Processing Networks

This state-of-the-art account unifies material developed in journal articles over the last 35 years, with two central thrusts: It describes a broad class of system models that the authors call “stochastic processing networks” (SPNs), which include queueing networks and bandwidth sharing networks as prominent special cases; and in that context, it explains and illustrates a method for stability analysis based on fluid models.

The central mathematical result is a theorem that can be paraphrased as follows: if the fluid model derived from an SPN is stable, then the SPN itself is stable. Two topics discussed in detail are (a) the derivation of fluid models by means of fluid limit analysis and (b) stability analysis for fluid models using Lyapunov functions. With regard to applications, there are chapters devoted to max-weight and back-pressure control, proportionally fair resource allocation, data center operations, and flow management in packet networks. Geared toward researchers and graduate students in engineering and applied mathematics, especially in electrical engineering and computer science, this compact text gives readers full command of the methods.

J. G. DAI received his PhD in mathematics from Stanford University. He is currently Presidential Chair Professor in the School of Data Science at the Chinese University of Hong Kong, Shenzhen. He is also the Leon C. Welch Professor of Engineering in the School of Operations Research and Information Engineering at Cornell University. He was honored by the Applied Probability Society of the Institute for Operations Research and the Management Sciences (INFORMS) with its Erlang Prize (1998) and with two Best Publication Awards (1997 and 2017). In 2018, he received the Achievement Award from the Association for Computing Machinery (ACM) SIGMETRICS. Professor Dai served as editor-in-chief of *Mathematics of Operations Research* from 2012 to 2018.

J. MICHAEL HARRISON earned degrees in industrial engineering and operations research before joining the faculty of Stanford University’s Graduate School of Business, where he served for 43 years. His research concerns stochastic models in business and engineering, including mathematical finance and processing network theory. His previous books include *Brownian Models of Performance and Control* (2013). Professor Harrison has been honored by INFORMS with its Expository Writing Award (1998), the Lanchester Prize for best research publication (2001), and the John von Neumann Theory Prize (2004); he was elected to the U.S. National Academy of Engineering in 2008.

Processing Networks

Fluid Models and Stability

J. G. DAI
The Chinese University of Hong Kong, Shenzhen and Cornell University

J. MICHAEL HARRISON
Stanford University

CAMBRIDGE UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre,
New Delhi – 110025, India

79 Anson Road, #06–04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of
education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781108488891
DOI: 10.1017/9781108772662

First published 2020

Printed in the United Kingdom by TJ International, Padstow Cornwall

A catalogue record for this publication is available from the British Library.

Library of Congress Cataloging-in-Publication Data
Names: Dai, J. G. (Jiangang), 1962– author. | Harrison, J. Michael, 1944– author.
Title: Processing networks : fluid models and stability / Jim Dai,
J. Michael Harrison.
Description: Cambridge ; New York, NY : Cambridge University Press, 2020. |
Includes bibliographical references and index.
Identifiers: LCCN 2020012552 | ISBN 9781108488891 (hardback) |
ISBN 9781108772662 (epub)
Subjects: LCSH: Stochastic processes. | Linear programming. | Queuing
networks (Data transmission) | Fluid dynamics. | Stability.
Classification: LCC QA274.A1 D35 2017 | DDC 519.2/3–dc23
LC record available at https://lccn.loc.gov/2020012552

ISBN 978-1-108-48889-1 Hardback

Cambridge University Press
978-1-108-48889-1 — Processing Networks
J. G. Dai , J. Michael Harrison
Frontmatter
More Information

To Liqin and Kevin, Elena and Sasha

Contents

Website

http://www.spnbook.org

This book is accompanied by the above website. The website provides corrections of mistakes and other resources that should be useful to both readers and instructors.

Cambridge University Press
978-1-108-48889-1 — Processing Networks
J. G. Dai , J. Michael Harrison
Frontmatter
More Information

Preface

This book has two purposes. First, it describes a broad class of mathematical system models, called stochastic processing networks (SPNs), that are useful as representations of service systems, industrial processes, and digital systems for computing and communication. The SPN models to be considered include such features as simultaneous resource possession, multi-input operations, and alternative processing modes. No comparably general treatment of network models has appeared previously in book format.

Second, it develops a fluid model methodology for proving SPN stability, by which we mean proving positive recurrence of the Markov chain describing the SPN. Specifically, we develop a theorem that can be informally paraphrased as follows: if the fluid model derived from an SPN is stable (as that phrase is defined later in this preface), then the SPN itself is stable. The significance of that result lies in the relative tractability of fluid models: proving fluid model stability is invariably easier than proving positive recurrence of the Markov chain for which it serves as a surrogate.

As multiple examples will show, proving fluid model stability for a complex SPN can still be challenging, requiring the construction of a suitable Lyapunov function. A large part of the book is aimed at demonstrating how the theorem has been used and can be used to analyze systems of contemporary interest, especially computing and communication networks.

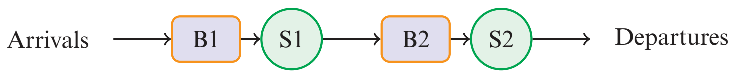

Figure 1 Tandem processing system.

Tandem example. To understand the content of the theorem paraphrased above, it is useful to consider a concrete example. Figure 1 depicts a tandem processing system with two servers, labeled S1 and S2. Jobs arrive from the outside world one at a time. Each job is processed first by S1, then by S2, and then exits. If a job arriving at either server finds that server idle, then the processing of that job begins immediately. On the other hand, if a job arriving at either server finds that server busy, then the job waits in a corresponding buffer (B1 or B2). Each server processes jobs from its associated buffer on a first-in-first-out (FIFO) basis, and continues working at full capacity so long as there is any job available for it to process. As a matter of convention, when reference is made later to "jobs currently occupying B1," that is understood to include not only waiting jobs but also the job being processed by S1, if there is one, and similarly for B2. For concreteness, let us assume that the external arrival process is Poisson with arrival rate λ_1; that S1 processing times, also called service times, are independent and identically distributed (i.i.d.) with some phase-type distribution (see Appendix D, Section D.8, for the meaning of that term) having mean m_1; and that S2 service times are i.i.d. and exponentially distributed with mean m_2.

Tandem fluid model. Associated with this discrete-flow network is a continuous-flow model, or fluid model, that consists of the following equations: for $t \geq 0$,

(1) $Z_1(t) = Z_1(0) + \lambda_1 t - \mu_1 T_1(t) \geq 0,$

(2) $Z_2(t) = Z_2(0) + \mu_1 T_1(t) - \mu_2 T_2(t) \geq 0,$

(3) $T_i(0) = 0, \quad 0 \leq T_i(t) - T_i(s) \leq t - s \quad \text{for } 0 \leq s \leq t, \quad i = 1, 2,$

(4) $Z_i(u) > 0$ for all $u \in [s,t]$ implies that $T_i(t) - T_i(s) = t - s$ for $0 \leq s \leq t$ and $i = 1, 2$.

Here $Z_i(t)$ is interpreted as the fluid content in buffer i at time t, and $T_i(t)$ is the cumulative amount of time that server i is busy up to time t ($i = 1, 2$). The parameter λ_1 is the arrival rate of fluid from the outside, and $\mu_i := m_i^{-1}$ is the processing speed of server i ($i = 1, 2$). Equations (1) and (2) are the flow balance equations, while (3) expresses service capacity constraints, and (4) dictates that each server operate at full capacity whenever its buffer is nonempty.

The fluid model is said to be stable if, for each solution Z of equations (1) through (4), there exists a time $\delta > 0$ such that $Z_1(t) = Z_2(t) = 0$ for

$t \geq \delta$. One can prove that the fluid model is stable if and only if the following conditions are satisfied:

$$\lambda_1 m_1 < 1 \quad \text{and} \quad \lambda_1 m_2 < 1. \tag{5}$$

Sequential decomposition of the tandem fluid model. There are many ways to prove that statement. Perhaps the simplest is to analyze the tandem fluid model sequentially, first for fluid in buffer 1, then for fluid in buffer 2. Given that $\lambda_1 < \mu_1$, it is easy to show that there exists a time $\delta_1 \geq 0$ such that $Z_1(t) = 0$ for $t \geq \delta_1$. After time δ_1, the fluid flowing into B1 instantaneously passes into B2, and thus the arrival rate to B2 is λ_1. One can again analyze buffer 2 in isolation, showing there exists $\delta_2 \geq \delta_1$ such that $Z_2(t) = 0$ for $t \geq \delta_2$.

Direct analysis of the discrete-flow model. Under the distributional assumptions stated earlier, we can model the tandem processing system as a continuous-time Markov chain $\{X(t) = (Z_1(t), \eta(t), Z_2(t)), t \geq 0\}$, where $Z_i(t)$ is the number of jobs occupying buffer i at time t, and $\eta(t)$ is a finite-valued phase indicator (see Section D.8) for the S1 service currently under way, if any. (When B1 is empty, $\eta(t) = 0$.) How does one prove that the Markov chain X is positive recurrent under condition (5)?

The first thing to say is that there exists no analog of the sequential decomposition approach we have described. That is, in the discrete-flow setting, stability analysis is *not* decomposable, despite the feedforward structure (that is, unidirectional flow) that is the salient feature of our example. Rather, with rare exceptions, the approach adopted by researchers is to apply the Foster–Lyapunov criterion described in Appendix D, Section D.7. In this approach, the analyst must identify a test function V, hereafter called a Lyapunov function, that satisfies the Foster–Lyapunov drift condition (D.36). For our tandem processing example, one can construct a Lyapunov function of the quadratic form

$$V(Z_1, \eta, Z_2) = Z_1^2 + a(Z_1 + Z_2)^2,$$

where a is any constant satisfying $0 < a < \mu_1/\lambda_1 - 1$. An analysis undertaken in Section 8.2, culminating in the inequality (8.19), will show that this function V satisfies the Foster–Lyapunov drift condition, thus proving the positive recurrence of X.

Lyapunov functions for fluid models. The sequential decomposition described earlier to prove stability for our tandem fluid model extends in a direct way to *any* feedforward fluid model. For a general fluid model, however, one proves stability in very much the same way as for a general Markov chain, namely, by identifying a Lyapunov function that satisfies an appropriate drift condition. That general theory will be developed in Chapter 8.

As an example, the simple linear function $V(Z) = Z_1 + Z_2$ satisfies the drift condition for our tandem fluid model, given that (5) holds. However, for reasons explained in Section 8.2, it does *not* satisfy the drift condition for the discrete-flow tandem model. Thus the simplest known Lyapunov function for the discrete-flow tandem model is quadratic, while that for its fluid analog is linear. This illustrates a phenomenon that has often been observed in the analysis of particular model structures: in cases where a Lyapunov function is known both for a discrete-flow SPN model and for the fluid model derived from it, the latter function is substantially simpler.

Control policies and stability conditions. In the preceding paragraphs, we have discussed the stability problem for processing networks as if it were simply one of analysis, that is, as if the central problem were to rigorously prove stability under a given control policy. In general, however, a system designer or system manager first confronts a problem of *synthesis*, namely, he or she must first devise a dynamic control policy, which may be called a *network protocol* or *network algorithm* in a digital system context.

In our tandem example, we have specified FIFO processing by both servers, but the same fluid equations are valid for other *nonidling* policies as well. (Here the term "nonidling" means that each server works at full capacity whenever there is accessible work for that server to do.) Specifically, the fluid model equations (1) and (2) remain valid under any policy such that the number of partially completed jobs at any given time is bounded by some constant L, (3) is valid under any policy, and (4) is valid under any nonidling policy, as will be shown in Section 7.1.

In this book, virtually all effort and attention is directed to the analytical problem of proving stability for a given policy. As a preliminary, we develop in Chapter 5 a general stability condition analogous to (5), based on what is called a *static planning problem*. That condition involves only first-order system data (average arrival rates, average processing times, and routing probabilities or average output quantities), and it is shown to be *necessary* for stability under any control policy.

One tends to feel intuitively that the condition is also sufficient for existence of a policy that achieves stability, but examples will show that expectation is not always correct.

Dominance of the fluid approach. Over the last 25 years, fluid model methodology has come to dominate in studies of network stability, allowing successful treatment of model families that have defied direct analysis. Notable in this regard are the feedforward networks referred to earlier. The method of sequential decomposition makes fluid model stability proofs almost trivial for such networks (see Section 8.3 for elaboration), whereas the feedforward structure may be of little or no help in direct analysis. This contrast is illustrated well by the recent work of Massoulié and Xu (2018) on information processing systems.

There are also important families of non-feedforward networks for which fluid models have been analyzed successfully to prove stability, but no method is known for direct analysis. Another way of saying this is that, for some important families of non-feedforward networks, Lyapunov functions have been successfully devised for their fluid model analogs, but not for the discrete-flow models themselves. This is true, for example, of the FIFO Kelly networks analyzed by Bramson (1996a), also treated in section 5.3 of the monograph by Bramson (2008). Another example is a packet switched communication network with what Walton (2015) called *random proportional scheduling*, to be treated in Chapter 12 of this book. In both those cases, an entropy-type Lyapunov function (see Section 10.5) provides the key to fluid analysis, and there is no known analog for the discrete-flow model itself.

Acknowledgments In writing this book, we have benefited from consultations with many colleagues, including Søren Asmussen, Maury Bramson, Anton Braverman, Doug Down, Sergey Foss, Peter Glynn, Mor Harchol-Balter, Frank Kelly, Sunil Kumar, Yi Lu, Andre Milzarek, Balaji Prabhakar, Devavrat Shah, R. Srikant, Jean Walrand, Neil Walton, Ruth Williams, Qiaomin Xie, Kuang Xu, Lei Ying, Elena Yudovina, Yuan Zhong, and Bert Zwart. Shuangchi He helped with some figures in the book, and Cornell PhD students Chang Cao and Mark Gluzman helped with finalizing the format of references.

Guide to Notation and Terminology

We use $\mathbb{R}$ to denote the set of real numbers, $\mathbb{R}_+$ the set of nonnegative real numbers, $\mathbb{Z}_+$ the set of nonnegative integers, and $\mathbb{N}$ the positive integers. A prime is used to denote the transpose of a vector or matrix, and vectors should be envisioned as column vectors unless something is said to the contrary. For an integer $d > 0$ and a vector $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, we define the norm $|x| = \sum_{i=1}^{d} |x_i|$, and for two vectors $x, y \in \mathbb{R}^d$, we define the inner product $x \cdot y = \sum_{i=1}^{d} x_i y_i$. For two vectors $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^d$, we write $x \leq y$ to mean that $x_i \leq y_i$ for each $i = 1, \ldots, d$.

The relationship $A := B$ means that A equals B by definition. The letter e is occasionally used to denote the vector $(1, \ldots, 1)$, and we denote by e^j a vector with a 1 as its jth component and all other components equal to zero; in each case, the dimension of the vector should be clear from context. For $x, y \in \mathbb{R}$, we use $x \vee y$ to denote $\max(x, y)$, and $x \wedge y$ to denote $\min(x, y)$.

A square, nonnegative matrix is said to be *substochastic* if each of its row sums is ≤ 1, and to be *stochastic* if each of its row sums is $= 1$. The *spectral radius* of a $d \times d$ substochastic matrix P is $\max_{1 \leq i \leq d} |\lambda_i|$, where $\lambda_1, \ldots, \lambda_d$ are the eigenvalues of P. A substochastic matrix P is said to be *transient* if its spectral radius is < 1, or equivalently, if $P^n \to 0$ as $n \to \infty$.

Throughout the book, we denote by $\mathbb{P}(\cdot)$ the probability measure underlying a model, and by $\mathbb{E}(\cdot)$ the corresponding expectation operator. That is, $\mathbb{P}(A)$ denotes the probability of an event A, and $\mathbb{E}(X)$ is the expected value of a random variable X. For an event A and random variable X, we define the partial expectation $\mathbb{E}(X; A) = \int_A X d\mathbb{P}$.

Phase-type distributions are defined and discussed in Appendix D, Section D.8, where we introduce the following notation for three specific families of nonnegative, univariate distributions: $\exp(r)$ denotes an *exponential* distribution with rate parameter $r > 0$; Erlang$(2, r)$ denotes an *Erlang* distribution with shape parameter 2 and rate parameter $r > 0$; and $H_d(p, \gamma)$ denotes a *hyperexponential* distribution (see Section D.8 for details).

Stochastic processing networks, also referred to frequently as SPN models, are formally defined in Chapter 2, Sections 2.1 through 2.4, where we introduce notation for model data and model-related processes that continues throughout the entire book. In particular, the uppercase Roman letters A, B, D, E, F, I, J, K, N, S, T, and Z are given more or less permanent meanings in those sections, but such symbols may be reused with new meanings in the appendices. Sets are most often denoted by uppercase script letters; three that appear frequently are $\mathscr{I} = \{1, \ldots, I\}$, $\mathscr{J} = \{1, \ldots, J\}$, and $\mathscr{K} = \{1, \ldots, K\}$.

For a function $f : \mathbb{R}_+ \to \mathbb{R}^d$, we use $\dot{f}(t)$ to denote the derivative of f at t. A point $t > 0$ is said to be a *regular point* for f if f is differentiable at t. When the function f is clear from the context, we sometimes call $t > 0$ a regular point without further qualification. Whenever the symbol $\dot{f}(t)$ is used, it is assumed that t is a regular point of f. Occasionally we also use $\frac{d}{dt} f(t)$ to denote $\dot{f}(t)$.