Stochastic Linear Control with a Communication Channel." Accepted to the IEEE Transactions on Automatic Control Special Issue on Networked Control, 2004.
[35] S. Vembu, S. Verdu, and Y. Steinberg, "The Source-Channel Separation Theorem Revisited," IEEE Trans. Info. Theory, vol. IT-41, pp. 44 - 54, January 1995.
[36] H. Witsenhausen, "Separation of Estimation and Control for Discrete Time Systems." Proceedings of the IEEE, Volume 59, No. 11, November 1971.
[37] W. Wong and R. Brockett, "Systems with Finite Communication Bandwidth Constraints I: State Estimation Problems," IEEE Trans. on Automatic Control, Vol. 42(9), pp. 1294-1299, Sept. 1997.
[38] W. Wong and R. Brockett, "Systems with Finite Communication Bandwidth Constraints II: Stabilization with Limited Information Feedback," IEEE Trans. on Automatic Control, Vol. 44 (5), pp. 1049 -1053, May 1999.

# An Introduction to Nonlinear Fault Diagnosis with an Application to a Congested Internet Router*

Michel Fliess[1], Cédric Join[2], and Hugues Mounier[3]

[1] Laboratoire STIX, École polytechnique, 91128 Palaiseau, France.
michel.fliess@stix.polytechnique.fr
[2] Centre de Recherche en Automatique de Nancy (CRAN), Université Henri Poincaré (Nancy I), BP 239, 54506 Vandœuvre-lés-Nancy, France.
cedric.join@cran.uhp-nancy.fr
[3] Institut d'Électronique Fondamentale (IEF), Bât. 220, Université Paris-Sud, 91405 Orsay, France. Hugues.Mounier@ief.u-psud.fr

## 1 Introduction

We aim at extending a recent linear theory on fault diagnosis [8, 9] to nonlinear systems by utilising, like Staroswiecki *et al.* [39], and Diop *et al.* [4], techniques stemming from differential algebra. Recall that differential algebra, which is mainly due to [35] and [29], was introduced in control by one of us [5, 6]. Besides clarifying several structural questions, differential algebra helped in the discovery [10] of *(differential) flatness*, which is immensely useful in concrete applications (see [37]). Here differential algebra yields clear-cut and straightforward definitions of key concepts such as *detectability*, *isolability*, and *parity equations* for fault variables, which are mimicking [9]. See [3, 18, 25, 26, 27, 28] for other approaches of nonlinear fault diagnosis.

Contrarily to the linear case [9], the derivatives of the control and sensor variables in the fault "indicators", *i.e.*, the *residuals*, cannot be removed via integrations. As done by Sira-Ramírez and one of us (see [15, 16, 38]) with state variable estimation, we are obtaining good approximations of derivatives of time signals, even when they are noisy. This permits to consider parity equations as efficient residuals.

As a concrete illustration we are considering a simplified model [22] of a congested or heavily loaded internet router. Detected defaults are abnormal queue length variation (at the router) and additional packet loss. Our algorithm yields fast computations and can easily be embedded in existing TCP protocols[4].

* Work partially supported by by the *action spécifique* "*Méthodes algbriques pour les systmes de communication numriques*" (CNRS, RTP 24, France). One author (CJ) is acknowledging the hospitality of the *Laboratoire STIX* at the *École polytechnique*.

[4] Note that our estimation techniques, which are stemming from [14], have been most successfully applied in signal processing (see, *e.g.*, [13]).

S. Tarbouriech et al. (Eds.): Advances in Communication Control Networks, LNCIS 308, pp. 327–343, 2004.

Our paper is organised as follows. After recalling in section 2 basic definitions of nonlinear systems in the differential algebraic setting, section 3 is devoted to the introduction of key properties of fault variables. A brief summary of our estimation techniques is provided in section 4. The illustrative example with convincing computer simulations is developed in section 5. Some directions for future works are listed in the conclusion.

## 2 Differential fields[5]

### 2.1 Basic definitions

A *differential ring* $R$ (see, *e.g.*, [29] and [1]) will be here a commutative ring[6] of characteristic zero, which is equipped with a single *derivation* $\frac{d}{dt} : R \to R$ such that, $\forall\, a, b \in R$,

- $\frac{d}{dt}(a+b) = \dot{a} + \dot{b}$,
- $\frac{d}{dt}(ab) = \dot{a}b + a\dot{b}$.

where $\frac{da}{dt} = \dot{a}$, $\frac{d^{\nu}a}{dt^{\nu}} = a^{(\nu)}$, $\nu \geq 0$. A *differential field* is a differential ring which is a field. A *constant* of $R$ is an element $c \in R$ such that $\dot{c} = 0$. A *(differential) ring* (resp. *field*) *of constants* is a differential ring (resp. field) which only contains constants. The set of all constant elements of $R$ is a subring (resp. subfield), which is called the *subring* (resp.*subfield*) *of constants*.

A *differential ring* (resp. *field*) *extension* is given by two differential rings (resp. fields) $R_1$, $R_2$, such that

- $R_1 \subseteq R_2$,
- the derivation of $R_1$ is the restriction to $R_1$ of the derivation of $R_2$.

**Notation** *If $R_1$ and $R_2$ are differential fields, the corresponding field extension is often written $R_2/R_1$.*

**Notation** *Let $S$ be a subset of $R_2$. Write $R_1\{S\}$ (resp. $R_1\langle S\rangle$) the differential subring (resp. subfield) of $R_2$ generated by $R_1$ and $S$.*

A *differential ideal* $\mathfrak{I}$ of $R$ is an ideal which is also a differential subring. It is said to be *prime* if, and only if, $\mathfrak{I}$ is prime in the usual sense.

### 2.2 Field extensions

The differential field extension $L/K$ is said to be *finitely generated* if, and only if, there exists a finite subset $S \subset L$ such that $L = K\langle S\rangle$. An element of $L$ is said to be *differentially algebraic* over $K$ if, and only if, it satisfies an algebraic differential

[5] See [2] for a most readable introduction to the subject. We follow here quite closely [10].

[6] See, *e.g.*, [23] for basic notions in commutative algebra.

equation with coefficients in $K$. It is said to be *differentially transcendental* if, and only if, it is not differentially algebraic. The extension $L/K$ is said to be *differentially algebraic* if, and only if, any element of $L$ is differentially algebraic over $K$. An extension which is not differentially algebraic is said to be *differentially transcendental*.

The following result is playing an important rôle:

**Proposition 1.** *A finitely generated differential extension $L/K$ is differentially algebraic if, and only if, its transcendence degree is finite.*

A set $\{\xi_\iota \mid \iota \in I\}$ of elements in $L$ is said to be *differentially algebraically independent over $K$* if, and only if, the set $\{\xi_\iota^{(\nu)} \mid \iota \in I, \nu \geq 0\}$ of derivatives of any order is algebraically independent over $K$. If a set is not differentially algebraically independent over $K$, it is *differentially algebraically dependent over $K$*. An independent set which is maximal with respect to inclusion is called a *differential transcendence basis*. The cardinalities of two such bases are equal. This cardinality is the *differential transcendence degree* of the extension $L/K$. Note that this degree is 0 if, and only if, $L/K$ is differentially algebraic.

### 2.3 Kähler differentials

The analogue of differential calculus in commutative algebra (see, *e.g.*, [21, 23]) has been extended to differential algebra in [24]. Consider again the extension $L/K$. Denote by

- $L[\frac{d}{dt}]$ the set of linear differential operators $\sum_{\text{finite}} a_\alpha \frac{d^\alpha}{dt^\alpha}$, $a_\alpha \in L$, which is a left and right principal ideal ring (see, *e.g.*, [32]);
- $\Omega_{L/K}$ the left $L[\frac{d}{dt}]$-module of Kähler differentials of the extension $L/K$;
- $d_{L/K}x \in \Omega_{L/K}$ the (Kähler) differential of $x \in L$.

**Proposition 2.** *The next two properties are equivalent:*

1. *The set $\{x_\iota \mid \iota \in I\} \subset L$ is differentially algebraically dependent.*
2. *The set $\{d_{L/K}x_\iota \mid \iota \in I\}$ is $L[\frac{d}{dt}]$-linearly dependent.*

The next corollary is a direct consequence from Propositions 1 and 2.

**Corollary 1.** *The module $\Omega_{L/K}$ satisfies the following properties:*

- *The* rank *[32] of $\Omega_{L/K}$ is equal to the differential transcendence degree of $L/K$.*
- *$\Omega_{L/K}$ is* torsion[7] *if, and only if, $L/K$ is differentially algebraic.*
- $\dim_L(\Omega_{L/K}) = \operatorname{tr\,d}^\circ(L/K)$. *It is therefore finite if, and only if, $L/K$ is differentially algebraic.*
- *$\Omega_{L/K} = \{0\}$ if, and only if, $L/K$ is algebraic.*

[7] See, *e.g.*, [32].

### 2.4 Nonlinear systems

Let $k$ be a given differential ground field. A *(nonlinear) system* is a finitely generated differential extension $K/k$. Set $K = k\langle S, \pi, \mathbf{W}\rangle$ where[8]

1. $S$ is a finite set of system variables,
2. $\pi = (\pi_1, \ldots, \pi_r)$ denotes the *perturbation*, or *disturbance*, variables,
3. $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_q)$ denotes the *fault* variables.

They satisfy the following properties:

- The perturbation and fault variables do not "interact", *i.e.*, the differential extensions $k\langle\pi\rangle/k$ and $k\langle\mathbf{W}\rangle/k$ are *linearly disjoint*[9].
- The fault variables are assumed to be *independent*, *i.e.*, $\mathbf{W}$ is a differential transcendence basis of $k\langle\mathbf{W}\rangle/k$.
- $k\{S, \mathbf{W}\} \cap (\pi) = \{0\}$, where the differential ideal $(\pi) \subset k\{S, \pi, \mathbf{W}\}$ generated by $\pi$ is prime. Write $k\{S^{\text{nom}}, \mathbf{W}^{\text{nom}}\} = k\{S, \pi, \mathbf{W}\}/(\pi)$, where the *nominal* system and fault variables $S^{\text{nom}}$, $\mathbf{W}^{\text{nom}}$ are the canonical images of $S$, $\mathbf{W}$. To those nominal variables corresponds the *nominal system* $K^{\text{nom}}/k$, where $K^{\text{nom}} = k\langle S^{\text{nom}}, \mathbf{W}^{\text{nom}}\rangle$ is the quotient field of $k\{S^{\text{nom}}, \mathbf{W}^{\text{nom}}\}$.
- $k\{S^{\text{nom}}\} \cap (\mathbf{W}^{\text{nom}}) = \{0\}$, where the differential ideal $(\mathbf{W}^{\text{nom}}) \subset k\{S^{\text{nom}}, \mathbf{W}^{\text{nom}}\}$ generated by $\mathbf{W}^{\text{nom}}$ is prime. Write

$$k\{S^{\text{pure}}\} = k\{S^{\text{nom}}, \mathbf{W}^{\text{nom}}\}/(\mathbf{W}^{\text{nom}})$$

  where the *pure* system variables $S^{\text{pure}}$ are the canonical images of $S^{\text{nom}}$. To those pure variables corresponds the *pure system* $K^{\text{pure}}/k$, where $K^{\text{pure}} = k\langle S^{\text{pure}}\rangle$ is the quotient field of $k\{S^{\text{pure}}\}$.

A *dynamics* is a system where a finite subset $\mathbf{u} = (u_1, \ldots, u_m) \subset S$ of *control* variables has been distinguished, such that the extension $K^{\text{pure}}/k\langle\mathbf{u}^{\text{pure}}\rangle$ is differentially algebraic. The control variables verify the next two properties:

- they do not interact with the fault variables, *i.e.*, the fields $k\langle\mathbf{u}^{\mathbf{nom}}\rangle$ and $k\langle\mathbf{W}^{\mathbf{nom}}\rangle$ are linearly disjoint over $k$.
- they are *independent*, *i.e.*, the components of $\mathbf{u}$ are differentially algebraically independent over $k$.

An *input-output system* is a dynamics where a finite subset $\mathbf{y} = (y_1, \ldots, y_p) \subset S$ of output variables has been distinguished. Only input-output systems will be considered in the sequel.

[8] The distinction below between these three types of variables is a direct generalisation of [9].

[9] See, *e.g.*, [23].

### 2.5 Variational system

Call $\Omega_{K/k}$ (resp. $\Omega_{K^{\text{nom}}/k}$, $\Omega_{K^{\text{pure}}/k}$) the *variational*, or *linearized*, *system* (resp. *nominal system*, *pure system*) of system $K/k$. Proposition 2 yields for pure systems

$$A \begin{pmatrix} d_{K^{\text{pure}}/k} y_1^{\text{pure}} \\ \vdots \\ d_{K^{\text{pure}}/k} y_p^{\text{pure}} \end{pmatrix} = B \begin{pmatrix} d_{K^{\text{pure}}/k} u_1^{\text{pure}} \\ \vdots \\ d_{K^{\text{pure}}/k} u_m^{\text{pure}} \end{pmatrix} \tag{1}$$

where

- $A \in K[\frac{d}{dt}]^{p\times p}$ is full rank,
- $B \in K[\frac{d}{dt}]^{p\times m}$.

The *pure transfer matrix*[10] is the matrix $A^{-1}B \in K(s)^{p\times m}$, where $K(s)$, $s = \frac{d}{dt}$, is the skew quotient field[11] of $K[\frac{d}{dt}]$.

## 3 Fundamental properties of fault variables

### 3.1 Detectability

The fault variable $\mathbf{w}_\mathfrak{l}$, $\mathfrak{l} = 1, \ldots, q$, is said to be *detectable* if, and only if, the field extension $K^{\text{nom}}/k\langle \mathbf{u}^{\text{nom}}, \mathbf{W}_\mathfrak{l}^{\text{nom}} \rangle$, where $\mathbf{W}_\mathfrak{l}^{\text{nom}} = \mathbf{W}^{\text{nom}} \backslash \{\mathbf{w}_\mathfrak{l}^{\text{nom}}\}$, is differentially transcendental. It means that $\mathbf{w}_\mathfrak{l}$ is indeed "influencing" the output. When considering the variational nominal system, formula (1) yields

$$\begin{pmatrix} d_{K^{\text{nom}}/k} y_1^{\text{nom}} \\ \vdots \\ d_{K^{\text{nom}}/k} y_p^{\text{nom}} \end{pmatrix} = T_{\mathbf{u}} \begin{pmatrix} d_{K^{\text{nom}}/k} u_1^{\text{nom}} \\ \vdots \\ d_{K^{\text{nom}}/k} u_m^{\text{nom}} \end{pmatrix} + T_{\mathbf{W}} \begin{pmatrix} d_{K^{\text{nom}}/k} \mathbf{w}_1^{\text{nom}} \\ \vdots \\ d_{K^{\text{nom}}/k} \mathbf{w}_q^{\text{nom}} \end{pmatrix}$$

where $T_{\mathbf{u}} \in K(s)^{p\times m}$, $T_{\mathbf{W}} \in K(s)^{p\times q}$. Call $T_{\mathbf{W}}$ the *fault transfer matrix*. The next result is clear:

**Proposition 3.** *The fault variable $\mathbf{w}_\mathfrak{l}$ is detectable if, and only if, the $\mathfrak{l}^{th}$ column of the fault transfer matrix $T_{\mathbf{W}}$ is non-zero.*

### 3.2 Isolability and parity equations

A subset $\mathbf{W}' = (\mathbf{w}_{\mathfrak{l}_1}, \ldots, \mathbf{w}_{\mathfrak{l}_{q'}})$ of the set $\mathbf{W}$ of fault variables is said to be *isolable* if, and only if, the extension $k\langle \mathbf{u}^{\text{nom}}, \mathbf{y}^{\text{nom}}, \mathbf{W}'^{\text{nom}} \rangle / k\langle \mathbf{u}^{\text{nom}}, \mathbf{y}^{\text{nom}} \rangle$ is differentially algebraic. It means that any component of $\mathbf{W}'^{\text{nom}}$ satisfies a *parity equation*, *i.e.*, an algebraic differential equations where the coefficients belong to $k\langle \mathbf{u}^{\text{nom}}, \mathbf{y}^{\text{nom}} \rangle$.

[10] See [7] for more details on transfer matrices of time-varying linear systems.
[11] See, *e.g.*, [32].

## 4 Residuals and nonlinear estimation

### 4.1 From parity equations to residuals

Any parity equation contains according to section 3.2 derivatives of control and sensor variables up to some finite order. We will be able to consider such a parity equation as a suitable fault "indicator", *i.e.*, a *residual*, if those derivatives are accurately estimated in a noisy environment. This has been achieved in [15, 16, 38] and will be briefly recalled below.

### 4.2 Estimation of the derivatives[12]

Consider a real-valued time function $x(t)$ which is assumed to be analytic on some interval $t_1 \leq t \leq t_2$. Assume for simplicity's sake that $x(t)$ is analytic around $t = 0$ and introduce its truncated Taylor expansion

$$x(t) = \sum_{\nu=0}^{N} x^{(\nu)}(0)\frac{t^\nu}{\nu!} + O(t^{\nu+1})$$

Approximate $x(t)$ in the interval $(0,\varepsilon)$, $\varepsilon > 0$, by a polynomial $x_N(t) = \sum_{\nu=0}^{N} x^{(\nu)}(0)\frac{t^\nu}{\nu!}$ of degree $N$. The usual rules of symbolic calculus in Schwartz's distributions theory [36] yield

$$x_N^{(N+1)}(t) = x(0)\delta^{(N)} + \dot{x}(0)\delta^{(N-1)} + \cdots + x^{(N)}(0)\delta$$

where $\delta$ is the Dirac measure at 0. From $t\delta = 0$, $t\delta^{(\alpha)} = -\alpha\delta^{(\alpha-1)}$, $\alpha \geq 1$, we obtain the following triangular system of linear equations for determining estimated values $[x^{(\nu)}(0)]_e$ of the derivatives[13] $x^{(\nu)}(0)$:

$$t^\alpha x^{(N+1)}(t) = t^\alpha \left( [x(0)]_e\delta^{(N)} + [\dot{x}(0)]_e\delta^{(N-1)} + \cdots + [x^{(N)}(0)]_e\delta \right) \qquad \alpha = 0,\ldots,N \tag{2}$$

The time derivatives of $x(t)$ and the Dirac measures and its derivatives are removed by integrating with respect to time both sides of equation (2) at least $N$ times:

$$\int^{(\nu)} \tau_1^\alpha x^{(N+1)}(\tau_1) = \int^{(\nu)} \tau_1^\alpha \left( [x(0)]_e\delta^{(N)} + [\dot{x}(0)]_e\delta^{(N-1)} + \cdots + [x^{(N)}(0)]_e\delta \right) \qquad \nu \geq N,\ \alpha = 0,\ldots,N \tag{3}$$

where $\int^{(\nu)} = \int_0^t \int_0^{\tau_{\nu-1}} \cdots \int_0^{\tau_1}$. A quite accurate value of the estimates may be obtained with a small time window $[0,t]$. Those iterated integrals are moreover low pass filters. They are attenuating high frequency noises, which are usually dealt with in a statistical setting.

[12] Estimating the derivatives of a given noisy time signal up to some finite order is a classic problem of applied analysis (see, *e.g.*, [30]), which still is actively investigated.

[13] Those quantities are *linearly identifiable* [14, 15].

# 5 Application to a simplified model of an internet router

## 5.1 Introduction

Accurate modelling of network behaviour, as well as efficient and QoS aware congestion control methods is still an active field of research [17, 19, 31, 33]. The advocated diagnosis method could allow the detection of abnormal increasing of data in queue of a router.
In order to illustrate the advantage of our technique to fault diagnose, we use the fluid-flow model developed in [33] for modelling TCP flows and active queue management (AQM) controller at a congested router.

## 5.2 TCP/RED model

Note that RED is a usual mechanism to control traffic, potentially solving problems like flow synchronisation, correlation of drop events while providing consistently high link utilisation. According to some assumptions, the differential equations modelling the interaction of a set of TCP flows and AQM routers in a network are given by (see [22, 33, 34] for more details):

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)W(t-R(t)}{2R(t-R(t))}P(t-R(t)) \tag{4a}$$

$$\dot{Q}(t) = \frac{W(t)}{R(t)}N(t) - C \tag{4b}$$

where

$W(t)$ is the length of TCP window,
$R(t)$ is the round trip time,
$Q(t)$ is the queue length at the heaviest loaded node,
$P(t)$ is the packet discard function,
$N(t)$ is the number of connections at the bottleneck node,
$C$ is the average link capacity,
$R(t)$ is related to the queue length through

$$R(t) = T + \frac{Q(t)}{C}$$

$T$ is the round trip propagation time.

Assume that

- the round trip time is dominated by the propagation delay $R$, which is then supposed to be constant,
- the window size is sufficiently large, *i.e.*, $W \gg 1$,
- the number of TCP flows varies only slowly, *i.e.*, $N$ is supposed to be a constant.

Equation (4) may be simplified and yields

$$\dot{W}(t) = \frac{1}{R} - \frac{(W(t))^2}{2R} P(t-R) \tag{5a}$$

$$\dot{Q}(t) = \frac{W(t)}{R} N - C \tag{5b}$$

*Remark 1.* The nonlinear delay system (5) may be easily reduced to the formalism of sections 2 and 3 by setting $u(t) = P(t-R)$.

### 5.3 Fault description

In this chapter two types of fault are considered.

#### Anomaly of TCP packets treatment

Equation (5) becomes

$$\dot{W}(t) = \frac{1}{R} - \frac{(W(t))^2}{2R} P(t-R) \tag{6a}$$

$$\dot{Q}(t) = \frac{W(t)}{R} N - C + F(t) \tag{6b}$$

where $F(t)$, $F(t) \geq 0$, is the fault signal, which stands for an abnormal decreasing of TCP packets treatment.

#### Abnormal packet losses

Another way of rendering packet losses would be to add an external loss source $E(t)$ to equation (5a)

$$\dot{W}(t) = \frac{1}{R} - \frac{(W(t))^2}{2R} (P(t-R) + E(t)) \tag{7a}$$

$$\dot{Q}(t) = \frac{W(t)}{R} N - C \tag{7b}$$

### 5.4 Fault diagnosis

According to equation (6b) the fault signal satisfies:

$$F(t) = \dot{Q}(t) - \frac{W(t)}{R} N + C$$

In a classical way, $W(t)$, $Q(t)$, $R$, $N$ and $C$ are assumed to be known (or measured). Thus we only need the time derivative of $Q(t)$ for the fault estimate.

*Remark 2.* The queue length $Q(t)$ is only available, strictly speaking, after a delay $\tau$ (the delay from the congested or heavily loaded router back to the source). This variable could also be well estimated using the techniques developed in [15, 16, 38].

Consider, following section 4, the approximation

$$Q(t) = a_0 + a_1 t + a_2 \frac{t^2}{2} + a_3 \frac{t^3}{6} + O(t^4)$$

which yields the estimate

$$[\dot{Q}(t)]_e = a_1 + a_2 t + a_3 \frac{t^2}{2}$$

The unknown coefficients are given by

$$a_1 = \frac{24}{t^4}\left(\frac{1}{2}t^3 Q(t) - \frac{11}{2}\int^{(1)}\left(t^2 Q(t)\right) + 14\int^{(2)}(tQ(t)) - 6\int^{(3)}(Q(t))\right) \tag{8a}$$

$$a_2 = \frac{120}{t^5}\left(-\frac{1}{2}t^3 Q(t) + 5\int^{(1)}\left(t^2 Q(t)\right) - 11\int^{(2)}(tQ(t)) + 4\int^{(3)}(Q(t))\right) \tag{8b}$$

$$a_3 = \frac{720}{t^6}\left(\frac{1}{6}t^3 Q(t) - \frac{3}{2}\int^{(1)}\left(t^2 Q(t)\right) + 3\int^{(2)}(tQ(t)) - \int^{(3)}(Q(t))\right) \tag{8c}$$

The time derivative of $Q(t)$ is then approximated as follows:

$$[\dot{Q}(t)]_e = \frac{1}{t^4}\left(12t^3 Q(t) - 72\int^{(1)}\left(t^2 Q(t)\right) + 96\int^{(2)}(tQ(t)) - 24\int^{(3)}(Q(t))\right)$$

This estimation is periodically reset in order to track the variations of $\dot{Q}(t)$. A convenient residual is given by

$$[F(t)]_e = [\dot{Q}(t)]_e - \frac{W(t)}{R}N + C$$

It is largely different from zero when a fault occurs.

*Remark 3.* The external source $E(t)$ could be jointly estimated with $F(t)$ using the same techniques

$$[E(t)]_e = \frac{2 - 2R[\dot{W}(t)]_e - (W(t))^2 P(t-R)}{(W(t))^2}$$

### 5.5 Numerical simulations

As usual, an active queue management is associated with the router. A classical example of an AQM policy is the Random Early Discard (RED). Then $P(t-R) = p(q(t-R))$ where

$$p(x(t)) = \begin{cases} 0, & 0 \leqslant x(t) < T_{min} \\ \dfrac{x(t) - T_{min}}{T_{max} - T_{min}} P_{max}, & T_{min} \leqslant x(t) \leqslant T_{max} \\ 1, & T_{max} < x(t) \end{cases}$$

In what follows, the input RED control is denoted by $u(t)$. For the numerical simulations we took a bandwidth capacity of 15 Mb/s and a packet size of 500 Bytes.

**Congestion-free case**

*Simulation parameters:*
$N = 40$ *number of TCP connections;*
$C = 3750$ *Packets/s link capacity;*
$T_{min} = 0$*;*
$T_{max} = 200$*;*
$P_{max} = 1$*;*
*A zero mean random variable with a variance of value 1 packet is added on the measured output* $Q(t)$*. This noise affects the feedback control input which, evidently, also depends on* $Q(t)$*.*

Two cases are presented:

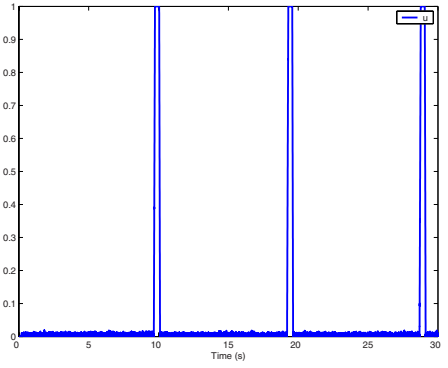
- Long distance communication (Europe $\leftrightarrow$ United States):
  The round trip time is firstly taken as $R = 200$ms. Simulations are presented in Figure 1. The router exhibits a periodical behaviour. Due to the delay $R$, no operating point is reached.

- Medium distance communication (Europe $\leftrightarrow$ Europe):
  In the remaining figures, the round trip time is taken as $R = 20$ms. Simulations presented in Figure 2 show that an equilibrium point is reached. The TCP window and the queue length become constant after a while.

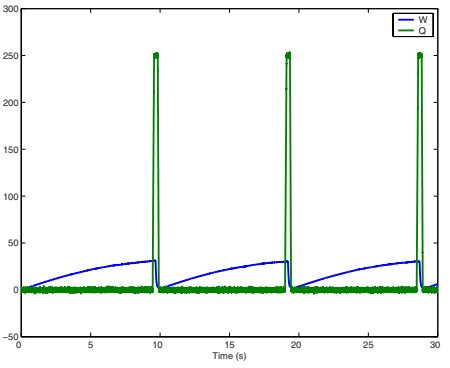
**Fault diagnosis**

Our results on fault isolation are illustrated in two scenarios.

**Experiment 1:** The fault $F$ (described in equation (6)) occurs at time $t = 7.5$ s and its magnitude is approximatively 20% of $Q(t)$. Its estimation is presented in figure 3. The behaviour of the estimation enables the distinction between the fault free case (see 3-(c)) and the faulty case (see 3-(d)).




*(a)* Input RED control




*(b)* Outputs $W(t)$ (sawtooth plot) and $Q(t)$ (flat with peaks plot)

**Fig. 1.** Simulation results when delay $R$ is quite large

**Experiment 2:** The fault $E$ (described in equation (7)) occurs at time $t = 7.5$ s and its magnitude is approximatively 20% of $u(t) = P(t - R)$. The fault estimation $[E(t)]_e$ permits (compare 4-(a) and 4-(b)) to detect the fault.

### Unknown input estimation

In figure 5, the estimation of the number $N$ of connection is also done online. Naturally the fault $F$ biases the estimation (see 5-(c)). In this case, the fault occurrence is interpreted as new TCP connections.

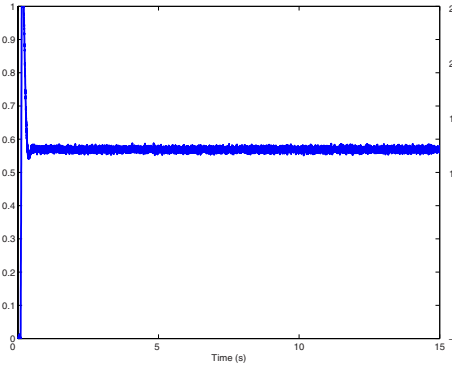
Let now the TCP connection number changes as follows

$$N(t) = \begin{cases} 40, & 0s < t < 4.5s \\ 45, & 4.5s \leqslant t \leqslant 9s \\ 35, & 9s < t \leqslant 15s \end{cases}$$

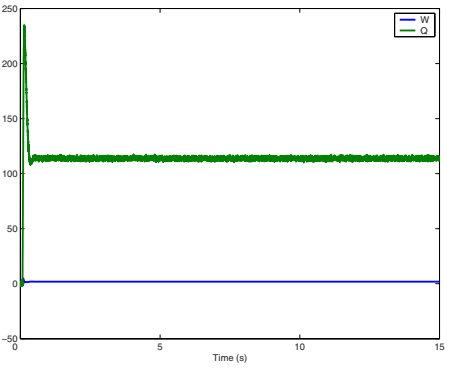
Router and estimation behaviour are shown in figure 6. Each change of $N(t)$ results in a new equilibrium point. The estimation of $N(t)$ is done quickly and with a good accuracy.

## 6 Conclusion

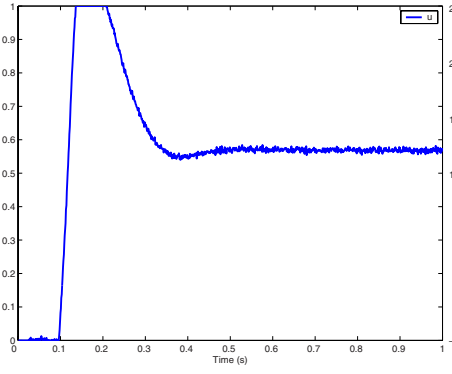
A forthcoming publication will provide a complete solution of the most celebrated case-study in nonlinear fault diagnosis, namely the three-tank problem (see [40, 41,



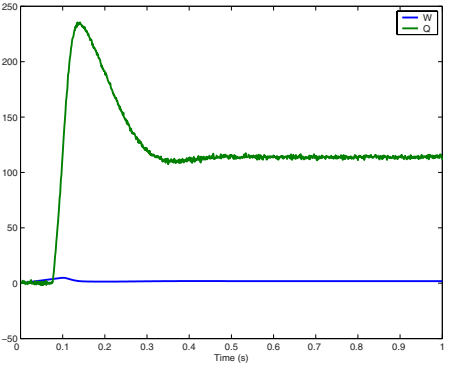

*(a)* Input RED control




*(b)* Outputs $W(t)$ (bottom) and $Q(t)$ (top)



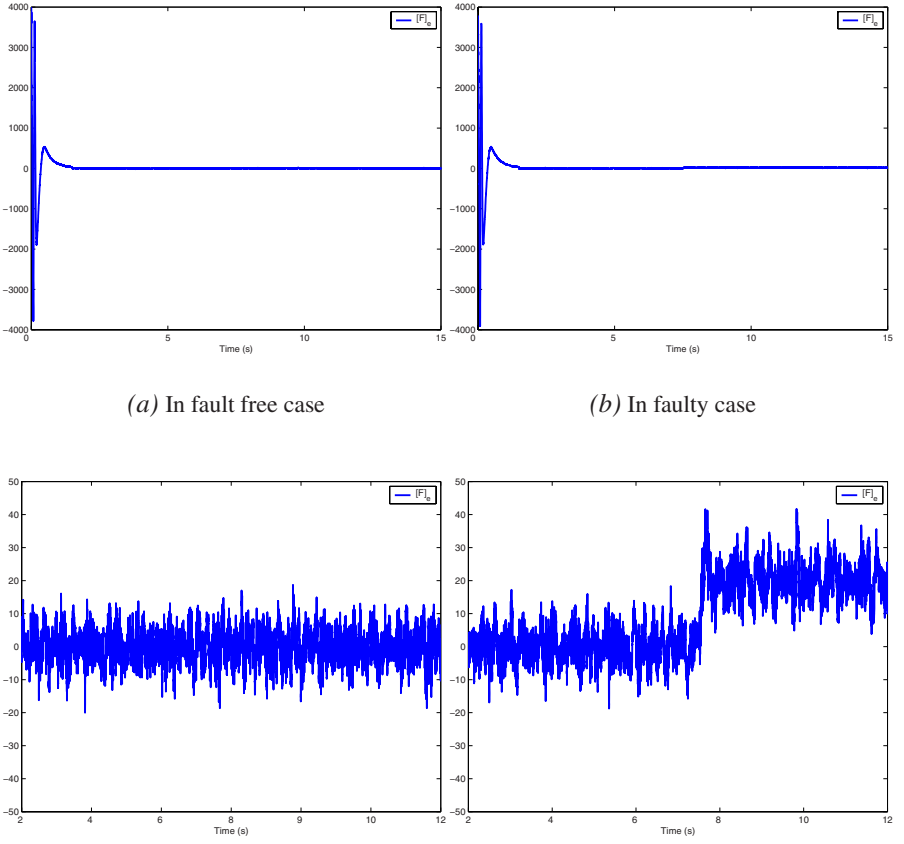

*(c)* Zoom on input RED control




*(d)* Zoom on outputs $W(t)$ (bottom) and $Q(t)$ (top)

**Fig. 2.** Case where the delay $R$ is smaller

42]). Fault diagnosis techniques are particularly well suited to cases where physics is yielding natural conservation laws. We obtain in the context of internet routers the conservation of packet numbers, *i.e.*, the analogue of fluid volume conservation. This feature is quite apparent in [20] and the same techniques as above could be applied with great benefit.




*(a)* In fault free case *(b)* In faulty case

*(c)* Zoom on (a) *(d)* Zoom on (b)

**Fig. 3.** Fault estimation

## References

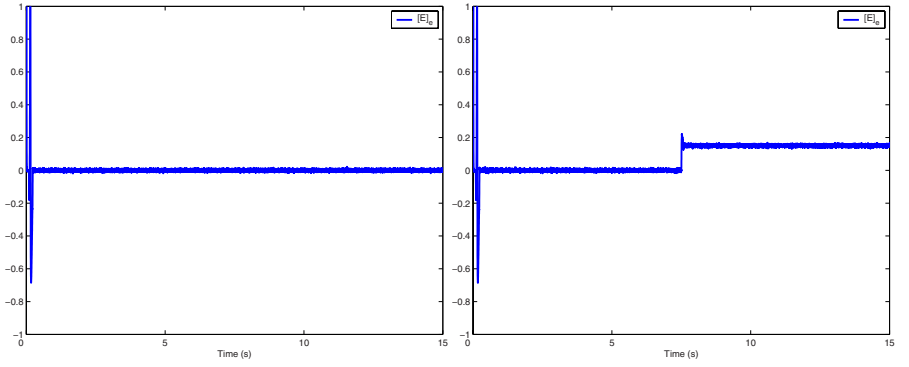
[1] A. Buium, *Differential Algebra and Diophantine Geometry*, Hermann, Paris, 1994.

[2] E. Delaleau, *Algèbre différentielle, in* Mathématiques pour les Systèmes Dynamiques, J.P. Richard Ed., vol. 2, chap. 6, pp. 245-268, Hermès, Paris, 2002.

[3] J. Chen, R. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer, Boston, 1999.

[4] S. Diop and R. Martinez-Guerra, An algebraic and data derivative information approach to nonlinear system diagnosis, *Proc. 6th Europ. Control Conf.*, Porto, 2001.

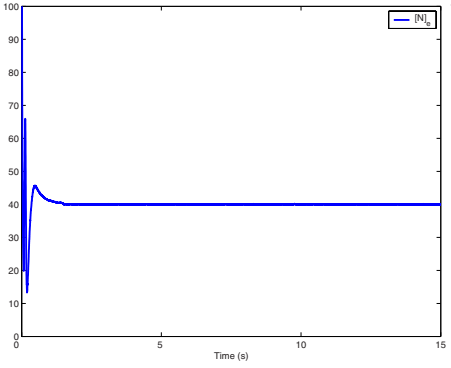
[5] M. Fliess, Automatique et corps différentiels, *Forum Math.*, **1**, 1989, 227-238.



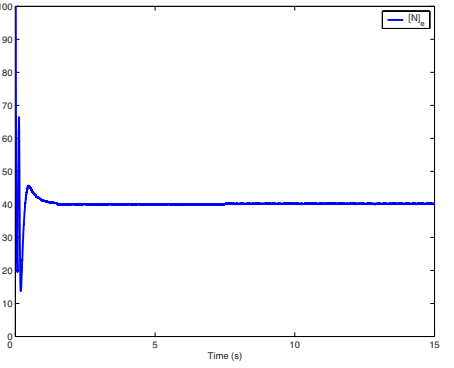

*(a)* In fault free case *(b)* In faulty case

**Fig. 4.** Fault estimation

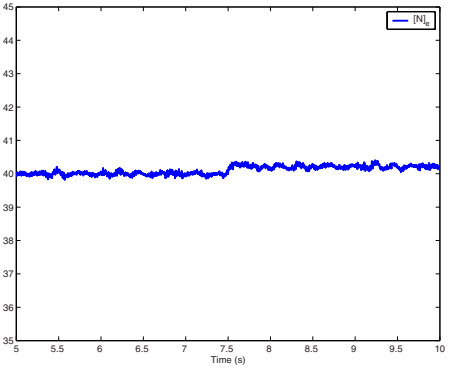
[6] M. Fliess, Controller canonical forms for linear and nonlinear dynamics, *IEEE Trans. Automat. Control*, **33**, 1990, 994-1001.
[7] M. Fliess, Une interprétation algébrique de la transformation de Laplace et des matrices de transfert, *Linear Algebra Appl.*, **203-204**, 1994, 429-442.
[8] M. Fliess, C. Join, An algebraic approach to fault diagnosis for linear systems, *CESA Proc.*, Lille, 2003.
[9] M. Fliess, C. Join, H. Sira-Ramírez, Robust residual generation for linear fault diagnosis: an algebraic setting with examples, *submitted for publication*.
[10] M. Fliess, J. Lévine, P. Martin, P. Rouchon, Flatness and defect of non-linear systems: introductory theory and examples, *Internat. J. Control*, **61**, 1995, 1327-1361.
[11] M. Fliess, J. Lévine, P. Martin, P. Rouchon, Deux applications de la géométrie locale des diffiétés, *Ann. Inst. H. Poincaré Phys. Théor.*, **66**, 1997, 275-292.
[12] M. Fliess, J. Lévine, P. Martin, P. Rouchon, A Lie-Bäcklund approach to equivalence and flatness of nonlinear systems, *IEEE Trans. Automat. Control*, **44**, 1999, 922-937.
[13] M. Fliess, M. Mboup, H. Mounier, H. Sira-Ramírez, Questioning some paradigms of signal processing via concrete examples, in *Algebraic Methods in Flatness, Signal Processing and State Estimators*, H. Sira-Ramírez, G. Silva-Navarro (Eds), Editorial Lagares, México, 2003, pp. 1-21.
[14] M. Fliess, H. Sira-Ramírez, An algebraic framework for linear identification, *ESAIM Control Optim. Calc. Variat.*, **9**, 2003, 151-168.
[15] M. Fliess, H. Sira-Ramírez, Reconstructeurs d'états, *C.R. Acad. Sci. Paris*, **I-338**, 2004, 91-96.
[16] M. Fliess, H. Sira-Ramírez, Control via state estimations of flat systems, *Proc. NOLCOS*, Stuttgart, 2004.




*(a)* In fault free case




*(b)* In faulty case




*(c)* Zoom on (b)

**Fig. 5.** On line estimation of number of connections

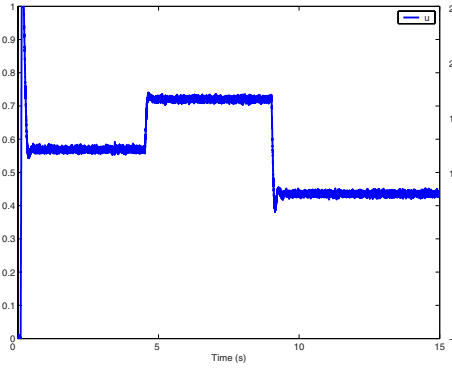
[17] S. Floyd, J. Padhye, and J. Widmer, Equation-based congestion control for unicast applications, *Proc. SIGCOMM'00*, 2000.

[18] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*, Marcel Dekker, New York, 1998.

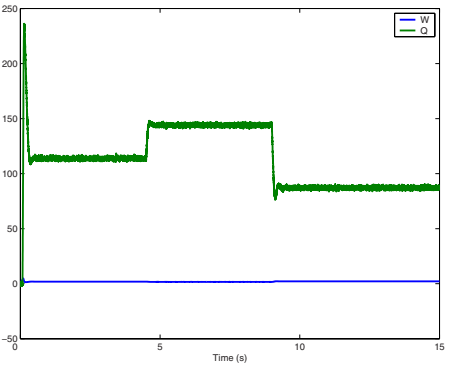
[19] L. A. Grieco, S. Mascolo, R. Ferorelli, Additive-increase/adaptive-decrease congestion control: a mathematical model and its experimental validation, *Proc. IEEE Internat. Symp. Computer Communications*, Giardini Naxos, 2002.

[20] V. Guffens, G. Bastin, H. Mounier, Using token leaky buckets for congestion feedback control in packets switched networks with guaranteed boundedness of buffer queues, *Proc 7th Europ. Control Conf.*, Cambridge, 2003.

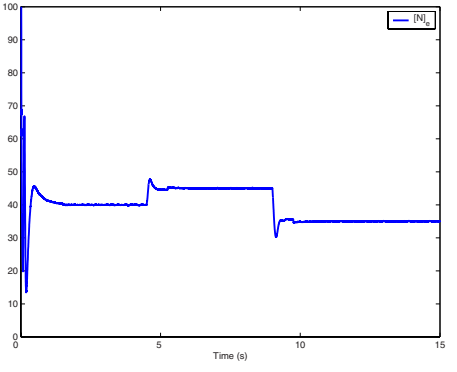
[21] R. Hartshorne, *Algebraic Geometry*, Springer, New York, 1977.




*(a)* Input RED control




*(b)* Outputs $W(t)$ (bottom) and $Q(t)$ (top)




*(c)* Zoom on (b)

**Fig. 6.** System behaviour when the number of TCP connections changes

[22] C.V. Hollot, V. Misra, D. Towsley, W.B. Gong, Analysis and design of controllers for aqm routers supporting tcp flows, *IEEE Trans. Automat. Control*, **47**, 2002, 945-959.

[23] N. Jacobson, *Basic Algebra, I & II*, $2^{nd}$ ed., Freeman, New York, 1985.

[24] J. Johnson, Kähler differentials and differential algebra, *Ann. Math.*, **89**, 1969, 92-98.

[25] C. Join, *Diagnotic des systèmes non linéaires - Contribution aux méthodes de découplage*, Thèse, Université Henri Poincaré, Nancy, 2002.

[26] C. Join, J. Jouffroy, J.-C. Ponsart, J. Lottin, Synthse d'un filtre isolateur de dfauts  l'aide de la thorie de la contraction, *Actes CIFA*, Nantes, 2002.

[27] C. Join, J.-C. Ponsart, D. Sauter, H. Jamouli, Fault decoupling via generalized output injection, *Proc. 7th Europ. Control Conf.*, Cambridge, 2003.
[28] C. Join, J.-C. Ponsart, D. Sauter, Diagnotic des systèmes non linéaires - Contribution aux méthodes de découplage, *J. Europ. Systmes Automatiss*, **37**, 2003, 1323-1328.
[29] E.R. Kolchin, *Differential Algebra and Algebraic Groups*, Academic Press, New York, 1973.
[30] C. Lanczos, *Applied Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1957.
[31] B. Liu, D.R. Figuereido, Y. Guo, J. Kurose, D. Towsley, A study of networks simulation effciency: fluid simulation vs. packet-level simulation, in *Proc. IEEE INFOCOM'01*, 2001.
[32] J. McConnell, J. Robson, *Noncommutative Noetherian Rings*, Amer. Math. Soc., Providence, RI, 2000.
[33] V. Misra, W.B. Gong, D. Towsley, Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED, *Proc. SIGCOMM00*, 2000.
[34] H. Mounier, G. Bastin, V. Vèque, Round trip time TCP tracking: a first step towards QoS pricing, *Int. J. Syst. Sci.*, **34**, 2003, 607-614.
[35] J.F. Ritt, *Differential Algebra*, Amer. Math. Soc., New York, 1950.
[36] L. Schwartz, *Théorie des distributions*, $2^e$ éd., Hermann, Paris, 1966.
[37] H. Sira-Ramírez, S. Agrawal, *Differential Flatness*, Marcel Dekker, New York, 2004.
[38] H. Sira-Ramírez, M. Fliess, On the output feedback control of a synchronous generator, *submitted for publication*.
[39] M. Staroswiecki, G. Comtet-Varga, Analytic redundancy for fault detection and isolation in algebraic dynamic system, *Automatica*, **37**, 2001, 687-699.
[40] D. Theilliol, H. Noura, J.-C. Ponsart, Fault diagnosis and accommodation of a three-tank system based on analytical redundancy, *ISA Transactions*, **41**, 2002.
[41] D. Zhou, G. Wang, S. Ding, Sensor fault tolerant control of nonlinear systems with application to a three-tank-systems, *Proc. 4th IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes SAFEPROCESS*, Budapest, 2000.
[42] A. Zolghadri, D. Henry, M. Monsion, Design of nonlinear observers for fault diagnosis: a case study, *Control Engineering Practice*, **4**, 1996, 1535-1544.

# List of Contributors

**C.T. Abdallah** (University of New Mexico, Albuquerque, USA) - Chapters 8, 11, 12, 14
**M. Ariola** (Università degli Studi di Napoli, Italy) - Chapter 8
**N.E. Barabanov** (Electrotechnical University, St. Petersburg, Russia) - Chapter 13
**S.C. Bengea** (Purdue University, USA) - Chapter 14
**J.D. Birdwell** (University of Tennessee, Knoxville, USA) - Chapters 11, 12
**C.G. Cassandras** (Boston University, USA) - Chapter 1
**F. Chatté** (HeuDiaSyC, France) - Chapter 5
**H. Che** (University of Texas, USA) - Chapter 6
**J. Chiasson** (University of Tennessee, Knoxville, USA) - Chapters 11, 12
**H. Choe** (Caltech, USA) - Chapter 2
**N. Chopra** (University of Illinois, USA) - Chapter 13
**R.A. DeCarlo** (Purdue University, USA) - Chapter 14
**S. Dhakal** (University of New Mexico, Albuquerque, USA) - Chapter 11
**N. Elia** (Iowa State University, USA) - Chapter 15
**M. Fliess** (Ecole Polytechnique, Palaiseau, France) - Chapter 16
**J. Ghanem** (University of New Mexico, Albuquerque, USA) - Chapter 11
**K. Gu** (Southern Illinois University, USA) - Chapter 5
**R. Hammi** (L2TI - Université Paris Nord, France)- Chapter 9
**M.M. Hayat** (University of New Mexico, Albuquerque, USA) - Chapters 11, 12
**P.F. Hokayem** (University of Illinois, USA) - Chapter 14
**H. Jerez** (University of New Mexico, Albuquerque, USA) - Chapter 11
**C. Join** (CRAN, Nancy, France) - Chapter 16
**C. Kulcsár** (L2TI - Université Paris Nord, France)- Chapter 9
**C.M. Lagoa** (The Pennsylvania State University, USA) - Chapter 6
**S.H. Low** (Caltech, USA) - Chapter 2
**T. Luzyanina** (K.U. Leuven, Belgium) - Chapter 5
**S. Mascolo** (Politecnico di Bari, Italy) - Chapter 7
**F. Mazenc** (INRIA-INRA, France) - Chapter 5
**D. Melchor-Aguilar** (IPICyT, Mexico) - Chapter 5
**W. Michiels** (K.U. Leuven, Belgium) - Chapter 5

**H. Mounier** (IEF - Université Paris Sud, France) - Chapter 16
**B.A. Movsichoff** (The Pennsylvania State University, USA) - Chapter 6
**S.I. Niculescu** (HeuDiaSyC, France) - Chapter 5
**R. Ortega** (LSS-CNRS, Gif sur Yvette, France) - Chapter 13
**H. Özbay** (Ohio University, USA) - Chapter 3
**F. Paganini** (UCLA, USA) - Chapter 10
**C. Panayiotou** (University of Cyprus, Cyprus) - Chapter 1
**H.F. Raynaud** (L2TI - Université Paris Nord, France)- Chapter 9
**M.W. Spong** (University of Illinois, USA) - Chapter 13
**R. Srikant** (University of Illinois, USA) - Chapter 4
**G. Sun** (Boston University, USA) - Chapter 1
**Z. Tang** (University of Tennessee, Knoxville, USA) - Chapter 12
**S. Tarbouriech** (LAAS-CNRS, Toulouse, France) - Chapter 8
**S. Tatikonda** (Yale University, USA) - Chapter 15
**P. Yan** (Ohio University, USA) - Chapter 3
**Z. Wang** (UCLA, USA) - Chapter 10
**Y. Wardi** (Georgiatech, USA) - Chapter 1

# Lecture Notes in Control and Information Sciences

## Edited by M. Thoma and M. Morari

Further volumes of this series can be found on our homepage: springeronline.com

**2001–2004 Published Titles:**

**Vol. 257:** Moallem, M.; Patel, R.V.; Khorasani, K.
Flexible-link Robot Manipulators
176 p. 2001 [1-85233-333-2]

**Vol. 258:** Isidori, A.; Lamnabhi-Lagarrigue, F.; Respondek, W. (Eds)
Nonlinear Control in the Year 2000 Volume 1
616 p. 2001 [1-85233-363-4]

**Vol. 259:** Isidori, A.; Lamnabhi-Lagarrigue, F.; Respondek, W. (Eds)
Nonlinear Control in the Year 2000 Volume 2
640 p. 2001 [1-85233-364-2]

**Vol. 260:** Kugi, A.
Non-linear Control Based on Physical Models
192 p. 2001 [1-85233-329-4]

**Vol. 261:** Talebi, H.A.; Patel, R.V.; Khorasani, K.
Control of Flexible-link Manipulators
Using Neural Networks
168 p. 2001 [1-85233-409-6]

**Vol. 262:** Dixon, W.; Dawson, D.M.; Zergeroglu, E.; Behal, A.
Nonlinear Control of Wheeled Mobile Robots
216 p. 2001 [1-85233-414-2]

**Vol. 263:** Galkowski, K.
State-space Realization of Linear 2-D Systems with Extensions to the General nD (n$>$2) Case
248 p. 2001 [1-85233-410-X]

**Vol. 264:** Baños, A.; Lamnabhi-Lagarrigue, F.; Montoya, F.J
Advances in the Control of Nonlinear Systems
344 p. 2001 [1-85233-378-2]

**Vol. 265:** Ichikawa, A.; Katayama, H.
Linear Time Varying Systems and Sampled-data Systems
376 p. 2001 [1-85233-439-8]

**Vol. 266:** Stramigioli, S.
Modeling and IPC Control of Interactive Mechanical Systems – A Coordinate-free Approach
296 p. 2001 [1-85233-395-2]

**Vol. 267:** Bacciotti, A.; Rosier, L.
Liapunov Functions and Stability in Control Theory
224 p. 2001 [1-85233-419-3]

**Vol. 268:** Moheimani, S.O.R. (Ed)
Perspectives in Robust Control
390 p. 2001 [1-85233-452-5]

**Vol. 269:** Niculescu, S.-I.
Delay Effects on Stability
400 p. 2001 [1-85233-291-316]

**Vol. 270:** Nicosia, S. et al.
RAMSETE
294 p. 2001 [3-540-42090-8]

**Vol. 271:** Rus, D.; Singh, S.
Experimental Robotics VII
585 p. 2001 [3-540-42104-1]

**Vol. 272:** Yang, T.
Impulsive Control Theory
363 p. 2001 [3-540-42296-X]

**Vol. 273:** Colonius, F.; Grüne, L. (Eds)
Dynamics, Bifurcations, and Control
312 p. 2002 [3-540-42560-9]

**Vol. 274:** Yu, X.; Xu, J.-X. (Eds)
Variable Structure Systems:
Towards the $21^{\mathrm{st}}$ Century
420 p. 2002 [3-540-42965-4]

**Vol. 275:** Ishii, H.; Francis, B.A.
Limited Data Rate in Control Systems with Networks
171 p. 2002 [3-540-43237-X]

**Vol. 276:** Bubnicki, Z.
Uncertain Logics, Variables and Systems
142 p. 2002 [3-540-43235-3]

**Vol. 277:** Sasane, A.
Hankel Norm Approximation
for Infinite-Dimensional Systems
150 p. 2002 [3-540-43327-9]

**Vol. 278:** Chunling D. and Lihua X. (Eds)
$H_\infty$ Control and Filtering of
Two-dimensional Systems
161 p. 2002 [3-540-43329-5]

**Vol. 279:** Engell, S.; Frehse, G.; Schnieder, E. (Eds)
Modelling, Analysis, and Design of Hybrid Systems
516 p. 2002 [3-540-43812-2]

**Vol. 280:** Pasik-Duncan, B. (Ed)
Stochastic Theory and Control
564 p. 2002 [3-540-43777-0]

**Vol. 281:** Zinober A.; Owens D. (Eds)
Nonlinear and Adaptive Control
416 p. 2002 [3-540-43240-X]

**Vol. 282:** Schröder, J.
Modelling, State Observation and
Diagnosis of Quantised Systems
368 p. 2003 [3-540-44075-5]

**Vol. 283:** Fielding, Ch. et al. (Eds)
Advanced Techniques for Clearance of
Flight Control Laws
480 p. 2003 [3-540-44054-2]

**Vol. 284:** Johansson, M.
Piecewise Linear Control Systems
216 p. 2003 [3-540-44124-7]

**Vol. 285:** Wang, Q.-G.
Decoupling Control
373 p. 2003 [3-540-44128-X]

**Vol. 286:** Rantzer, A. and Byrnes C.I. (Eds)
Directions in Mathematical Systems
Theory and Optimization
399 p. 2003 [3-540-00065-8]

**Vol. 287:** Mahmoud, M.M.; Jiang, J. and Zhang, Y.
Active Fault Tolerant Control Systems
239 p. 2003 [3-540-00318-5]

**Vol. 288:** Taware, A. and Tao, G.
Control of Sandwich Nonlinear Systems
393 p. 2003 [3-540-44115-8]

**Vol. 289:** Giarré, L. and Bamieh, B.
Multidisciplinary Research in Control
237 p. 2003 [3-540-00917-5]

**Vol. 290:** Borrelli, F.
Constrained Optimal Control
of Linear and Hybrid Systems
237 p. 2003 [3-540-00257-X]

**Vol. 291:** Xu, J.-X. and Tan, Y.
Linear and Nonlinear Iterative Learning Control
189 p. 2003 [3-540-40173-3]

**Vol. 292:** Chen, G. and Yu, X.
Chaos Control
380 p. 2003 [3-540-40405-8]

**Vol. 293:** Chen, G. and Hill, D.J.
Bifurcation Control
320 p. 2003 [3-540-40341-8]

**Vol. 294:** Benvenuti, L.; De Santis, A.; Farina, L. (Eds)
Positive Systems: Theory and Applications (POSTA 2003)
414 p. 2003 [3-540-40342-6]

**Vol. 295:** Kang, W.; Xiao, M.; Borges, C. (Eds)
New Trends in Nonlinear Dynamics and Control,
and their Applications
365 p. 2003 [3-540-10474-0]

**Vol. 296:** Matsuo, T.; Hasegawa, Y.
Realization Theory of Discrete-Time Dynamical Systems
235 p. 2003 [3-540-40675-1]

**Vol. 297:** Damm, T.
Rational Matrix Equations in Stochastic Control
219 p. 2004 [3-540-20516-0]

**Vol. 298:** Choi, Y.; Chung, W.K.
PID Trajectory Tracking Control for Mechanical Systems
127 p. 2004 [3-540-20567-5]

**Vol. 299:** Tarn, T.-J.; Chen, S.-B.; Zhou, C. (Eds.)
Robotic Welding, Intelligence and Automation
214 p. 2004 [3-540-20804-6]

**Vol. 300:** Nakamura, M.; Goto, S.; Kyura, N.; Zhang, T.
Mechatronic Servo System Control
Problems in Industries and their Theoretical Solutions
212 p. 2004 [3-540-21096-2]

**Vol. 301:** de Queiroz, M.; Malisoff, M.; Wolenski, P. (Eds.)
Optimal Control, Stabilization and Nonsmooth Analysis
373 p. 2004 [3-540-21330-9]

**Vol. 302:** Filatov, N.M.; Unbehauen, H.
Adaptive Dual Control: Theory and Applications
237 p. 2004 [3-540-21373-2]

**Vol. 303:** Mahmoud, M.S.
Resilient Control of Uncertain Dynamical Systems
278 p. 2004 [3-540-21351-1]

**Vol. 304:** Margaris, N.I.
Theory of the Non-linear Analog Phase Locked Loop
303 p. 2004 [3-540-21339-2]

**Vol. 305:** Nebylov, A.
Ensuring Control Accuracy
256 p. 2004 [3-540-21876-9]

**Vol. 306:** Bien, Z.Z.; Stefanov, D. (Eds.)
Advances in Rehabilitation
472 p. 2004 [3-540-21986-2]

**Vol. 307:** Kwon, S.J.; Chung, W.K.
Perturbation Compensator based Robust Tracking
Control and State Estimation of Mechanical Systems
158 p. 2004 [3-540-22077-1]