CONGESTION-INDUCED COLLAPSE IN NETWORKS: MANAGING FAILURE CASCADES IN COMPLEX SYSTEMS AND INFRASTRUCTURE PROTECTION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF MANAGEMENT SCIENCE & ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

David L. Alderson
May 2003

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Nicholas Bambos
(Principal Co-Advisor)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

William J. Perry
(Principal Co-Advisor)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Peter W. Glynn

Approved for the University Committee on Graduate Studies.

Preface

The national infrastructure systems of the United States form a complex mesh of interdependent networks. While technology advances in these infrastructures have brought great efficiencies to modern life, our economic and social welfare have become inextricably dependent on them. Over the last decade, there has been growing concern over the vulnerability of national infrastructure systems to accidents, failures, and attacks. While there is significant evidence to support the belief that catastrophic cascading failures can and will happen in the future, there is so far little theory to understand how and why these failure cascades occur and propagate. There is even less known about what to do about them.

This thesis lends perspective and insight into this problem through the study of congestion-induced network failures. The objective is to develop a framework through which a comprehensive treatment of this broad and important problem will be possible. Particular attention is placed on the identification tensions and tradeoffs in complex network management and design, as well as on the development of robust management strategies for critical infrastructure systems.

Acknowledgements

The completion of this thesis would not have been possible without the encouragement and support of many colleagues and friends.

I would first like to thank my principal co-advisors, Nicholas Bambos and William Perry. I would like to thank Dr. Perry for his enthusiasm and entrepreneurial spirit for tackling important, yet difficult problems. I might not have engaged this broad and challenging topic had he not shown consistent faith in my ability to do so. I would like to thank Dr. Bambos for sharing his vision and passion for problems in networking and also for teaching me how to tell a good story.

I would also like to thank several other individuals who have played a formal role in this research. Foremost among these, I would like to thank Peter Glynn for serving as a reader of this thesis and for his ongoing patience and generosity of spirit in working with me. From Dr. Glynn I have developed a fondness for applied probability, and it is because of his example that I strive to be a better student and teacher of it. I would also like to thank the members of my oral defense committee: Frank Kelly, Michael May, and Jim Gibbons, who served as my committee chairman. It is with great pleasure that I deliver on my promise to them.

It is with deep gratitude that I thank Tom Byers for his ongoing mentorship and enduring friendship. As my own personal Morrie Schwartz, Dr. Byers has given me the best example of how to be a great teacher and person, and I hope to live up to his expectations.

The ideas represented here are the result of countless conversations and interactions with colleagues from diverse backgrounds. I would like to thank Mike Connolly and Tom Haley of the Union Pacific Railroad for their willingness to teach me about

the modern railroad business. The modeling insight for this problem was inspired by conversations about railroads, and I look forward to taking this research full circle in addressing some of UP's operational challenges. I would also like to thank the Santa Fe Institute and the Institute for Pure and Applied Mathematics (IPAM) at UCLA for sponsoring me as a visiting reseacher during my graduate tenure and facilitating this dialogue. Finally, I would like to thank John Lilly and the good people at Reactivity, Inc., the best damn startup for whom I could have possibly worked. Particular recognition goes to Graham Miller, who has helped me with more than a programming problem or two.

There are several other individuals at Stanford who deserve special recognition. I would like to give special thanks to Deborah Gordon and Lorrie Papadakis for looking after me with care these last several years and providing their constant and friendly support. I am fortunate to have been blessed with so many good friends without whom this experience as a graduate student would not have been as educational or as fun: John McConnell, Celestino Martinez, Maria Alejandra Quijada, Heleen Kist, Aaron Scurlock, Alysia Desbiens, and Matt Brennan. And as anyone who knows me will attest, this thesis would not have been possible without warm environment and friendly baristas of the Starbucks Coffee Company.

Finally, I would like to thank my friends and family for their love, patience, and support during my tenure in graduate school. I would like to thank Lisa for her eternal love and affection. This dissertation is dedicated to the memory of my grandfather who taught me, among many things, that a project is not done until it is done right.

Contents

List of Tables

List of Figures

Chapter 1

Network Infrastructure Vulnerability

Networks are the fabric of the modern world around us. When we turn on the lights, or talk on the phone, or drive on the highway, or surf the Internet, we are using a system that was designed and built as a network. In fact, nearly all of our national infrastructures, from telecommunications to transportation, are built in this manner. There are good reasons for this. Networks have the economic property that allows increasingly efficient sharing of resources as the size of the network grows. However, network structures are inherently complex. The possible number of interactions within a network grows rapidly in the number of components and the number of relationships between components. The resulting complexity creates significant challenges for the design, prediction, and control of network behavior. Complexity may even contribute to network fragility. This tradeoff between efficiency, complexity, and fragility in networks is particularly evident in the infrastructure systems that connect our nation.

1.1 National Infrastructure Systems

Throughout our nation's history, the development of infrastructure networks has brought great advances in efficiency. The completion of the trans-continental rail-

road in 1869 reduced the travel time between New York and San Francisco from approximately five months to approximately five days. Along with it came the completion of transcontinental telegraph service that ushered a new era in communications. Over the last 100 years, there have been enormous achievements made in connecting the United States both physically and informationally. Some of these include transcontinental postal service, the public telephone network, modern air transportation, overnight delivery, the commercialization of the Internet, and wireless voice and data services. Indeed, much of the economic and social life of this country is based upon these infrastructures.

The success in these infrastructures has led to a critical dependence upon them. In fact, our reliance on these infrastructures is so great that the systematic disruption of any one of them can have catastrophic economic and social consequences. Despite the obvious nature of this dependence, large-scale failures within these systems have already happened, and there may even be good reason to believe that such failures will continue to happen in the future.

1.1.1 Evidence of Large-Scale Failures

In the last decade alone, most of the aforementioned infrastructure systems have experienced major disruptions of some form. Consider the following examples from data communications, transportation, and electric power.

Failures in Data Networks

In February 2001, a switch malfunction within AT&T's Asynchronous Transfer Mode (ATM) data network led to the overloading of 7% of their switches and a disruption to 5% of all virtual circuits within the network [87]. ATM is a technology that uses virtual circuits to transmit data packets of information and achieves high bandwidth, low delay performance for applications such as streaming media and video conferencing. The cause of the switch malfunction was attributed to a traffic aberration that the switch experienced following a cut in a fiber optic line earlier in the day [38]. As a result, the errant switch repeatedly sent out messages to its neighbors that net-

work trunks were available and then unavailable, a situation known as "thrashing". The switch did this until it eventually overloaded its CPU and memory, when it ultimately crashed. In response to this failure, other switches tried to reroute the traffic but became overloaded as well. The incident disrupted the network for four hours and affected an undisclosed number of customers, some of whom did not receive the "all clear" message from AT&T until the next day [88].

AT&T also suffered a major disruption to its frame relay network in April 1998. In contrast to ATM technology, frame relay is a packet-based technology used to transmit bulk data, such as in backoffice business operations. This incident was initiated by a bug in a software upgrade to one of AT&T's Cisco switches [39]. The software bug created a fault that generated a tremendous number of administrative messages to the other switches. As a result, the network became overloaded and stopped routing data for up to 26 hours. The incident affected 100% of AT&T's then 6,600 customers (an estimated 45% of the total market) and seriously disrupted the business operations of customers such as Wells Fargo Bank, Wal-Mart, and Unisys Corporation [119, 102].

A similar incident affected MCI's frame relay network in August 1999. Again, the cause was a bug in a software upgrade (this time to a Lucent switch). In this case, the incident affected only 30% of MCI's frame relay network (the network was partitioned), but the incident was not resolved for a total of ten days [25, 26, 116]. More than 3000 customers were affected, including America Online and the Chicago Board of Trade, who lost more than 250 trading workstations in Paris, Toyko, London, and at sites throughout the United States [96, 95, 51].

It is perhaps obvious that large-scale disruptions to data networks can have serious economic consequences to businesses that rely upon them. Even so, it is worth noting the magnitude of such events. While it is hard to estimate the impact of disruptions to *internal* operations, such as were experienced in the aforementioned cases, there have been estimates of the impact of downtime to *externally facing* business operations, such as those used in e-commerce. According to a study published by Dataquest, the range of costs associated with an hour of downtime varies from as low as $14,500 per hour to more than $7 million per hour depending on the particular business sector [43].

The Stratus Group[1] estimates that one minute of downtime at Federal Express costs the company $1 million. At Visa, which averages approximately 5,000 transactions per second, Stratus estimates the cost of downtime at $10 million per minute [40].

Union Pacific Service Crisis

From June 1997 to December 1998, the Union Pacific Railroad[2] experienced a severe disruption in its ability to provide timely service to its shippers. Initially triggered by a single derailment within a critical train yard outside of Houston, the disruption was fueled by a number of complicating factors including extreme weather conditions, unusual operating conditions among competing railroads, labor troubles, and political conditions in Mexico. The net result was the buildup and eventual spread of congestion from the Houston and Gulf Coast area to the Central Corridor (Kansas-Nebraska-Wyoming) and eventually Southern California. During this time period the economic impact to Union Pacific was tremendous. UPRR's parent company, Union Pacific Corporation, reported a swing in its income from continuing operations of more than $430 million in FY 1997 to losses of more than $630 million in FY 1998 [1], and media reports estimated the losses to UP's shippers at several billions of dollars [122].

Electric Power Outages

In November 1965, an overcurrent relay on a single transmission line triggered a major electric power outage throughout the Northeastern United States, including New York City, and Canada. More than 30 million people were left without power for up to 13 hours. The incident became known as the Great Northeast Blackout and incurred losses of an estimated $100 million [36, 112].

During the summer of 1996, local faults on two separate occasions led to major outages within the Western Systems Coordinating Council[3] portion of the national

[1]The Stratus Group is a technology consulting firm, part of Stratus Technologies.

[2]Union Pacific is the largest of four cargo railroad transportation companies currently operating in the United States.

[3]The WSCC is one of three major interconnet regions that partition the electric power grid of the continental United States.

power grid. On July 3 of that year, a tree fell onto a major power line in Idaho, causing power failures across a 15-state area, including Canada and Mexico, and affecting more than 2 million people [92]. Then, just weeks later on August 10, another incident near the California-Oregon border led to wide-spread outage over a seven state area and affecting more than 7.5 million people [113, 45]. The economic losses for this second outage were estimated at more than $100 million to electricity producers and more than $1 billion to consumers [24]. In both cases, record heat throughout the West was a contributing factor, as the power grid was already operating near critical load when the failures occurred.

2001 Baltimore Tunnel Accident

In July 2001, a train carrying industrial solvents and corrosive chemicals wrecked inside the 1.7-mile Howard Street Tunnel below downtown Baltimore [81]. The train caught fire and began issuing hazardous smoke into the Baltimore Harbor [97, 72]. The train was also carrying plywood and paper that provided ample fuel for the fire that lasted for more than 5 days [15, 99, 98]. The wreck affected nearly all rail traffic on the eastern seaboard, causing major delays and rerouting of trains [90]. However, there were a number of other consequences. The accident cut through three major fiber-optic lines for the East Coast that also ran through the tunnel (WorldCom Inc.'s UUNet, Metromedia Fiber Network and PSINet Inc.) resulting in the worst internet congestion in more than 3 years time [65, 49]. The wreck also caused a water main break above the tunnel that flooded a major intersection within the downtown area for more than 100 hours, shutting down all automobile and train traffic [70]. At the time of the accident, the Baltimore Orioles, whose home field of Camden Yards is directly above the tunnel, were in the middle of a doubleheader game. The stadium had to be evacuated, and because all traffic out of the downtown area was shut down, thousands were stranded without hotel space [108]. People who did find space in hotels around the harbor couldn't flush their toilets, and brown water ran from their faucets. More than 1,200 downtown Baltimore Gas and Electric customers were left without power, of which two nearby office towers were shutdown for several days.

1.1.2 Similarities Among Infrastructures

The above incidents are examples of large-scale failures in infrastructure networks. Each of the affected systems is very different—they have dissimilar structure on many scales and are often based on entirely different physical principles. There is no reason to believe *a priori* that their behavior should look anything alike. Yet, it is interesting to note the qualitative (if not more substantial) similarities in the failure behavior observed among all of them.

1. *The initiating event for each incident is a seemingly minor, local disturbance.* There is no reason to suspect in advance that such a local event could have systemwide consequences. Such disturbances are well within the design specifications for the system and are to be expected.

2. *The large-scale nature of the incident is not because of a single point of failure for the entire system.* Instead, it is the connectivity *within* and *between* these systems that allows the failure to propagate. That is, connectivity allows for the possibility of a *cascading failure*.

3. *The failure spreads quickly, before repairs can be made.* In general, it is this interaction of time scales that allows it to spread, as we will discuss later.

4. *Once the large scale failure behavior has been initiated, it takes a life of its own.* Usually, fixing the initiating failure does not fix the larger scale problem.

From these similarities, we hypothesize that there is something in the common *network structure* of these systems that contributes to the similarities in their failure behavior.

1.1.3 A Common Information Infrastructure

All of the aforementioned infrastructure systems have "grown up" in a manner that was largely independent of one another. However, with the growth of the Internet and the advent of a common "information infrastructure", there has been an ongoing drive to increase the connectivity within and between these systems. The motivation

for this ongoing drive is to leverage efficiencies between these systems. Although this effort has been successful in realizing many of these new efficiencies for daily life, it has come at a cost of greater overall system complexity.

Whether the former independence of these infrastructure systems was a historical artifact or part of an intentional design is irrelevant. The point is that separation between systems provides natural protective barriers between them, and the removal of these barriers through their systematic interconnection may be cause for concern. Indeed, many researchers and policy makers are questioning the extent to which the perceived fragility for these systems is a direct consequence of their interconnected nature.

1.1.4 Government Interest in Infrastructure Protection

Within the last several years, there has been growing interest by the federal government in the extent to which these national infrastructure systems are at risk to large-scale failures. In particular, one of the key issues has been the extent to which the interdependence between infrastructure systems makes them vulnerable to accidents, failures, and attacks.

Presidential Commission on Critical Infrastructure Protection

One of the first and best known initiatives within the United States for the assessment of large-scale vulnerability to interconnected infrastructures was the Presidential Commission on Critical Infrastructure Protection (PCCIP)[4]. Based upon the premise that national security is a shared responsibility, the PCCIP was a committee of experts from private industry, academic research communities, and government agencies. In their published report, submitted in October 1997, they concluded that while there is no imminent danger of catastrophic failure, the evolving complexity of these systems is creating an increasing number of threats [85]. The report also emphasized the

[4]The PCCIP was established in July 1996 by Presidential Executive Order 13010 and given a charter to formulate a comprehensive national strategy for protecting national infrastructures from physical and "cyber" threats.

growing research and development challenges associated with the design and control of large-scale networked systems.

The PCCIP had a number of direct outcomes on the perspective of the federal government toward infrastructure protection. Most notably, two new agencies were created by Presidential Decision Directive 63 (PDD-63) to address specific needs outlined by the PCCIP. PDD-63 established the National Infrastructure Protection Center (NIPC) to within the FBI to serve as the "government's focal point for threat assessment, warning, investigation, and response for threats or attacks against our critical infrastructures".[5] In addition, PDD-63 established the Critical Infrastructure Assurance Office (CIAO) within the Commerce Department as an inter-departmental policy coordination agency.

The efforts of the PCCIP have also directly impacted academic and industry research initiatives. In its final report, the PCCIP recognized a need for "improved simulation and modeling capability to understand the effects of interconnected and fully indepdendent infrastructures".[6] In particular, during the formal workshops both before and after the submission of the PCCIP report, there was a stated need for better understanding of the causes and behavior of cascading failures [5]. Some of these needs have started to be addressed through recent interdisciplinary research initiatives targeted at the large-scale behavior of network systems [7, 69].

Aftermath of September 11, 2001

In the wake of the terrorist attacks on the United States on September 11, 2001, there has been renewed attention on the protection of our national infrastructures. Initiatives such as the creation of the Department of Homeland Security are evidence of a fundamental change in the way we think about protection from terrorism—the responsibility for protection has shifted from local policing to the authority of the federal government. As our national adversaries obtain greater sophistication, it is clear that we must consider a range of possible threats. While protecting ourselves against the use of weapons of mass destruction will remain a high priority, it is becoming

[5] See NIPC web site, http://www.nipc.gov

[6] See [86], page 8.

apparent that we must also protect ourselves against the intentional exploitation of fragilities in our interconnected infrastructure systems.

1.2 Research Objectives

This research has been motivated by a desire to understand the extent to which the network structure of these systems contributes to their vulnerability. This thesis is meant to be a first step toward this broad, ambitious goal by achieving the following objectives.

1. *Frame the the large-scale failure and recovery behavior of infrastructure systems as a problem in network dynamics.*

2. *Identify and model the issues of load sensitivity in network components.* Develop an analytical framework that enables the quantitative and qualitative understanding of how network components can fail from overloading. Establish control mechanisms that optimize component performance while guarding against collapse.

3. *Identify the issues of network connectivity and load sensitivity that lead to cascading failures.* Extend the single component model to address canonical forms of network systems.

4. *Identify and develop domain-specific applications for further study.*

Throughout this this thesis, a number of reoccurring themes will emerge, including the operating and design tradeoffs between efficiency and robustness, the use of global versus local control, and the role of information for achieveing near-optimal solutions.

1.3 Outline for this Document

This thesis is organized in the following manner. Chapter 2 develops the methodological perspective for this study based on ideas from network dynamics and problems

in flow networks. In Chapter 3, we present the basic model for understanding the behavior of a congestion-sensitive network component. We use a dynamical systems approach to thinking about component behavior, and develop a qualitative understanding of congestion collapse. In Chapter 4, we assume complete predictability and deterministic operation of the stand-alone network component in order to develop policies for its optimal control. Then in Chapter 5 we extend this framework to include stochastic component behavior—that is, we consider cases when the inputs or behavior of the system can only be characterized probabilistically. We characterize the form of an optimal control policy for this uncertain case. In Chapter 6, we discuss birth-death processes as a specific type of stochastic model that yields insight into the behavior and optimal control of our system. In Chapter 7, we consider basic models for network systems. We consider network systems comprised of components in parallel, and show how such models appropriately represent certain types of load balancing computer systems. We also examine systems comprised of components in tandem, and we show how such models are appropriate for transportation systems like the Union Pacific Sunset Route. We conclude in Chapter 8 with a summary of our contributions and an outline of ongoing and future research initiatives in this area.

Chapter 2

Methodological Background

Traditional concepts from the theory of networks provide a convenient foundation for the study of network infrastructure behavior and the potential for cascading failures. Even so, addressing the large-scale vulnerability of national infrastructure systems is a daunting task. Each of the aforementioned systems (e.g. transportation, telecommunication, electric power) has a great deal of specialized structure that makes them dissimilar at highly detailed levels of comparison. So while it is clear that most national infrastructure systems exhibit strong network characteristics, it is not clear *a priori* that these characteristics have a decisive role in their vulnerability or robustness to cascading failures.

The purpose of this chapter is twofold. First, we introduce more formally the concept of cascading failures, and we review some of the previous efforts to understand their behavior. Second, we show that these previous efforts have been incomplete, since they have not addressed the large class of problems represented by network flow models. As a result, we will show that there is a need for the development of an analytical framework for the investigation of cascading failure behavior in network flow systems.

2.1 What is a Cascading Failure?

The notion of cascading behavior is a familiar one—one often thinks of a cascading waterfall, or the domino effect. That is, an initial disturbance causes a local failure, which leads to another failure, and then another, and so on. But what exactly is a cascading failure, and what is required to model it? In order to investigate the behavior of cascading failures, it is important to be precise in our definition. In particular, we must do the following.

1. Define our network components and what it means for them to *fail*.
2. Describe the interdependence between components, as well as the *mechanism* by which the failure of one component can lead to the failure of another. In general, we assume that this mechanism is part of the normal interaction of components within the system, so the cascading behavior is a natural by-product of this interaction.
3. Characterize the *large-scale behavior* of the cascading phenomenon.

Any model for casacading failures should do these three things. Thus, a cascading failure starts with an *initiating event*, and in most cases it does not matter whether this event is the result of an *accident*, internal *failure*, or explicit *attack*. Notice also that by this definition, a cascading failure is *not* any of the following: (1) a single point of failure; (2) the occurrence of multiple, concurrent failures; or (3) a contagion phenomenon, such as might be exhibited by a computer virus, that changes the behavior of the system as it spreads.

In the next section we review some previous work in modeling cascading failure behavior. Before doing so, however, it is worthwhile to frame cascading failures within the broader context of network dynamics.

2.1.1 Network Dynamics

In this study of cascading failures, we are interested in the *dynamic* properties of the network. But what does one mean by *network dynamics*? Network dynamics generally means one of two things: (1) dynamics *on* networks, or (2) dynamics *of*

networks.[1] In the context of dynamics *on* networks, one generally assumes that the network topology is static, and the feature of interest is the behavior of the system on top of that fixed topology. In contrast, when considering the dynamics *of* networks, the topic of interest is how the network topology itself evolves over time.

Where do cascading failures fit in the context of network dynamics? Cascading failures are a by-product of the interaction between dynamics on networks and dynamics of networks. If it is reasonable to take the perspective that the failure of a network component is equivalent to its removal, then a component failure is a change to its topology. However, a change in topology undoubtedly also affects the behavior on top of that topology. If that modified behavior in turn leads to yet another component failure (say, from overload), then the behavior on the network has affected its topology as well. In the case where this process repeats itself again and again, we say that a cascading failure has occurred. In this manner, a cascading failure can be interpreted as a positive feedback loop between changes in the dynamics on and of the network.[2]

2.2 Previous Work in Cascading Failures

The literature specifically focused on cascading failure behavior is rather sparse, and most of the previous investigations have been limited to specific application domains. Foremost among these is the electric power industry, where cascading failures have been of great interest since the 1960s.[3] More recently, in the aftermath of the Western Power Outages of 1996, there have been several directed efforts to gain insight into the large-scale fragility of the electric power grid to cascading failures. For example, the use of simulation models to investigate the swing-equation dynamics of electric power systems has shown that the distribution of failure sizes is well-approximated

[1] This distinction was previously made by Duncan Watts and Jim Crutchfield at the Santa Fe Institute.

[2] A complementary perspective can be taken for the process of network *recovery* and *growth*, where it is possible for feedback to accelerate the process by which repaired or new nodes join the active network topology.

[3] The Northeast Blackout of 1967 led to the formation of the North American Electric Reliability Council (NERC), which is chartered with the overall prevention of cascading failures within the grid.

by a power-law [47], a result that is consistent with the empirical data for outages in the U.S. over the last 100 years [6]. A separate investigation has focused on pinpointing the most vulnerable locations in a real power system, determining appropriate measures of vulnerability, and evaluating various solutions for economical protection systems upgrades [125].

In the following sections, we review several past efforts to investigate real (and theoretical) cascading failure behavior. Our focus in on the models proposed and the results obtained, as these investigations will be the starting point for the development for our own modeling framework.

2.2.1 A Binary-State Threshold Model

The simplest model for a network component is a *binary* representation, in which the state of the component can have one of two discrete values. A simple model of this type was used by Watts [126] to characterize cascading behavior in power grids and the propagation of fads in social networks. This approach had been used previously to model 'bandwagon effects' in social behavior, particularly the collective behavior of consumer demand (see [55] and references therein). In this modeling framework, the binary state of a network node is affected only by the collective state of its immediate neighbors. The specific relationship is given by a *threshold function*, which specifies how many neighboring nodes need to be in a certain state to induce a node to be in the same state.

- In the context of electric power grids, each node represents a piece of equipment and a link between them represents the presence of a physical wired connection. The binary state of a node represents its {normal, failed} state. The threshold function represents the number of neighbors that need to fail before the node fails.

- In the context of social networks, nodes represent individuals, links between nodes represent social relationships between them, and the state of each node represents some social choice, for example the choice between {Coke, Pepsi} for

cola preference. The threshold function indicates the number of one's friends that must be of the other type before one will switch brand loyalty.

Using a mathematical framework from random graph theory and problems in percolation, Watts focuses on developing measures for the probability that a cascade will result from the change in value of a single node as well as measure for the expected size of a cascade once it is triggered. His results show that cascades tend to be infrequent, yet large when they occur. Also, large cascades in these models are generally hard to predict, a result that is consistent with previous results (again, see [55]) that show the propensity of these types of models to chaotic dynamics.

While interesting from the theoretical perspective, this framework is of limited practical value to the study of critical infrastructure systems. A primary reason for this is that it is doubtful that real infrastructure systems are subject to the same type of threshold mechanism except at perhaps a phenomenological level. In addition, the nature of the mathematical results of this approach are derived from an assumption that the network topology is consistent with that of a random graph. Real infrastructure networks, like other highly engineered systems, do not satisfy this assumption. Nonetheless, this framework successfully illustrates that cascading failure behavior is possible even from simple interactions among network components, and it supports the need for careful treatment of the qualitative and quantitative aspects of complex network systems.

2.2.2 A Finite-State Probability Model

Another approach to modeling the behavior of an individual network component is to consider a *probabilistic* representation, such as afforded by a Markov chain. Using this perspective, the state of a component can vary among a finite number of possible values, and the transition between these values is governed by the probability of "jumping" from one value to another. This approach allows for a great deal of modeling flexibility in terms of the number of states and the parameters affecting the behavior which can be incorporated into the probabilities. It is possible to use this type of model to consider systems of networked components, however the drawback

of such an approach is that it requires complete specification of the probabilistic interactions between states. Since the total complexity of the framework increases as a product of the number of network components and the number of states per network component, even systems of moderate size can become fairly unmanageable.

A novel framework called the *Influence Model* was introduced in the PhD dissertation by Asavathiratham [9] to represent the discrete-time dynamics of networked, interacting Markov chains. By imposing certain constraints on the nature of the interaction between network components, the method allows for tractable analysis of large-scale network behavior. In particular, the model assumes that each node is represented by a Markov chain whose transition matrix is dynamically influenced by the state of its neighbors. The nature of this influence for a given network node in a single iteration of the method is as follows. According to a predetermined probability vector, either the node or one of its neighbors is chosen, and then the transition matrix for the original node is influenced by the state of the node that was selected. Since a node that influences itself behaves exactly as an isolated Markov chain, the Influence Model natually allows for the consideration of varying levels of interdependence between network components.

The Influence Model has been used effectively to investigate a number of aspects of cascading failure behavior. In a network of simple binary {Normal, Failed} components, the model dynamics under certain parameters have shown that network connectivity is two edged sword [10]. In particular, the presence of neighboring nodes can be *helpful* (in that it allows failed nodes to recover more quickly), yet also *harmful* (in that it may cause Normal nodes to fail more quickly). It is believed that similar tradeoffs are present in electric power grids, where the load carried by electric components depends on the availability of neighboring components, and overloading of components can lead to failure. The Influence Model has been applied to networks of homogeneous and nonhomogeneous network components [10], it has also been used to examine the large-scale dynamics of network growth and failure in general complex systems [105]. A recent case study of the Influence Model was used to investigate the tradeoffs in policies for allocating resources for optimal maintenance and repair of nodes [106].

The challenge in the practical application of the Influence Model is identifying the appropriate influences at work for real infrastructure systems. Again, the approach gives up some flexibility in the type of interactions that it can represent in exchange for greater analytical tractability. Nonetheless, it has been shown to be an effective, yet general framework for answering questions about the nature of the global network behavior that arises from spatial and temporal interactions among its components.

2.2.3 Other Related Work

Survivable Networks

Within the telecommunication and computer networking communities, significant attention has been giving to the study of "survivable networks" or "fault-tolerant networks" [2]. Generally speaking, the underlying assumption in this work is that network failures occur indepedently of one another. Then the focus is to study the likelihood and consquences of particular failure scenarios, with a goal of designing networks that are protected for all likely scenarios. This work is very interesting and has yielded many contributions to the resilience of computer and telecommunications networking technologies. However, it precludes by assumption the possibility of cascading failures and is therefore of limited value for this study.

Vulnerability and Robustness of Complex Systems

Within the last decade, there has been tremendous interest from the scientific community in understanding the large-scale vulnerability of complex systems. This interest has been motivated by numerous observations that failure event sizes in natural and engineered systems exhibit great variability—that is, the size and frequency of these events can often be described in terms of *power laws* which indicate that catastrophic events are likely to occur, albeit rarely.

Currently there are at least two possible explanations for the presence of these large failure events. The first is called *self-organized criticality (SOC)* [14] and is a product of the physics community. The focus of SOC is on the *phase transitions* that occur between order and disorder in natural, random systems. SOC postulates

that there is a direct relationship between the presence of these phase transitions and overall system complexity. Using the methods of statistical mechanics, advocates of SOC have demonstrated a tendency for many random systems to self-organize at these critical thresholds, at which one can observe power-law behavior (for an example in the context of forest fires, see [77]). This viewpoint has represented the conventional perspective of the physics community over the last decade.

An alternative explanation, known as *highly optimized tolerance (HOT)* [30], claims that the commonly-observed highly variable event sizes in systems optimized by engineering design are the result of tradeoffs between yield, cost of resources, and tolerance to risk. As a result, HOT systems are characterized by high performance, highly structured internal complexity, apparently simple and robust external behavior, and have the risk of hopefully rare but potentially catastrophic cascading failures initiated by possibly quite small perturbations [31]. By emphasizing the importance of design, structure, and optimization, the HOT concept provides a framework in which robustness in complex systems is a constrained and limited quantity that must be diligently managed. To date, the HOT concept has proven to be a powerful and predictive theory for generating power law event sizes [130].

Several other approaches have proposed qualitative explanations for the presence of large failure events. One such explanation is the concept of *normal accidents* [89] and has evolved from the sociology domain. The notion of normal accidents says qualitatively that systems with complex (nonlinear) interactions and tight coupling are more apt to experience system-level failures that disrupt the ongoing ability of the system to perform its intended task. The suggestion from this type of analysis is that organizations or infrastructure systems whose failure can have catastrophic potential (e.g. nuclear power plants, petrochemical manufacturing) should be abandoned or restricted until such time as their design and management can be improved so as to minimize this potential. Most of the other work on cascading failures has been phenomenological, rather than technical—particularly in the public policy work on protecting critical infrastructures [74, 85].

In all of the aforementioned theories, the general consensus is that there is a strong association between the growing complexity in our large-scale systems and a growing

vulnerability to catastrophic events.

2.2.4 A Need for Something More

While the aforementioned models for cascading failures provide a useful starting point, they are of limited use for the investigation of critical infrastructure systems. Most of the quantitative models have been developed specifically for application to electric power systems. However, the physics of electricity dynamics are rather different than the "physics" of other infrastructure systems. As a result, there is a need for an alternative approach, particularly one that addresses the needs of transportation and telecommunication systems. One promising approach for doing this comes from the perspective of *network flow models*.

2.3 Network Flow Models

This section reviews some of the basics of network flow models and outlines a framework for studying cascading failures in this context. In particular, we argue that *congestion* is a primary mechanism by which cascading failures can occur in flow networks.

2.3.1 Resource Allocation in Networks

Networks allow the sharing of distributed resources. Typically, these resources provide services in the form of *transport* and *processing*. For example, the telephone network enables the sharing of telephone circuits (providing transport of voice, data, and control signals) and telephone switching equipment (providing call setup and route processing). While it is prohibitively expensive for an individual to build her own telephone lines to place her calls, the network allows her to share a collection of circuits with millions of other users in a manner that reduces the overall cost.

In the context of a network, we often measure a resource in terms of its *capacity* and its use in terms of the *load* that is placed upon it. By extension, a network has an aggregate capacity and its total usage can be measured in terms of its aggregate load.

Observe that this total load is *distributed* among the individual resources within the network. In fact, many networking problems are concerned with finding a "good" distribution of load among the resources within the network. Thus, one can equivalently think of load distribution or resource allocation.

In general, a network structure becomes increasingly cost efficient as sharing enables higher utilization of expensive resources. Therefore, if minimizing the operating cost of a network is of primary concern, the network manager is going to have incentive to maximize its utilization. In the absence of disruptions, such policies are obvious. However, most real networks are susceptible to the occasional *failure* (loss) of a resource. One strategy for handling the occasional loss of network resources is to maintain some *reserve capacity* in the network, thereby allowing for redundancy. The choice of optimal amount and distribution of reserve capacity will depend on the *operating policy* of the network manager, the traffic characteristics of the network, and her beliefs about potential failures.

2.3.2 Congestion in Networks

Most networks have a finite number of available resources. Often, this is the result of high resource cost or resource scarcity. (As noted before, it is these properties that motivate the sharing of resources via the network structure in the first place.) Recall that the use of a resource is measured in terms of its load, and the maximum amount of load that can be handled by a resource is its capacity. What happens when there is more demand for a resource than capacity? The typical convention in network systems is that a resource will accept load up to its capacity, then any additional demand will either be blocked or forced to wait until the resource becomes available at some time in the future. When this occurs, the resource is said to experience *congestion*. This type of time-sharing means that critical resources are fully utilized under conditions of congestion. Along these lines, the severity of congestion is typically measured in terms of the quantity of load that is waiting. As a result of having a limited number of finite capacity network resources, a key issue for networks is the extent to which contention for these limited resources results in large-scale congestion within

the system.

Congestion is a key challenge for network managers. When there is contention for critical network resources, the performance of the entire network can seriously deteriorate. The congested resources act as bottlenecks preventing other parts of the network from being completely utilized. The deterioration of network performance can occur even in the absence of failures in the network components themselves. This phenomenon is simply the result of the combinatorial nature of resource dependence in network structures, and a great deal of attention in the study of networks has been devoted to understanding this aspect.

2.3.3 Hierarchy of Network Flow Models

In order to answer key operational and network management questions, engineers and applied mathematicians have developed many types of network flow models. Network flow models are sometimes called "stock-and-flow models" because they are interested in a particular quantity (the stock) and its movement (the flow) through the network. In this context, utilization of network resources comes in the form of storage, processing, and transport. Network flow problems are a particular type of resource allocation problem in which there is a notion of *conservation of flow* within the network. That is, stock is neither created nor destroyed as it moves through the network. Our convention in this study will be to refer simply to *network flow models* or *flow networks* when speaking of this broad class of models. Flow network modeling has been successfully used for many applications including transportation, telecommunications, production planning, and inventory control.

There is a hierarchy of network flow models that has been used to investigate phenomena at different levels of granularity.[4] Some of these models and their corresponding uses are listed in the table below.

[4]This taxonomy has been adapted from that in [48].

Model Granularity	Quantity of Interest	Model Type
coarse	long-term averages	static flow
↑	time-dependent averages	flow approximation
medium	averages and variances	diffusion approximation
↓	probability distributions	queueing
fine	event sequences	simulation

Table 1. Hierarchy of network flow models.

Often, models at different levels of granularity are associated with network management decisions on different time scales. Generally speaking, decisions for operations on short time scales, such as real-time processing and routing, are modeled using fine granularity. Conversely, decisions for management on long time scales, such as network capacity planning, are modeled using coarse granularity.

2.3.4 Failures in Flow Networks

While the relationship between congestion and large-scale network performance has been a key issue in the study of flow networks, there has been no treatment of an explicit relationship between congestion and failure in individual network components. That is, while the performance of the network may degrade with congestion, the individual components are generally assumed to continue to function properly.[5] In this study, we will consider *congestion-induced failure* in network components, and show that this mechanism is sufficient to induce cascading failures in flow network systems.

Although cascading failures in flow networks have not received explicit treatment within the aforementioned hierarchy of models, it should be intuitively clear that large-scale network failure and recovery occur over medium time scales. That is, we expect the spread of component failures from congestion to occur on time scales that are longer than the movement of individual flows and shorter than the time scales on which the network can be provisioned with additional resources. The point is

[5]In cases where the resulting performance degradation is severe, the appearance to the network user may be that the system has indeed failed even in the absence of individual component failures.

that failure and recovery sits in the middle of this model hierarchy, and we will use a number of these models in the course of this investigation. These notions will become clearer as we begin to build our model of cascading failures.

2.4 Cascading Failures in Flow Networks

We need to answer the following questions en route to the development of a framework for cascading failures in network flow systems.

1. What is the mechanism that leads to the failure of a network flow component?
2. How does the failure of one component lead to the failure of another?
3. Under what conditions will failure actually spread?

The following sections provide an overview of the approach and set the stage for the introduction of a quantitative model in the next chapter.

2.4.1 A Simple Model of Component Failure

We will assume that network nodes can fail as a result of *overload*. The details of how this happens is the subject of the next chapter. In the meantime, it is sufficient to characterize a component as being in one of the following states.

- *normal*—the component is uncongested and has spare capacity for additional load
- *congested*—the component is overloaded and does not have any spare capacity, performance may also be suffering
- *failed*—the component has ceased operating and will remain so in the absence of repair

In the context of the finite-state models mentioned earlier in the chapter, we have a simple model for the lifecycle of an isolated network component. The possible transitions from one state to another is illustrated in Figure 2.1.

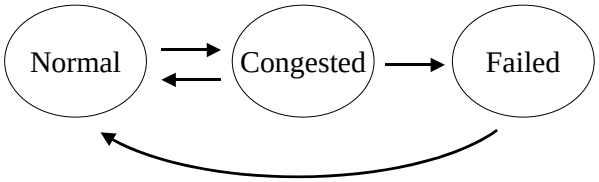


Figure 2.1: *Simplified failure state machine.*

2.4.2 Component Interdependence

In a network flow system, individual network components are inherently linked by the flow conservation principle. The total system load must be distributed among all components, and it cannot be reduced by changes in the operating policy. If a component fails, its load must be redistributed among the other components. A key challenge for network management is the manner in which load is distributed (and redistributed) among the network resources. In the event that this (re)distribution leads to additional overload in other parts of the network, then additional failures may result.

2.4.3 Conditions for Failure Cascade

Cascading behavior starts with the failure of a single network component. In order for the cascade to occur, however, the failure of must spread to another node before the failed node can recover or is repaired. Consider the following simplified scenario. When the single failure occurs, the network begins to redistribute the total system load over the remaining components. At the same time, the failed component may also begin the process of repair. If the repair happens quickly, it may be possible for the network as a whole to return to its original allocation of load. If the repair takes a relatively long time, then the network as a whole must find a different distribution of load. If the new distribution results in other congested nodes, then other failures are possible, and a cascade may result.

2.5 Chapter Summary

By extending existing network flow models to include congestion-induced failure of network components, it is possible to develop a comprehensive framework for the study of cascading failures. As indicated above, a key challenge for network managers is load allocation and redistribution among network resources in the presence of a failure. As we develop our understanding of cascading failure behavior in flow networks, our interest will be focused on issues of network operation, control, and design. Of particular interest will be the *tradeoff between efficiency and robustness* for individual components and the network as a whole. Our objective will be to identify and develop appropriate management policies that lead to desired performance.

Chapter 3

The Congestion-Sensitive Network Element

The purpose of this chapter is to develop the basic model of the congestion-sensitive network component. In this chapter, we will restrict our attention to models in which all system dynamics and inputs are deterministic.

3.1 Basic Model

Consider an input-output system whose state is characterized in terms of its current amount of work, denoted in continuous time as $x(t)$. The behavior of this system is relatively simple: work arrives, work is processed, and work departs. Mathematically, the system evolves in continuous time according to the following equation.

$$\dot{x}(t) = A(t) - D(t)$$

Here, $x(t)$ is the amount of work in the system at time t and $A(t)$, $D(t)$ are the respective arrival, departure rates of work in the system at time t. We will denote such a system a *processing system* or generically just a *processor*. The setup of the system is illustrated in Figure 3.1 below.

Dynamical systems of this type have been studied for more than 50 years. Some-

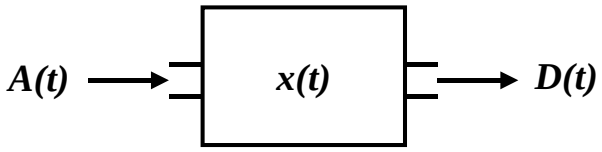


Figure 3.1: *Basic input-output processing system.*

times, these systems are known as *stock-and-flow* systems [50]. Such input-output systems have been used to model system behavior in production systems, population dynamics, fluid flows, queueing behavior, and many other important phenomena in economics and engineering [76, 34].

3.1.1 Workload Functions

The departure rate for a system of this type depends on the system state and system time; that is it is of the form $D(t) = w(x(t), t)$. The physical interpretation is that the system output is a function of both the current work and the current time. For the purposes of this study, we will restrict ourselves to stationary systems, so $D(t) = w(x(t))$. We will call the function w a *workload function*. It relates the amount of work (or load) in the system to its output rate. Although it is not required by the mathematics, we generally assume that a workload function satisfies the condition $0 \leq w(x) \leq x$ for $x \geq 0$, which corresponds to a physical notion that the system cannot output more work than it has. In such cases, clearly $w(0) = 0$.

Input-output processing systems can be characterized in terms of their workload function. For example, if the workload function has a finite maximum, then we say that the processor is *capacitated*, meaning that there is a finite limit to the amount of work than can be processed per unit time. A common form of a capacitated workload function is $w(x) = \min(x, K)$, where K represents the finite capacity of the system. An example of a system with this form is the K-server queue, in which each server processes work at unit rate. The corresponding workload function is is illustrated in Figure 3.2.

To say that a workload function is capacitated or uncapacitated is an incomplete characterization. Of equal (or greater) concern is the manner in which the output

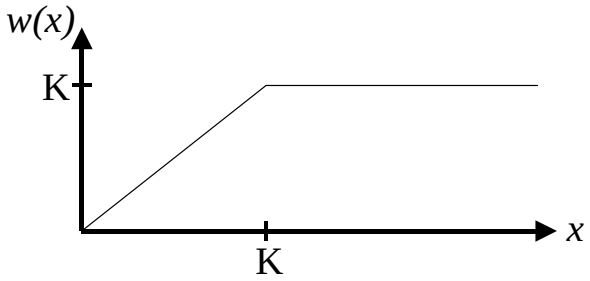


Figure 3.2: *The capacitated workload function* $w(x) = \min(x, K)$.

rate changes with the amount of load on the system. If for any range of x, $dw/dx < 0$ we say that the processor is *sensitive to congestion*. For the purposes of this study, we will consider workload functions that are *quasiconcave* in x. For univariate functions $w(x)$, this means that there exists a point x^* such that when $x \leq x^*$ the function $w(x)$ is nondecreasing and when $x \geq x^*$ the function $w(x)$ is nonincreasing.

Our particular interest is in processors that are susceptible to *congestion collapse*. That is, we are interested in cases where the workload function has the limiting behavior $w(x) \to \Omega \geq 0$ as $x \to \infty$. Quite often, we can identify a finite state $M < \infty$ for which $w(M) = \Omega$. We call M the *collapse point* of the system. In many practical systems, M represents the point at which operator intervention is required to return the system to normal operation. For example, consider the case of a busy roadway or intersection in which police officers need to direct traffic out of a traffic jam. An alternative example occurs in the context of a computer network whenever a system administrator needs to reboot a computer server or other piece of data communications equipment. In these cases where the collapse point M is known, we are interested in the evolution of the system on the restricted state space $[0, M]$. If at some time $t^* > 0$, we have $x(t^*) = M$, then we say that the system has *collapsed* at time t^*.

Of particular interest for this investigation are the following three congestion-sensitive workload functions.

Workload Function 1

The first workload function of interest is defined by the following relationship.

$$w_1(x) = xe^{-bx}$$

This function is continuous with continuous first derivative on the interval $x \geq 0$. Observe that it satisfies our definition for a valid workload function (since $xe^{-bx} \leq x$ for $x \geq 0$ and $\lim_{x\to\infty} xe^{-bx} = 0$). Furthermore, the first derivative of this function is

$$\begin{aligned} w_1'(x) &= x(-b)e^{-bx} + e^{-bx} \\ &= e^{-bx}(1 - bx). \end{aligned}$$

The function attains a maximum when $x^* = 1/b$ and acheives a corresponding value $w_1(x^*) = 1/(eb)$. See Figure 3.3 for an illustration. Observe that when $x < 1/b$ then

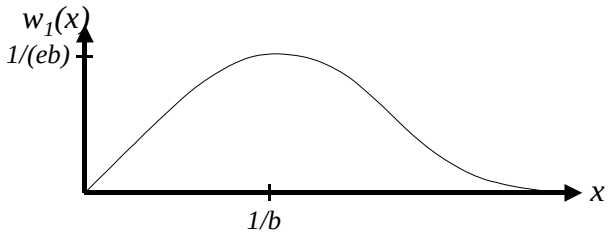


Figure 3.3: *Workload function* $w_1(x) = xe^{-bx}$.

$w_1'(x) > 0$ and when $x > 1/b$ then $w_1'(x) < 0$, so the function w_1 is a quasiconcave function.

Workload Function 2

The second workload function of interest is defined as the following.

$$w_2(x) = \begin{cases} x\left(1 - (x/M)^p\right) & 0 \leq x \leq M \\ 0 & x > M \end{cases}$$

Observe that this function satisfies the conditions for a valid workload function. The first derivative of this function is

$$\begin{aligned} w_2'(x) &= \frac{d}{dx}\left[x - \frac{1}{M^p}\, x^{p+1}\right] \\ &= 1 - (p+1)\left(\frac{x}{M}\right)^p. \end{aligned}$$

The function attains a maximum when $x^* = M\left(\frac{1}{p+1}\right)^{1/p}$ and acheives a corresponding value $w_2(x^*) = M\left(\frac{1}{p+1}\right)^{1/p}\left[1 - \frac{1}{p+1}\right]$. Clearly, when $x < M\left(\frac{1}{p+1}\right)^{1/p}$ then $w_2'(x) > 0$ and when $x > M\left(\frac{1}{p+1}\right)^{1/p}$ then $w_2'(x) < 0$, so the function w_2 is quasiconcave. The point M is the finite collapse point of the system.

Note also that when $p = 1$ the function is a parabola, and it attains maximum at $x^* = M/2$ at value $w_2(x^*) = M/4$. Now consider the limiting behavior of $w_2(x)$ as $p \to \infty$. Since

$$\lim_{p\to\infty} \frac{1}{p+1} = 0 \quad \text{and} \quad \lim_{p\to\infty} \left(\frac{1}{p+1}\right)^{1/p} = 1,$$

it is easy to see that

$$\lim_{p\to\infty} x\left(1 - \left(\frac{x}{M}\right)^p\right) = x\left(1 - \delta_M(x)\right) \quad \text{where} \quad \delta_M(x) = \begin{cases} 1 & x \geq M \\ 0 & x \leq M \end{cases}$$

and

$$\lim_{p\to\infty} x^* = M \quad \text{and} \quad \lim_{p\to\infty} w_2(x^*) = M.$$

This limiting behavior is summarized in Figure 3.4 below.

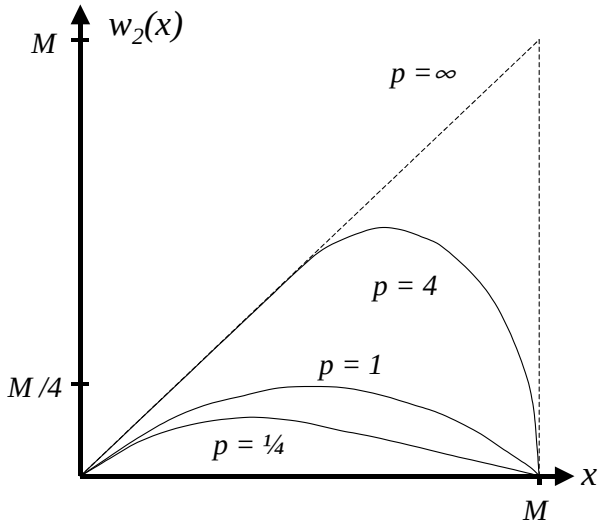


Figure 3.4: *The limiting behavior of workload function* $w_2(x)$ *as* $p \to \infty$.

Workload Function 3

The third workload function of interest is defined by the following relationship.

$$w_3(x) = \begin{cases} \frac{c}{a}\, x & 0 \le x \le a \\ c - \frac{c}{b}\,(x-a) & a \le x \le a+b \\ 0 & x > a+b \end{cases}$$

This piecewise linear function is illustrated in Figure 3.5 below. The first derivative of this function obviously is

$$w_3'(x) = \begin{cases} \frac{c}{a} & 0 \le x \le a \\ -\frac{c}{b} & a \le x \le a+b \\ 0 & x > a+b \end{cases}$$

and the function attains a maximum at $x^* = a$ with value $w_3(x^*) = c$. Clearly, this function is quasiconcave and has finite collapse point $M = a + b$. Provided that $c/a \le 1$, this function satisfies our conditions as a valid workload function, namely

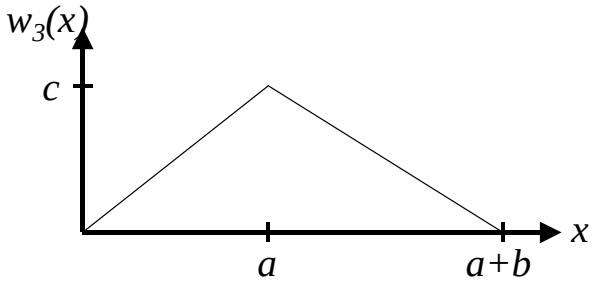


Figure 3.5: *Piecewise linear workload function* $w_3(x)$.

$w_3(x) \leq x$ for all $x \geq 0$ and $w_3(x) \to 0$ as $x \to \infty$.

Comments

These three functions are continuous and they all have the same basic unimodal shape, however each provides distinct insights. The function w_1 is continuously differentiable for the entire state space $x \geq 0$, and its shape has made it an attractive candidate for modeling of many processes in computer science. This function has been of particular utility in the analysis of random access network protocols [20, 100], most notably the ALOHA and ethernet protocols for which the system throughput is modeled using the functional form $w(x) = G(x)e^{-G(x)}$. The parameterization of the function w_2 makes it of great use in modeling a number of real-world processes. A number of transportation systems can be modeled using variations of w_2, and for the value $p = 1$ this function corresponds to the classic Greenshields model for transportation [84, 78]. However, the analysis of this function suffers because it is non-differentiable at the collapse point M. The piecewise linear nature of the function w_3 allows for convenient analysis within each of the intervals, but again suffers because of the discontinuity in its derivative at interval boundaries.

Collectively, these functions provide a flexible framework for the study of congestion-sensitive processors. Depending on the application under study, one of the above functions may be more appropriate than the others. For the remainder of this study, we will assume that all real-world processors of interest have workload functions that are reasonably approximated by one of these three forms.

3.1.2 Research Questions

The objective of the remainder of this chapter is to formalize these notions and to provide mathematical insight into the following research questions.

1. What is the behavior of a congestion-sensitive system under arbitrary known input?
2. Under what conditions will a system collapse? Or, what conditions will ensure that the system does not collapse?
3. What forms of control need to be applied to prevent collapse?
4. What should be done to optimize the performance of such a system?

The remainder of this chapter is organized in the following manner. Next in Section 3.2, we provide a qualitative analysis of the workload function and develop some basic intuition for the use of input rate control for managing system behavior. In Section 3.3 we obtain specifc results for the behavior of uncontrolled systems. This understanding is fundamental to the development of optimal control strategies in the next two chapters.

3.2 Qualitative Analysis

Before proceeding with the formal analysis of systems with congestion-sensitive workload functions, it is worthwhile to provide a qualitative summary of their behavior. There are two reasons for this. First, some of the the aforementioned research questions can be answered immediately using a straightforward qualitative approach. Second, this qualitative analysis will help to motivate our interest in its study and provide insight into the tensions and tradeoffs inherent in these systems.

3.2.1 Constant Input

Consider the steady state behavior of a processor under constant input rate λ. Again, we assume that the processor has a workload function consistent with the unimodal shape described previously. While a thorough treatment of the equilibria and stability

of systems under constant input will be treated later, there is tremendous insight that is immediate. Specifically, this insight can be obtained by simultaneously plotting the workload $w(x)$ and arrival rate $A(x) = \lambda$ (in this form to indicate that the arrival rate does not change with the state of the system) and examining their intersection points. Again, let x^* be the point at which $w(x)$ is maximized (i. e. $w(x^*) \geq w(x)$ for $x \geq 0$).

The first observation is that if $\lambda > w(x^*)$ then the system is unstable—the constant arrival rate is always greater than the amount of work that can be handled by the processor. We therefore consider the case where $0 < \lambda \leq w(x^*)$. Given the unimodal nature of the workload function, it should be apparent that there will be two points of intersection, x_1 and x_2, satisfying the general relationship $0 < x_1 \leq x^* \leq x_2$. We can gain additional insight by considering the overall behavior of the system for different values of x. Specifically, for $0 \leq x < x_1$, we observe that $w(x) < \lambda$ so x is *increasing* in that interval. Similarly, for $x_1 < x < x_2,\ w(x) > \lambda$ so x is *decreasing*. Finally, for $x > x_2,\ w(x) < \lambda$ so x is *increasing*. Figure 3.6 illustrates this analysis. We therefore conclude that x_1 is a *stable* equilibrium point and x_2 is an *unstable* equilibrium point.

We summarize this system behavior as follows. For a constant input intensity $\lambda \leq w(x^*)$, if the initial system state $x(0)$ satisfies $0 \leq x < x_2$ then the system will evolve to x_1. However, if $x(0) > x_2$ then x will grow without bound and the system will collapse.

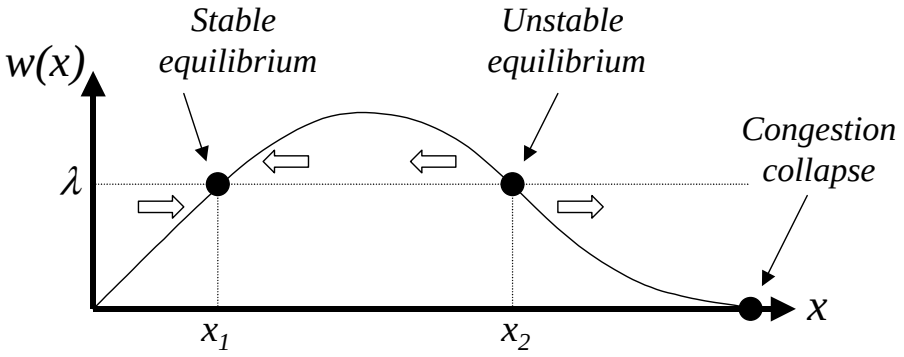


Figure 3.6: *Congestion-sensitive system under constant input rate.*

3.2.2 Operating Regimes

This preliminary analysis has implications for the operation of congestion-sensitive processing systems. Consider the case of an operator who has the ability to select the input level λ. From the previous analysis, it should be clear that

- there is a maximum allowable constant input rate, and
- whether or not a given input level leads to good (stable) behavior or bad (unstable) behavior depends on the current system state x.

In other words, the operator must choose an operating policy that depends on the current state of the system. One natural characterization of system state is to consider the following intervals (or *zones*) illustrated in Figure 3.7.

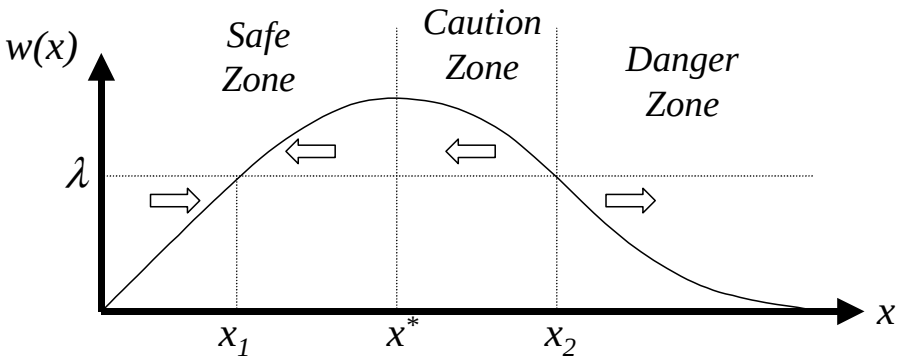


Figure 3.7: *Operating zones.*

- *Safe Zone*, $0 \leq x \leq x^*$. In this interval the system is uncongested. Increases in the input level result in increases in the overall system output rate. The system is stable to most perturbations in system state and tends to return to the stable equilibrium point x_1.
- *Caution Zone*, $x^* < x \leq x_2$. In this interval, the system is slightly congested. Additional increases in the input level result in *decreases* in the overall system output rate. The system tends toward the stable equilibrium point x_1, however large perturbations that take the system above the unstable equilibrium point x_2 will result in unstable behavior.

- *Danger Zone*, $x > x_2$. The system is severely congested, and increases in the overall input rate will dramatically reduce the overall system output rate. Furthermore, the system is unstable and will tend to the point of overall system collapse without a reduction in the input rate.

Based on this preliminary analysis, it is reasonable to believe that input rate control may be an effective means of manging the overall behavior of the system. Specifically, for cases when the system state is beyond the unstable equilibrium ($x > x_2$), the only way for recovery to occur is to severely restrict the input rate. Such strategy works because changing the input rate also changes the equilibrium points. For example, a change in the input rate from λ to λ' results in a change in the equilibrium points from x_1 and x_2 to x_1' and x_2'. Figure 3.8 illustrates this below.

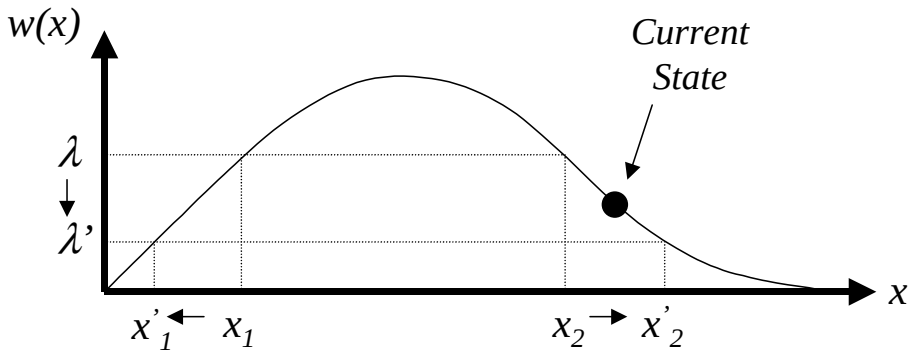


Figure 3.8: *System recovery via input rate reduction.*

3.2.3 Efficiency vs. Robustness

There is another caveat to the use of input rate control in this system. Consider a system under constant input rate $\lambda < w(x^*)$ and operating near the stable equilibrium point x_1. Now, assume the objective of the system operator is to maximize the efficiency of the system. It should be clear that the current operating point $x = x_1$ is *inefficient*, since the system has additional output capacity. If the system operator decides to increase the input level to $\lambda' > \lambda$ what happens? The stable equilibrium point increases from x_1 to x_1' and the system output rate increases. However, there

is another effect of this change. Recall that if the system state is perturbed such that $x > x_2$, then without a corresponding change in the input rate the system becomes unstable. Thus, for a system operating about the stable equilibrium point x_1, the distance $x_2 - x_1$ can be thought of as the *safety margin*—it represents the maximum perturbation that can be absorbed by the system without losing stability. For example, define $\phi(\lambda)$ to be the safety margin, and observe that $\phi(\lambda) \searrow 0$ as $\lambda \nearrow w(x^*)$. Thus, an increase in the input rate results in a *decrease* in the safety margin. If the size of the safety margin is a measure of system *robustness*, then it is clear that the price for increased efficiency is a loss of robustness. Figure 3.9 illustrates this tradeoff.

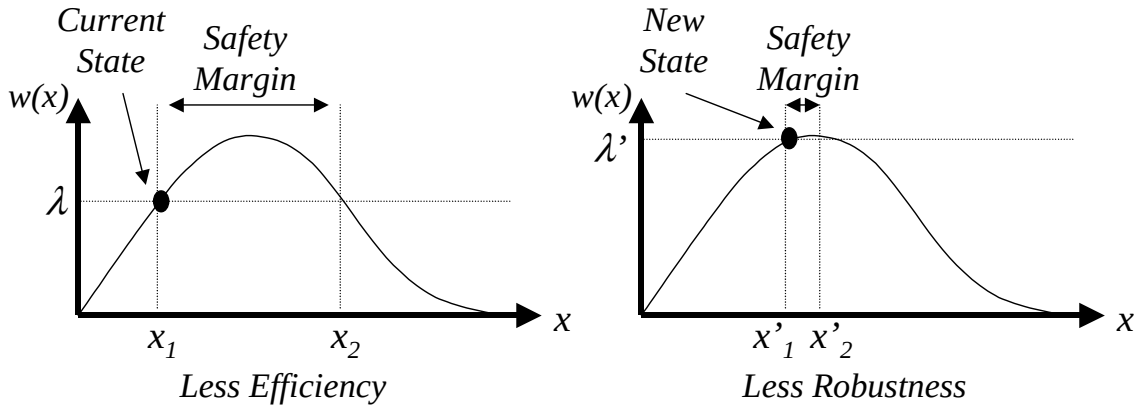


Figure 3.9: *Increasing input rate to raise efficiency results in a loss of robustness.*

The qualitative analysis presented here provides a great deal of insight into the behavior of congestion-sensitive systems under constant input. Furthermore, it illuminates some of the tradeoffs and tensions inherent to an operator who has the ability to set the input rate. However, in many real world applications a system operator will not be able to select this rate but must deal with a stream of external arrivals. In general, the arrival patterns for this incoming work may be irregular, and the operator may have limited ability to control the system under this input. Understanding the overall system behavior in the presence of arbitrary arrivals is the subject of the next section.

3.3 Uncontrolled System Behavior

Consider a congestion sensitive input-output system that does not use input control. This type of system according to the relation

$$\frac{dx}{dt} = A(t) - w(x)$$

where $w(x)$ is the workload function and $A(t)$ is the arrival stream. This uncontrolled system will evolve deterministically as a function of its initial state $x(0)$ and the arrival stream $\{A(t), t \geq 0\}$. Given a particular workload function, we would like to answer the following questions.

1. For which input streams is an exact solution $x(t)$ available?

2. As a step toward input control, consider the ability to accept or reject the entire arrival sequence $\{A(t), t \geq 0\}$. We call the sequence $A(t)$ *admissible* if it does not result in congestion collapse; that is, if $x(t) < M$ for all $t \geq 0$ under input $A(t)$. The relevant question then becomes can we obtain conditions on *admissibility* of input streams?

3.3.1 General Solution

Under the assumption that $w(x)$ exhibits the type of congestion sensitivity described previously and $A(t)$ is arbitrary, we have a system described by a nonlinear first order ordinary differential equation of the form

$$P(x,t)dt + Q(x,t)dx = 0$$

where $P(x,t) = w(x) - A(t)$ and $Q(x,t) = 1$. Using the standard approaches for integration, we observe that an equation of this type is solved exactly by a function $R(x,t) = \kappa$ (for an arbitray constant κ) if

$$\frac{\partial R(x,t)}{\partial t} = P(x,t) \quad \text{and} \quad \frac{\partial R(x,t)}{\partial x} = Q(x,t).$$

This is true if and only if $\partial P/\partial x = \partial Q/\partial t$. In our case, $\partial P/\partial x = w'(x)$ and $\partial Q/\partial t = 0$ so this equation is not exact. However, its relatively simple form may allow for the identification of an integrating factor $\mu(x,t)$ so that the equation

$$\mu(x,t)\left\{P(x,t)dt + Q(x,t)dx\right\} = 0$$

is exact. Thus, we require

$$\begin{aligned} \frac{\partial}{\partial x}\left[\mu(x,t)\left\{w(x) - A(t)\right\}\right] &= \frac{\partial}{\partial t}\left[\mu(x,t)\right] \\ \mu(x,t)w'(x) + \frac{\partial \mu(x,t)}{\partial x}\left\{w(x) - A(t)\right\} &= \frac{\partial \mu(x,t)}{\partial t} \end{aligned}$$

Thus, we can find the general integrating factor by solving this nonlinear, first-order partial differential equation. In general, this task is quite difficult, and we must look at specific arrival and workload functions for convenient structure.

3.3.2 Solution for Piecewise Constant Input

In cases where one can reasonably assume input to be piecewise constant, then the solution is simplified. For any constant input value λ, the system is governed by the ordinary differential equation (ODE)

$$\frac{dx}{dt} = \lambda - w(x).$$

This equation is separable, and can be solved by integrating both sides.

$$\int \frac{dx}{\lambda - w(x)} = \int dt$$

The extent to which this approach is convenient depends on the form of the workload function.

3.3.3 Solution for Workload Function 1

Recall that under this workload function the dynamical system of interest evolves according to

$$\frac{dx}{dt} = A(t) - xe^{-bx}$$

for $x(t) \geq 0$. Alhtough this workload function has the nicest shape from a modeling perspective, this ODE is the least tractable. In particular, consider the seemingly simple case of zero input ($A(t) = 0$). Under this assumption, the ODE simplifies greatly to

$$\frac{e^{bx}}{x}\, dx = -dt$$

which can be solved by integrating both sides. However, the integral of the LHS has a power series solution of the form

$$\begin{aligned} \int \frac{e^{bx}}{x}\, dx &= \log x + \frac{bx}{1} + \frac{1}{2}\frac{(bx)^2}{1 \cdot 2} + \frac{1}{3}\frac{(bx)^2}{1 \cdot 2 \cdot 3} + \dots \\ &= \log x + \sum_{j=1}^{\infty} \frac{1}{j}\frac{(bx)^j}{j!}. \end{aligned}$$

Unfortunately, this form does not lend itself nicely to an analtyical solution for $x(t)$. However, it is possible that other variations of this functional form might yield to simplified analysis. For example, all functions with the more general form

$$w(x) = G(x)e^{G(x)}$$

will exhibit the same qualitative shape of $w_1(x)$. While it is possible that a convenient functional form might be obtained during the normal course of modeling a specific application, it is beyond the scope of this study to exhaustively explore and analyze all possible functions.

3.3.4 Solution for Workload Function 2

Recall that under this workload function the dynamical system of interest evolves according to

$$\frac{dx}{dt} = A(t) - x\left(1 - \left(\frac{x}{M}\right)^p\right)$$

on the interval $0 \leq x \leq M$. This system can be rewritten in the more familiar form

$$\left\{x\left(1 - \left(\frac{x}{M}\right)^p\right) - A(t)\right\} dt + dx = 0.$$

However, the general solution to this ODE is also difficult to solve. Fortunately, there is much that can be said about the behavior of the system for particular values of $A(t)$ or p.

Zero Input Case

When $A(t) = 0$, the ODE simplifies greatly to

$$\frac{dx}{dt} = x\left(1 - \left(\frac{x}{M}\right)^p\right)$$

and can be solved explicitly. With initial condition $x(0) = x_0$, we obtain the following.

$$x(t) = \frac{x_0 M}{(x_0^p + (M^p - x_0^p)e^{pt})^{1/p}}$$

Observe that in the case where $p = 1$, this is the familar form of logistic decay.

$$x(t) = \frac{x_0 M}{x_0 + (M - x_0)e^t}$$

Quadratic Case ($p = 1$)

It is perhaps not surprising that additional analysis is possible when the system is quadratic in x (when $p = 1$). Before proceeding with this analysis below, we remark that it is convenient to rescale the system by letting $y = x/M$. Then, substituting

$x = My$ and $dx/dt = M dy/dt$, we obtain

$$\frac{dy}{dt} = \frac{1}{M} A(t) - y\,(1 - y^p)$$

where $0 \leq y \leq 1$. It should be clear that this form is equivalent to our original system. We therefore restrict the subsequent discussion of this system to be defined by y. This system can be analyzed for constant input $A(t)$.

Dynamics Under Constant Input

Note that when $p = 1$ the function is a parabola, and in the rescaled system the workload function attains maximum at $y^* = 1/2$ with value $w_2(y^*) = 1/4$. Let λ represent the rate of constant input, rescaled so that a value $\lambda = 1$ corresponds to the maximum feasible arrival rate. Specifically, for $p = 1$ and a constant arrival rate for the original system of $A(t) = \kappa$, then $\lambda = \kappa/M$ and the system evolves according to

$$\frac{dy}{dt} = \frac{\lambda}{4} - y + y^2$$

where the state space is restricted to $0 \leq y \leq 1$ and $0 \leq \lambda \leq 1$ correspond to feasible arrival rates.

It should be clear that $\lambda > 1$ corresponds to arrival rates that are greater than the maximum achievable output rate of the system, so the system is unstable whenever $\lambda > 1$. Next, we discuss the equilibria and stability of this system under a constant arrival intensity λ, where $0 \leq \lambda < 1$. Equilibrium occurs whenever $dy/dt = 0$, that is, when

$$\frac{\lambda}{4} - y + y^2 = 0$$

which corresponds to the fixed points $y^* = \frac{1}{2} \pm \frac{1}{2}\sqrt{1 - \lambda}$.

One approach to calculate the stability of this system is to linearize around the fixed points and inspect the first derivative of the linearized system. However, from our previous analysis we already concluded that

- $y_1^* = \frac{1}{2} - \frac{1}{2}\sqrt{1 - \lambda}$ is a *stable* equilibrium point, and

- $y_2^* = \frac{1}{2} + \frac{1}{2}\sqrt{1-\lambda}$ is an *unstable* equilibrium point.

Furthermore, recall that we defined the *safety margin* as the distance between the equilibrium points and that this safety margin was a function of the arrival rate. Thus, define $\phi(\lambda)$ to be the safety margin, and observe that for $p = 1$ it has functional form $\phi(\lambda) = \sqrt{1-\lambda}$. These relationships are illustrated in Figure 3.10 below.

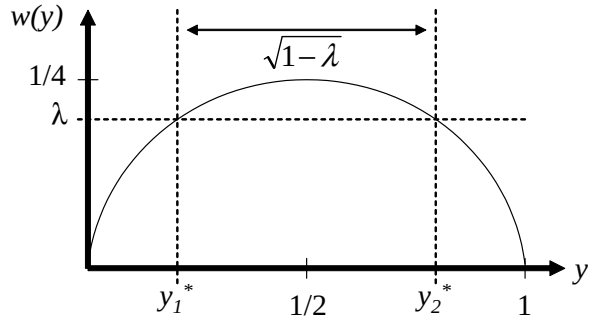


Figure 3.10: *Equilibrium points and safety margin for the normalized parabolic workload function* w_2.

In order to obtain relationships that describe the exact dynamics of this system under constant input, we must consider three cases. In all cases, we only consider dynamics of the state variable on the interval $0 \leq y \leq 1$ with initial state $y(0) = y_0$.

Case 1. $0 \leq \lambda < 1$

$$y(t) = \frac{(\frac{1}{2} + \frac{1}{2}\sqrt{1-\lambda})(y_0 - \frac{1}{2} + \frac{1}{2}\sqrt{1-\lambda}) + (-\frac{1}{2} + \frac{1}{2}\sqrt{1-\lambda})(y_0 - \frac{1}{2} - \frac{1}{2}\sqrt{1-\lambda})\ e^{\sqrt{1-\lambda}\ t}}{(y_0 - \frac{1}{2} + \frac{1}{2}\sqrt{1-\lambda}) - (y_0 - \frac{1}{2} - \frac{1}{2}\sqrt{1-\lambda})\ e^{\sqrt{1-\lambda}\ t}}$$

Observe that when $\lambda = 0$ this reduces to the previously discussed logistic decay function.

$$y(t) = \frac{y_0}{x_0 + (1-y_0)e^t}$$

Case 2. $\lambda = 1$

$$y(t) = \frac{y_0 - \frac{1}{2}}{1 - (y_0 - \frac{1}{2})t} + \frac{1}{2}$$

Case 3. $\lambda > 1$

$$y(t) = \frac{1}{2}\sqrt{1-\lambda}\ \tan\left[\frac{1}{2}\sqrt{1-\lambda}\ t + \tan^{-1}\left(\frac{y_0 - \frac{1}{2}}{\frac{1}{2}\sqrt{1-\lambda}}\right)\right] + \frac{1}{2}$$

The derivation of these relationships is rather tedious and therefore omitted here.

Input Admissibility

Recall that our interest is in knowing whether or not a particular input stream is admissible. In the case when $p = 1$ under constant input $A(t) = \lambda$, we know that the unstable equilibrium point occurs at $y_2^* = \frac{1}{2} + \frac{1}{2}\sqrt{1-\lambda}$. Thus, if the system is currently at point $y > 1/2$ an input rate is admissible if it satisfies $\lambda < y(1 - y^p)$. Of course, if $y < 1/2$, then we require only that $\lambda < 1$.

3.3.5 Solution for Workload Function 3

Recall the piecewise linear form of this workload function.

$$w_3(x) = \begin{cases} \frac{c}{a}\ x & 0 \le x \le a \\ c - \frac{c}{b}\,(x-a) & a \le x \le a+b \\ 0 & x > a+b \end{cases}$$

The advantage of this function is that the behavior of the system within each interval is *linear* and can be analyzed completely. The drawback is that one must keep track of how the system switches between one interval and another. For the time being, we consider behavior of the system within each interval.

Behavior on $0 \le x \le a$

In this region of state space, the system is governed by

$$\frac{dx}{dt} = A(t) - \frac{c}{a}\ x$$

which for $x(t_0) = x_0$ is solved by the following system trajectory.

$$x(t) \;=\; x_0\; e^{-(c/a)(t-t_0)} + e^{-(c/a)t} \int_{t_0}^{t} A(\tau)\; e^{(c/a)\tau} d\tau$$

For constant input rate $A(t) = \lambda$ this further simplifies to the following.

$$x(t) \;=\; \frac{a}{c}\;\lambda + \left[x_0 - \frac{a}{c}\;\lambda\right] e^{-(c/a)(t-t_0)}$$

Behavior on $a \leq x \leq a + b$

In this region of state space, the system is governed by

$$\frac{dx}{dt} = A(t) - c + \frac{c}{b}\;\;(x - a)$$

which for $x(t_0) = x_0$ is solved by a similar system trajectory.

$$x(t) \;=\; a + b + (x_0 - a - b)\;\; e^{(c/b)(t-t_0)} + e^{(c/b)t} \int_{t_0}^{t} A(\tau)\; e^{-(c/b)\tau} d\tau$$

For constant input rate $A(t) = \lambda$ this further simplifies to the following.

$$x(t) \;=\; a + b - \frac{b}{c}\;\lambda + \left[x_0 - a - b + \frac{b}{c}\;\lambda\right] e^{(c/b)(t-t_0)}$$

Constant Input

The preceeding analysis can be clearly understood if we examine the particular case when $A(t) = \lambda$. Consider the equilibrium points for the system.

- For $x \in [0, a]$, $dx/dt = 0$ when $\lambda - (c/a)x^* = 0$. So we have $x_1^* = (a/c)\lambda$.

- For $x \in [a, a+b]$, $dx/dt = 0$ when $\lambda - c + (c/b)(x^* - a) = 0$. So we have $x_2^* = a + b - (b/c)\lambda$.

Recall that x_1^* is a stable equilibrium point and x_2^* is an unstable equilibrium point. This function is illustrated in Figure 3.11 below.

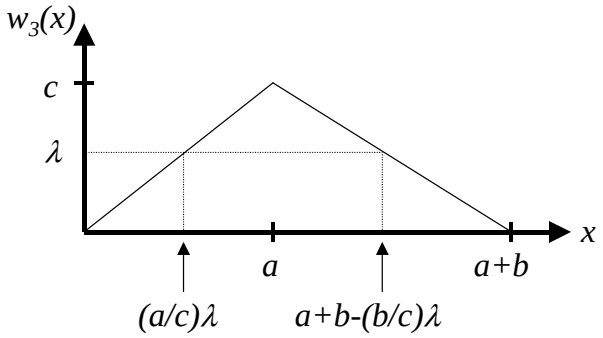


Figure 3.11: *Equilibrium points for piecewise linear workload function $w_3(x)$ under constant input.*

Now, using our previously derived results we observe that for $A(t) = \lambda$ the system evolves on the interval $[0, a]$ according to this equation.

$$x(t) \;=\; \frac{a}{c}\,\lambda + \left[x_0 - \frac{a}{c}\,\lambda\right] e^{-(c/a)(t-t_0)}$$

Observe that the first term of this expression is the equilibrium point x_1^*, and the second term represents the distance away from the equilibrium. Observe that this distance decreases exponentially in time, consistent with our notion of its stability.

Similarly, on the interval $[a, a+b]$ the system evolves according to the following equation.

$$x(t) \;=\; a + b - \frac{b}{c}\,\lambda + \left(x_0 - a - b + \frac{b}{c}\,\lambda\right) e^{(c/b)(t-t_0)}$$

Again, the first term is the equilibrium point and the second term is the distance away from it. Observe that here, this distance increases in time, consistent with our notion of the instability around x_2^*.

Input Function Admissibility

To ascertain whether or not a particular input sequence $A(t)$ is admissible, we need to answer two questions.

1. If $x(t_0) \in [\ 0, a]$, when and under what conditions will the sytem reach $x(t) = a$ and cross into the interval $[a, a+b]$?

2. If $x(t_0) \in [a, a+b]$ what guarantees that the system will not reach $x(t) = a+b$?

We address these question below.

Hitting Time to $x(t) = a$

Assume that the system starts at $x(t_0) = x_0 \in (0, a)$. Let t^* be defined as the first time the system reaches $x(t^*) = a$. We know that the time t^* satisfies this relationship.

$$\begin{aligned} x(t^*) &= x_0\ e^{-(c/a)(t^*-t_0)} + e^{-(c/a)t^*} \int_{t_0}^{t^*} A(\tau)\ e^{(c/a)\tau} d\tau \\ a &= e^{-(c/a)t^*} \left[x_0 e^{(c/a)t_0} + \int_{t_0}^{t^*} A(\tau)\ e^{(c/a)\tau} d\tau \right] \end{aligned}$$

This equation can be solved for t^* for any given function $A(t)$ that is integrable. The condition that $t^* < \infty$ means that the system will enter the interval $[a, a+b]$ under input $A(t)$. Again, if we assume that $A(t) = \lambda$ (a constant) then we obtain

$$t^* = -\frac{a}{c} \ln \left[e^{-(c/a)t_0} \left(\frac{a - \frac{a}{c}\ \lambda}{x_0 - \frac{a}{c}\ \lambda} \right) \right].$$

Note that if $\lambda \leq c$ then $t^* = \infty$.

Admissibility for $a \leq x \leq a+b$

The condition for admissibility is that $x(t) < a+b$ for all $t \geq 0$. Assume that the system starts at $x(t_0) = x_0 \in (a, a+b)$. Admissibility requires that the following hold

for all values $t \geq 0$.

$$(x_0 - a - b)\ e^{(c/b)(t-t_0)} + e^{(c/b)t} \int_{t_0}^{t} A(\tau)\ e^{-(c/b)\tau} d\tau \ < \ 0$$

This simplifies to the following.

$$\int_{t_0}^{t} A(\tau)\ e^{-(c/b)\tau} d\tau \ < \ (a + b - x_0)\ e^{-(c/b)t_0}$$

Again, assuming $A(t) = \lambda$, we have the requirement that

$$-(b/c)\lambda e^{-(c/b)(t-t_0)} < (a + b - x_0)\ e^{-(c/b)t_0}.$$

Collectively, the above workload functions provide a reasonable family of congestion-sensitive behaviors that may be of use in modeling real input-output systems. While their analysis is not always convenient, they share the same basic qualitative behavior. Depending on the domain-specific application under study, it may be worthwhile to pursue further the type of analysis presented here.

3.4 Chapter Summary

In this chapter, we have developed a simple deterministic model for a congestion-sensitive input-output system. We have shown that a key to understanding the behavior is the shape of the workload function. Using qualitative and quantitative analysis, we have developed a basic understanding of the response of the system to varying levels of input, and we have characterized the behavior when the system experiences congestion collapse. We developed conditions under which arrival sequences can be admitted without modification. For cases in which an arrival sequence is not admissible, we identified input control as a primary means by which overall system behavior can be managed. In the next chapter, we will continue in this direction and develop adaptive measures based on input control to achieve optimal system performance.

Chapter 4

Deterministic Models

4.1 Optimal Control for Deterministic Input

Our previous analyses for the qualitative and quantitative behavior assumed that system input was constant, and moreover that the system operator could choose this constant level of input. For most real systems, however, it is more likely that the input stream $A(t)$ is exogenous and time varying. A system operator may be limited in her ability to control the arrivals to the system.

In this section, we consider the case of an external arrival stream $A(t)$ that is deterministically known in advance by the operator. In general, we assume that the input stream $A(t)$ is piecewise continuous. In addition, we assume that the system operator can influence the behavior of the system by choosing from a set of admissible controls. Following the convention in [110, 19], we define an *admissible control* $u(t)$ to be a piecewise continuous function on $[0, T]$ satisfying $u(t) \in \mathcal{U}(t)$. Here, we define $\mathcal{U}(t) = [0, A(t)]$ such that $u(t)$ represents the instantaneous rate of admitted arrivals to the system, subject to the condition $0 \leq u(t) \leq A(t)$. Depending on the context, other control rules may also reasonable.[1] In applying this type of control, we assume that all restricted arrivals are lost to the system. The system evolves according to

[1]For example, one could define the control as restricting the *proportional* input to the system. To do so, one would define $\mathcal{U}(t) = [0, 1]$, and then the system would evolve accoring to $\frac{dx}{dt} = u(t)A(t) - w(x)$.

the simplified dynamics

$$\frac{dx}{dt} = u(t) - w(x)$$

where $0 \leq u(t) \leq A(t)$. Again, we assume that $u(t)$ is piecewise continuous and therefore admissible.

There are several system objectives that may be of interest depending on the application. Generally speaking, two primary quantities of interest for processing systems are *delay* and *throughput*, and it will be on the latter that we will focus. Specifically, we will consider maximizing the output rate $w(x(t))$ over a finite time horizon T. The *deterministic* optimal control problem is given as follows

$$\begin{aligned} \max_{0 \leq u(t) \leq A(t)} \quad & \int_{t=0}^{T} w(x(t))dt \\ s.t. \quad & \dot{x}(t) = u(t) - w(x(t)) \\ & x(0) \text{ and } \{A(t), t \in [0,T]\} \text{ given} \end{aligned}$$

where $\dot{x}(t) = dx/dt$. In words, the challenge is to choose the optimal input rate $u^*(t)$ that maximizes the output rate of work over the finite time interval $[0, T]$. In general, we may be interested in maximizing some other function of system output, denoted $R(x) = r(w(x))$, in which case the objective function changes accordingly.

The above formulation is not completely rigorous. Since the control function $u(t)$ can have discontinuous jumps, the system trajectory will not be differentiable everywhere. This condition will be exacerbated if the workload function w is not continuously differentiable with respect to x. A more appropriate formulation would assert that the state of the system at any time t is given by the following integral condition

$$x(t) = x(0) + \int_0^t (u(t) - w(x(t)))\, dt.$$

Unfortunately, a more rigorous treatment of this issue is beyond the scope of this thesis.

4.1.1 Direct Approach

If we can solve the ODE $\dot{x}(t) = u(t) - w(x(t))$ (with boundary condition $x(0) = x_0$ and subject to $0 \leq u(t) \leq A(t)$) for a solution $x(t) = f(x_0, u(t))$, then we can reduce the previous formulation to an optimization problem of a single variable, namely

$$\max_{0 \leq u(t) \leq A(t)} \int_{t=0}^{T} w\left(f(x_0, u(t))\right).$$

There are many well established methods for solving this type of nonlinear optimization. Unfortunately, we know that analytic solutions to the ODE for workload functions $w_1(x)$ and $w_2(x)$ are difficult, and even the complete solution to the ODE with workload function $w_3(x)$ is inconvenient (since we must keep track of the switching between the two linear regions). As a result, we seek other more general approaches.

4.1.2 Necessary Conditions

Under the assumption that the control $u(t)$ is piecewise continuous and the system trajectory $x(t)$ is twice continuously differentiable (see [19] or Appendix C of [110]), we know by the Maximum Principle that the optimal control trajectory $\{u^*(t), t \in [0, T]\}$ must satisfy the condition

$$u^*(t) = \arg \max_{0 \leq u(t) \leq A(t)} H(x^*(t), u(t), \gamma(t))$$

where $x^*(t)$ is the optimal system trajectory, $H(x, u, \gamma)$ is the *Hamiltonian* given by

$$\begin{aligned} H(x, u, \gamma) &= w(x) + \gamma^T [u - w(x)] \\ &= w(x) - \gamma w(x) + \gamma u \end{aligned}$$

and $\gamma(t)$ is the solution to the *adjoint equation*

$$\begin{aligned} \dot{\gamma}(t) &= -H_x(x(t), u(t), \gamma(t)) \\ &= w_x(\gamma - 1) \end{aligned}$$

with terminal condition $\gamma(T) = 0$. Here, we use the notation $H_x = \partial H/\partial x$ and $w_x = \partial w/\partial x$. Since the optimal control trajectory must maximize the Hamiltonian, it is clear that $u^*(t)$ must satisfy the following relationship.

$$u^*(t) = \begin{cases} A(t) & \text{if } \gamma(t) > 0 \\ 0 & \text{if } \gamma(t) < 0 \end{cases}$$

Observe that the optimal control variable $u^*(t)$ is unspecified whenever $\gamma(t) = 0$. The complete solution is obtained by simultaneously solving the following equations.

$$\begin{aligned} \dot{x^*} &= u^* - w(x^*) \\ \dot{\gamma} &= \gamma w_x(x^*) - R_x(x^*) \\ u^*(t) &= \begin{cases} A(t) & \gamma(t) > 0 \\ 0 & \gamma(t) < 0 \end{cases} \\ x^*(0) &= x_0 \\ \gamma(T) &= 0 \end{aligned}$$

Again, $R_x = \partial R/\partial x$, and $x^*(t)$ represents the optimal trajectory of the system. The general analytic solution to this problem is difficult. However, the above system of equations can be approximated by an appropriate discrete version. Specifically, approximate dx/dt by

$$\frac{\Delta x}{\Delta t} = \frac{x(t + \Delta t) - x(t))}{\Delta t}$$

and approximate $d\gamma/dt$ by

$$\frac{\Delta \gamma}{\Delta t} = \frac{\gamma(t + \Delta t) - \gamma(t))}{\Delta t}$$

Then, the above system of equations is represented as

$$\begin{aligned} x(t+\Delta t) &= x(t) + [\ u(t) - w(x(t))\ ]\,\Delta t \\ \gamma(t+\Delta t) &= \gamma(t) + [\ \gamma(t) w_x(x(t)) - R_x(x(t))\ ]\,\Delta t \\ u(t) &= \begin{cases} A(t) & \gamma(t) > 0 \\ 0 & \gamma(t) < 0 \end{cases} \\ x(0) &= x_0 \\ \gamma(T) &= 0 \end{aligned}$$

Given an initial value $x(0) = x_0$, this system of equations can be solved numerically by selecting the value of $\gamma(0)$ that results in $\gamma(T) = 0$. However, there is a better approach for attaining a numerical solution based on ideas from *dynamic programming*.

4.1.3 Numerical Solution via DP

The above problem can be solved using a dynamic programming approach. Let $V(x,t)$ equal the value of the optimal cost-to-go function for the system at state x at time t. Consider a discrete version of the system in which the system moves by an amount Δx in time Δt. The system evolves according to

$$\Delta x = [u(t) - w(x)]\,\Delta t$$

and the available control is subject to the constraint $0 \le u(t) \le A(t)$. The optimal cost-to-go function then satisfies

$$\begin{aligned} V(x,t) &= \max\{w(x)\Delta t + V(x+\Delta x, t+\Delta t)\} \\ &= \max_{0\le u(t)\le A(t)} \{w(x)\Delta t + V(x+[u(t)-w(x)]\Delta t, t+\Delta t)\} \\ &= w(x)\Delta t + \max_{0\le u(t)\le A(t)} \{V(x+[u(t)-w(x)]\Delta t, t+\Delta t)\} \end{aligned}$$

with terminal condition $V(x,T) = 0$. We will use the DP algorithm to solve this finite-horizon problem by iterating recursively backwards to obtain the solution for $V(x_0, 0)$. However, before proceeding with the dynamic programming algorithm to solve this discrete version of the optimal control problem, it is worth noting that our formulation naturally leads to a nonlinear partial differential equation (PDE). By taking partial derivatives of the above expression with respect to t we obtain the following relationship

$$\begin{aligned}\frac{\partial V}{\partial t} &= w(x) + \frac{\partial V}{\partial x}\left(A(t) - w(x)\right)\ I(\partial V/\partial x > 0) \\ &\quad -\frac{\partial V}{\partial x}\left(w(x)\right)\ I(\partial V/\partial x < 0)\end{aligned}$$

where $I(\cdot)$ is the *indicator function*. This is the *Hamilton-Jacobi-Bellman (HJB) Equation*. It is a PDE that states conditions that must be satisfied by the optimal value function for all x, t. Note that the switching boundary $\partial V/\partial x = 0$ effectively decouples the system dynamics into two simplified regions.

$$\begin{aligned}\frac{\partial V}{\partial x} > 0 \quad &\rightarrow \quad \frac{\partial V}{\partial t} = w(x) + \frac{\partial V}{\partial x}\left(A(t) - w(x)\right) \\ \frac{\partial V}{\partial x} < 0 \quad &\rightarrow \quad \frac{\partial V}{\partial t} = w(x) - \frac{\partial V}{\partial x} w(x)\end{aligned}$$

If we can solve the PDE for each region for the corresponding $V(x,t)$ then we know when $\partial V/\partial x > 0$ and $\partial V/\partial x < 0$, which gives the optimal control.

Discrete Time System

Consider the case where the time interval $[0, T]$ is broken into N subintervals of length $\Delta t = T/N$. Let the system evolve in discrete time with steps indexed by $n \in [0, N]$. Here, we will maintain the convention of counting "up" from an initial period 0 up to a final period N. Let x_n be the state of the system in period n. Similarly, let A_n denote the arrivals for period n, and let u_n be the the input control for period n,

subject to the constraint $0 \leq u_n \leq A_n$. Thus, the system evolves according to

$$x_{n+1} = x_n - w(x_n) + u_n$$

Define $J_n(x) = V(x, n)$ to be the discrete value of the optimal cost-to-go function when the system is in state x in period n. Starting with the system in a terminal state x_N, we can solve the dynamic program recursively backwards to obtain the following.

$$\begin{aligned} J_N(x_N) &= w(x_N) \\ J_{N-1}(x_{N-1}) &= w(x_{N-1}) + \max_{0 \leq u_{N-1} \leq A_{N-1}} J_N(x_{N-1} - w(x_{N-1}) + u_{N-1}) \end{aligned}$$

At each time step, we choose the input control so as to maximize a function that combines the system output in the next step and the optimal system trajectory beyond the next step. If we cared only about maximizing system output in the next step, we could restrict attention to the workload function $w(x)$. Since this function is maximized at x^* (defined such that $w(x^*) \geq w(x), \forall x$), all we would need to do is to choose an input that gets the system as close as possible to the optimal output level. In the absence of constraints on u_k, this would mean $u_k^* = x^* - x_k + w(x_k)$ for all periods $k \in [0, N]$. However, this is not possible in general, since $0 \leq u_k \leq A_k$. It should therefore be clear that the following holds in time period $N - 1$.

$$u_{N-1}^* = \begin{cases} 0 & x_{N-1} - w(x_{N-1}) \geq x^* \\ \min(x^* - x_{N-1} + w(x_{N-1}), A_{N-1}) & x_{N-1} - w(x_{N-1}) < x^* \end{cases}$$

Clearly, the optimal control in period k will depend *both* on the current state x_k *and* the future arrivals $\{A_{k+1}, A_{k+2}, \dots\}$. The general relationship for our backward

recursive solution to the discrete dynamic program is the following.

$$
\begin{aligned}
J_k(x_k) &= w(x_k) + \max_{0 \le u_k \le A_k} J_{k+1}(x_{k+1}) \\
&= w(x_k) + \max_{0 \le u_k \le A_k} \left[w(x_{k+1}) + \max_{0 \le u_{k+1} \le A_{k+1}} J_{k+2}(x_{k+2}) \right] \\
&= w(x_k) + \max_{0 \le u_k \le A_k} \left[w(x_{k+1}) + \max_{0 \le u_{k+1} \le A_{k+1}} \left[w(x_{k+2}) + \dots \right. \right. \\
&\qquad \left. \left. \cdots + \max_{0 \le u_{N-2} \le A_{N-2}} \left[w(x_{N-1}) + \max_{0 \le u_{N-1} \le A_{N-1}} w(x_N) \right] \dots \right] \right] \\
&= w(x_k) + \max_{\substack{0 \le u_k \le A_k, \ \dots \\ 0 \le u_{N-1} \le A_{N-1}}} \left[w(x_{k+1}) + w(x_{k+2}) + \cdots + w(x_{N-1}) + w(x_N) \right]
\end{aligned}
$$

The solution to the original problem is $J_0(x_0)$. The optimal control vector $u^* = \{u_0^*, u_1^*, \dots, u_{N-1}^*\}$ is a function of x_0 and the arrival sequence $\{A_0, A_1, \dots, A_{N-1}\}$.

Example: Optimal Control For Arbitrary Input

This type of optimal control problem is best illustrated by an example, such as is depicted in Figure 4.1. We are given a system with piecewise linear workload function of type w_3 with parameters $a = 1.5$, $b = 1.5$, and $c = 0.2$ (Figure 4.1a). This system lives within the interval $x \in [0, 3]$. The system maximizes its output rate when $x = 1.5$ and achieves a maximum output rate of 0.2. Assume the system starts empty $x(0) = 0$ and time evolves in discrete increments of $\Delta t = 0.1$. The system receives an arbitrary input sequence $A(t)$ with the following characteristics (Figure 4.1b). In each time interval either a single unit of work arrives or no work arrives. This arrival sequence is generated by a random process in which an arrival occurs in each time period with probability $p = 0.25$, so we expect the average arrival rate to be 0.25. Observe that the average arrival rate is greater than the maximum output rate of the system, so admission control is required to maintain stability. We assume that admission control in each period is constrained to admit all or nothing, that is $u(t) \in \{0, 1\}$. We assume that the arrival sequence is known in advance and that we are interested in maximizing the throughput of a congestion-sensitive system over

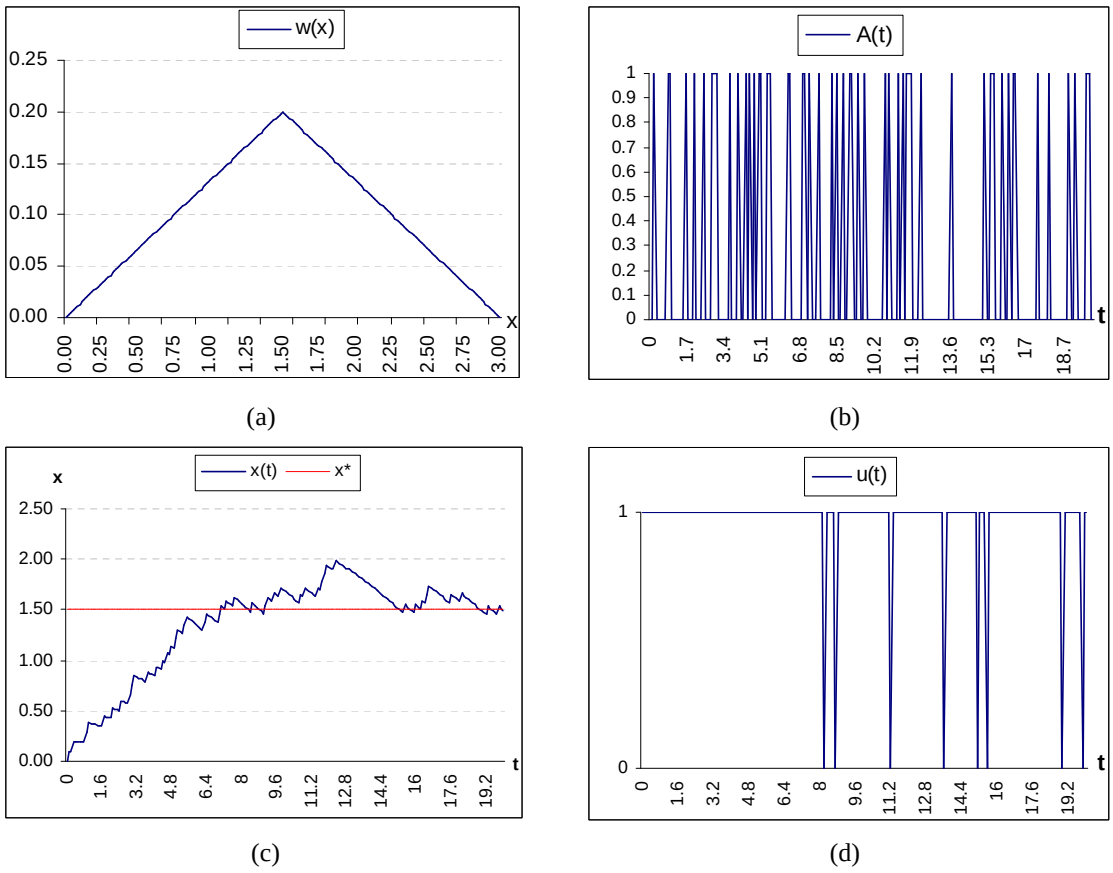


Figure 4.1: *Example of deterministic optimal control solution from DP.* (a) The piecewise linear workload function $w(x)$. (b) The arbitrary input sequence $A(t)$. (c) The optimal system trajectory $x(t)$. (d) The optimal control sequence $u(t)$.

finite time horizon $T = 20$.

We can use the DP algorithm outlined in the previous section to solve this problem. The solution is given by the optimal system trajectory (Figure 4.1c) which results from the optimal control sequence (Figure 4.1d). We observe a few qualitative features of this solution. First, since the system starts empty the optimal control is to admit all arrivals until a point at which $x(t) > x^*$ and the system experiences congestion. After that, control is used to maintain the system near the optimal operating point x^*. Furthermore, the system becomes most congested just prior to the time interval $t \in [12, 15]$ during which there are few arrivals.

From a preliminary inspection of the optimal trajectory, we speculate that the optimal system trajectory is one in which the system moves to its optimal operating point x^* as quickly as possible and then remains near there for all operating time. Furthermore, there appear to be two primary forces that affect the ability of the system to remain at the optimal operating point. The first is *congestion*, resulting from *too much* inventory at a given point in time. The second is *starvation*, resulting from *too little* inventory at a given point in time. Here we are interested in the tradeoff between congestion and starvation in overall system dynamics. We further explore these ideas below.

4.2 Tradeoff: Congestion vs. Starvation

We have demonstrated previously that input control is effective at preventing situations in which there are too many arrivals. By blocking excess arrivals, congestion can be avoided. Unfortunately, input control can do little to avoid cases in which there are too few arrivals, a situation known as *starvation*. However, if periods of starvation can be anticipated, it might be possible to choose an input policy that compensates for upcoming starvation by admitting more input than would otherwise be optimal. To explore this tradeoff, we consider the following canonical example.

4.2.1 The Case of On-Off-On Arrivals

Recall the dynamics of the system under the piecewise linear function

$$w_3(x) = \begin{cases} \frac{c}{a}\, x & 0 \le x \le a \\ c - \frac{c}{b}\,(x-a) & a \le x \le a+b \\ 0 & x > a+b \end{cases}$$

and assume that we are interested in solving the aforementioned deterministic optimization problem

$$\begin{array}{rc} \max\limits_{u(t)} & \int_{t=0}^{T} w(x(t))dt \\ s.t. & \dot{x}(t) = u(t) - w(x(t)) \\ & 0 \leq u(t) \leq A(t), t \in [0,T] \\ & x(0) \text{ and } \{A(t), t \in [0,T]\} \text{ given} \end{array}$$

Now, consider the case where the system starts at the optimal operating point $x(0) = x^* = a$ and the arrival stream has the following form.

$$A(t) = \begin{cases} \lambda & 0 \leq t < t_1 \\ 0 & t_1 \leq t < t_2 \\ \lambda & t > t_2 \end{cases}$$

Here λ is a constant satisfying $\lambda > w(x^*) = c$. In words, there is a constant input stream that shuts off at time t_1 and then turns on again at time t_2. We call this input stream an "on-off-on" arrival sequence. Let $\tau_1 = t_2 - t_1$ denote the length of the "off" interval for which there are no arrivals.

Research Question *What is the optimal system trajectory for a system under this input stream?*

System Dynamics and Output for $0 \leq x(t) \leq a$

Recall, that we know the exact evolution of the system within both intervals. In particular, when the system starts at $x(t_0) = x_0 \in [0, a]$ and remains in this interval under constant input rate $A(t) = \lambda$ the system evolves according to

$$x(t) \;=\; \frac{a}{c}\,\lambda + \left[x_0 - \frac{a}{c}\,\lambda\right] e^{-(c/a)(t-t_0)}.$$

We are also interested in the system output during this period. Let $J[t_0, t]$ be the contribution to the objective function during the interval $[t_0, t]$. From the above

dynamics, we have the following relationship.

$$\begin{aligned}
J[t_0,t] &= \int_{t_0}^{t} w(x(s))ds \\
&= \int_{t_0}^{t} \frac{c}{a}\, x(s)ds \\
&= \frac{c}{a}\int_{t_0}^{t} \left(\frac{a}{c}\,\lambda + \left[x_0 - \frac{a}{c}\,\lambda\right] e^{-(c/a)(s-t_0)}\right) ds \\
&= \lambda(t-t_0) + \frac{c}{a}\left[x_0 - \frac{a}{c}\,\lambda\right]\int_{t_0}^{t} e^{-(c/a)(s-t_0)}ds \\
&= \lambda(t-t_0) + \frac{c}{a}\left[x_0 - \frac{a}{c}\,\lambda\right]\left(-\frac{a}{c}\right)\left(e^{-(c/a)(t-t_0)} - 1\right) \\
&= \lambda(t-t_0) - \left[x_0 - \frac{a}{c}\,\lambda\right]\left(e^{-(c/a)(t-t_0)} - 1\right)
\end{aligned}$$

System Dynamics and Output for $a \le x(t) \le a+b$

Similarly, for starting point $x(t_0) = x_0 \in [a, a+b]$ and under constant input rate $A(t) = \lambda$ the system evolves within the interval according to

$$x(t) = a + b - \frac{b}{c}\,\lambda + \left[x_0 - a - b + \frac{b}{c}\,\lambda\right] e^{(c/b)(t-t_0)}.$$

The corresponding output during this interval is the following.

$$\begin{aligned}
J[t_0,t] &= \int_{t_0}^{t} w(x(s))ds \\
&= \int_{t_0}^{t} \left[c - \frac{c}{b}\ (x(s)-a)\right] ds \\
&= \left(c + \frac{ac}{b}\right)(t-t_0) - \frac{c}{b}\int_{t_0}^{t} x(s)ds \\
&= \left(c + \frac{ac}{b}\right)(t-t_0) \\
&\qquad -\frac{c}{b}\int_{t_0}^{t} \left(a+b-\frac{b}{c}\ \lambda + \left[x_0 - a - b + \frac{b}{c}\ \lambda\right] e^{(c/b)(s-t_0)}\right) ds \\
&= \left(c + \frac{ac}{b} - \frac{c}{b}\left(a+b-\frac{b}{c}\ \lambda\right)\right)(t-t_0) \\
&\qquad -\frac{c}{b}\left[x_0 - a - b + \frac{b}{c}\ \lambda\right]\int_{t_0}^{t} \left(e^{(c/b)(s-t_0)}\right) ds \\
&= \lambda(t-t_0) - \frac{c}{b}\left[x_0 - a - b + \frac{b}{c}\ \lambda\right]\frac{b}{c}\left(e^{(c/b)(t-t_0)} - 1\right) \\
&= \lambda(t-t_0) - \left[x_0 - a - b + \frac{b}{c}\ \lambda\right]\left(e^{(c/b)(t-t_0)} - 1\right)
\end{aligned}$$

4.2.2 Baseline Trajectory

As a baseline for comparison, consider the following operating policy for a system starting at $x(0) = x^*$ and subject to the arrival stream described above.

- During the interval $0 \le t < t_1$ when $A(t) = \lambda$, maintain $x(t) = x^*$ by admitting $u(t) = w(x^*)$.
- During the interval $t_1 \le t < t_2$ when $A(t) = 0$, as the system continues to process output, the system state $x(t)$ will decrease to a level $x^* - x_2$.
- For $t \ge t_2$ when $A(t) = \lambda$, choose an input rate that returns the system to the optimal operating point $x(t) = x^*$ as quickly as possible and then maintains that optimal operating point for all time.
  - Choose $u(t) = \lambda$ for an amount of time τ_2 until the system returns to the optimal operating point at time $t_3 = t_2 = \tau_2$.

– For $t \geq t_3$ choose $u(t) = w(x^*)$.

We can compute the exact trajectory and system output that results from this policy.

Behavior for $t \leq t_1$

Trajectory As previously discussed, this is simply $x(t) = x^* = a$.

Throughput Over the interval $t \in [0, t_1]$ the total throughput is $J[0, t_1] = w(x^*)t_1 = ct_1$.

Behavior for $t_1 \leq t \leq t_2$

Trajectory Starting with $x(t_1) = x^*$ and with $A(t) = 0$ for $t \in [t_1, t_2)$, we can compute

$$\begin{aligned} x(t_2) &= x^* e^{-(c/a)(t_2 - t_1)} \\ &= x^* - x^* \left(1 - e^{-(c/a)(t_2 - t_1)}\right) \\ &= x^* - x_2 \end{aligned}$$

where $x_2 = x^* \left(1 - e^{-(c/a)(t_2 - t_1)}\right)$. Observe that $0 \leq x_2 \leq x^* = a$. Specifically, as $(t_2 - t_1) \to 0$ then $x_2 \to 0$ and as $(t_2 - t_1) \to \infty$ then $x_2 \to x^* = a$.

Throughput Starting with $x(t_1) = x^*$, we can compute the throughput in the interval $[t_1, t_2)$.

$$\begin{aligned} J(t_1, t_2) &= x^* \left[1 - e^{-(c/a)(t_2 - t_1)}\right] \\ &= x_2 \end{aligned}$$

Behavior for $t > t_2$

Trajectory Starting at $x(t_2) = x^* - x_2$ and with $A(t) = \lambda$ for $t \geq t_2$, we can compute

$$x(t) = \frac{a}{c}\,\lambda + \left[x^* - x_2 - \frac{a}{c}\,\lambda\right] e^{-(c/a)(t-t_2)}$$

for $t \geq t_2$. And from this it is clear that the system returns to the optimal operating point after an amount of time τ_2, where $x(t_2 + \tau_2) = x^*$.

$$\begin{aligned}
x(t_2+\tau_2) &= \frac{a}{c}\,\lambda + \left[x^* - x_2 - \frac{a}{c}\,\lambda\right] e^{-(c/a)(t_2+\tau_2-t_2)} \\
x^* &= \frac{a}{c}\,\lambda + \left[x^* - x_2 - \frac{a}{c}\,\lambda\right] e^{-(c/a)(\tau_2)} \\
e^{-(c/a)(\tau_2)} &= \frac{x^* - \frac{a}{c}\,\lambda}{x^* - x_2 - \frac{a}{c}\,\lambda} \\
\tau_2 &= -\frac{a}{c}\,\ln\left[\frac{x^* - \frac{a}{c}\,\lambda}{x^* - x_2 - \frac{a}{c}\,\lambda}\right] \\
&= -\frac{a}{c}\,\ln\left[\frac{x^* - \frac{a}{c}\,\lambda}{x^* e^{-(c/a)(t_2-t_1)} - \frac{a}{c}\,\lambda}\right] \\
&= -\frac{a}{c}\,\ln\left[\frac{1 - \frac{\lambda}{c}}{e^{-(c/a)(t_2-t_1)} - \frac{\lambda}{c}}\right]
\end{aligned}$$

Throughput Starting with $x(t_2) = x^* - x_2$ and for $t \in [t_2, t_2 + \tau_2)$, we can compute the corresponding contribution to throughput.

$$\begin{aligned}
J(t_2, t_2+\tau_2) &= \lambda\,\tau_2 - \left[x^* - x_2 - \frac{a}{c}\,\lambda\right]\left[e^{-(c/a)\tau_2} - 1\right] \\
&= \lambda\,\tau_2 - \left(x^* - \frac{a}{c}\,\lambda\right) + \left[x^* e^{-(c/a)(t_2-t_1)} - \frac{a}{c}\,\lambda\right] \\
&= \lambda\,\tau_2 - x^*\left(1 - e^{-(c/a)(t_2-t_1)}\right) \\
&= \lambda\,\tau_2 - x_2
\end{aligned}$$

t	A(t)	u(t)	x(t)
$0 \leq t < t_1$	λ	$w(x^*)$	x^*
$t_1 \leq t < t_2$	0	0	$x^* e^{-(c/a)(t-t_1)}$
$t = t_2$	λ	λ	$x^* - x_2$
$t_2 < t < t_2 + \tau_2$	λ	λ	$\frac{a}{c}\ \lambda + \left[x^* - x_2 - \frac{a}{c}\ \lambda\right] e^{-(c/a)(t-t_2)}$
$t = t_3 = t_2 + \tau_2$	λ	$w(x^*)$	x^*
$t > t_2 + \tau_2$	λ	$w(x^*)$	x^*

Table 4.1: *System under baseline trajectory.*

Summary

The system behavior under this baseline trajectory is summarized in Table 4.1 and in Figure 4.2 below. System output over the various time intervals is

$$\begin{aligned} J[0, t_1] &= ct_1, \\ J[t_1, t_2] &= x_2, \\ J[t_2, t_2 + \tau_2] &= \lambda\ \tau_2 - x_2. \end{aligned}$$

The total throughput of the system is therefore as follows.

$$\begin{aligned} J[0, t_2 + \tau_2] &= J[0, t_1] + J[t_1, t_2] + J[t_2, t_2 + \tau_2] \\ &= ct_1 + \lambda\ \tau_2 \\ &= ct_1 - \lambda\ \frac{a}{c}\ \ln\left[\frac{1 - \frac{\lambda}{c}}{e^{-(c/a)(t_2 - t_1)} - \frac{\lambda}{c}}\right] \end{aligned}$$

We define the *normalized* throughput of the system as

$$\frac{J[0, t_2 + \tau_2]}{t_2 + \tau_2} = \frac{ct_1 + \lambda\ \tau_2}{t_2 + \tau_2}.$$

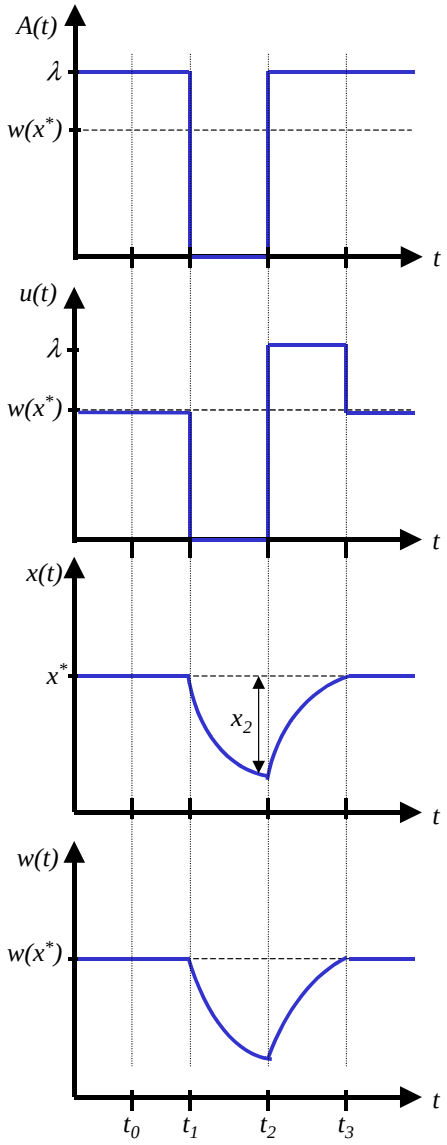


Figure 4.2: *System under baseline trajectory.*

4.2.3 Modified Trajectory

Now consider a modified operating policy. Again, we assume that the system begins operation at $x(0) = x^*$ and is subject to the same "on-off-on" arrival stream.

- During the interval $0 \leq t < t_0 < t_1$ when $A(t) = \lambda$, maintain $x(t) = x^*$ by admitting $u(t) = w(x^*)$.
- During the interval $t_0 \leq t < t_1$ when $A(t) = \lambda$, choose an input rate $u(t) = \lambda$ so as to accrue an extra x_1 inventory [so that $x(t_1) = x^* + x_1$]. Note that the length of the interval $\tau_1 = t_1 - t_0$ is defined by the amount of extra inventory x_1 to accrue.
- During the interval $t_1 \leq t < t_2$ when $A(t) = 0$, as the system continues to process output, the system state $x(t)$ will decrease to a level $x^* - x_2$.
- For $t \geq t_2$ when $A(t) = \lambda$, choose an input rate that returns the system to the optimal operating point $x(t) = x^*$ as quickly as possible and then maintains that optimal operating point for all time.
  - Choose $u(t) = \lambda$ for an amount of time τ_2 until the system returns to the optimal operating point at time $t_3 = t_2 = \tau_2$.
  - For $t \geq t_3$ choose $u(t) = w(x^*)$.

Again, we can compute the exact trajectory that results from this policy as well as the total throughput for the system. It should be clear that the baseline trajectory is a special case of this strategy for which $x_1 = 0$. In fact, the value x_1 parameterizes the entire problem.

Behavior for $t < t_0$

Trajectory As previously discussed, this is simply $x(t) = x^* = a$.

Throughput Total throughput in this interval is $J[0, t_0] = w(x^*)t_0 = ct_0$.

Behavior for $t_0 \leq t < t_1$

Trajectory Behavior in this interval is governed by

$$x^* + x_1 = a + b - \frac{b}{c}\,\lambda + \left[x^* - a - b + \frac{b}{c}\,\lambda\right] e^{(c/b)(t_1 - t_0)}$$

which yields the following relationship for t_0.

$$\begin{aligned} e^{(c/b)(t_1-t_0)} &= \frac{x_1 - b + \frac{b}{c}\,\lambda}{-b + \frac{b}{c}\,\lambda} \\ t_1 - t_0 &= \frac{b}{c}\,\ln\left[\frac{x_1 + b(\frac{\lambda}{c} - 1)}{b(\frac{\lambda}{c} - 1)}\right] \\ t_0 &= t_1 - \frac{b}{c}\,\ln\left[\frac{x_1 + b(\frac{\lambda}{c} - 1)}{b(\frac{\lambda}{c} - 1)}\right] \end{aligned}$$

Define $\tau_0 = t_1 - t_0$. Thus, $t_0 = t_1 - \tau_0$.

Throughput Total throughput in this interval is as follows.

$$\begin{aligned} J[t_0, t_1] &= \lambda(t_1 - t_0) - \left[x^* - a - b + \frac{b}{c}\,\lambda\right]\left(e^{(c/b)(t_1-t_0)} - 1\right) \\ &= \lambda\,\tau_0 - \left[\frac{b}{c}\,\lambda - b\right]\left(\frac{x_1 - b + \frac{b}{c}\,\lambda}{-b + \frac{b}{c}\,\lambda} - 1\right) \\ &= \lambda\,\tau_0 - \left[x_1 - b + \frac{b}{c}\,\lambda\right] + \left[\frac{b}{c}\,\lambda - b\right] \\ &= \lambda\,\tau_0 - x_1 \end{aligned}$$

Behavior for $t_1 \leq t \leq t_2$

Trajectory Since the system is now starting with $x(t_1) = x^* + x_1$ its dynamics under $A(t) = 0$ for $t \in [t_1, t_2)$ are more complicated. Specifically, consider two subintervals. First let $[t_1, t_1 + \hat{\tau})$ be the time interval in which the system evolves from $x^* + x_1$ to x^*. Then, let $[t_1 + \hat{\tau}, t_2]$ be the time interval in which the system evolves from x^* to $x^* - x_2$. The system is governed by different dynamics on each subinterval.

During $t \in [t_1, t_1 + \hat{\tau}]$ the system evolves according to the familiar equation.

$$x(t) = a + b + [x^* + x_1 - a - b]\, e^{(c/b)(t-t_1)}$$

Since $x(t_1 + \hat{\tau}) = x^* = a$, we can solve for $\hat{\tau}$.

$$\begin{aligned} x^* &= a + b - [x^* + x_1 - a - b]\, e^{(c/b)(t_1+\hat{\tau}-t_1)} \\ e^{(c/b)\hat{\tau}} &= \frac{b}{b - x_1} \\ \hat{\tau} &= \frac{b}{c}\, \ln\left[\frac{b}{b - x_1}\right] \end{aligned}$$

Note that we assume that $\hat{\tau} < \tau_1$. This implies a relationship for x_1.

$$\begin{aligned} \frac{b}{c}\, \ln\left[\frac{b}{b - x_1}\right] &< \tau_1 \\ \frac{b}{b - x_1} &< e^{(c/b)\tau_1} \\ x_1 &< b\left(1 - e^{-(c/b)\tau_1}\right) \end{aligned}$$

During $t \in [t_1 + \hat{\tau}, t_2]$ the system evolves according to the given equation.

$$x(t) = x^* e^{-(c/a)(t-(t_1+\hat{\tau}))}$$

At the point $t = t_2$, the system is at

$$\begin{aligned} x(t_2) &= x^* e^{-(c/a)(t_2-(t_1+\hat{\tau}))} \\ &= x^*\ e^{-(c/a)(t_2-t_1)}\ e^{(c/a)\hat{\tau}} \end{aligned}$$

but note that

$$\begin{aligned} e^{(c/a)\hat{\tau}} &= e^{(c/a)(b/c)\ln(b/(b-x_1))} \\ &= e^{(b/a)\ln(b/(b-x_1))} \\ &= \ln\left[\left(\frac{b}{b - x_1}\right)^{b/a}\right] \end{aligned}$$

so we have

$$\begin{aligned} x(t_2) &= x^* \, e^{-(c/a)(t_2-t_1)} \ln\left[\left(\frac{b}{b-x_1}\right)^{b/a}\right] \\ &= x^* - x^*\left[1 - e^{-(c/a)(t_2-t_1)} \ln\left(\frac{b}{b-x_1}\right)^{b/a}\right] \\ &= x^* - x_2 \end{aligned}$$

where $x_2 = x^*\left[1 - e^{-(c/a)(t_2-t_1)} \ln\left(\frac{b}{b-x_1}\right)^{b/a}\right]$. Again, observe that $0 \leq x_2 \leq x^*$.

Throughput Again, we consider the subintervals $[t_1, t_1+\hat{\tau})$ and $[t_1+\hat{\tau}, t_2)$ separately. Starting with $x(t_1) = x^* + x_1$ and for $t \in [t_1, t_1+\hat{\tau})$, we can compute $J(t_1, t_1+\hat{\tau})$.

$$\begin{aligned} J(t_1, t_1+\hat{\tau}) &= -\frac{c}{b}\left[x^* + x_1 - a - b\right]\frac{b}{c}\left(e^{(c/b)\hat{\tau}} - 1\right) \\ &= [b - x_1]\left(\frac{b}{b-x_1} - 1\right) \\ &= x_1 \end{aligned}$$

Starting with $x(t_1+\hat{\tau}) = x^*$ and for $t \in [t_1+\hat{\tau}, t_2)$, we can compute $J(t_1+\hat{\tau}, t_2)$.

$$\begin{aligned} J(t_1+\hat{\tau}, t_2) &= x^*\left[1 - e^{-(c/a)(t_2-(t_1+\hat{\tau}))}\right] \\ &= x^*\left[1 - e^{-(c/a)(t_2-t_1)} \ln\left(\frac{b}{b-x_1}\right)^{b/a}\right] \\ &= x_2 \end{aligned}$$

Collectively, we have the following.

$$\begin{aligned} J(t_1, t_2) &= J(t_1, t_1+\hat{\tau}) + J(t_1+\hat{\tau}, t_2) \\ &= x_1 + x_2 \end{aligned}$$

Behavior for $t > t_2$

This case is exactly the same as for the baseline case, except that the starting point $x(t_2) = x^* - x_2$ is now different. The relevant equations are as follows.

Trajectory For $t \geq t_2$ the system evolves according to

$$x(t) = \frac{a}{c}\,\lambda + \left[x^* - x_2 - \frac{a}{c}\,\lambda\right] e^{-(c/a)(t-t_2)}.$$

Throughput Starting with $x(t_2) = x^* - x_2$ system output in the interval $t \in [t_2, t_2 + \tau_2)$ is

$$J(t_2, t_2 + \tau_2) = \lambda\,\tau_2 - x_2$$

where

$$\tau_2 = -\frac{a}{c}\,\ln\left[\frac{x^* - \frac{a}{c}\,\lambda}{x^* - x_2 - \frac{a}{c}\,\lambda}\right]$$

and in this case

$$x^* - x_2 = x^* e^{-(c/a)(t_2 - t_1)}\,\ln\left(\frac{b}{b - x_1}\right)^{b/a}.$$

Summary

We similarly summarize the behavior of this modified trajectory in Table 4.2.
System output over the various time intervals is

$$\begin{aligned} J[0, t_0] &= ct_0 \\ &= c(t_1 - \tau_0), \\ J[t_0, t_1] &= \lambda\,\tau_0 - x_1, \\ J[t_1, t_2] &= x_1 + x_2, \\ J[t_2, t_2 + \tau_2] &= \lambda\,\tau_2 - x_2, \end{aligned}$$

t	A(t)	u(t)	x(t)
$0 \leq t < t_0$	λ	$w(x^*)$	x^*
$t_0 \leq t < t_1$	λ	λ	x^*
$t = t_1$	0	0	$x^* + x_1$
$t_1 \leq t < t_1 + \hat{\tau}$	0	0	$x^* e^{-(c/a)(t-t_1)}$
$t_1 + \hat{\tau} \leq t < t_2$	0	0	$x^* e^{-(c/a)(t-t_1)}$
$t = t_2$	λ	λ	$x^* - x_2$
$t_2 < t < t_2 + \tau_2$	λ	λ	$\frac{a}{c}\ \lambda + \left[x^* - x_2 - \frac{a}{c}\ \lambda\right] e^{-(c/a)(t-t_2)}$
$t = t_2 + \tau_2 = t_3$	λ	$w(x^*)$	x^*
$t > t_3$	λ	$w(x^*)$	x^*

Table 4.2: *System under modfied trajectory.*

so the total throughput of the system is

$$\begin{aligned} J[0, t_2 + \tau_2] &= J[0, t_0] + J[t_0, t_1] + J[t_1, t_2] + J[t_2, t_2 + \tau_2] \\ &= c(t_1 - \tau_0) + \lambda\ \tau_0 + \lambda\ \tau_2 \end{aligned}$$

where

$$\begin{aligned} \tau_0 &= \frac{b}{c}\ \ln\left[\frac{\frac{x_1}{b} + \frac{\lambda}{c} - 1}{\frac{\lambda}{c} - 1}\right] \\ \tau_2 &= -\frac{a}{c}\ \ln\left[\frac{1 - \frac{\lambda}{c}}{e^{-(c/a)(t_2 - t_1)} \ln\left(\frac{b}{b - x_1}\right)^{b/a} - \frac{\lambda}{c}}\right]. \end{aligned}$$

Observe that the system output for the modified system reduces to the baseline case whenever $x_1 = 0$. A graphical comparison of the system controls, trajectories and outputs under the modified and the baseline policies is presented in Figure 4.3.

As before, the *normalized* throughput of the system is given as follows.

$$\frac{J[0, t_2 + \tau_2]}{t_2 + \tau_2} = \frac{c(t_1 - \tau_0) + \lambda(\tau_0 + \tau_2)}{t_2 + \tau_2}$$

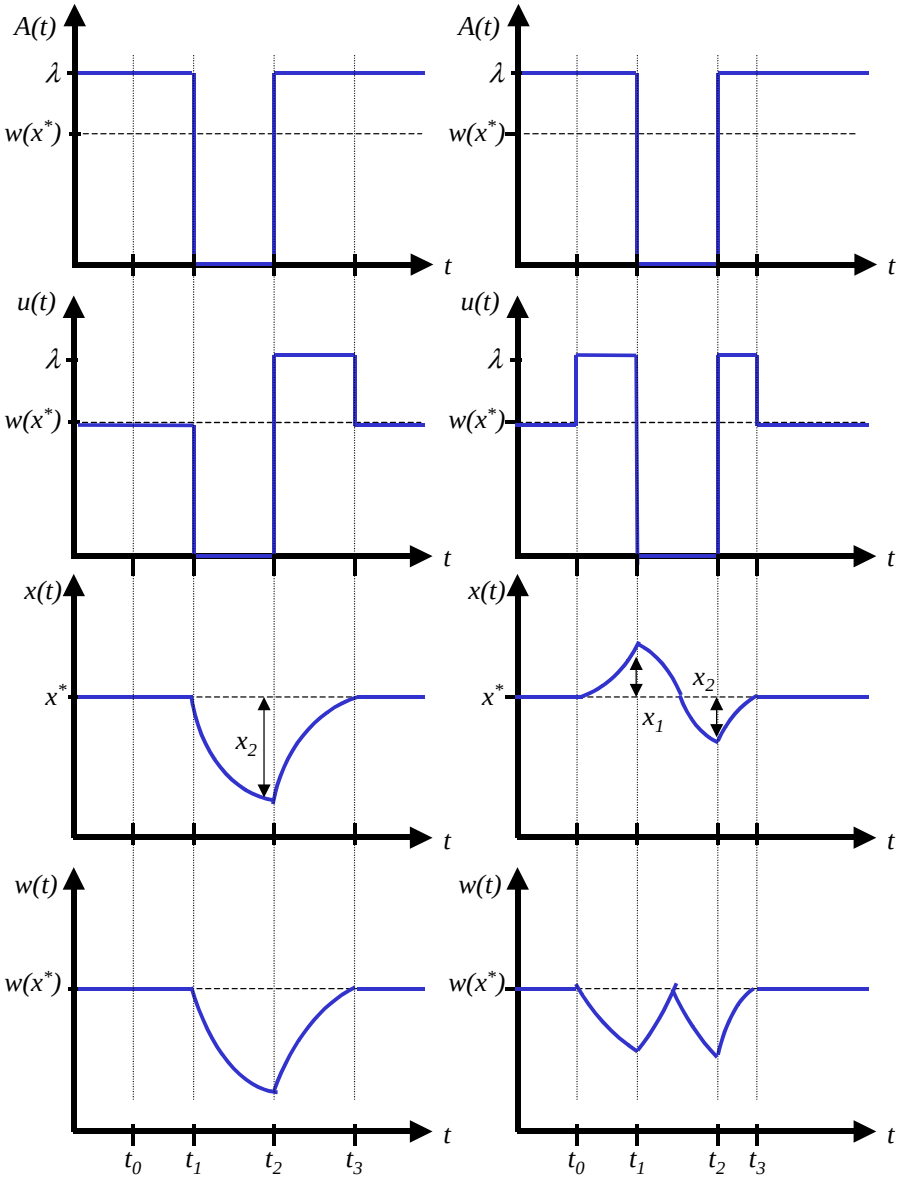


Figure 4.3: *Comparison of baseline and modified policies.*

4.2.4 Optimal Policy

It is now clear that the entire behavior of the system under the modified policy is parameterized by the amount of excess inventory x_1 to have on hand at time t_1 when the system input is shut off. Other critical parameters of the problem are the arrival rate λ and the duration of the "off" interval τ_1, as well as the parameters of the workload function a, b, and c.

We have thus defined an optimization problem in a single variable.[2] Specifically, let $\tilde{J}(x_1)$ be the value of the normalized system throughput.

$$\tilde{J}(x_1) = \frac{c(t_1 - \tau_0) + \lambda(\tau_0 + \tau_2)}{t_2 + \tau_2}$$

Our objective is to select the optimal value of x_1 so as to maximize the normalized throughput

$$\max_{0 \le x_1 \le b} \tilde{J}(x_1)$$

where parameters τ_0, τ_2 are defined as

$$\begin{aligned} \tau_0 &= \frac{b}{c} \ln \left[\frac{\frac{x_1}{b} + \frac{\lambda}{c} - 1}{\frac{\lambda}{c} - 1} \right], \\ \tau_2 &= -\frac{a}{c} \ln \left[\frac{1 - \frac{\lambda}{c}}{e^{-(c/a)(t_2 - t_1)} \ln \left(\frac{b}{b - x_1} \right)^{b/a} - \frac{\lambda}{c}} \right]. \end{aligned}$$

Let $\tilde{J}(x_1)$ be equal to the value of the objective function for a given value of x_1. It is possible to plot the shape of the function $\tilde{J}(x_1)$. For example, consider the case where $a = b = c = 1$, $\lambda = 2$, $t_1 = 1$ and $t_2 = 2$. Then the function $\tilde{J}(x_1)$ has the following shape for values $0 \le x_1 \le b\left(1 - e^{-(c/b)\tau_1}\right) = 0.632$.

In this case, we observe that the function is maximized at $x_1^* = 0.366$ and achieves value $\tilde{J}(x_1^*) = 0.856$.

[2] We note that a rigorous proof is needed to assert that the true optimization problem over all possible control functions can be reduced to an optimization of a single variable. Unfortunately, such a proof is beyond the scope of this thesis.

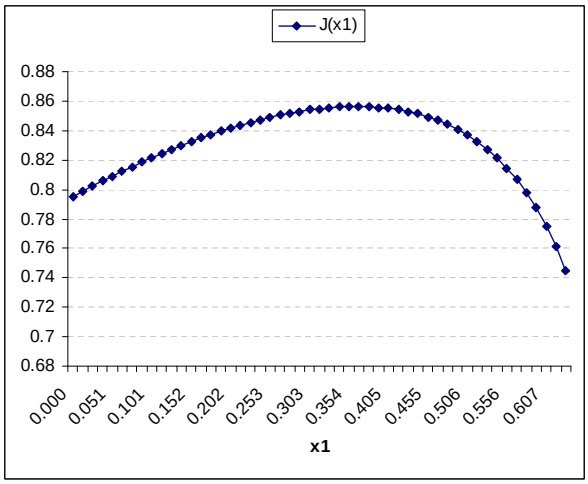


Figure 4.4: *System throughput as function of excess inventory* x_1.

Validation with DP

In this section we have solved for the optimal trajectory of a congestion sensitive system in the presence of a canonical "on-off" input sequence by formulating an alternative optimization problem which can be solved analytically. However, this same problem can be solved by the deterministic DP algorithm. If our analysis is correct, the two solutions should be in agreement. We take a simple example as appropriate evidence. Specifically, consider the previous case where $a = b = c = 1$, $\lambda = 2$, $t_1 = 1$ and $t_2 = 2$. For this problem, we showed by analysis that the objective function $\tilde{J}(x_1)$ was solved with value $x_1^* = 0.366$. Corresponding to this solution is an optimal system trajectory $x^*(t)$. In Figure 4.5, we compare this optimal trajectory to the trajectory obtained by deterministic DP. Although there are some minor discrepancies between the two (most likely caused by discretization of the problem in the DP), we conclude that these solutions are in close agreement.

4.2.5 Sensitivity Analysis

The analytical solution to the simple on-off arrival sequence provides a convenient framework for investigating the *sensitivity* of the optimal input control. In particular, we would like to gain insight into the following questions.

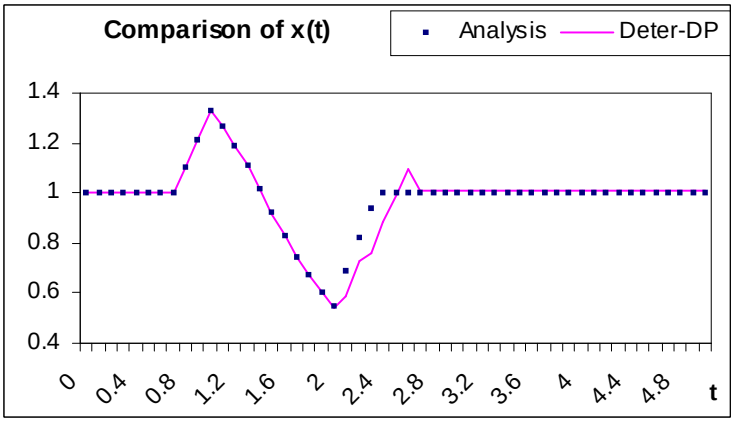


Figure 4.5: *Trajectory Comparison.* The deterministic DP algorithm finds nearly the same solution predicted by our analysis for the canonical on-off input.

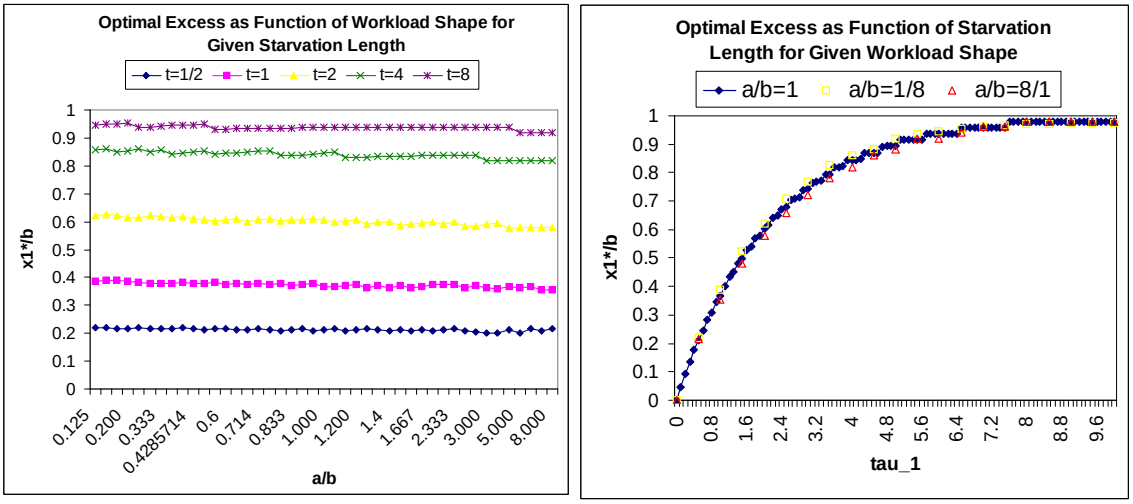
1. What is the effect the shape of the workload function on the optimal amount of excess inventory x_1^*?
2. What is the effect of the magnitude and duration of the starvation period on the optimal amount of excess inventory x_1^*?

We consider each of these questions briefly.

Workload Sensitivity

Recall that the shape of the workload function is given by the parameters a, b, c. Recall that these parameters partition the system state space $[0, a+b]$ into two regions: an *uncongested region* given by $[0, a]$, and a *congested region* given by $[a, a+b]$. The system workload function is maximized when $x = a$ at value $w(a) = c$. As already discussed, when the system is at a point $x = x* + x_1$, the system throughput is $c - (c/b)(x - a)$ Similarly, when the system is at a point $x = x^* - x_2$, the system throughput is $c - (c/a)(a - x)$. In this manner, the ratio a/b is a measure of the *relative expense* of having too little work versus too much work in the system. When $a/b < 1$ then the throughput penalty for too little work is less than the penalty for too much, and vice versa for $a/b > 1$.

As discussed previously, we would like to know much excess work x_1 to have on hand at the start of a starvation period of duration τ_1. Recall that we must have



(a) (b)

Figure 4.6: *Sensitivity to Starvation Duration.* As duration of starvation period increases, so does the optimal amount of relative excess x_1^*/b, independently of the relative shape of the workload function.

$0 \leq x_1 \leq b$. Thus, x_1^*/b represents the *relative excess* that should be accumulated prior to a starvation period. Figure 4.6a plots the optimal relative excess x_1^*/b as a function of the ratio a/b for starvation lengths $\tau_1 = 0.5, 1, 2, 4, 8$. It is interesting to observe that the this value is fairly insensitive to changes in the ratio a/b and seems to depend wholly on the value of τ_1. This relationship is confirmed in Figure 4.6b where plot the optimal relative excess as a function of τ_1.

Another important feature of the workload function is its maximum processing capacity, represented by the parameter c. It seems likely that changes in c may have a dramatic effect on the form of the optimal control. Consider the ratio $(a + b)/c$, which is a measure of the relative buffering capacity versus processing capacity of the system. When $a + b$ is large, the system can hold a large amount of work (although it may process it slowly). When c is large, the system can process a large amount of work per unit time. The role of the parameter c is illustrated in Figure 4.7. Again, we consider the relative amount of optimal excess x_1^*/b as our measure of interest. In Figure 4.7a, we see that the optimal excess is sensitive to the ratio $(a + b)/c$ but insensitive to the ratio a/b (as shown before). In Figure 4.7b, we see the functional

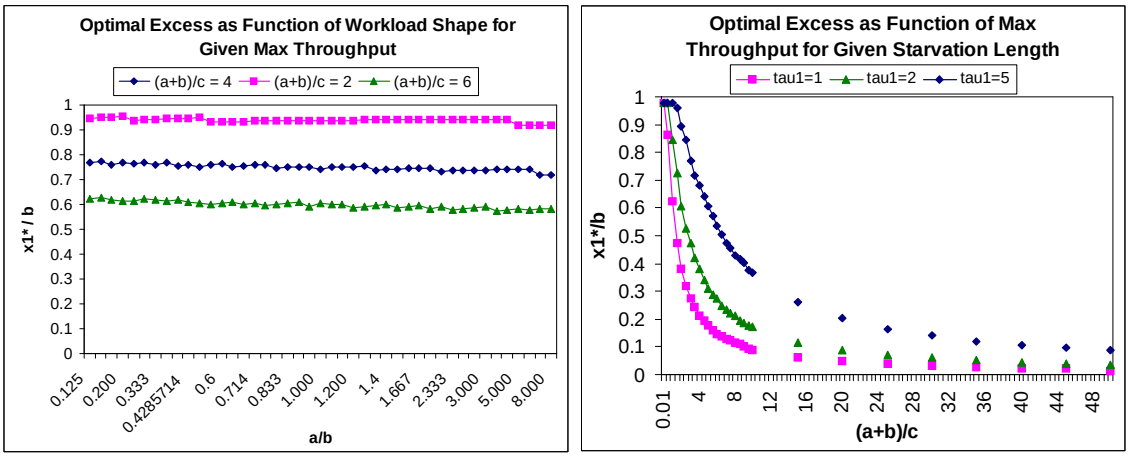


Figure 4.7: *Sensitivity to Throughput Capacity.* As the maximum processing speed of the system increases, the optimal amount of relative excess x_1^*/b decreases. This relationship is independent of the relative shape of the workload function.

role of the parameter c. Specifically, when c is relatively large ($(a+b)/c$ is relatively small), the optimal control is to take on a large amount of excess in anticipation of a starvation event. This makes sense since the system can process this excess work more quickly during the same starvation interval. Conversely, when c is relatively small ($(a+b)/c$ is relatively large), the optimal control is to take on very little excess in anticipation of a starvation event. In this case, the system is unable to process this excess work quickly during the starvation period.

From this analysis, we conclude that the two most important factors in determining the amount of excess work to take in the anticipation of a starvation event are the the duration of the starvation and the relative speed at which the system can process work. Other factors, such as the relative shape of the workload function are less important.

4.3 Chapter Summary

In this chapter, we developed optimal control policies for arrival sequences that are known in advance. We showed that the tradeoff between congestion and starvation is a fundamental tradeoff that must be understood for managers whose objective is to

maximize system throughput. We used both analytical and numerical approaches for determining the optimal control for particular arrival sequences, and we investigated the sensitivity of these control policies to key system and input parameters.

Chapter 5

Stochastic Models

The models of the previous chapter provide a basic understanding of the system dynamics and management tradeoffs for the congestion-sensitive system when future arrivals are known in advance and system processing is deterministic. For most real applications, however, system behavior is rarely known with such certainty. Often, the system inputs are not known in advance, and system processing is often subject to disruptions. In the language of our previous continuous-time model, we need to consider input-output systems that evolve according to

$$\dot{x}(t) = A(t) - w(x(t), \xi(t))$$

where the arrival sequence $A(t)$ is unknown and processing is subject to an unknown disturbance $\xi(t)$. When faced with systems of this type, can we use admission control to manage system behavior, and if so, what form should this control take? In this chapter we extend our previous results to address systems that operate under uncertainty. We observe that while the same tradeoff between congestion and starvation holds in the stochastic case, the functional form of the optimal policy is different.

5.1 Stochastic Models in Discrete-Time

Consider a discrete-time input-output system where X_t denotes the state of the system at time t. Let the system evolve according to

$$X_{t+1} = X_t - D_t + A_t$$

where A_t, D_t represent the respective number of system arrivals and departures in period t. This discrete model is equivalent to the version presented in the previous chapter when $\Delta t = 1$. We can simplify the model even further by restricting the system to live within a discrete state space $S = \{0, 1, 2, \dots\}$. We therefore require here that $A_t, D_t \in \{0, 1, 2, \dots\}$ with $D_t \leq X_t$. Thus, we have a system where work arrives and departs in discrete units.

We can make this model stochastic by explicitly incorporating the uncertainty associated with system arrivals and departures. Here, we treat A_t and D_t as random variables on which we define the appropriate probability mass functions. Let $P_{A_t}(k) = P\{A_t = k\}$ and $P_{D_t}(k) = P\{D_t = k\}$. Consider a particular case when $E[D_t] = w(X_t)$, where w is the workload function described in detail previously. Then for a congestion-sensitive workload function, we have $E[D_t] \to \Omega$ as $X_t \to \infty$, and we expect that the system will exhibit the same congestion-sensitive behavior demonstrated in its deterministic counterpart. Furthermore, a stochastic system of this type will exhibit the same instability and be susceptible to congestion collapse.

5.1.1 Example: single server queue

Consider now a particular instance of the model described above. Assume that in each period a single arrival occurs independently with probability p (no arrival with probability $1-p$). That is, $A_0, A_1, A_2, \dots$ are independent and identically distributed Bernoulli random variables with success probability p. Furthermore, assume that the in each period when the system is in state x a single departure occurs with probability q_x (no departure with probability $1-q_x$).[1] We require that $0 \leq q_x \leq 1$ for all $x > 0$ and $q_0 = 0$ (no departures from an empty system). The resulting system can be modeled

[1] For the sake of simplicity, we have restricted $A_t, D_t \in \{0, 1\}$, however, this need not be the case.

as a discrete time Markov chain (DTMC) with following transition probabilities.

$$\begin{aligned} P\{X_{t+1} = 1 | X_t = 0\} &= p \\ P\{X_{t+1} = 0 | X_t = 0\} &= 1 - p \end{aligned}$$

and for $i > 0$

$$\begin{aligned} P\{X_{t+1} = i + 1 | X_t = i\} &= p(1 - q_i) \\ P\{X_{t+1} = i - 1 | X_t = i\} &= (1 - p)q_i \\ P\{X_{t+1} = i | X_t = i\} &= pq_i + (1 - p)(1 - q_i) \end{aligned}$$

This system is a reasonable model for a discrete-time *queueing system*, where the value X_t is the number of customers (jobs, work, etc.) in the system, and in each time period there is at most a single arrival or departure. A diagram of possible state transitions is illustrated in Figure 5.1.

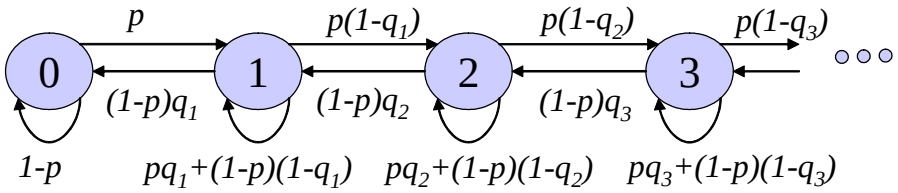


Figure 5.1: *DTMC model for single-server queue.*

Observe that the expected number of arrivals and departures per period depends on the current system state. Based on this setup, $\tilde{A}(x) \equiv E[A_t | X_t = x] = p$ and $\tilde{D}(x) \equiv E[D_t | X_t = x] = q_x$. Furthermore, $E[A_t - D_t | X_t = x] = p - q_x$ is the *expected drift* of the system.

Suppose that the departure probabilities take the particular form $q_x = \beta_1 x e^{-\beta_2 x}$ for some values $\beta_1, \beta_2 > 0$, with β_2 chosen such that $0 \leq q_x \leq 1$ for all $x \in \{0, 1, 2, \dots\}$. A concurrent plot of expected arrival and departure rates as a function of system state shows a familiar picture, illustrated in Figure 5.2, for which our qualitative analysis suggests system instability. Although the stochastic nature of the system will allow the system to explore all values in the state space S, the system drift will tend to push the system toward the point x_1^* and away from the point x_2^*. Furthermore, the

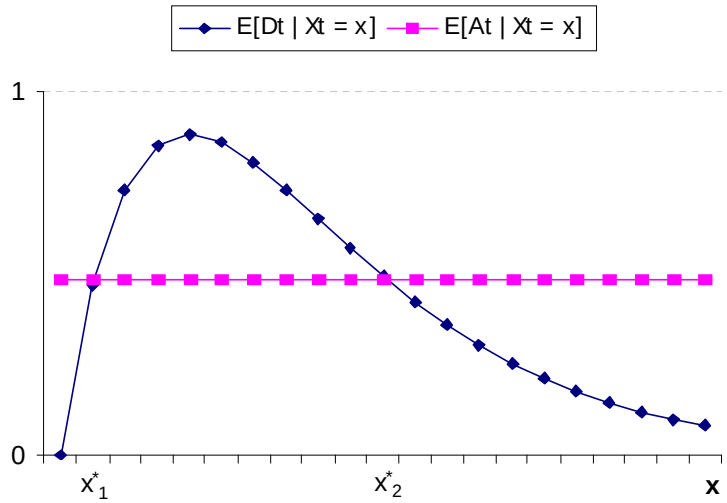


Figure 5.2: *Expected arrival and departure probabilities as a function of system state.*

system is inherently unstable: the system will reach the unstable equilibrium point x_x^* in finite time, after which the system is expected to grow without bound. In other words, this system is inherently unstable and will collapse in finite time. We will show this rigorously in what follows. In the meantime, we consider admission control as an approach to stabilizing the overall behavior of the system. Again, we use dynamic programming.

5.2 DP and Markovian Decisions

Consider a scenario in which during each interval a system operator can decide whether or not to admit any arrival. Assume that blocked arrivals are lost to the system. If the system operator wishes to maximize the throughput of the system, what form should the admission policy take?

Consider a finite stage dynamic program of the following form. Let $J_n(x)$ represent the expected value of the objective function when the system is in state x with n periods remaining[2] and the optimal admission policy is used. Let $A_n(x)$ and $D_n(x)$

[2]The deterministic dynamic program of the last chapter has an equivalent formulation in which the periods are indexed in decreasing order. For the remainder of this thesis, we will maintain this

be random variables representing the number of arrivals and departures that occur when there are n periods to go. Based on this setup, we have the following.

$$\begin{aligned} J_0(x) &= E\{D_0(x)\} = q_x \\ J_n(x) &= E\{D_n(x)\} + \max\left[E\{J_{n-1}(x - D_n(x))\},\right. \\ &\qquad \left.(1-p)E\{J_{n-1}(x - D_n(x))\} + p\ E\{J_{n-1}(x - D_n(x) + 1)\}\right] \end{aligned}$$

where

$$\begin{aligned} E\{J_{n-1}(x - D_n(x))\} &= q_x J_{n-1}(x-1) + (1-q_x)J_{n-1}(x) \\ E\{J_{n-1}(x - D_n(x) + 1)\} &= q_x J_{n-1}(x) + (1-q_x)J_{n-1}(x+1) \end{aligned}$$

Thus, our control rule chooses among the better expected cost-to-go with $n-1$ periods for the case of an admitted arrival and a blocked arrival. The value of the optimal cost-to-go function with n periods to go is equal to the expected number of depatures in that period plus the better optimal cost-to-go with $n-1$ periods remaining.

Observe that the n-stage objective function can be rewritten as follows.

$$\begin{aligned} J_n(x) &= E\{D_n(x)\} + E\{J_{n-1}(x - D_n(x))\} + \\ &\qquad p \cdot \max\left[0, E\{J_{n-1}(x - D_n(x) + 1)\} - E\{J_{n-1}(x - D_n(x))\}\right] \end{aligned}$$

The term $E\{J_{n-1}(x - D_n(x) + 1)\} - E\{J_{n-1}(x - D_n(x))\}$ is equal to the expected marginal improvement in next period's objective from admitting an additional job when there are n periods remaining. So we will choose to block an incoming arrival whenever

$$\begin{aligned} E\{J_{n-1}(x - D_n(x) + 1)\} &< E\{J_{n-1}(x - D_n(x))\} \\ q_x J_{n-1}(x) + (1-q_x)J_{n-1}(x+1) &< q_x J_{n-1}(x-1) + (1-q_x)J_{n-1}(x) \\ (1-q_x)\left(J_{n-1}(x+1) - J_{n-1}(x)\right) &< q_x\left(J_{n-1}(x) - J_{n-1}(x-1)\right) \\ \frac{J_{n-1}(x+1) - J_{n-1}(x)}{J_{n-1}(x) - J_{n-1}(x-1)} &< \frac{q_x}{1-q_x} \end{aligned}$$

provided that $J_{n-1}(x) - J_{n-1}(x-1) \neq 0$ and $q_x \neq 1$. When $q_x = 1$, we block an arrival whenever $J_{n-1}(x) < J_{n-1}(x-1)$. This says that whether or not we choose

convention of a decreasing index.

to block an incoming arrival can be interpreted in terms of the ratio between the marginal increase to the objective for the *next* arrival and the marginal increase to the objective for the *last* arrival. In particular, whenever this ratio is less than the relative likelihood of a departure in the next period, we will choose to block the arrival.

What does this ratio tell us about the form of the optimal value function? Since the departure probabilities q_x are quasiconcave on $S = \{0, 1, 2, \dots\}$, we know that there exists an x^* such that q_x is nondecreasing whenever $x \leq x^*$ and q_x is nonincreasing whenever $x \geq x^*$. It then follows that the ratio $q_x/(1 - q_x)$ is nonincreasing when $x \geq x^*$. It seems reasonable therefore that under appropriate conditions on the objective functions $J_n(x)$ that there will exist some finite value of x above which it is never optimal to admit additional arrivals.

5.3 Form of Optimal Control Policy

Based on the quasiconcavity of the the departure probabilities q_x, we have speculated that the optimal input control can be understood in terms of an admission *threshold*. That is, we believe that in any time period n, there exists a threshold value θ_n such that it is optimal to admit new arrivals when $x_n \leq \theta_n$ and it is optimal to block new arrivals when $x_n > \theta_n$. We can formalize this notion by establishing an *optimal control policy* for the congestion-sensitive processing system. In this section, we investigate the form of the optimal control policy for both the finite horizon and infinite horizon problems. However, we first review some of the previous work that has been conducted on the use of admission control in stochastic processing systems.

5.3.1 Previous Work on Optimal Control in Queueing

The study of optimal control for input-output systems has been conducted primarily in the context of queueing systems. For queueing systems with finite service capacity and a first-in first-out (FIFO) service discipline, individuals joining the queue impose a penalty on *future* arrivals in the form of increased waiting times for those customers.

In this manner, congestion results in decreased service for customers although the performance of the server remains unaffected. This simple form of congestion has been a dominant theme in the literature on queueing.

In the context of a queueing system, optimal control generally means maximizing the performance of the queue with respect to a particular system objective. In this study, we are concerned with the maximizing the throughput of the input-output system. Let $Z(t)$ equal the total aggregate throughput of the system over the continuous time interval $[0, t]$, and let α denote the decision policy to be used for controlling the system. Within the literature, there are two standard representations for system throughput. The first is the *expected total discounted reward over an infinite horizon*, which we denote here by $V_x^\beta(\alpha)$ and with the following form.

$$V_x^\beta(\alpha) = E_\alpha \left\{ \int_0^\infty e^{-\beta t} dZ(t) \middle| X_0 = x \right\}$$

where α is our decision policy, β is the discount rate, and x is the starting point of the system. An alternate approach is to consider the *long-run average expected throughput* denoted here by $V_x(\alpha)$ and represented as follows.

$$V_x(\alpha) = \lim_{t\to\infty} \inf \frac{1}{t} \, E_\alpha \left\{ Z(t) \mid X_0 = x \right\}$$

For a review of the formulation and known results for these control problems, consult Stidham and Prabhu [114] or Bertsekas [19].

From the outset, it has been understood that admission control could be an effective means for managing the overall congestion of a queueing system. There are two basic ways in which admission control can be implemented. In the first, a system operator or manager makes an admission decision for each potential arrival to the system. That arrival is either admitted to the system or denied entry (we say that the arrival is *blocked*) and leaves the system never to return. The system operator makes admission decisions in a manner that tries to optimize the performance of the system as a whole. The admission policy chosen is called the *socially optimal* solution. An alternate perspective is one in which customers arriving to the system are given a choice to enter the system or to *balk*. In this model, each customer makes an admission decision based on what is *individually optimal* to her, without regard to

the performance of the system as a whole.

Naor [82] was among the first to formalize these ideas with quantitative analysis. He showed that within a static context the optimal admission policy for each perspective of the threshold type. Denoting by θ_S the threshold for the socially optimal policy, the system operator will choose to admit a customer only when the current number $n < \theta_S$. Similarly with θ_I denoting the admission threshold for individuals, an arriving customer will choose to join the system only when $n < \theta_I$. Naor showed that in general $\theta_S < \theta_I$, meaning that individually optimized decisions tend to overload the system beyond the socially optimal level. However, by imposing a *toll* or other pricing mechanism on the individual at the time of arrival to reflect the external affect of that customer's arrival on the system as a whole, it is possible to induce socially optimal behavior among individual decision makers.

The use of admission control in queueing has figured prominently in subsequent work on control of queueing systems. Miller [79] developed policies for admitting customers with variable rewards in order to maximize reward per unit time. Yechiali [133, 134] developed optimal admission policies for exponential queues with general arrival processes. Blackburn [21] considered optimal control of a queue in which customers can renege as well as balk. Stidham and Prabhu [114] provide a comprehensive review of the various formulations and solutions on optimal control of queueing.

Lippman and Stidham [73] investigated the distinction between individual and social optimization. Low [75] developed dynamic pricing policies for admission control. Johansen and Stidham [63] extended the analysis on admission control to general input-output systems. Serfozo [109] generalized the analysis for both admission and service rate control to a broad class of stochastic processes, including random walks, birth-death processes, and queueing systems. Stidham [115] provides a nice summary of the models and results in the study of admission control for queueing systems.

While admission control is an effective means for managing the behavior of queueing systems, alternative approaches do exist. For example, methods that control the service rate [56, 135, 41, 21, 111, 107, 44] can be effective at minimizing congestion and achieving desired system performance. A partial review of the literature on the analysis and design of control mechanisms for single server queues is available from

Crabill, Gross and Magazine [42]. Again, a unified mathematical treatment of the control mechanisms in queueing systems and similar stochastic processes is available from Serfozo [109]. The utility of each control mechanism depends on the constraints of the application under investigation.

In this research, we are investigating a system in which the performance of the server itself is sensitive to congestion. In this manner, the arrival of an additional customer to the system imposes a penalty on *current customers* in the system in addition to future arrivals. For this reason, we do not consider altering the behavior of the server and instead focus exclusively on the use of admission control to manage the behavior of our congestion-sensitive processing systems.

5.3.2 Finite Horizon Problems

For the finite horizon problems discussed thus far, we have been interested in the optimal β-discounted expected throughput, given as follows.

$$V_{x,T}^{\beta}(\alpha) = E_{\alpha}\left\{ \int_0^T e^{-\beta t} dZ(t) \middle| X_0 = x \right\}$$

As before, β is the discount rate, and x is the starting point of the system. We seek an admission policy α that depends on the current level of work in the system and the number of periods remaining and maximizes this throughput.

In the context of our discrete-time queueing problem discussed previously, we choose $\beta = 1$ and represent system throughput as follows

$$V_{x,n}(\alpha) = E_{\alpha}\left\{ \sum_{k=0}^{n} q_{n-k} \middle| X_0 = x \right\}$$

We know that for optimal policy α^*, we have $V_{x,n}(\alpha^*) = J_n(x)$ where, $J_n(x)$ satisfies the DP recursive relationship from before.

$$\begin{aligned} J_n(x) &= E\{D_n(x)\} + E\{J_{n-1}(x - D_n(x))\} + \\ &\quad p \cdot \max\left[0, E\{J_{n-1}(x - D_n(x) + 1)\} - E\{J_{n-1}(x - D_n(x))\}\right] \end{aligned}$$

That is, we will choose the input control $\alpha_n(x) \in \{0, 1\}$ when the current level of work is x and there are n periods remaining. The vector $\alpha = \{\alpha_n, \alpha_{n-1}, \ldots, \alpha_0\}$ represents

the optimal control policy for the system. Again, we believe that each element of this policy $\alpha_n(x)$ is a threshold function in x.

$$\alpha_n(x) = \begin{cases} 1 & x \leq \theta_n \\ 0 & x > \theta_n \end{cases}$$

One approach to validating this conjecture would be to show that $\alpha_n(x)$ is non-decreasing in x. To do this, we would need to construct an inductive proof in the style of [114]. Essentially, we look to prove that the optimal value function $J_n(x)$ is quasiconcave on the countable state space $\{0, 1, 2, \dots\}$. Although a formal proof of this type is beyond the current scope of this thesis, we point out a few features in the sketch of this proof that make this approach promising.

As a basis, note that $J_0(x) = q_x$ is quasiconcave, and therefore a threshold policy is optimal in the terminal stage. To make the inductive step, we need to show that for $J_{n-1}(x)$ quasiconcave that $J_n(x)$ is also quasiconcave. To do this, recall the form of the n-stage objective function in slightly modified form

$$J_n(x) \;=\; q_x + (1-p)\; h(x) + p\;\max\left[h(x), h(x+1)\right]$$

where we have defined the functions

$$\begin{aligned} h(x) &= q_x J_{n-1}(x-1) + (1-q_x) J_{n-1}(x) \\ h(x+1) &= q_x J_{n-1}(x) + (1-q_x) J_{n-1}(x+1) \end{aligned}$$

The crux of the argument is in showing that the function $h(x)$ is indeed quasiconvex in x. While we have not yet proved this to be true, similar arguments have been made by Stidham and Prabhu [114], Topkis [120], Serfozo [109], and Stidham [115].

5.3.3 Infinite Horizon Problems

Our analysis of the optimal control policy is greatly simplified for the case of an infinite horizon problem because the number of periods remaining at every stage is infinite and therefore the same. It is therefore possible to restrict attention to the

identification of a *stationary* control policy of the form $\alpha = \{\alpha^*, \alpha^*, \alpha^*, \dots\}$

$$\alpha^*(x) = \begin{cases} 1 & x \leq \theta^* \\ 0 & x > \theta^* \end{cases}$$

where there is a single threshold value θ^*. As before, we consider the case of a congestion-sensitive queue in which for each time period a single arrival occurs independently with probability p and a single departure occurs independently with state-dependent probability q_x.

For the case of the discounted expected reward over an infinite horizon, one approach to obtaining the optimal control policy is to continue the induction for the finite horizon problem in the limit as $n \to \infty$. While it is known that an optimal stationary policy for this type of problem exists [22], there is additional work in the analysis of the value function to obtain its form. This approach is equivalent to the method of *value iteration* discussed below.

A more straightforward approach is available if we consider the *long-run average expected throughput* for the system. Let $J_x(\alpha)$ represent the value function when the system is in state x and stationary policy α is used. For the DTMC $\{x_k, k \geq 0\}$, this is the average system output per stage under policy α. For an initial state x_0, the value function equals the following.

$$J_{x_0}(\alpha) = \lim_{t \to \infty} \frac{1}{t} \; E\left\{ \sum_{k=0}^{t-1} q_{x_k} \middle| \, x_0 \right\}$$

This system evolves according to transition probabilities $P_{ij}(\alpha(i))$ when action $\alpha(i)$ is taken in state i. Using the standard matrix forms

$$J(\alpha) = \begin{pmatrix} J_i(\alpha) \end{pmatrix} \qquad P(\alpha) = \begin{pmatrix} P_{ij}(\alpha(i)) \end{pmatrix} \qquad q = \begin{pmatrix} q_i \end{pmatrix}$$

we obtain

$$J(\alpha) = \left(\lim_{t \to \infty} \frac{1}{t} \; \sum_{k=0}^{t-1} P^k(\alpha) \right) q$$

where $P^k(\alpha)$ is the matrix of k-step transition probabilities under policy α. Since

this matrix is stochastic, it is known [18] that the following limit

$$P^*(\alpha) = \lim_{t\to\infty} \frac{1}{t} \sum_{k=0}^{\infty} P^k(\alpha)$$

exists. $P^*(\alpha)$ is the matrix of steady-state probabilities under policy α for each initial state. We therefore have $J(\alpha) = P^*(\alpha)\ q$.

Under the additional restriction that the policy α yields a recurrent Markov chain, these steady-state probabilities will be independent of the initial state. Let $\gamma(\alpha)$ be the average output rate under stationary policy α. This implies that $J_i(\alpha) = \gamma(\alpha)$ for all i. It turns out that this additional restriction is a necessary condition for an optimal policy.

Optimal Nonrandomized Stationary Policy

We will now show conclusively that a nonrandomized threshold policy is indeed optimal for this alternative, but equivalent, form of this stochastic control problem evolving over an infinite horizon.

Proposition 1. *Among the class of nonrandomized stationary policies, a threshold policy yields the optimal average output rate for the congestion-sensitive system.*

Proof. We first argue that any stationary policy that does not admit a positive recurrent Markov chain cannot be optimal. Since we have assumed that $q_x \to 0$ as $x \to \infty$, it is clear that any policy that allows the amount of work in the system to grow without bound will result in suboptimal performance. Thus, a necessary condition for optimality is that the policy results in a recurrent Markov chain.

We can characterize the recurrent class of the Markov chain in the following manner. For any stationary policy α, let $n = \min\{i : \alpha(i) = 0\}$. That is, n is the smallest amount of work in the system for which the system under policy α blocks incoming arrivals. In the simple case where there exists a finite failure state N, it is clear that an optimal policy must choose $n < N$. In any case, a policy that does not allow divergence must choose $n < \infty$. Under any such policy α, given an initial starting point $i < n$ the k-step transition probability $P_{ij}^k(\alpha) = 0$ for all $j > n$ and all values

of k. Thus, a necessary condition for an optimal policy is that with probability 1 the system will become trapped within a recurrent class $\{0, 1, 2, \ldots, n\}$.

We now show that among the stationary policies that admits a recurrent Markov chain, a threshold policy yields the optimal average output. Consider a state $s \in \{0, 1, 2, \ldots, n\}$. For any initial state i, we define the *hitting time* to state s as t_{is}, given by

$$t_{is} = \min\{t : x_t = s, x_{t-1} \neq s, \ldots, x_1 \neq s, x_0 = i\}.$$

The expected hitting time from state i clearly depends on the policy α that is being used. In order for a policy α to yield a recurrent system, we must have $E[t_{is}(\alpha)] < \infty$. That is, under an optimal stationary policy α the expected time to reach state s from any initial state i must be finite.

The average output rate for all policies in this family will be equal to $J_i(\alpha)$.

$$\begin{aligned} J_i(\alpha) &= \lim_{t\to\infty} \frac{1}{t}\, E\left\{ \sum_{k=0}^{t_{is}} q_{x_k} + \sum_{k=t_{is}}^{t} q_{x_k} \middle| x_0 = i \right\} \\ &= \lim_{t\to\infty} \frac{1}{t}\, E\left\{ \sum_{k=0}^{t_{is}} q_{x_k} \middle| x_0 = i \right\} + \lim_{t\to\infty} \frac{1}{t}\, E\left\{ \sum_{k=t_{is}}^{t} q_{x_k} \middle| x_{t_{is}} = s \right\} \\ &= \lim_{t\to\infty} \frac{1}{t}\, E\left\{ \sum_{k=0}^{t} q_{x_k} \middle| x_0 = s \right\} \\ &= J_s(\alpha) \end{aligned}$$

Since, the contribution by the initial transient phase of any sample path of the system is zero, all policies which result in the recurrent class $\{0, 1, 2, \ldots, n\}$ will result in the same average output rate. Thus, given an optimal value of n^*, a stationary policy of the form

$$\alpha(i) = \begin{cases} 1 & 0 \leq i < n^* \\ 0 & i \geq n^* \end{cases}$$

will yield an optimal average output rate. Since this is a nonrandomized threshold policy, the proof is complete. ■

Bellman's Equation

In this section, we review the connection to Bellman's Equation and discuss some of the computational methods available for determining the particular optimal policy for a given problem. The discussion here is adapted from that in [19, 103].

Let $\gamma(\alpha)$ be the average output per stage achieved under policy α. We have shown that as long as policy α results in the single recurrent class $\{0, 1, 2, \ldots, n\}$ that the average output per stage is independent of the initial state. That is, $J_i(\alpha) = \gamma(\alpha)$ for all $i \in S$. This average cost per stage satisfies Bellman's Equation

$$\gamma(\alpha) + h_i(\alpha) = q_i + \sum_{j=0}^{n} P_{ij}(\alpha(i))\ h_j(\alpha)$$

where the vector $h(\alpha)$ has interpretation as the *differential* or *relative cost vector*. For any chosen state $s \in \{0, 1, 2, \ldots, n\}$, we can think of $h_i(\alpha)$ as the expected amount of system output that will occur as the system moves from i to s. Since we know that $E[t_{is}(\alpha)] < \infty$ for all i, the values for $h_i(\alpha)$ are finite for all $i \neq s$. These costs are then uniquely solved by the following system of equations.

$$\begin{aligned} \gamma(\alpha) + h_i(\alpha) &= q_i + \sum_{j=0}^{n} P_{ij}(\alpha(i))\ h_j(\alpha), \quad \text{for } i \neq s \\ h_s(\alpha) &= 0 \end{aligned}$$

This leads naturally to a number of approaches to computing the optimal policy for a given problem. These methods are detailed in [58, 19, 18] but are summarized here for completeness.

- *Value Iteration.* This method is the simplest approach to computing the optimal policy. Essentially, one selects an arbitrary terminal cost function J_0 and then successively computes the optimal k-stage costs using the DP recursion.

  $$J_i^{k+1} = \min_{u \in U(i)} \left\{ q_i + \sum_{j=0}^{n} P_{ij}(\alpha(i))\ J_j^k \right\}$$

  Under the conditions that Bellman's equation can be solved for some vector h^*,

then the successive costs of the value iteration will converge.

$$\frac{J_i^k}{k} \to \gamma^*$$

The drawbacks of this approach is that the components of J^k may diverge to $+\infty$ or $-\infty$, resulting in numerical overflow or underflow issues. Also, this approach does not solve for the optimal differential costs h^*. A variant of this method, known as *Relative Value Iteration* overcomes these drawbacks by iterating on the relative value $h_i^k = J_i^k - J_s^k$, where s is some fixed state in the recurrent class.

- *Policy Iteration*. This method is characterized by a stationary policy α^k during iteration k. For each iteration, we compute the corresponding average cost $\gamma^k(\alpha)$ and differential costs $h_i^k(\alpha)$ satisfying the equations.

$$\begin{aligned} \gamma^k(\alpha) + h_i^k(\alpha) &= q_i + \sum_{j=0}^{n} P_{ij}(\alpha^k(i))\ h_j^k(\alpha), \quad \text{for } i \neq s \\ h_s^k(\alpha) &= 0 \end{aligned}$$

We then find a new stationary policy α^{k+1} that satisfies for all i

$$q_i + \sum_{j=0}^{n} P_{ij}(\alpha^{k+1}(i))\ h_j^k(\alpha) = \min_{u \in U(i)} \left\{ q_i + \sum_{j=0}^{n} P_{ij}(u)\ J_j^k \right\}$$

The algorithm terminates when $\gamma^{k+1}(\alpha) = \gamma^k(\alpha)$ and $h_i^{k+1}(\alpha) = h_i^k(\alpha)$ for all i. Under the condition that all stationary policies α result in a recurrent Markov chain, the policy iteration method will obtain the optimal policy in a finite number of steps.

Randomized Stationary Policies

An alternative approach in solving for the optimal policy would be to consider the family of *randomized* stationary policies. It seems plausible that a randomized policy, in which the likelihood of admitting an arrival gradually decreases with increased work, might outperform a nonrandomized policy with a sharp threshold. Here, we repeat the analysis of [103] to answer conclusively in the negative. We employ a linear

programming formulation to make our point.

Consider the case in which the admission policy when in state i is given by value $\tilde{\alpha}(i) \in [0,1]$. That is, $\tilde{\alpha}(i)$ represents the *probability* of choosing $\alpha(i) = 1$ from the action space $\{0,1\}$. Let π_i be the stationary probability of being in state i. As before, we restrict our attention to randomized policies that admit a recurrent steady-state solution on the restricted state space $\{0,1,2,\ldots,n\}$. As a result, we know that these stationary probabilities are guaranteed to exist and must satisfy $\sum_{i=0}^{n} \pi_i = 1$, with $\pi_i = 0$ for $i > n$.

Define π_i^a to be the probability of being in state i *and* choosing action $a \in \{0,1\}$. That is, $\pi_i^1 = \pi_i \; \tilde{\alpha}(i)$ and $\pi_i^0 = \pi_i \; (1 - \tilde{\alpha}(i))$. The average system output is therefore

$$\sum_{i=0}^{n} \left(\pi_i^0 + \pi_i^1\right) q_i = \sum_{i=0}^{n} \pi_i q_i.$$

In this manner, we can formulate the following linear program to optimize the average system output.

$$\begin{aligned}
\max \quad & \sum_{i=0}^{n} \left(\pi_i^0 + \pi_i^1\right) q_i \\
s.t. \quad & \pi_i^0 + \pi_i^1 = \sum_{j=0}^{n} \left(\pi_j^0 \; P_{ji}(0) + \pi_j^1 \; P_{ji}(1)\right), i \in \{0,1,\ldots,n\} \\
& \sum_{i=0}^{n} \pi_i^0 + \pi_i^1 = 1 \\
& \pi_i^0, \pi_i^1 \geq 0, i \in \{0,1,\ldots,n\}
\end{aligned}$$

It has been shown [19, 17] that the *dual linear program* of this problem is the following.

$$\begin{aligned}
\min \quad & \gamma \\
s.t. \quad & \gamma + h_i \geq q_i + \sum_{j=0}^{n} P_{ij}(u) \; h_j, i \in S, u \in \{0,1\} \\
& \gamma \text{ unrestricted}
\end{aligned}$$

In other words, a solution (γ, h) of this dual program is feasible if it satisfies

$$\gamma + h_i \geq q_i + \sum_{j=0}^{n} P_{ij}(\alpha(i)) \; h_j$$

for all nonrandomized stationary policies α. A feasible solution must therefore satisfy $\gamma \geq \gamma^*$ where (γ^*, h^*) is the optimal solution. However, we know that for the optimal nonrandomized policy α^*, we have

$$\gamma(\alpha^*) + h_i(\alpha^*) = q_i + \sum_{j=0}^{n} P_{ij}(\alpha^*(i))\ h_j(\alpha^*).$$

Therefore, the optimal nonrandomized policy α^* with differential costs $h(\alpha^*)$ solves the dual program. By duality theory, we know that the minimal value of the dual cannot be less than the maximal value of the primal problem. Therefore, we have

$$\sum_{i=0}^{n} \pi_i q_i \leq \gamma(\alpha^*)$$

which proves the following proposition.

Proposition 2. *A randomized stationary policy cannot outperform the optimal nonrandomized strategy for the congestion-sensitive queueing problem.*

Again, additional details of this analysis can be found in [19].

5.4 Value of Information

There is a difference between the solution obtained by the deterministic and stochastic DP algorithms that comes from a difference in the information that is available about future arrivals. An important question is, *How valuable is this information?* One approach is to consider DP formulations in which the amount of information that available is parameterized so as to allow for *partial information*. But what does it mean to have partial information about future arrivals? There are at least two possible answers.

1. *Partial* information about arrivals in *all* periods
2. *Complete* information about arrivals in *some* periods

We can consider two formulations that address these notions.

5.4.1 Blended Arrival Streams

Consider a system in which aggregate arrivals are comprised of two different streams. One stream is known deterministically to the controller, the second is known only probabilistically to the controller. We assume that the two streams have the same overall statistics. For example, they could be two independent realizations of the same sequence of random numbers. Denote the unknown arrival and departure sequences $\tilde{A}_t$ and $\tilde{D}_t$ respectively to distinguish them from their known counterparts.

- Let $\{A_k\}$ be the "known" arrival sequence
- Let $\{\tilde{A}_k\}$ be the "unknown" arrival sequence

Let β denote the fractional amount of input stream A_k. Then, the overall arrivals are given by

$$\hat{A}_k = \beta A_k + (1 - \beta)\tilde{A}_k.$$

In this manner, we can test the performance of the solution obtained by the DP algorithm for different values of β. Again, for $\beta = 1$, we have the deterministic arrival sequence studied in the previous chapter. For $\beta = 0$, we have the probabilistic arrival sequence given in the preceeding section of this chapter. This is illustrated in Figure 5.3.

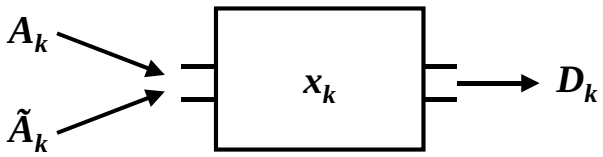


Figure 5.3: *Input-output system subject to blended arrival streams.*

This type of approach might be thought of as a deterministic arrival sequence with some stochastic noise on top of it. The parameter β controls the relative contribution of the signal versus the noise.

The behavior of stochastic processing systems in the presence of multiple arrival streams has been studied in the context of queueing. Blanc *et al* [23] studied the case of a multi-server queue with two separate unknown arrival streams, the first (type

1) subject to admission control and the other (type 2) uncontrolled. In their model, the system is rewarded only for admission of type 1 arrivals, and the decision horizon is infinite. They showed that for both the case of long-run discounted reward and long-run average reward the optimal admission policy for the type 1 arrivals is of the threshold type, and they characterized the threshold values for each case.

For the case of separate known (deterministic) and unknown (stochastic) arrival streams, it seems likely that the optimal policy will use a threshold mechanism to control admission to the system. In particular, it is reasonable to choose $\alpha(x_k) = \min\{[\beta - x_k]^+, A_k + \tilde{A}_k\}$. Then, we will want to admit an amount $u_k = \alpha(x_k)$ so that the system evolves according to $x_{k+1} = x_k - w(x_k) + u_k$. In the case when some arrivals are blocked, it is also necessary to specify whether the blocked arrivals will be from A_k, from $\tilde{A}_k$, or from both. While it seems reasonable to give *priority* to to deterministic arrivals, a thorough investigation of these issues is required before the advantages are apparent.

Numerical results for this type of formulation are possible, but they are beyond the current scope of this research. However, we can speculate on the effect of β on the performance of the solution obtained by the DP algorithm. Let $J^\beta(x_0)$ be the value of the solution obtained by the DP algorithm for initial system value x_0 and under arrival uncertainty β. In general, we expect that $J^\beta(x_0)$ is nondecreasing in β, and at a minimum we would like to be able to show that $J^1(x_0) \geq J^0(x_0)$. We would like to illustrate this sensitivity using a graph like the one shown in Figure 5.4.

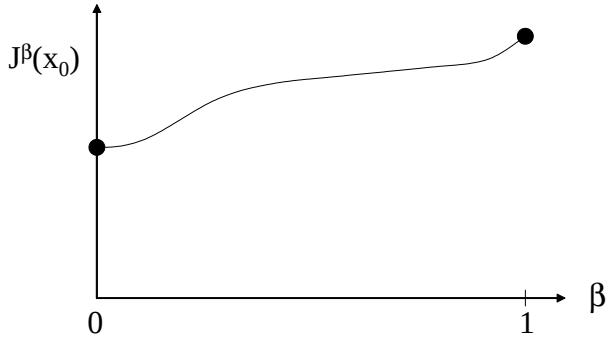


Figure 5.4: *Anticipated sensitivity of optimal DP solution to values of β.*

5.4.2 Horizon Methods

Another way to try to understand the value of information is to consider a case in which the system can observe with certainty the future arrivals for a finite number of time periods into the future. Such approaches are akin to *forecasting methods* in DP, such as those studied in [19].

Consider the deterministic N-period finite horizon problem from Section 4.1.3. Here, we write the optimal cost-to-go $J_k(x_k)$ in the following modified form.

$$\begin{aligned} J_k(x_k) &= w(x_k) + \max_{0\le u_k\le 1}\Big[w(x_{k+1}) + \max_{0\le u_{k+1}\le 1}\Big[w(x_{k+2}) + \dots \\ &\quad \cdots + \max_{0\le u_{N-2}\le 1}\Big[w(x_{N-1}) + \max_{0\le u_{N-1}\le 1} w(x_N)\Big]\dots\Big]\Big] \end{aligned}$$

In this formulation, we use a *proportional input control* such that the system evolves according to $x_{k+1} = x_k - w(x_k) + u_k A_k$, with $0 \le u_k \le 1$. For the stochastic version of this problem, the optimal cost-to-go function $\tilde{J}_k(x_k)$ takes the following form.

$$\begin{aligned} \tilde{J}_k(x_k) &= w(x_k) + \max_{0\le u_k\le 1} E_{\tilde{A}_k}\Big\{w(x_{k+1}) + \max_{0\le u_{k+1}\le 1} E_{\tilde{A}_{k+1}}\Big\{w(x_{k+2}) + \dots \\ &\quad \dots + \max_{0\le u_{N-2}\le 1} E_{\tilde{A}_{N-2}}\Big\{w(x_{N-1}) + \max_{0\le u_{N-1}\le 1} E_{\tilde{A}_{N-1}}\Big\{w(x_N)\Big\}\Big\}\dots\Big\}\Big\} \end{aligned}$$

Define $J_k^i(x_k)$ to be the value of the optimal cost-to-go function when the incoming arrivals are known deterministically for the next i periods, and known only probabilistically after that. The recursive relationship is given by

$$\begin{aligned} J_k^i(x_k) &= w(x_k) + \max_{0\le u_k\le 1} J_{k+1}^{i-1}(x_{k+1}), \text{ for } i > 0, \\ J_k^0(x_k) &= w(x_k) + \max_{0\le u_k\le 1} E_{A_k}\left\{J_{k+1}(x_{k+1})\right\}. \end{aligned}$$

In this context, the parameter i can be interpreted as the *horizon* over which incoming arrivals are known. The optimal control sequence with a fixed horizon i is given by

$$u_k^*(x_k) = \arg\max_{0\le u_k\le 1} J_{k+1}^i(x_k - w(x_k) + u_k A_k).$$

In other words, the optimal control is selected based on a horizon of length i.

The drawback of this formulation is that in order to compute J_k^i one must first compute $J_{k+1}^{i-1}, J_{k+2}^{i-2}, \dots J_{k+i}^0$. However, we can compute these values recursively start-

ing with $i = 0$.

$$\begin{aligned}
J_k^0(x_k) &= w(x_k) + \max_{0 \le u_k \le 1} E_{A_k} \{J_{k+1}(x_{k+1})\} \\
&= w(x_k) + \max_{0 \le u_k \le 1} E_{A_k} \{J_{k+1}(x_k - w(x_k) + u_k A_k)\} \\
J_k^1(x_k) &= w(x_k) + \max_{0 \le u_k \le 1} \left[J_{k+1}^0(x_{k+1})\right] \\
&= w(x_k) + \max_{0 \le u_k \le 1} \left[w(x_{k+1}) + \max_{0 \le u_{k+1} \le 1} E_{A_{k+1}} \{J_{k+2}(x_{k+2})\}\right]
\end{aligned}$$

For general horizon length i,

$$\begin{aligned}
J_k^i(x_k) &= w(x_k) + \max_{0 \le u_k \le 1} \left[J_{k+1}^{i-1}(x_{k+1})\right] \\
&= w(x_k) + \max_{0 \le u_k \le 1} \left[w(x_{k+1}) + \max_{0 \le u_{k+1} \le 1} \left[w(x_{k+2}) + \ldots \right.\right. \\
&\quad \left.\left. \cdots + \max_{0 \le u_{k+j} \le 1} \left[w(x_{k+j}) + \max_{0 \le u_{k+j+1} \le 1} E_{A_{k+j+1}} \{J_{k+j+2}(x_{k+j+2})\}\right] \ldots \right]\right] \\
&= w(x_k) + \max_{0 \le u_k \le 1} \left[w(x_{k+1}) + \max_{0 \le u_{k+1} \le 1} \left[w(x_{k+2}) + \ldots \max_{0 \le u_{k+j} \le 1} \left[w(x_{k+j}) + \right.\right.\right. \\
&\quad \max_{0 \le u_{k+j+1} \le 1} E_{A_{k+j+1}} \left\{w(x_{k+j+2}) + \max_{0 \le u_{k+j+2} \le 1} E_{A_{k+j+2}} \left\{w(x_{k+j+3}) + \ldots \right.\right. \\
&\quad \left.\left.\left.\left. \cdots + \max_{0 \le u_{N-1} \le 1} E_{A_{N-1}} \left\{w(x_N)\right\}\right\} \ldots \right\}\right\}\right] \ldots \Bigg]\Bigg]
\end{aligned}$$

and up to a total horizon length of $N - k$

$$\begin{aligned}
J_k^{N-k}(x_k) &= w(x_k) + \max_{0 \le u_k \le 1} \left[w(x_{k+1}) + \max_{0 \le u_{k+1} \le 1} \left[w(x_{k+2}) + \ldots \right.\right. \\
&\quad \left.\left. \cdots + \max_{0 \le u_{N-2} \le 1} \left[w(x_{N-1}) + \max_{0 \le u_{N-1} \le 1} w(x_N)\right] \ldots \right]\right]
\end{aligned}$$

where again, $x_{k+1} = x_k - w(x_k) + u_k A_k$. It should be clear that $J_k^{N-k}(x_k) = J_k(x_k)$ (the deterministic DP formulation) and $J_k^0(x_k) = \tilde{J}_k(x_k)$ (the stochastic formulation). In this manner, it is possible to interpolate between a horizon of length zero and a (complete) horizon of length N.

While the numerical results of this formulation are still pending, we expect the following effect of an increased horizon i on the performance of the solution. Let $J^i(x_0)$ be the value of the solution obtained by the DP algorithm for initial system value x_0

and under finite horizon i. In general, we expect that $J^i(x_0)$ is nondecreasing in i, and we also expect that the marginal benefit from increasing the horizon diminishes as the horizon grows. Thus, we expect $J^i(x_0)$ to be concave in i, as illustrated by the hypothetical graph in Figure 5.5.

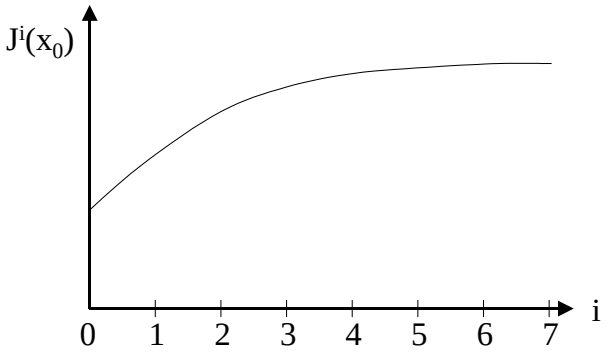


Figure 5.5: *Diminishing returns for increasing horizon.*

It seems likely that the marginal benefit will decrease quickly over relatively small value of i. In cases where there is a cost associated with increasing the horizon, the ability to identify an optimal horizon will be of great importance.

5.5 Chapter Summary

In this chapter, we investigated models in which system behavior is stochastic. Leveraging modeling insight from the literature on queueing theory, we showed that the optimal admission policy for the congestion-sensitive processor is of the threshold type. The uncertain nature of stochastic system behavior prompted us to ask questions about the value of information about future arrivals to the system. We extended traditional analysis to speculate on the answer to this question.

Chapter 6

Birth-Death Models

Consider a continuous-time version of the model presented in the last chapter. In this model, arrivals occur randomly but now within a continuum of possible time values, generally assumed to be the open real interval $[0, \infty)$, and according to a known probability distribution. System departures are similarly stochastic and in continuous time, however the probability of a departure at any moment in time depends on the current amount of work in the system.

As long as we assume that the system evolves on the discrete state space $S = \{0, 1, 2, \dots\}$ (that is, the work in the system can be intepreted in terms of discrete jobs, customers, etc.) and the probability of future events depends only on the present system state and not the past, then our system is a Continuous-Time Markov Chain (CTMC). Since the system evolves in continuous time, it is probabilistically impossible for more than one event to occur at any moment in time. Assuming that our model precludes the arrival or departure of work in *batches*, then our system falls within a special class of CTMC models, known as *birth-death processes*. In the remainder of this section, we will consider several variations of birth-death models that yield additional insight into our input-output systems.

6.1 Birth-Death Models

A *birth-death system* is a particular type of continuous-time Markov chain (CTMC) in which the behavior of the system is restricted to unit increments, called *births*, or unit decrements, called *deaths*. Birth-death models have been used extensively to study population dynamics and a host of other important applications, including queueing and inventory systems. For a complete review of the basics of Markov chains and birth-death processes, see [104, 117, 66].

In modeling our processing system, we choose the system variable $X(t)$ to represent the amount of work in the system at time t. We assume that the system is constrained to live within the state space $S = \{0, 1, 2, \dots\}$. We further assume that time evolves continuously and that changes in the value of $X(t)$ are restricted to unit changes. In other words, if $X(t) = x > 0$, the next jump can only be to either $x+1$ or $x-1$. We characterize the jump behavior in terms of *birth rates* and *death rates*, each of which depend on the current system state. Following the standard convention for notation, let λ_x and μ_x be the corresponding birth and death rates when the system is in state $x \in S$. Mathematically, these transition rates are known to satisfy for each $x \in S$ the relationship

$$\begin{aligned} \lambda_x &= \lim_{\epsilon \to 0} \frac{P\{X(t+\epsilon) = x+1 | X(t) = x\}}{\epsilon}, \\ \mu_x &= \lim_{\epsilon \to 0} \frac{P\{X(t+\epsilon) = x-1 | X(t) = x\}}{\epsilon}, \end{aligned}$$

with the additional requirement that $\mu_0 = 0$. The structure of these transition rates is illustrated in Figure 6.1.

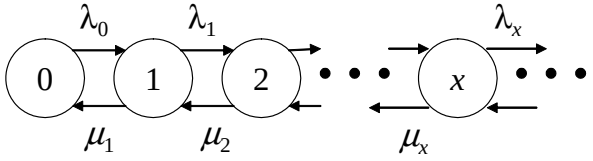


Figure 6.1: *Birth-death chain.*

We assume that the value of *probability transition function* $P_{i,j}(s) \equiv P\{X(t+s) = j | X(t) = i\}$ is independent of t (stationary) and does not depend on the state of the

system prior to being in state i (by the Markov property). Under these assumptions, the holding time in state x before the next event is an exponential random variable with parameter ν_x, defined as follows.

$$\nu_x = \begin{cases} \lambda_0 & x = 0 \\ \lambda_x + \mu_x & x > 0 \end{cases}$$

Let $P_{i,j}$ be the probability that the system when in state i makes its next jump to state j, for all $i, j \in S$. For the birth-death chain, these probabilities are given by

$$\begin{aligned} P_{x,x+1} &= \frac{\lambda_x}{\lambda_x + \mu_x} \text{ for } x \geq 0, \\ P_{x,x-1} &= \frac{\mu_x}{\lambda_x + \mu_x} \text{ for } x > 0, \end{aligned}$$

with all other transition probabilities equal to zero.

In modeling our congestion-sensitive input-output system, we choose the birth rates so as to correspond to the external arrival process and the death rates as to correspond to the workload function. In this manner, birth-death models are advantageous in that our analysis is equally tractable for any of the workload functions described in Chapter 3.

In this context, our objective is to maximize the *death rate* of the system as it evolves over time through the use of an appropriate admission control policy. One approach to this control problem is to again consider the DP formulation of the previous section.

It is well known that every CTMC can be interpreted in terms of an embedded DTMC with exponential holding times. In our model, the expected holding time when in state x is given by $\frac{1}{\lambda+\mu_x}$, and during this period departures occur at rate μ_x. Thus, the expected number of departures between successive jumps of the embedded DTMC is $\mu_x(\frac{1}{\lambda+\mu_x})$, which is equivalent to the probability that the next event is a departure. Let $t \in T$ index the set of times at which jumps of the embedded DTMC occur. If we consider a DP in which each stages corresponds to a jump of the embedded DTMC,

we obtain the following recursive relationship for the value function.

$$\begin{aligned} V_t(x) &= \mu_x \left(\frac{1}{\lambda+\mu_x}\right) + \left(\frac{\mu_x}{\lambda+\mu_x}\right) V_{t+1}\left((x-1)^+\right) \\ &\qquad + \left(\frac{\lambda}{\lambda+\mu_x}\right) \max\left[V_{t+1}(x), V_{t+1}(x+1)\right] \\ &= \left(\frac{\mu_x}{\lambda+\mu_x}\right)\left(1 + V_{t+1}\left((x-1)^+\right)\right) \\ &\quad + \left(\frac{\lambda}{\lambda+\mu_x}\right) \max\left[V_{t+1}(x), V_{t+1}(x+1)\right] \end{aligned}$$

For the infinite horizon problem we know that the optimal admission control is of the threshold type. Again, for the case of the finite horizon problem it may be possible to exploit the specific structure of this objective function to show by induction that quasiconcavity in μ_x implies quasiconcavity for all functions $V_t(x)$. However, there is another simpler approach for showing this result based on the steady state behavior of the birth-death process.

6.2 Steady State Distributions for B-D Models

Of primary interest is the long-run behavior of the birth-death process. Define π_j to be the *limiting probability for state* j, defined as

$$\pi_j = \lim_{t\to\infty} P_{ij}(t).$$

In other words, π_j represents the long-run likelihood that the system will be found in state j. An important issue for all CTMCs is knowing under what conditions the limiting probabilities exist. For a general CTMC, a limiting probability distribution exists when the Markov chain is irreducible and recurrent. In the case of a birth-death process this condition is equivalent to the requirement that $\lambda_n, \mu_n > 0$ for all n and furthermore that the these birth and death rates are such that

$$\sum_{n=1}^{\infty} \frac{\lambda_0\lambda_1\ldots\lambda_{n-1}}{\mu_1\mu_2\ldots\mu_n} < \infty.$$

It is well known that this last requirement is sufficient for the existence of the limiting probabilities. For an accessible proof of this condition, see [104].

It should be noted that these birth-death rates for a congestion-sensitive system do not satify the sufficiency condition for the existence of limiting probabilities. To see this, define the term ρ_n as $\rho_n = \frac{\lambda_0 \lambda_1 \ldots \lambda_{n-1}}{\mu_1 \mu_2 \ldots \mu_n}$, then sufficiency condition is equivalent to $\sum_{n=1}^{\infty} \rho_n < \infty$. Recall that the quasiconcavity of μ_n implies that there exists some n^* such that when $n > n^*$ then μ_n is nonincreasing in n. Suppose there exists a point $N > n^*$ such that $\mu_N < \lambda$. Since $\lambda_x = \lambda$ is fixed, then ρ_n is nondecreasing for $N > n^*$. Separating the infinite series into parts

$$\sum_{n=1}^{\infty} \rho_n = \sum_{n=1}^{N} \rho_n + \sum_{n=N+1}^{\infty} \rho_n,$$

it is clear that the first series is positive and finite, whereas the second series diverges. Thus, we have proved the following proposition.

Proposition 3. *A birth-death process with constant birth rate λ and quasiconcave death rates μ_n is stable if and only if there exists $N < \infty$ such that $\mu_n > \lambda$ for $n > N$.*

In the case where death rates are quasiconcave in n and $\mu_n \to \Omega < \lambda$ as $n \to \infty$ this condition clearly does not hold. Again, this shows that in the absence of control the congestion-sensitive processing system is unstable. In these cases where limiting probabilities do not exist, we pursue to types of analysis. First, we use the probabilistic structure of the birth and death rates to quantify the transient behavior of the system. Second, we consider modifications to the probabilistic structure of the problem in order to induce the desired stable behavior. We consider each in turn.

6.2.1 Transient Behavior and Hitting Times

In the trivial case when $\mu_n \to 0$ as $n \to M < \infty$, then the collapse point M is a trapping state of the Markov chain. This is equivalent to saying that the CTMC is nonrecurrent. While the limiting probabilities do not exist, it is possible for us to characterize the mean *hitting time* of the system to state M. In this case, we have a chain that looks like the following.

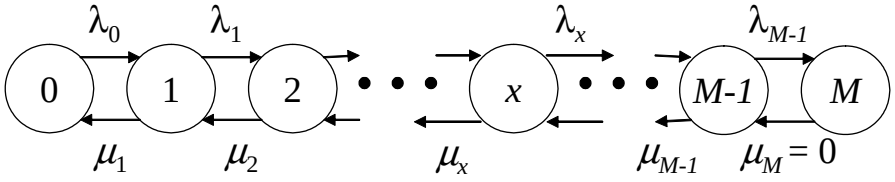


Figure 6.2: *Birth-death chain with trapping state at M.*

The following analysis is elementary and of the style found in [104], however we repeat it for completeness and for use in additional analysis that is to follow. Define y_i as the mean hitting time to state M, given that the system is currently in state i. Observe the following general relationships.

$$\begin{aligned} y_0 &= \frac{1}{\lambda_0} + y_1, \\ y_i &= \frac{1}{\lambda_i + \mu_i} + \frac{\lambda_i}{\lambda_i + \mu_i}\, y_{i+1} + \frac{\mu_i}{\lambda_i + \mu_i}\, y_{i-1}, \text{ for } 0 < i < M, \\ y_M &= 0. \end{aligned}$$

Rearranging terms we obtain

$$\begin{aligned} y_0 - y_1 &= \frac{1}{\lambda_0}, \\ y_i - y_{i+1} &= \frac{1}{\lambda_i} + \frac{\mu_i}{\lambda_i}\, (y_{i-1} - y_i), \text{ for } 0 < i < M. \end{aligned}$$

Letting $z_i = y_i - y_{i+1}$, we have

$$\begin{aligned} z_0 &= \frac{1}{\lambda_0}, \\ z_i &= \frac{1}{\lambda_i} + \frac{\mu_i}{\lambda_i}\, z_{i-1}, \text{ for } 0 < i < M. \end{aligned}$$

We can therefore solve the recursive relationship

$$\begin{aligned} z_0 &= \frac{1}{\lambda_0}, \\ z_1 &= \frac{1}{\lambda_1} + \frac{\mu_1}{\lambda_1}\, z_0, \\ z_2 &= \frac{1}{\lambda_2} + \frac{\mu_2}{\lambda_2}\, z_1 \\ &= \frac{1}{\lambda_2} + \frac{\mu_2}{\lambda_2\lambda_1} + \frac{\mu_2\mu_1}{\lambda_2\lambda_1\lambda_0}, \\ &\vdots \\ z_{M-1} &= \sum_{i=1}^{M-1} \frac{1}{\lambda_i} \prod_{j=i+1}^{M-1} \frac{\mu_i}{\lambda_i} + \left(\prod_{j=1}^{M-1} \frac{\mu_j}{\lambda_j} \right) \frac{1}{\lambda_0}. \end{aligned}$$

But we also know that $z_{M-1} = y_{M-1} - y_M = y_{M-1}$, since $y_M = 0$. So we have

$$y_{M-1} = \sum_{i=1}^{M-1} \frac{1}{\lambda_i} \prod_{j=i+1}^{M-1} \frac{\mu_i}{\lambda_i} + \left(\prod_{j=1}^{M-1} \frac{\mu_j}{\lambda_j} \right) \frac{1}{\lambda_0}$$

and we can solve for each y_i using the backwards recursion $y_i = z_i + y_{i+1}$. While perhaps tedious, these particular computations are easily obtained by the use of a spreadsheet or similar numerical method.

Consider as an example the specific case of a birth-death system having constant birth rate and death rates in accordance with the piecewise linear workload function.

$$\begin{aligned} \lambda_x &= \lambda && x \geq 0 \\ \mu_x &= \begin{cases} \frac{c}{a}x & 0 \leq x \leq a \\ c - \frac{c}{b}(x-a) & a < x \leq a+b \\ 0 & x > a+b \end{cases} \end{aligned}$$

Note that this model is consistent with our previous model for the congestion-sensitive input-output system under constant arrivals and with piecewise linear workload function w_3. Assume for this example that $\lambda = 3$ and these death rates are characterized by parameters $a = 15$, $b = 10$, and $c = 6$. The numerical results for the mean hitting times for the states of this system are shown below in Figure 6.3.

The shape of the curve of mean hitting times has clear interpretation. As long as the system remains to the "left" of the unstable equilibrium point, the mean hitting

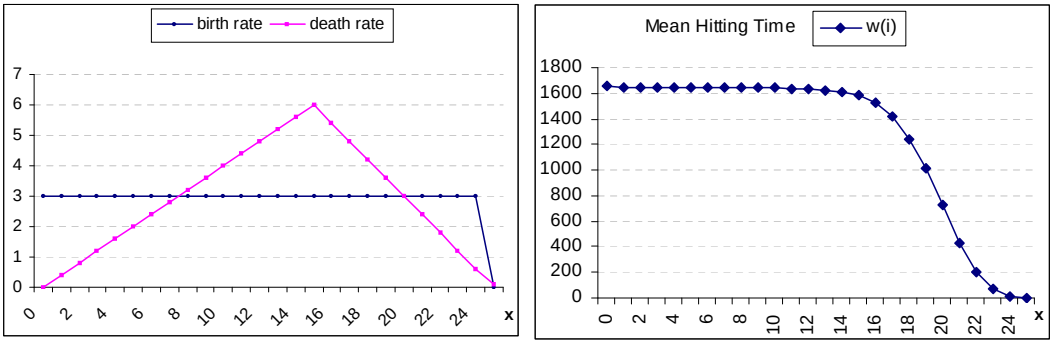


Figure 6.3: *Example of birth-death chain and corresponding mean hitting times.*

time is large and relatively insensitive to the actual state. However, as soon as the system moves to the "right" of the unstable equilibrium point the system is expected to move rapidly toward the trapping state, and the corresponding mean hitting time drops quickly.

Also of interest is the effect of changes in the birth rate λ to the mean hitting times. Previously, we commented that increases to the arrival rate resulted in a loss of robustness. Figure 6.4 shows the numerical results for the associated mean hitting times when $\lambda = 4, 5, 6, 7$.

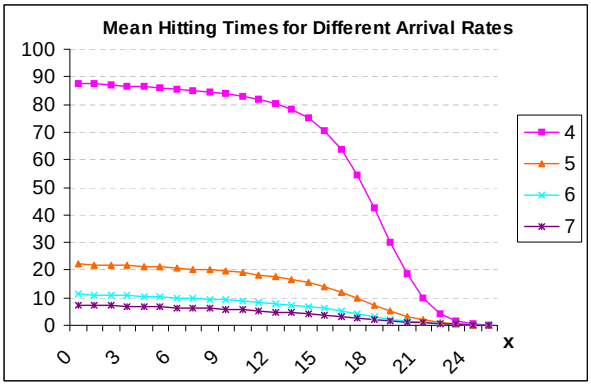


Figure 6.4: *Sensitivity of mean hitting time to increases in birth rate.*

A comparison of these results supports our previous assertion. Let $y_0(\lambda)$ represent the mean hitting time from state 0 for a given λ. From Figure 6.3, we have $y_0(3) = 1651$. And from Figure 6.4, we observe that $y_0(4) = 88$, $y_0(5) = 22.2$, $y_0(6) = 11.2$, and

$y_0(7) = 7.4$. In this manner, it is clear that as the arrival rate approaches the system maximum, the increase in system fragility is severe.

6.2.2 Achieving System Stability in Birth-Death Models

It should be clear from this analysis that the nonexistence of limiting probabilities equates to system instability. That is, any birth-death system with constant birth rate and quasiconcave (congestion-sensitive) death rates will reach the collapse state in finite time. Our question of interest is therefore: *How should we modify the structure of the birth and death rates so that the sufficiency condition for limiting probabilities is satisfied?*

As before, we will consider admission control as the primary means by which we influence system behavior. Also, since we have restricted our system to stationary transition probabilities, we will similarly consider only stationary admission policies. In the context of our birth-death system, this implies for each state n we will define an *admission probability* a_n. That is, births in our system will now occur at rate $a_n\lambda$. Our condition for system stability requires that

$$\sum_{n=1}^{\infty} \lambda^n \frac{a_0 a_1 \dots a_{n-1}}{\mu_1 \mu_2 \dots \mu_n} < \infty,$$

or that the terms $a_{n-1} < \mu_n/\lambda$ as $n \to \infty$.

Previously, we showed that a nonrandomized threshold policy of the form

$$a_n = \begin{cases} 1 & 0 \le n < N \\ 0 & n \ge N \end{cases}$$

achieves the optimal system throughput. This type of policy restricts our Markov chain to the finite state space $S = \{0, 1, 2, \dots, N\}$ and so the sufficiency condition for limiting probabilities is clearly satisfied.

6.3 Birth-Death Chains with Reflection

The use of a nonrandomized threshold policy to prevent the system from nearing the collapse state is equivalent to creating a *reflecting barrier* for the system. As noted

above, we create the reflecting barrier by modifying the birth rates so that arrivals are blocked when the population size is N, for some value $N < M$. Assuming that the system has a piecewise linear workload function, we have the following modified birth and death rates.

$$\lambda_x = \begin{cases} \lambda & 0 \leq x < N \\ 0 & x \geq N \end{cases} \qquad \mu_x = \begin{cases} \frac{c}{a}x & 0 \leq x \leq a \\ c - \frac{c}{b}(x-a) & a < x \leq a+b = M \\ 0 & x > M \end{cases}$$

The modified chain is illustrated in Figure 6.5. In the presence of the reflecting

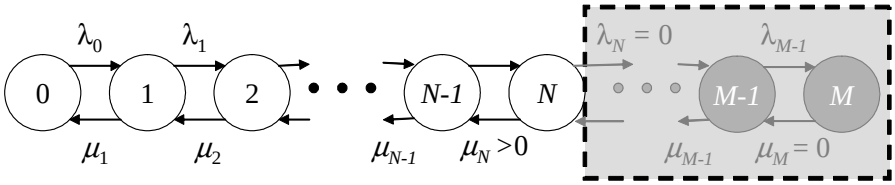


Figure 6.5: *Birth-death chain with reflecting barrier.*

barrier, it is relatively easy to compute the stationary probabilities.

$$\begin{aligned} \pi_n &= \frac{\lambda_{n-1}\lambda_{n-2}\ldots\lambda_1\lambda_0}{\mu_n\mu_{n-1}\ldots\mu_2\mu_1}\pi_0 = \rho_n\pi_0 \\ \pi_0 &= \frac{1}{1+\sum_{n=1}^{N}\frac{\lambda_0\lambda_1\ldots\lambda_{n-1}}{\mu_1\mu_2\ldots\mu_n}} = \frac{1}{1+\sum_{n=1}^{N}\rho_n} \end{aligned}$$

Again, it should be clear why we require that $\sum_{n=1}^{N}\rho_n < \infty$.

Once more consider our previous example of a birth-death chain with birth rate $\lambda = 3$ and piecewise linear death rates having parameters $a = 15$, $b = 10$, and $c = 6$. From our previous analysis, we know that there is a stable equilibrium at $x_1^* = \frac{a}{c}\,\lambda = \frac{15}{6}\,3 = 7.5$ and an unstable equilibrium at $x_2^* = a+b-\frac{b}{c}\,\lambda = 15+10-\frac{10}{6}\,3 = 20$. The steady-state probabilities for the system will depend on whether the reflecting barrier N lies to the left or right of the unstable equilibrium point. We consider successive choices for the reflecting barrier $N = \{24, 23, 22, 20\}$. The results of a numerical solution to this example are illustrated in Figure 6.6 below.

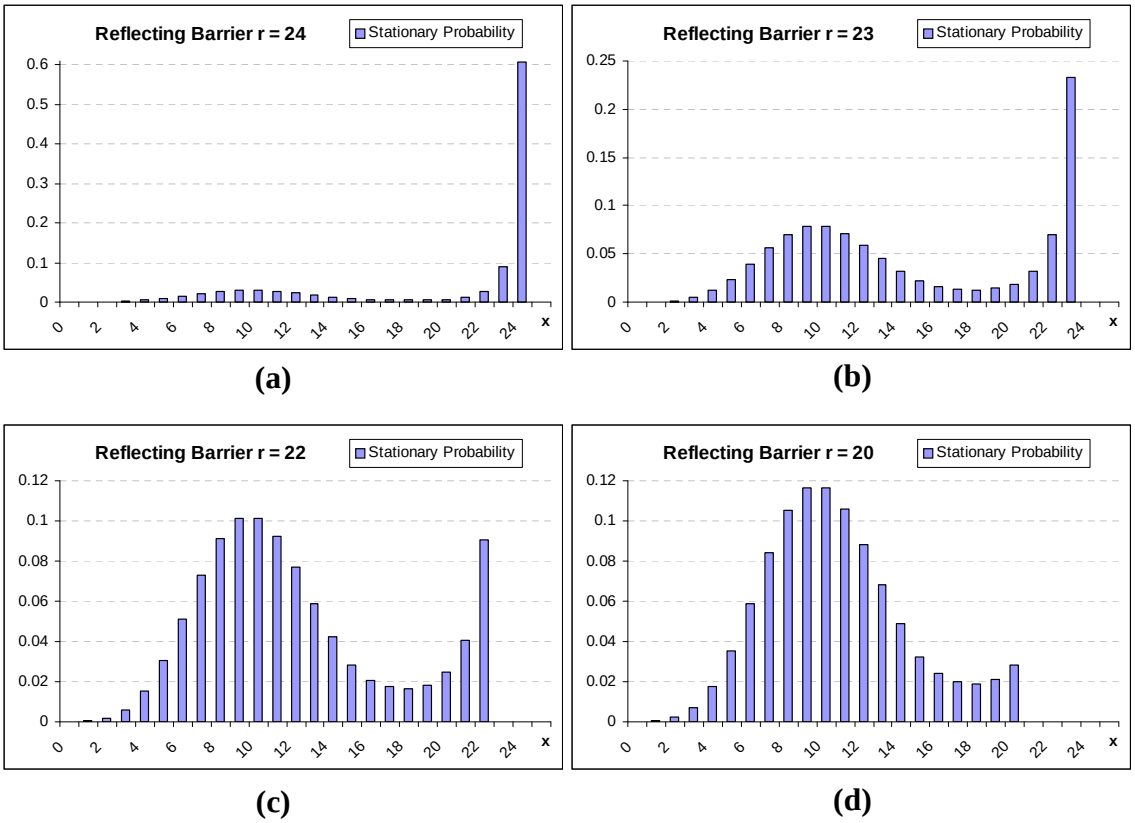


Figure 6.6: *Birth-death chain with reflection at* r. Four values for the relecting barrier: (a) $r = 24$, (b) $r = 23$, (c) $r = 22$, and (d) $r = 20$. As the reflecting barrier is lowered to coincide with the unstable equilibrium point $x_2^* = 20$, the mass of the system's stationary distribution shifts Note the change in scale on the vertical axis for each graph.

6.3.1 Optimal Reflecting Barrier for Birth-Death Chains

Given the ability to admit or block arrivals for any particular state i, what can we say about (1) the *existence* of an optimal reflecting barrier, and (2) the *location* of an optimal reflecting barrier? To answer these questions, recall that the limiting probability distribution for a birth-death system is characterized by its birth rates, death rates, and the location of the reflecting barrier N. Consider the average death

rate of the system with reflecting barrier at N, given by

$$J(N) = \sum_{i=0}^{N} \mu_i \ \pi^N(i)$$

where π^N is the steady-state distribution over a state space $\{0, 1, 2, \ldots, N\}$. Clearly $J(0) = 0$, so we assert that $J(1) > J(0)$. We therefore ask the question *What happens to the average death rate if the upper barrier N is increased to $N + 1$?* When the reflecting barrier is increased to $N + 1$, the system admits a new set of steady state probabilities, denoted $\pi^{N+1}(i)$ to reflect that this system lives on the expanded state space $\{0, 1, 2, \ldots, N + 1\}$. The change to the average death rate is given as follows.

$$\begin{aligned} J(N+1) - J(N) &= \sum_{i=0}^{N+1} \mu_i \ \pi^{N+1}(i) - \sum_{i=0}^{N} \mu_i \ \pi^N(i) \\ &= \sum_{i=0}^{N} \mu_i \left[\pi^{N+1}(i) - \pi^N(i)\right] + \mu_{N+1} \ \pi^{N+1}(N+1) \end{aligned}$$

Observe that for any state $i \in \{0, 1, 2, \ldots, N\}$ the change in value to the associated steady-state probability is as follows.

$$\begin{aligned} \pi^{N+1}(i) - \pi^N(i) &= \rho_i \left(\frac{1}{1 + \sum_{n=1}^{N+1} \rho_n} \right) - \rho_i \left(\frac{1}{1 + \sum_{n=1}^{N} \rho_n} \right) \\ &= \rho_i \left(\frac{\left(1 + \sum_{n=1}^{N} \rho_n\right) - \left(1 + \sum_{n=1}^{N+1} \rho_n\right)}{\left(1 + \sum_{n=1}^{N+1} \rho_n\right)\left(1 + \sum_{n=1}^{N} \rho_n\right)} \right) \\ &= \frac{-\rho_i \ \rho_{N+1}}{\left(1 + \sum_{n=1}^{N+1} \rho_n\right)\left(1 + \sum_{n=1}^{N} \rho_n\right)} \\ &= -\pi^{N+1}(N+1) \ \pi^N(i) \end{aligned}$$

This yields

$$\begin{aligned} J(N+1) - J(N) &= \sum_{i=0}^{N} \mu_i \left[-\pi^{N+1}(N+1) \ \pi^N(i)\right] + \mu_{N+1} \ \pi^{N+1}(N+1) \\ &= \pi^{N+1}(N+1) \left(\mu_{N+1} - \sum_{i=0}^{N} \mu_i \pi^N(i) \right). \end{aligned}$$

Consider the expression within the parentheses. The first term μ_{N+1} is the death rate when the system is in state $N+1$. The second term $\sum_{i=0}^{N} \mu_i \pi^N(i)$ is the average output rate of the system over the initial state space $\{0, 1, 2, \ldots, N\}$. This implies that the net change to the average system death rate can increase only when the output rate μ_{N+1} is greater than the average death rate obtained over the previous states $\{0, 1, 2, \ldots, N\}$.

Another interpretation of this result is that the net change to average output rate can be understood in terms of the stationary probability mass that is moved away from the states $\{0, 1, 2, \ldots, N\}$ and redistributed to state $N+1$. The amount of the probability mass that is moved is exactly $\pi^{N+1}(N+1)$, and it is taken from the states $\{0, 1, 2, \ldots, N\}$ in the same proportion as the previous distribution π^N.

Recall that the death rates μ_n are quasiconcave in n. That is, there exists a point n^* such that μ_n is nondecreasing on the interval $0 \leq n \leq n^*$ and μ_n is nonincreasing on the interval $n^* \leq n$. Since μ_n is nondecreasing on $0 \leq n \leq n^*$, it is clear that the average death rate of the system can will never worsen by increasing N over this interval. However, since μ_n is nonincreasing on the interval $n^* \leq n$, there will be a finite point N^* above which the average death rate cannot be improved. Thus, we have again shown that a threshold policy is always optimal for the birth-death chain.

In this manner, we have shown that in order to maximize the death rate of the the birth-death system described here, one should use a threshold policy in the form of a reflecting barrier N satisfying $\mu^* < N < M$. In fact, the barrier satisfies the following proposition.

Proposition 4. *An optimal barrier N^* exists and satisfies $x_2^* \leq N^* < M$, where x_2^* is the unstable equilibrium point and M is the collapse point.*

Proof. We have already shown that $N^* < M$, so we turn to proving that $N^* \geq x_2^*$. Let $\bar{\lambda}$ and $\bar{\mu}$ represent the respective average birth and death rates for the system. For a system with reflecting barrier at N, let $B(N)$ denote the probability that an incoming arrival is blocked. Here, $B(N) = 1 - \pi^N(N)$. So $\bar{\lambda} = \lambda\ B(N) = \lambda\ (1 - \pi^N(N)) < \lambda$. When the reflecting barrier is at N, we also have $\bar{\mu} = \sum_{i=0}^{N} \mu_i \pi^N(i)$. In equilibrium, we know that $\bar{\mu} = \bar{\lambda} < \lambda$.

From our previous analysis, we have shown that increasing the reflecting barrier from N to $N+1$ results in an increase to the average death rate as long as $\mu_{N+1} > \bar{\mu}$. Recall that for a quasiconcave death rates μ_x, we defined the equilibrium points x_1^* and x_2^* such that (1) $\mu_x \leq \lambda$ for $0 \leq x < x_1^*$, (2) $\mu_x \geq \lambda$ for $x_1^* \leq x \leq x_2^*$, and (3) $\mu_x \leq \lambda$ for $x > x_2^*$. So if state $N+1 < x_2^*$, $\mu_{N+1} \geq \lambda > \bar{\mu}$, and increasing the reflecting barrier to $N+1$ will always increase the average death rate for the system. Thus, the reflecting barrier will always be greater than the unstable equilibrium point x_2^*, and the proof is complete. ■

6.4 Birth-Death Chains with Reset

While the attention in this thesis is primarily on the use of admission control to influence the behavior of a congestion-sensitive input-output system, there are a host of other types of control methods that may be of relevant to applications of interest. For example, consider instead the case when a congested system does not recover gracefully even in the absence of additional arrivals. In this instance, the best course of action for a system operator may be to *reset* the system to an empty state before admitting new arrivals. Usually when an input-output system is reset to an empty state all the work in the system at the time of reset is lost forever. A reset operation may be expensive or even infeasible for some types of systems (e.g. in transportation systems, it is infeasible to flush cars from a highway or trains from a segment of track), while it may be cheap, convenient, or even routine in other arenas (e.g. in computer networks, an overloaded server or router is often simply rebooted). Although this type of control is somewhat outside the primary scope of this work, systems with reset are easily investigated using a variation of the models discussed thus far, so we take a moment to examine them.

Systems with reset can be modeled using a variation of the birth-death formulation in the following manner. Instead of specifying a reflecting barrier for the birth-death process, we select a state N to serve as the *reset state*. We do this by setting the death rate $\mu_N = 0$ and adding an additional *reset rate* ω. That is, ω is the rate at which the system jumps from from state N back to state 0. All other transition rates

remain as before. This results in the following changes to the transition probabilities of the embedded DTMC.

$$\begin{aligned} P_{0,1} &= 1 \\ P_{x,x+1} &= \frac{\lambda_x}{\lambda_x + \mu_x} \text{ for } 0 < x < N \\ P_{x,x-1} &= \frac{\mu_x}{\lambda_x + \mu_x} \text{ for } 0 < x < N \\ P_{N,0} &= 1 \end{aligned}$$

Again, all other transition probabilities are equal to zero. The resulting chain is illustrated in Figure 6.7.

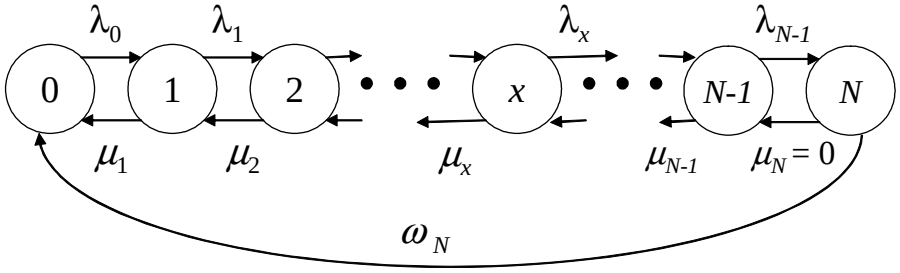


Figure 6.7: *Birth-death chain with reset.*

It should be noted that this system is no longer a birth-death chain, but its structure is similar enough that we can employ a similar technique for computing the limiting probabilities. In this case, the limiting probabilities are known to satisfy the

following relationships, given by the Chapman-Komolgorov equations.

$$\begin{aligned}
\pi_0(\lambda_0) &= \pi_1\mu_1 + \pi_N\omega \\
\pi_1(\lambda_1+\mu_1) &= \pi_0\lambda_0 + \pi_2\mu_2 \\
\pi_2(\lambda_2+\mu_2) &= \pi_1\lambda_1 + \pi_3\mu_3 \\
&\vdots \\
\pi_i(\lambda_i+\mu_i) &= \pi_{i-1}\lambda_{i-1} + \pi_{i+1}\mu_{i+1} \\
&\vdots \\
\pi_{N-2}(\lambda_{N-2}+\mu_{N-2}) &= \pi_{N-3}\lambda_{N-3} + \pi_{N-1}\mu_{N-1} \\
\pi_{N-1}(\lambda_{N-1}+\mu_{N-1}) &= \pi_{N-2}\lambda_{N-2} \\
\pi_N(\omega) &= \pi_{N-1}\lambda_{N-1}
\end{aligned}$$

Here, we define ρ_i such that $\pi_i = \rho_i\pi_{N-1}$ for all $i = 0, 1, 2, \ldots N$. Then,

$$\begin{aligned}
\rho_i &= \frac{1}{\lambda_i}\left[(\lambda_{i+1}+\mu_{i+1})\rho_{i+1} - \mu_{i+2}\rho_{i+2}\right] \text{ for } i = 0, 1, 2, \ldots N-3, \\
\rho_{N-2} &= \frac{1}{\lambda_{N-2}}\left[(\lambda_{N-1}+\mu_{N-1})\rho_{i+1}\right], \\
\rho_{N-1} &= 1, \\
\rho_N &= \frac{\lambda_{N-1}}{\omega}.
\end{aligned}$$

And since $\sum_{i=0}^{N}\pi_i = 1$, we have

$$\pi_{N-1}\sum_{i=0}^{N}\rho_i = 1,$$

which yields

$$\pi_{N-1} = \frac{1}{\sum_{i=0}^{N}\rho_i}.$$

Consider again the previous example with $N = 25$, constant birth rate $\lambda = 3$, and death rate parameters $a = 15$, $b = 10$, and $c = 6$. Here, we can observe the impact of the reset rate ω on the stationary distribution. Numerical results for this case are presented in Figure 6.9. Given that the system has entered state N, the amount of

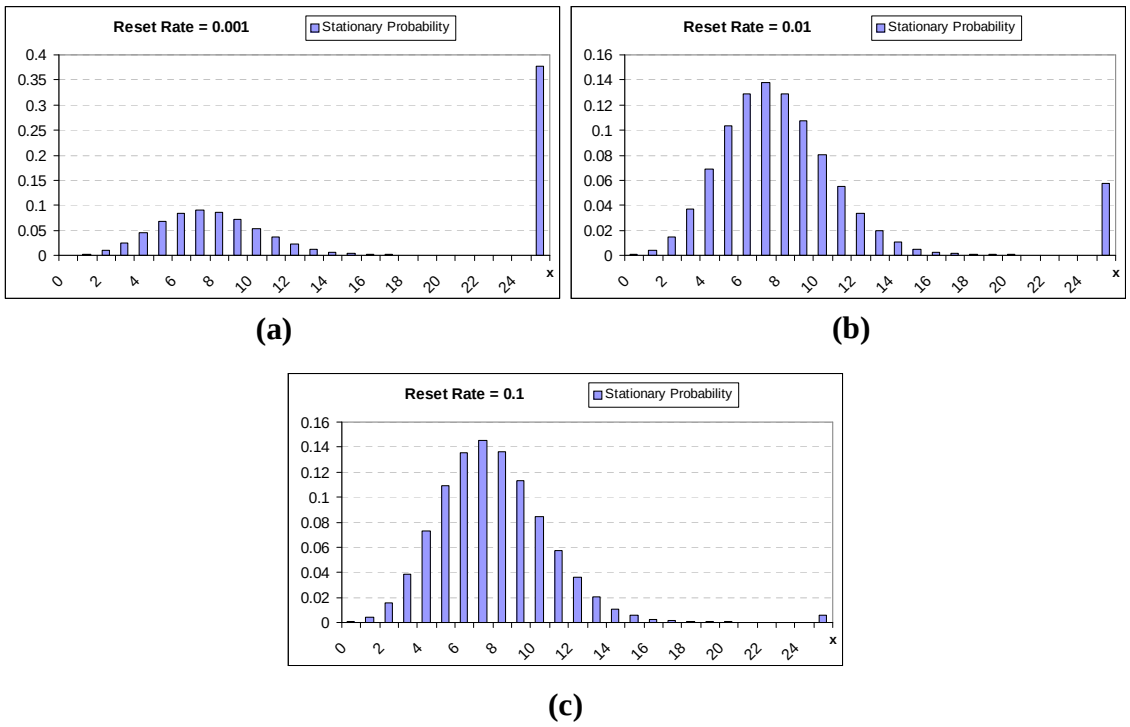


Figure 6.8: *Stationary distribution of birth-death chain with reset for various reset rates.* Three values for the relecting barrier: (a) $\omega = 0.001$, (b) $\omega = 0.01$, and (c) $\omega = 0.1$. Note the change in scale on the vertical axis for each graph.

time for the system to reset itself is the *holding time in state* N. Since holding times in a CTMC are exponential random variables, we know that the expected reset time of the process is equal to $1/\omega$.

We can classify the behavior of the system at all times as being in one of two *phases*. Either the system is *operating normally* or the system is in the process of *resetting itself* after a collapse. Because of the system's congestion-sensitive nature, we know that any phase in which the system is operating normally will be of finite length. That is, whenever the system is operating normally, there is only a finite amount of time remaining until the system reaches the reset state. Similarly, the expected reset time of the system is also finite. Denote a phase of normal operation

followed by a phase of resetting as a single *operating cycle* of the system. Since the process is a CTMC, we know that the process renegerates itself probabilistically after each operating cycle.

Because the system is nonproductive (that is, there is no output) during each resetting phase, an important issue is the relative proportion of time that the system spends in state N. Define $\pi_i(\omega)$ to be the stationary probability of being in state i for a particular value of the reset rate ω. In the above example, $\pi_{25}(\omega)$ is the stationary probability of being in the failed state, and the corresponding values for $\pi_i(\omega)$ are $\pi_{25}(0.1) = 0.006$, $\pi_{25}(0.01) = 0.057$, and $\pi_{25}(0.001) = 0.377$. Again, as the reset rate increases, the absolute and relative amount of time spent in the reset rate decreases.

Define T_ω to be the expected reset time, and recall that y_0 be the expected hitting time to state N given that the system starts in state 0. We know intutitively that $\pi_N(\omega)$ should be equal to

$$\pi_N(\omega) = \frac{T_\omega}{T_\omega + y_0}.$$

In other words, $\pi_i(\omega)$ is the expected proportion of time the system spends in the reset state relative to the total expected time for a single operating cycle. Similarly, we can define Z (need better notation) as the the *expected hitting cost* of the system, and the expected system output rate should then be

$$\frac{Z}{T_\omega + y_0}.$$

Since $T_\omega = 1/\omega$, it is always the case that a faster reset rate results in higher average system output rate. Another approach to measuring the average system output rate is to compute directly the expected output rate of the system, given by

$$\sum_{i=0}^{N-1} \mu_i \ \pi(i).$$

Using these metrics we can investigate the optimal reset point for the system.

6.4.1 Obtaining the Optimal Reset Point

Suppose we are given a system that needs to be reset whenever it enters a collapse state N. An important operational question is whether or not it is possible to improve the behavior of the system by proactively resetting the system *before* it reaches the collapse state N.

Just as with the case of the reflecting barrier, we are interested in selecting the location of the optimal reset point that maximizes the average system output rate

$$J(N) = \sum_{i=0}^{N-1} \mu_i \ \pi(i).$$

Note that in this case, the system output rate is zero when in the reset state N, so time spent in this state does not increase the average system output rate. If we assume that the reset rate ω is fixed but that the system operator can choose the state N from which the system is always reset, what value of N maximizes the average output rate?

One approach to answering this question is again to consider what happens when we increase the location of the reset point from N to $N + 1$. As before, we are interested in the following relationship.

$$J(N+1) - J(N) = \sum_{i=0}^{N-1} \mu_i \left[\pi^{N+1}(i) - \pi^N(i)\right] + \mu_{N+1}\pi^{N+1}(N)$$

Unfortunately, this line of analysis is not as tractable analytically as in the case of a reflecting barrier. However, we can evaluate it numerically without difficulty. Implementing the computations from (equation above) in a spreadsheet for the same numerical example, we obtain the following picture of the functional relationship for the average death rate $J(N)$.

From these numerical results, we make the following observations.

1. The optimal reset point generally appears to be greater than the unstable equilibrium point. This result is consistent with the proof for the location of the optimal reflecting barrier. It also makes intuitive sense, in that we only want to reset the system when we are fairly sure that the system is about to collapse.

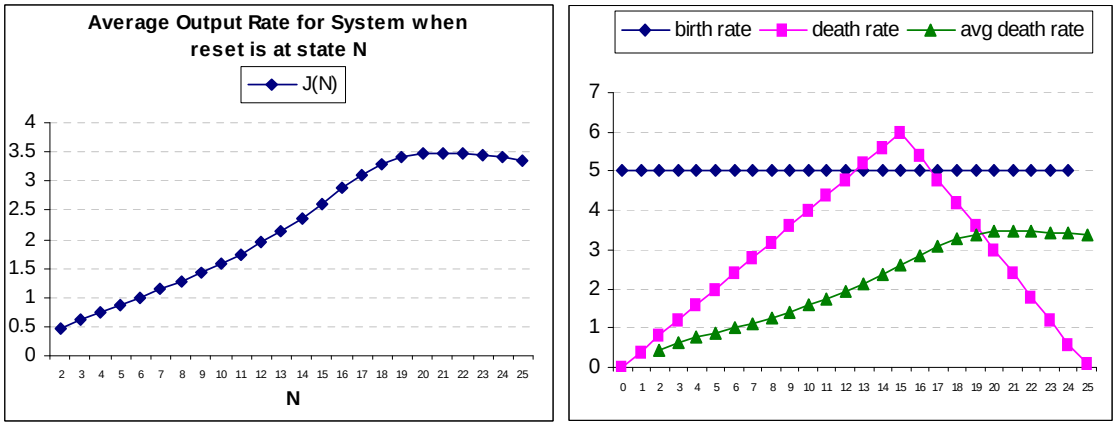


Figure 6.9: *Average System Output for Various System Reset Points.* On the left, average system output rate $J(N)$ is plotted for each reset value. On the right, average death rate for each reset point is plotted along with birth and death rates.

2. The location of the reset point seems to be independent of the reset rate ω.

While additional analysis in this direction is interesting, it is beyond the current scope of this thesis.

6.5 Chapter Summary

Using the framework of birth-death processes, we have extended our stochastic analysis of congestion-sensitive systems. We have reinforced our understanding for the unstable behavior of the system in the absence of admission control, and we have quantified this behavior in terms of hitting times to the collapse point. We have demonstrated again that the threshold policy for admission control is optimal and shown that the optimal reflecting barrier will be located at a value not less than the unstable equilibrium point. In addition, we examined a different kind of system control in which the system operator has the ability to reset the system to an initial state. Again, we quanitified the behavior of the system under this type of control policy and speculated on the location for the optimal reset point.

Chapter 7

Network Systems

While the behavior of stand alone, congestion-sensitive processors is interesting, our desire is to understand the large scale network behavior of interacting processors. In this chapter, we take the first steps toward this ultimate objective. We provide a framework in which we consider the dependence between interacting input-output systems. We identify several basic network forms, and we describe how their congestion-sensitive nature necessarily makes them susceptible to failure cascades. We describe how even these simple networks provide a realistic description of some important infrastructure systems. Furthermore, we leverage previous work on queueing networks in order to take the first steps in the development of control policies that are effective in mitigating the risk of cascading failures while still providing good network performance.

7.1 Parallel Systems

Perhaps the simplest configuration of interacting congestion-sensitive processors occurs when they operate in *parallel*. In this case, the network load is shared among the processors and the primary challenge is *load balancing* among each processor.

7.1.1 A Parallel Processing Model

Consider a system of J processors arranged in parallel. Let x_j be the amount of work in processor j and $w_j(x)$ be its corresponding workload function. Here, we will extend our notation for continuous time dynamical systems. Assume that arrivals occur at rate $A(t)$ and let $D_j(t)$ represent the departure rate of work from processor j. In this model we do not allow for a system operator to reject arrivals. Instead, the job of the input controller is to assign each unit of arriving work to one of the processors. In this manner, the input controller functions as a *load balancer* among the J processors. We assume that each assignment is permanent in the sense that work cannot be rebalanced at a later time. The dynamics of this system are illustrated in Figure 7.1 below.

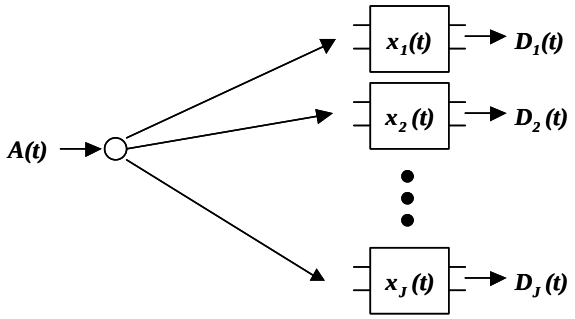


Figure 7.1: *Load balancing in parallel system.*

Let $u_j(t)$ represent the fractional amount of arrivals that are routed to processor j at time t. Of course, we require $\sum_{j=1}^{J} u_j(t) = 1$, for all $t \in [0, \infty)$. In the case where the allocation of work among processors is based on a fixed proportion (e.g. $u_j(t) = u_j$), then the behavior of each processor can be analyzed independently in the manner of the previous chapters. However, we know from our previous stochastic analysis that, in the absence of admission control, each standalone processor will experience congestion collapse in finite time.

As an operator of this system, we would like to leverage the following facts in the development of an optimal control policy. First, individual processors might not be identical. For example, the processors may differ in their maximum processing capacity, and this fact should be considered in the allocation of work to each processor.

Second and more importantly, it is likely that the loading among processors will not be uniform over time. It therefore seems reasonable that we should be able to achieve better system performance by using a *dynamic allocation scheme*. Of course, the form of this type of policy is likely to depend on the specific application of interest.

Load balancing problems such as this have been studied in the queueing literature and have found application in a number of computer problems. In the context of queueing, Winston [132] was the first to show that for a system with Poisson arrivals and several identical, exponential (congestion-*insensitive*) servers arranged in parallel, the *shortest line discipline* stochastically dominates other policies in maximizing the time-discounted number of customers served during any interval. In other words, the system maximizes its throughput in a stochastic sense if each arriving customer always joins the shortest queue. The intuitive appeal of this policy is obvious. Weber [127] extended this result to the case of a general arrival process and a general service distribution with a non-decreasing hazard rate. Ephremides, Variaya and Walrand [46] considered a simple system of two identical exponential servers. They showed that if the queue lengths can be observed by incoming arrivals, then the optimal policy is to join the shortest queue. However in the case where the queue lengths are not observed, then provided that the initial lengths of the queues was the same, they showed that the optimal policy is to alternate between the two queues. In another variation of the problem, Bell and Stidham [16] considered the case where arriving customers must choose among the servers using information about the service time distributions for each processor but without knowledge of existing congestion levels. They showed that the arrival patterns based on individual self-interest differ systematically from the socially optimal allocations chosen by an input controller.

In the context of computer applications, the above formulation is an appropriate model for the problem of assigning incoming World Wide Web (WWW) traffic to machines in a web server farm. Over the last several years, there has been tremendous interest in the characterization of web server behavior of in the presence of traffic overload [8] and on the development of methods for mitigating possibility of server collapse in these systems [59, 62, 33, 3]. A particularly important strategy is one in which server load is intelligently balanced among the individual machines [37, 35].

Traditional analysis in the context of computer load balancing has focused on the *relative* performance of a control policy rather than on its absolute performance [13, 129, 12, 4]. For example, one is often interested in the relative performance of one policy over another for a large ensemble of possible arrival streams, or a relative comparison of a given policy to the optimal policy for a known set of arrivals. In constructing the appropriate policy, there is a general tradeoff between finding the optimal allocation and an easily implemented algorithm, such as the Greedy algorithm [131].

7.1.2 Cascading Failures in Parallel Systems

During normal operation of a parallel system with congestion-sensitive components, individual processors will occasionally fail. An initial failure may be caused by congestion collapse within that processor or as the result of some exogenous shock. In either case, the important issue is how the system of processors responds in the presence of this failure. As long as the failed processor remains unavailable, the system will need to balance all future arrivals among the remaining processors. If the rebalancing is done poorly or perhaps too slowly, it is possible that another processsor can experience congestion collapse. If this process repeats with more and more processors failing in succession, we say that the system has suffered a cascading failure.

In order to investigate the potential for cascading failures in parallel processing systems, we are interested in the following questions.

1. What types of load balancing policies make sense for normal operation, and what is the behavior of a this congestion-sensitive parallel system under this policy?
2. Under what conditions will a single processor fail? How frequent are such failures likely to be?
3. Under what conditions will the system collapse from a cascading failure? Alternatively, what conditions will ensure that the system does not collapse?
4. What forms of control need to be applied to prevent collapse?

5. What should be done to optimize the throughput performance of such a system?

In a manner consistent with our previous analysis of stand alone processors, we intend to leverage known methods from the literature on stochastic control.

7.2 Tandem Systems

Another simple type of processing network is one in which the input-output systems are connected in *tandem*—that is, the output from one processor becomes the input to another. These systems have diverse applications, including transportation, manufacturing, and communication systems. Unfortunately, the dynamics of these systems are more complicated than the dynamics of parallel systems, as the following example illustrates.

7.2.1 2-Node Tandem System Model

Consider the simple case where two congestion-sensitive input-output systems are connected in tandem without intermediate buffering. That is, the output of the first processor is the input to the second processor. Let $x_1(t)$ denote the amount of the work in the first system at time t, with quantites $A_1(t)$, $D_1(t)$, and $w_1(x)$ denoting the respective arrival process, departure process, and workload function. We define the analogous quantities $x_2(t)$, $A_2(t)$, $D_2(t)$, and $w_2(x)$ for the second processor. Then, each system evolves according to its own ODE

$$\begin{aligned} \frac{dx_1}{dt} &= A_1(t) - D_1(t), \\ \frac{dx_2}{dt} &= A_2(t) - D_2(t). \end{aligned}$$

However, since they are connected in tandem $A_2(t) = D_1(t)$. Substituting the appropriate workload functions, we obtain the following.

$$\begin{aligned} \frac{dx_1}{dt} &= A_1(t) - w_1(x_1(t)), \\ \frac{dx_2}{dt} &= w_1(x_1(t)) - w_2(x_2(t)). \end{aligned}$$

Assume we allow for admission control to the first processor only. Let $u_1(t) \in [0,1]$ denote the fractional amount of input allowed at time t. If we are interested in maximizing the throughput of the system as before, we have the following optimal control problem.

$$\begin{aligned} \max_{u_1(t)} \quad & \int_{t=0}^{T} w_2(x_2(t))dt \\ s.t. \quad & \dot{x_1}(t) = u_1(t) - w_1(x_1(t)) \\ & \dot{x_2}(t) = w_1(x_1(t)) - w_2(x_2(t)) \\ & 0 \leq u_1(t) \leq A_1(t), t \in [0,T] \\ & x_1(0), x_2(0) \text{ given} \end{aligned}$$

The interacting aspects of these processors is illustrated in Figure 7.2.

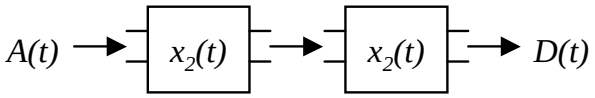


Figure 7.2: *Basic 2-node tandem system.*

As with our investigation of the single processor system, we are interested in the following questions.

1. What is the behavior of a this congestion-sensitive tandem system under arbitrary known input?
2. Under what conditions will the first processor fail? The second? How frequent are such failures likely to be?
3. Under what conditions will the system collapse from a cascading failure? Or, what conditions will ensure that the system does not collapse?
4. What forms of control need to be applied to prevent collapse?
5. What should be done to optimize the performance of such a system?

While a thorough treatment of these questions is beyond the scope of this current work, there is a tremendous amount of insight that is already available from our previous analysis. We use this opportunity to develop this insight in order that it might serve as a foundation for ongoing work.

7.2.2 Qualitative Analysis

We can begin to get insight into the behavior of a congestion-sensitive tandem system from the same type of qualitative analysis used previously. Imagine that we have two congestion-sensitive systems that operate deterministically but are connected in tandem in the manner described above. The system dynamics under constant input can be understood in terms of a *phase plot* for the two node system. In the case where the first processor is stable under constant arrival rate λ, we know that the output rate of the first processor (and therefore the input rate of the second processor) is also λ. We know that the first processor will have equilibrium points given by x_1^1 and x_1^2. Similarly, the second processor will have equilibrium points given by x_2^1 and x_2^2. Figure 7.3 illustrates the intersection of these equilibria on the $x_1 - x_2$ plane.

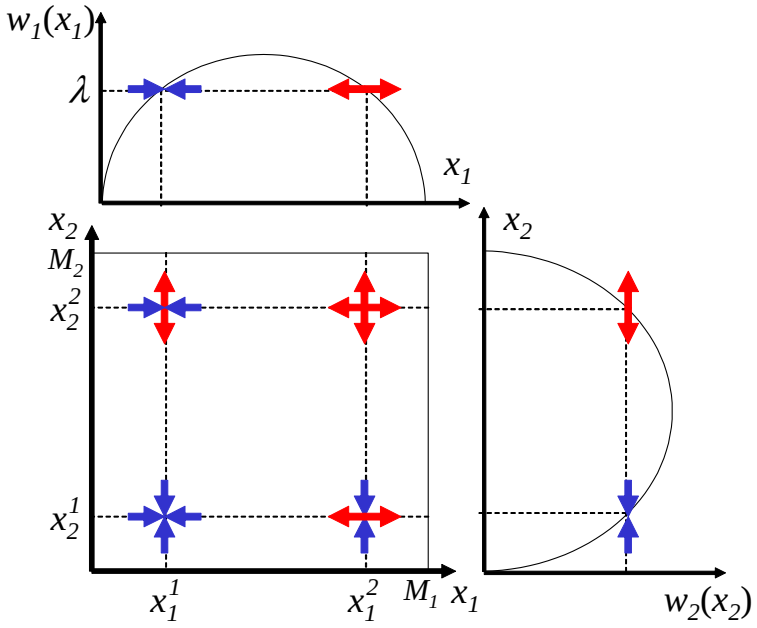


Figure 7.3: *Phase plot for 2-node tandem system under constant input λ.*

Using the same qualitative analysis as before, it is clear that

- (x_1^1, x_2^1) is a stable equilibrium for the 2-node system,
- (x_1^2, x_2^1), (x_1^1, x_2^2), and (x_1^2, x_2^2) are unstable equilibrium points.

As with the standalone processor, this system is clearly unstable in the absence of control. The development of an effective input control mechanism will require a thorough understanding of the system dynamics.

7.2.3 Birth-Death Formulation

It is relatively straightforward to consider an extension of the simple one-dimensional birth death model. Let $(n_1, n_2) \in \{0, 1, 2, \dots\} \times \{0, 1, 2, \dots\}$ be the population of at processors 1 and 2 respectively. The population for each processor is allowed unit increments and decrements, with the restriction that a departure from processor 1 means an arrival to processor 2. Assume that births occur at processor 1 with Poisson rate λ. Assume that deaths at processor 1 occur with rate $\mu_1(n_1)$ and that deaths from processor 2 occur with rate $\mu_2(n_2)$. In summary, we have a CTMC with the following possible transitions.

$$\begin{array}{lll} (\mathrm{n}_1, n_2) \to (n_1 + 1, n_2) & \text{with rate } \lambda & \mathrm{n}_1, n_2 \geq 0 \\ (\mathrm{n}_1, n_2) \to (n_1 - 1, n_2 + 1) & \text{with rate } \mu_1(n_1) & \mathrm{n}_1 \geq 1, n_2 \geq 0 \\ (\mathrm{n}_1, n_2) \to (n_1, n_2 - 1) & \text{with rate } \mu_2(n_2) & \mathrm{n}_1 \geq 0, n_2 \geq 1 \end{array}$$

When the death rates μ_1 and μ_2 correspond to congestion-sensitive workload functions, this type of system is going to be unstable in a manner consistent with the one-dimensional birth-death system and illustrated by our qualitative analysis above.

As before, we consider input control as a primary means by which we can establish system stability and manage system performance. Consider the case where we implement a *threshold* admission policy at *each* of the processors. Let N_1, N_2 be the thresholds imposed on the two systems. The system then evolves on the simplex (N_1, N_2) and has the following possible transitions.

$$\begin{array}{lll} (\mathrm{n}_1, n_2) \to (n_1 + 1, n_2) & \text{with rate } \lambda & 0 \leq n_1 < N_1, 0 \leq n_2 \leq N_2 \\ (\mathrm{n}_1, n_2) \to (n_1 - 1, n_2 + 1) & \text{with rate } \mu_1(n_1) & 1 \leq n_1 \leq N_1, 0 \leq n_2 < N_2 \\ (\mathrm{n}_1, n_2) \to (n_1 - 1, n_2) & \text{with rate } \mu_1(n_1) & 1 \leq n_1 \leq N_1, n_2 = N_2 \\ (\mathrm{n}_1, n_2) \to (n_1, n_2 - 1) & \text{with rate } \mu_2(n_2) & 0 \leq n_1 \leq N_1, 1 \leq n_2 N_2 \end{array}$$

In this case, system arrivals that occur when $n_1 = N_1$ are blocked and lost to the system forever. Similarly, a departure from processor 1 that occurs when $n_2 = N_2$ is blocked from entering the second processor and is lost to the system. The steady-state distributions for this Markov chain are easily obtained. As an example, consider the simple case where each node has an identical piecewise linear workload function. Figure 7.4 plots the associated steady-state probabilities. Observe that this distribution is consistent with the qualitative analysis of Figure 7.3.

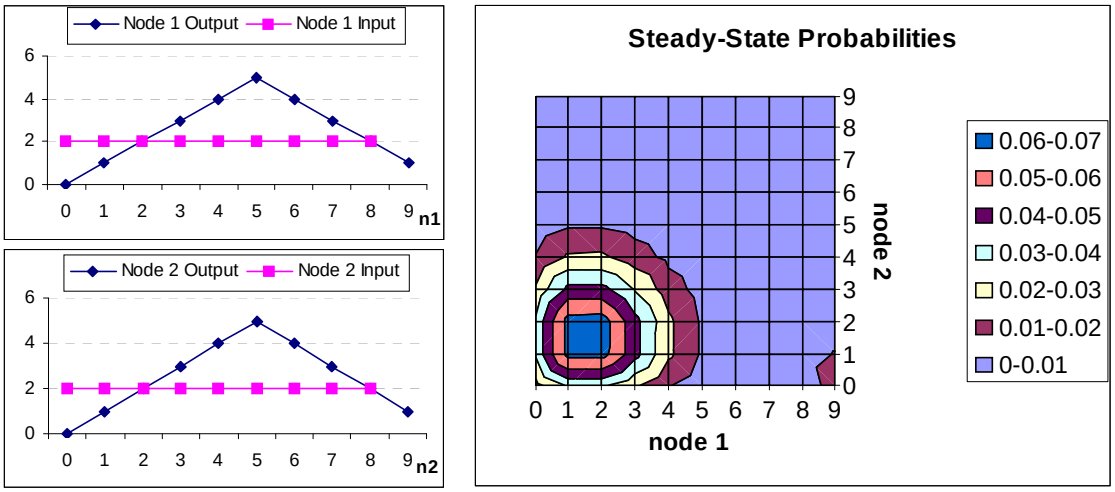


Figure 7.4: *Steady-state distributions for Markov chain model representing two queues in tandem.*

While this type of control does yield a stable system, it is somewhat heavy in its restrictions. A more appealing approach would impose admission control at only the first processor. However, it is unclear whether this simple admission policy would provide adequate control for the system. Although the formal development of optimal control mechanisms is beyond the current scope of this thesis, we leverage previous results from the vast literature on stochastic networks to build intuition for future work.

7.2.4 General Results in Stochastic Networks

The analysis of queueing networks has focused primarily on models which yield a *product-form* steady-state distribution. That is, the joint steady-state distribution for the system as a whole can be computed as the product of the steady-state distributions for the individual, stand alone network components. Systems with product-form distributions allow for easy computation of performance metrics.

The development of product-form results started with the early work of Burke [27, 28] who showed for a queueing system with Poisson arrivals and exponential service that the output process is also Poisson and independent of the state of the queue. Jackson [60, 61] extended this work to the network context with his development of joint probability distributions for queueing models representing production systems. Kelly [68] generalized the class of product form results using ideas from reversed processes. Recent treatments of queueing networks and product form solutions are available from Walrand [123], Gelenbe and Pujolle [52], and Chao, Miyazawa, and Pinedo [32].

Although the above results apply to general network forms, they have been used extensively in the analysis of tandem networks. Reich [93, 94] showed that for a tandem system of two or more exponential queues in equilibrium the customer sojourn times at each queue are independent. Weber [128] extended this result to show that the ordering of the queues does not affect the stationary output of the system. In other words, the queues are *interchangeable* in the stochastic sense. This result also hold for the case when the service times of all queues are deterministic. Earlier work by Tembe and Wolff [118] had shown that a tandem system comprised of a mix of queues with deterministic and exponential service distributions is not interchangeable. When this is the case, they provided methods for obtaining the optimal ordering of such queues. Walrand and Variaya [124] generalized the results of Reich to show that the sojourn times of customers in Jackson networks with non-overtaking paths are mutually independent. Lehtonen [71] uses sample path arguments to obtain this result for two queues in tandem and also shows that the overall departure process becomes stochastically faster as the servers become more homogeneous. Tsoucas and Walrand [121] extend this result to the general case of N queues in tandem. That is,

if the exponential service rate of each queue is given by $\mu_1, \mu_2, \ldots, \mu_N$, then subject to the constraint that $\mu_1 + \mu_2 + \cdots + \mu_N = K$ the departure process for the system is maximized when $\mu_i = K/N$ for $i = 1, 2, \ldots, N$.

The above results remain valid provided that there are no limits on the number of customers that can be at each queue—that is, as long as there is no blocking at one station by the next. Morse [80] was probably the first to consider the behavior of tandem queues without intermediate buffering. In this framework, a customer exiting service at one station can proceed to the next station only if there is available capacity at the next station. In the case where the next station is full, the customer remains with the server at the current station and blocks all subsequent customers in that queue. Avi-Itzhak & Yadin [11] considered the case of a tandem system in which there is no intermediate buffering between sequential queues. In their model, arrivals to the first queue are Poisson and there is infinite buffering at the first station. They showed for both the case of exponential and deterministic service times that the densities for customer sojourn times and number-in-system do not depend on the order of stations. Furthermore, they showed that interference between stations incresases with the difference in the exponential service parameters, and also that interference does not exist for deterministic service times. Hildebrand [57] presented methods for evaluating the capacity of tandem systems with finite intermediate buffering. Pinedo and Wolff [91] considered a case where service times at each station are the same for a given customer but vary from customer to customer. They examined both cases in which blocking between tandem station is and is not allowed. They showed that the performance of the tandem system generally improves as the arrival process becomes more regular and as the service distribution becomes less regular.

In a departure from the restrictions imposed by a need for exact analysis of queueing systems, Newell [83] considered the case of non-Poisson and non-stationary queueing systems using deterministic and diffusion approximations. His model specifically addresses the case of finite storage at each tandem queue. In general, he shows that tandem systems can be most easily understood by identifying the single bottleneck of the system, and in doing so the analysis is greatly simplified. Accordingly, he comments that analysis of systems with identical servers are the most difficult to an-

alyze in this manner. While he develops analytic approximations to many important queueing networks, he does not address the issues associated with optimal control of these systems.

The issue of optimal control for tandem queueing systems has also received attention. Rosberg, Varaiya and Walrand [101] considered the case of two queues in tandem where service rate control at the first queue is used to optimize expected discounted cost of the system as a whole. In their model, each queue has a buffer of infinite size but incurs a per unit holding cost for its inventory. The service rate $u \in [0, a]$ for the first queue is chosen according to the joint state (x_1, x_2) of the system. They showed that the optimal control is of the form $u = 0$ or $u = a$ according to a *switchover curve* S. In particular, the optimal policy chooses $u = a$ when $x_1 \geq S(x_2)$ and chooses $u = 0$ when $x_1 < S(x_2)$. Ghoneim and Stidham [53] consider an alternate model for the control of two queues in tandem. In their setup, each queue receives its own stream of external arrivals and can accept or reject each arrival from the outside. However, departures from the first queue always join the second queue, and there is no blocking or rejecting these customers from the second queue. For the finite horizon problem, they show that the optimal admission policies for each queue takes the form of a rejection region of the state space (x_1, x_2). This rejection region is essentially the two-dimensional analog of the threshold policy in use for a single queue.

7.2.5 A Need for Something More

While a review of this literature yields insight into the types analyses and control policies that are apt to be useful for an investigation of congestion-sensitive processors in tandem, the above results do not directly apply. The congestion-sensitive nature of the processor and its resulting instability mean that the basic results of Burke [27, 28, 29], Jackson [60, 61], and Kelly [68] will not hold. Nonetheless, it is reasonable to believe that the optimal control policy will correspond to some form of a switching curve for the tandem case, much in the same way that the optimal control was a threshold policy for the single processor case. The development of this

optimal admission control policy remains part of ongoing work.

7.2.6 Cascading Failures in Tandem Systems

In a tandem system of congestion-sensitive processors, it is possible that any of the processors along the chain can experience congestion collapse. This possibility results from fluctations in the work levels that flow through the system as well as differences in the workload functions of each processor. What happens when a processor fails? If a downstream processor fails, it is possible to have cascading effects upstream, particularly if individual processors are not allowed to implement their own input control. Similarly, when an upstream processor fails downstream nodes suffer from *starvation*. In general, the optimal control policy will carefully balance these tensions so as to maximize overall system performance.

7.2.7 Applications to Transportation Systems

Tandem systems provide a simple, yet reasonable model for the dynamics of certain transportation systems. For example, consider a segment of one-way traffic on a major highway. Vehicles arrive to the entrance point of the segment and then, after some period of time, exit the other end of the segment. The amount of time that it takes a vehicle in the system depends the segment length, the speed of the vehicle, and *the number of other vehicles in that segment*.

Transportation engineers have characterized this behavior in terms of the relationships between: number of items in the system or *density* (denoted n), the average processing rate or *velocity* (denoted v), and the vehicle output rate or *throughput* (denoted μ). Of course, these quantities depend also on the vehicle input rate, and the standard approach is to assume that the average velocity is inversely proportional to the density. For example, consider a simple case in which velocity decreases linearly with traffic density. That is, $v(n) = K(1 - \frac{n}{N})$, where K is a constant and N is the maximum possible density. Then, system throughput can be simply understood as the product of density and velocity. In other words, $\mu(n) = n\ v(n) = K\ n\ (1 - \frac{n}{N})$. Note that in this simple case, system throughput is quadratic in n, and it is easily

seen that when $K = 1$ throughput is maximized at $n^* = N/2$ with value $\mu^* = N/4$. The relationship for throughput and velocity can similarly be obtained. Figure 7.5 illustrates these relationships. They can be found in most introductory texts on

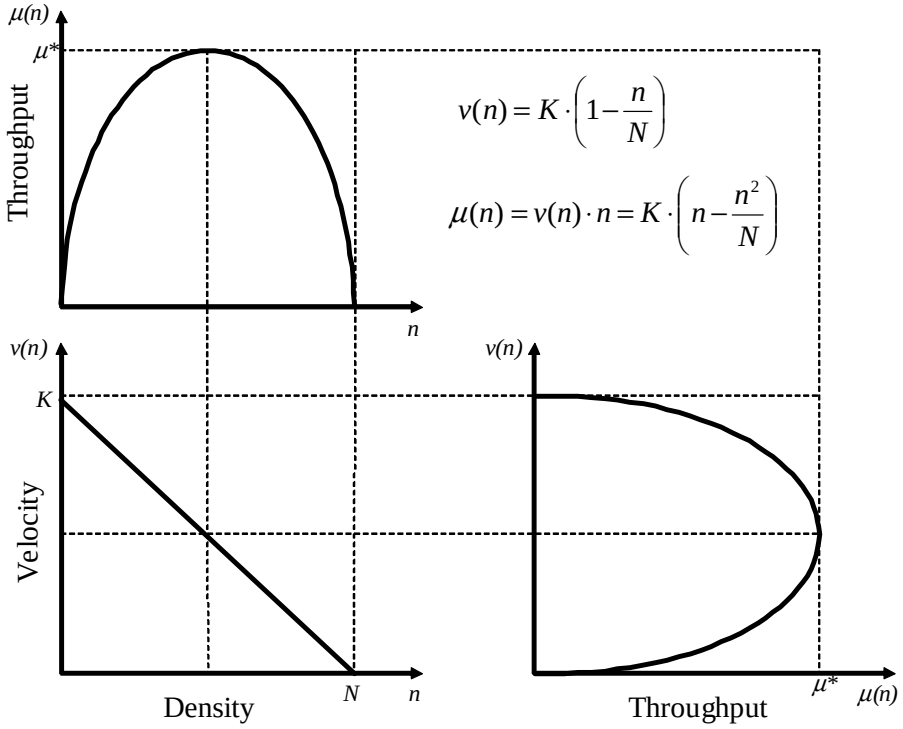


Figure 7.5: *Relationship between density, velocity, and throughput in transportation systems.*

transportation engineering [78].

It is important to recognize that the throughput-density relationship defined here corresponds exactly to the workload function that we have used throughout this thesis. In particular, the throughput relationship $\mu(n) = n\ v(n) = K\ n\ (1 - \frac{n}{N})$ is the famous Greenshields traffic model, and it correspond to our second workload function $w_2(x) = \{x\,(1 - (x/M)^p)$ when $p = 1$. In this manner, the simple tandem system can be directly applied to problems in highway traffic.

In a manner similar to highways, railroad systems can be examined in the context of tandem networks. For example, consider the *Sunset Route* that is owned and operated by the Union Pacific Railroad (UPRR). The Sunset Route is a stretch of

primarily single-track railroad that runs from Los Angeles to El Paso. Trains travel in both directions along this single track railroad, using track sidings to allow trains to pass one another in either direction. Because track sidings may be spaced at large distances, the speed (and therefore throughput) that is attainable by trains along the Sunset Route depends on the density of trains and the spacing of the sidings.

The general relationship between congestion-sensitivity and cascading failures in railroads has been understood since the time of the Union Pacific service crisis [64]. An important next step in the understanding of large-scale congestion behavior in railroads will come from the development of appropriate tandem network models. For example, the Sunset Route is conveniently modeled by a tandem system with two *classes* of traffic (one for each direction). Each segment of track is represented by a separate input-output system. For example, assuming that there are J such segments, then for any segment j one has *eastbound* traffic, denoted x_j^e, and *westbound* traffic, denoted x_j^w. Analogous system equations follow directly. It should be noted that this relatively simple model yields complex behavior, and that understanding how to manage this type of network is of primary importance to Union Pacific. A case study of the management issues for type of system is currently ongoing.

7.3 Chapter Summary

Our ultimate goal is to understand the collective behavior of congestion-sensitive processors that are connected in arbitrary ways. That is, we wish to consider network topologies that include a mixture tandem components, parallel components, and feedback. By studying these canonical network structures in isolation, we take the first steps in developing the necessary theoretical foundation for a broader theory of network dynamics. At the same time, these simple network forms are reasonable models for several important applications ranging from computer and communication networks to transportation systems.

Chapter 8

Where Do We Go From Here?

The intention of this thesis has been to develop an insight and understanding of congestion-induced failures that is sufficient for addressing the large-scale vulnerability issues of our national infrastructure systems. Of particular interest is the development of improved management guidelines for the complex networks of today and the identification of design principles for the robust networks of tomorrow.

8.1 Contributions of This Thesis

Although there remains significant work ahead in achieving this ambitious goal, this dissertation has already made several important contributions.

1. ***Articulating an agenda of significant scope and setting out in a direction to address key questions about the large-scale vulnerability of our infrastructure systems.***

Our national infrastructures are a complex network of networks. Their efficiencies have encouraged a great dependence on them, and it is this dependence that makes us vulnerable to accidents, failures, and attacks. While this problem has been recognized at the level of the federal government for nearly a decade, relatively little progress has been made to date in understanding the causes and behavior of large-scale failure

events. This thesis leverages an interdisciplinary perspective to develop an analytical framework for the study of congestion-sensitive network flow behavior.

2. *Isolating the canonical model for a network element that is simple but not simplistic.*

In this thesis, we have focused on systems consisting of processing elements that are sensitive to congestion. We have shown how these elements can be characterized in term of a workload function and have shown the existence of a broad class of such functions that are useful representing the realistic behavior of many important systems. We have seen that even when in isolation, these elements have distinct nonlinear behavior and are inhererntly unstable.

3. *Analyzing the model from various angles to develop a deep understanding of its associated performance.*

In our analysis of the congestion-sensitive network element, we have leveraged several different theoretical perspectives. We have looked at deterministic and stochastic system behavior, and we have investigated control strategies over finite and infinite time horizons. When the system objective is maximal throughput, we have interpreted the dilemma of input control as a tradeoff between congestion and starvation. We have leveraged popular stochastic birth-death models and diffusion models to reinforce this perspective and identify optimal control strategies.

4. *Extending the model by incorporating connections among several elements in order to understand large-scale network behavior.*

We have considered the simplest forms of feed-forward networks, namely those of tandem and parallel processing systems. We have demonstrated how the congestion-sensitive nature of individual elements naturally yields the potential for cascading failures in each of these network types. We have commented on the potential for emergent behavior in these types of systems and identified important features of desired performance, most notably the need for graceful degradation in the presence of overwhelming system load. We have speculated on the form of optimal control

for these basic systems, and we have initiated the research for an all encompassing general network framework.

5. ***Identifying significant applications of immediate importance, and setting in motion the process of studying them in order to produce management recommendations.***

The development of our modeling framework has been with an eye to address critical applications in computer, telecommuncation, and transportation systems. We have shown how even simple tandem network models are a realistic representation of transportations systems, such as the Union Pacific Railroad. Railroad systems in particular have been vulnerable to congestion-induced cascading failures, and insights into the management of these vulnerabilities are of immediate importance. Similarly, we have illustrated how simple parallel configuations of network elements can reasonably represent computer processing facilities, such as web server farms. The development of policies for optimal load balancing in these parallel environments and the evaluation of their stability and performance is of ongoing importance to information infrastructure providers.

6. ***Leveraging our management insights for complex networks with an understanding of engineering design to accelerate the design and development of robust networks for the future.***

We have initiated a study of issues for the *synthesis* of robust network systems. In particular, we recognize that there will be an ongoing tension between efficiency and robustness for our infrastructures. Through the study of our simple model, we have demonstrated how choices in this tradeoff can result in vulnerabilities to cascading failures. As these infrastructure systems continue to permeate the fabric of our daily lives, it will be essential that the architects of these systems possess a deep understanding of the large-scale implications of their design decisions.

8.2 Ongoing Research

While the conclusion of this thesis marks a completion of the first phase of this research, there are several important ongoing research threads that already constitute the next phase in this agenda.

- *A thorough treatment of the finite horizon problem.* In particular, we would like to develop an understanding of the optimal horizon admission policy as a function of both the current load and the time remaining. While long-lived or ongoing decision problems may be analyzed over an infinite horizon, the corresponding decision policies are often implemented over a finite horizon. Management decisions for infrastructure systems are made and then updated for finite rolling time horizons, based on the limited information that is available at each decision point. For this reason, it will be important to complete the aforementioned analysis for value of information, including the horizon method and blended arrival stream approach.

- The extension of the existing frameworks for the treatment of *general network structures*. The canonical models of the tandem and parallel processing networks extend naturally into a general network model for congestion-sensitive processing systems. We aim to understand the behavior and control issues associated with this broad network model in order to enable the treatment of a greater diversity of infrastructure applications. Of particular interest in this analysis will be understanding the differences between these network models and the results of traditional product-form queueing networks.

- The development of *diffusion-approximation models* to lend additional insight into the stochastic behavior of congestion-sensitive systems. Systems that evolve in continuous time and continuous space are often represented using *diffusion processes* [67, 54]. They are a natural extension of the birth-death processes that were examined in Chapter 6. Because they can be viewed as approximations to many other types of stochastic processes, diffusion processes are an important class of stochastic models. Diffusion processes have a rich mathematical

framework that allows for the easy calculation of several performance metrics of interest and enables greatly facilitated sensitivity analysis. Diffusion process models can also be extended to the types of network settings mentioned above.

- Application of our network framework to *a detailed study of the congestion-sensitivity challenges facing the Union Pacific railroad.* While executives at the UPRR have learned much from the 1997-1998 service crisis, their approach to managing the tradeoff between efficiency and robustness in their railroad operations remains heuristic. Because their business will continue to be subject to disruptions from weather, equipment failures, and labor issues, they have an immediate and ongoing need for a deeper quantitative and theoretical understanding of this issue. In order to apply the framework and theory developed in this thesis, we will require a specific understanding of how the low-level dynamics of particular train movement leads to large-scale congestion behavior. Of particular importance will be leveraging the aforementioned value of information results to understand how to they should schedule and manage a mixture of known shipments and last-minute shipments. We hope to identify appropriate incentive structures for demand shaping of customer arrivals, and look to deisng a management framework for integrating these incentives with overall operation of network flows. The ultimate goal for the design of a management system will be to integrate the theoretical insights for system operation with real-time network data.

- Application of the network model to *a detailed study of load balancing behavior in computer server farms.* Again, we require a detailed understanding of how the low level dynamics of server response to increased load can result in congestion collapse. As a first step, our goal is to identify the optimal control strategies for the single-location load balancing problem. An important extension of these results will be to the case of spatially-distributed load balancers, such as used in content distribution networks.

- Application of the network model to *a detailed study of interacting ethernet domains.* With the rise of gigabit ethernet as a major medium for local area

and even campus area networks, we can expect a growing ubiquity, complexity and reliance for interacting ethernet domains. Since shared access protocols like ethernet are known to be subject to the type of congestion-sensitivity discussed in this thesis, it seems reasonable to expect that networks of ethernet domains may be at risk to cascading congestion-collapse. We are investigating this conjecture using both theoretical and experimental approaches.

These projects are interesting and of immediate importance. They are also part of a natural progression to addressing a number of difficult, yet important issues regarding the infrastructure systems of the future.

8.3 Future Challenges

Dreamers of future infrastructure systems envision a world in which we live under the canopy of a network that connects everything and everyone. In this world, information is available instantaneously, goods and services flow without friction, and society proceeds in a manner that is uninhibihted by geographic distance. And of course, in this world "the network" is presumed to be robust and reliable.

While the prospect of this future world provides a compelling reason to continue the current drive to interconnect everything and everyone, we pause to outline some of the big, fundamental questions that remain unanswered.

- *Can we develop early-warning systems to alert network infrastructure operators to pending failures, and can we develop the appropriate management policies to avert catastrophe once it has begun?* For each of the current infrastructure systems, we require a theoretical understanding the patterns of failure and the specific progression of a large-scale failure. In order to achieve this, we require a theoretical model for cascading failures that specifically incorporates control strategies to arrest the cascade before it spreads out of control.

- *How do design and optimization at various levels of network structure contribute to the robustness and fragility of the system as a whole?* What is the role

of evolutionary design in the creation of robust infrastructure systems? Is it possible that there exists something akin to a highly-optimized tolerance (HOT) model for national infrastructure systems?

- *How do the results for congestion-sensitive flow networks extend to other types of transportation systems, including air transportation networks and urban highway systems?* The challenges faced in these industries are as important and perhaps more immediate than for railroad systems. For example, when there is a major snowstorm in chicago, flights throughout the country are disrupted. How should this be managed? Is there a way to develop a flow management scheme that achieves the necessary efficiencies for the airlines while allowing the operating schedule to remain robust to disturbances? Similarly, when there is a major accident at a critical highway or intersection that creates gridlock for major metropolitan areas, is there a routing scheme that could allow for dynamic rebalancing? Understanding the similarities and differences between these systems will be of primary importance to answering these questions.

- *To what extent can we leverage this analysis to understand other types of infrastructures which suffer collapse from starvation instead of congestion?* Examples of such systems include supply-chain systems and networks of financial markets whose robustness relies on the constant availability of commodity flows. In the rare event that such systems experience a shock, how does the industry as a whole react? Consider the following paraphrased examples that have occurred during the last decade. First, an earthquake in Taiwan destroys key silicon manufacturing facilities creating a shortage of memory chips, which creates a cascade of stockouts within a computer appliance industry that has been optimized for just-in-time manufacturing. Second, the collapse of a major, global financial institution sends shockwaves through the world banking system as major banks are exposed to an unanticipated risk. The cascading nature of these types of behaviors is clear. We believe that results from this thesis will be useful in developing appropriate models and insights.

This thesis is a necessary first step for addressing these fundamental questions.

8.4 Final Remarks

We are fortunate to live in a time of great advancement. The speed and scope of technology change has brought incredible efficiencies and conveniences that permeate all aspects of society. As the pace of this evolution accelerates, we expect that technological advances in the years to come will allow for us to interconnect even more people and devices, at faster speeds, and interacting in a more significant way with the physical environment around us. While we marvel in the infrastructures that bring us great capability and luxury, we must remember that our growing dependence makes us vulnerable as a result. In conclusion, we need to make sure that our theoretical understanding of complex networks keeps pace with these technological advances. It is our only hope for predicting what the nature and scope of the behavior of these systems will be.

We look forward to pursuing these interesting and important research questions in the years to come.

Bibliography

[1] Union Pacific Corporation 1998 Annual Report. Available electronically from http://www.unionpacific.com.

[2] *IEEE Communications Magazine, Survivability Issue*, August 1999.

[3] T.F. Abdelzaher and N. Bhatti. Web content adaptation to improve server overload behavior. In *Proc. of 2nd International World Wide Web Conference*, Toronto, Canada, May 1999.

[4] M. Alanyali and B. Hajek. On Simple Algorithms for Dynamic Load Balancing. In *Proc. IEEE Infocom*, pages 230–238, 1995.

[5] D. Alderson, D. D. Elliott, G. Grove, T. Holiday, S. J. Lukasik, and S. E. Goodman. Workshop on Protecting and Assuring Critical National Infrastructure: Next Steps, February 26-27, 1998. Technical report, Center for International Security and Arms Control, Stanford University, 1998.

[6] M. Amin. Toward Self-Healing Energy Infrastructure Systems. *IEEE Computer Applications on Power*, pages 20–28, January 2001.

[7] Massoud Amin. EPRI/DoD Complex Interactive Networks/Systems Initiative: Self-Healing Infrastructures. Keynote presentation at the 2nd DARPA-JFACC Symp. on Advances in Enterprise Control, Minneapolis, July 10-11, 2000.

[8] M.F. Arlitt and C.L. Williamson. Internet web servers: Workload characterization and performance implications. *IEEE/ACM Transactions on Networking*, 5(5), Oct 1997.

[9] C. Asavathiratham. *The Influence Model: A Tractable Representation for the Representation of the Dynamics of Networked Markov Chains*. PhD thesis, Massachusetts Institute of Technology, October 2000. EECS Dept.

[10] C. Asavathiratham, S. Roy, B. Lesieutre, and G. Verghese. The Influence Model. *IEEE Control Systems Magazine*, pages 52–64, December 2001.

[11] B. Avi-Itzhak and M. Yadin. A sequence of two servers with no intermediate queue. *Management Science*, 11(5):553–564, March 1965.

[12] B. Awerbuch and Y. Azar. Local optimization of global objectives: Competitive distributed deadlock resolution and resource allocation. In *Proc. of 35th Ann. IEEE Symp. on Foundations of Computer Science*, pages 240–249, 1994.

[13] Y. Azar. On-line load balancing. In A. Fiat and G. J. Woeginger, editors, *Online Algorithms: The State of the Art*, pages 178–195. Springer, 1998.

[14] P. Bak. *How Nature Works: The Science of Self-Organized Criticality*. Copernicus, New York, 1996.

[15] Molly Ball. Baltimore Firefighters' Job Near End After Wreck. *Washington Post*, 2001. July 23.

[16] Colin E. Bell and Shaler Stidham. Individual versus social optimization in the allocation of customers to alternative servers. *Management Science*, 29(7):831–839, July 1983.

[17] Bertsekas and Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

[18] Dimitri Bertsekas. *Dynamic Programming and Stochastic Control*. Academic Press, New York, 1976.

[19] Dimitri Bertsekas. *Dynamic Programming and Optimal Control*, volume 1 and 2. Athena Scientific, Belmont, Mass., 2001.

[20] Dimitri Bertsekas and Robert Gallagher. *Data Networks*. Prentice Hall, Upper Saddle Ridge, NJ, 2 edition, 1987.

[21] J. Blackburn. Optimal control of a single-server queue with balking and reneging. *Management Science*, 19:297–313, 1972.

[22] David Blackwell. Discounted dynamic programming. *Annals of Mathematical Statistics*, 36(1):226–235, February 1965.

[23] J.P.C. Blanc, Peter R. de Waal, Phillippe Nain, and Donald Towsley. Optimal control of admission to a multiserver queue with two arrival streams. *IEEE Transactions on Automatic Control*, 37(6):785–797, June 1992.

[24] Booz, Allen, and Hamilton Consulting. Economic Impacts of Infrastructure Failures. Report to the President's Commission on Critical Infrastructure Protection, 1997.

[25] J. Borland. MCI faces customer wrath after failures. *CNET News*, 1998. August 11.

[26] J. Borland. ISPs say MCI outage could kill businesses. *CNET News*, 1999. August 13.

[27] P.J. Burke. The output of a queueing system. *Operations Research*, 4:699–704, 1956.

[28] P.J. Burke. The output process of a stationary M/M/s queueing system. *Annals of Mathematical Statistics*, 39(4):1144–1152, August 1968.

[29] P.J. Burke. The dependence of sojourn times in tandem M/M/s queues. *Operations Research*, 17:754–755, 1969.

[30] J.M. Carlson and J.C. Doyle. Highly Optimized Tolerance: Robustness and design in complex systems. *Phys. Rev. Lett.*, 84:2529–2532, 2000.

[31] J.M. Carlson and J.C. Doyle. Complexity and Robustness. *Proc. Nat. Acad. Sci.*, 99, suppl. 1:2538–2545, 2002.

[32] Xiuli Chao, Masakiyo Miyazawa, and Michael Pinedo. *Queueing Networks: Customers, Signals and Product Form Solutions*. John Wiley and Sons, Chichester, England, 1999.

[33] H. Chen and P. Mohapatra. Session-based overload control in qos-aware web servers. In *Proc. IEEE Infocom 2000*, New York, June 2002.

[34] Charles M. Close and Dean K. Frederick. *Modeling and Analysis of Dynamic Systems*. Houghton Mifflin Company, Boston, 2 edition, 1993.

[35] M. Colajanni, P.S. Yu, and V. Cardellini. Dynamic load balancing in geographically distributed heterogeneous web servers. In *Proc. IEEE Int. Conf. on Distributed Computing Systems (ICDCS'98)*, pages 295–302, Amsterdam, The Netherlands, 1998.

[36] Central Maine Power Company. The Great Northeast Blackout of 1965. Available via WWW at http://www.cmpco.com/about/system/blackout.html.

[37] M. Conti, E. Gregori, and F. Panzieri. Load distribution among replicated web servers: A qos-based approach. In *Proc. 2nd ACM Workshop on Internet Server Performance (WISP'99)*, Atlanta, GA, 1999. ACM Press.

[38] J. Cope. Network Outage Hits ATT's ATM Users. *Computerworld Magazine*, 2001. February 26.

[39] AT&T Corporation. AT&T announces cause of frame-relay network outage. AT&T News Release, April 1998. April 22.

[40] Stratus Corporation. Stratus Uptime Advisor, 1996. Available electronically from http://www.stratus.com.

[41] Thomas B. Crabill. Optimal control of a service facility with variable exponential service times and constant arrival rate. *Management Science*, 18(9):560–566, May 1972.

[42] Thomas B. Crabill, Donald Gross, and Michael J. Magazine. A classified bibliography of research on optimal design and control of queues. *Operations Research*, 25(2):219–232, 1977.

[43] Dataquest. Perspective: Sept. 30, 1996. Technical report, Dataquest, 1996. As cited by Stratus Corporation in "Stratus Uptime Advisor".

[44] Bharat T. Doshi. Optimal control of the service rate in an M/G/1 queueing system. *Advances in Applied Probability*, 10:682–701, 1978.

[45] S. Engler. 'Domino effect' zapped power in West. *Cable News Network*, 1996. August 11.

[46] A. Ephremides, P. Variaya, and J. Walrand. A simple dynamic routing problem. *IEEE Transactions on Automatic Control*, AC-25(4):690–693, August 1980.

[47] P. Parrillo et. al. Simmulation-based analysis of cascading failure for power networks. Technical Report TR CIT-CDS-98-013 1998, California Institute of Technology, 1998.

[48] J. Filipiak. *Modelling and Control of Dynamic Flows in Communication Networks*. Springer-Verlag, Berlin, 1988.

[49] C. Benjamin Ford and Neil Adler. Train derailment in Baltimore reveals a fragile Net backbone. *Gazette Newspapers*, 2001. Article appeared on July 27, 2001.

[50] Jay W. Forrester. *Principles of Systems*. Wright-Allen Press, Cambridge, Mass., 2 edition, 1968.

[51] P. Fusco. MCI Worldcom Back Online. *ISP News*, 1999. August 16.

[52] E. Gelenbe and G. Pujolle. *Introduction to Queueing Networks*. John Wiley and Sons, Chichester, England, 2 edition, 1998.

[53] Hussein Ghoneim and Shaler Stidham. Control of arrivals to two queues in series. *European Journal of Operational Research*, 21:399–409, 1985.

[54] P. Glynn. Diffusion approximations. In D.P. Heyman and M.J. Sobel, editors, *Handbooks on OR & MS*, volume 2. Elsevier Science, North-Holland, 1990.

[55] M. Granovetter and R. Soong. Threshold Models of Interpersonal Effects in Consumer Demand. *Journal of Economic Behavior and Organization*, 7:83–99, 1986.

[56] Daniel P. Heyman. Optimal operating policies for M/G/1 queueing systems. *Operations Research*, 16:362–382, 1968.

[57] David K. Hildebrand. On the capacity of tandem server, finite queue, service systems. *Operations Research*, 16:72–82, 1968.

[58] Ronald Howard. *Dynamic Programming and Markov Processes*. M.I.T. Press, Cambridge, Mass., 1960.

[59] R. Iyer, V. Tewari, and K. Kant. Overload control mechanisms for web servers. In *Proc. of Workshop on Performance and QoS of Next Generation Networks*, pages 225–244, Nagoya, Japan, November 2000.

[60] James R. Jackson. Networks of waiting lines. *Operations Research*, 5:518–521, 1957.

[61] James R. Jackson. Jobshop-like queueing systems. *Management Science*, 10(1):131–142, October 1963.

[62] H. Jamjoom, J. Reumann, and K.G. Shin. Qguard: Protecting internet servers from overload. Technical report, University of Michigan Department of Computer Science and Engineering, 2000.

[63] Søren Glud Johansen and Shaler Stidham. Control of arrivals to a stochastic input-output system. *Advances in Applied Probability*, 12:972–999, 1980.

[64] Paul M. Julich. Qualitative Analysis of Congestion and the Recovery from Congestion in Rail Networks. Presentation to Union Pacific Railroad, September 1997.

[65] Sandeep Junnarkar. Fire's effects ripple onto the Net. *CNET News.com*, 2001. July 19.

[66] Samuel Karlin and Howard M. Taylor. *A First Course in Stochastic Processes*. Academic Press, New York, 1974.

[67] Samuel Karlin and Howard M. Taylor. *A Second Course in Stochastic Processes*. Academic Press, New York, 1981.

[68] F.P. Kelly. *Reversibility and Stochastic Networks*. F.P. Kelly, 1979.

[69] S. Lall. Architectures for Secture and Robust Distributed Infrastructures: Project Overview. Available electronically via http://element.stanford.edu/ lall/projects/architectures. AFOSR DoD URI, F49620-01-1-0365 (led by Stanford University).

[70] Lyndsey Layton and Don Phillips. Train Sets Tunnel Afire, Shuts Down Baltimore. *Washington Post*, 2001. Article appeared on July 19, 2001.

[71] Tapani Lehtonen. On the ordering of tandem queues with exponential servers. *Journal of Applied Probability*, 23:115–129, 1986.

[72] Laura Lewis. Train Derails Hazardous Chemicals Onboard; Two Cars Carrying Hydrochloric Acid. TheWBALChannel.com (WBAL Channel 11 News, Baltimore), 2001. July 19.

[73] Steven A. Lippman and Shaler Stidham. Individual versus social optimization in exponential congestion systems. *Operations Research*, 25(2):233–247, 1977.

[74] R. G. Little. Controlling Cascading Failure: Understanding the Vulnerabilities of Interconnected Infrastructures. Technical report, National Research Council, 2001.

[75] D. Low. Optimal dynamic pricing policies for an M/M/s queue. *Operations Research*, 22:545–561, 1974.

[76] David G. Luenberger. *Introduction to Dynamic Systems*. John Wiley and Sons, New York, 1979.

[77] Bruce D. Malamud, Gleb Morein, and Donald L. Turcotte. Forest Fires: An Example of Self-Organized Critical Behavior. *Science*, 281:1840–1842, September 18 1998.

[78] William R. McShane, Roger P. Roess, and Elena S. Prassas. *Traffic Engineering*. Prentice Hall, Upper Saddle Ridge, NJ, 2 edition, 1998.

[79] B. Miller. A queueing reward system with several customer classes. *Management Science*, 16(3):234–245, November 1969.

[80] Philip M. Morse. *Queues, Inventories and Maintenance*. John Wiley and Sons, 1958.

[81] Matthew Mosk and Michael E. Ruane. Baltimore Battles Train Fire, Spills. *Washington Post*, 2001. July 20.

[82] P. Naor. The regulation of queue size by levying tolls. *Econometrica*, 37(1):15–24, 1969.

[83] G.F. Newell. *Approximate Behavior of Tandem Queues*, volume 171 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin, 1979.

[84] G.F. Newell. *Traffic Flow on Transportation Networks*. The MIT Press, Cambridge, Mass., 1980.

[85] President's Commission on Critical Infrastructure Protection. Critical Foundations. Technical report, The White House, 1997.

[86] President's Commission on Critical Infrastructure Protection. Report Summary. Technical report, The White House, 1997.

[87] D. Pappalardo. ATM network outage at AT&T. *Network World*, 2001. February 21.

[88] D. Pappalardo. AT&T, customers grapple with ATM outage. *Network World*, 2001. February 26.

[89] C. Perrow. *Normal Accidents*. Princeton University Press, 1999.

[90] Don Phillips. CSX Warns Fire Will Delay Freight. *Washington Post*, 2001. July 20.

[91] Michael Pinedo and Ronald W. Wolff. A comparison between tandem queues with dependent and independent service times. *Operations Research*, 30(3):464–749, May-June 1982.

[92] S. Reed. Massive power outage in West still unexplained. Cable News Network, 1996. July 3.

[93] E. Reich. Waiting times when queues are in tandem. *Annals of Mathematical Statistics*, 28:768–773, 1957.

[94] E. Reich. Notes on queues in tandem. *Annals of Mathematical Statistics*, 34:338–341, 1963.

[95] J. Rendleman. MCI Worldcom blames Lucent software for outage. *eWeek*, 1999. August 17.

[96] J. Rendleman. MCI Worldcom still under the gun. *eWeek*, 1999. August 12.

[97] Manuel Roig-Franzia. Chemical Experts Take the Acid Test: Tunnel Spill Spurs High-Stakes Mission. *Washington Post*, 2001. July 20.

[98] Manuel Roig-Franzia and Molly Ball. In Baltimore, Toxic Scare Abates. *Washington Post*, 2001. July 22.

[99] Manuel Roig-Franzia and Michael E. Ruane. Firefighters Pull Out 22 Burned Freight Cars. *Washington Post*, 2001. July 21.

[100] Raphael Rom and Moshe Sidi. *Multiple Access Protocols: Performance and Analysis*. New York. Springer-Verlag, 1990.

[101] Zvi Rosenberg, Pravin Variaya, and Jean Walrand. Optimal control of queues in tandem. *IEEE Transactions on Automatic Control*, AC-27(3):600–610, June 1982.

[102] J. Rosenbush. AT&T crash exposed Achilles' heel. *USA Today*, 1999. January, 26.

[103] Sheldon Ross. *Applied Probability Models with Optimization Applications*. Holden-Day, San Francisco, 1970.

[104] S.M. Ross. *Introduction to Probability Models, 8th Ed.* Harcourt/Academic Press, San Diego, 2002.

[105] S. Roy, C. Asavathiratham, B. Lesieutre, and G. Verghese. Network Models: Growth, Dynamics, and Failure. In *Proc. of 34th Hawaii Int'l Conference on System Sciences - 2001*, January 2001.

[106] S. Roy, B. Lesieutre, and G. Verghese. Resource Allocation in Networks: A Case Study of the Influence Model.

[107] R. Schassberger. A note on optimal service selection in a single server queue. *Management Science*, 21(11):1326–1331, July 1975.

[108] Paul Schwartzman and Manuel Roig-Franzia. Taken Out of the Ballgame: Thousands of Orioles Fans, Others Chased From Downtown. *Washington Post*, 2001. Article appeared on July 19, 2001.

[109] Richard Serfozo. Optimal control of random walks, birth and death processes, and queues. *Advances in Applied Probability*, 13:61–83, 1981.

[110] Suresh P. Sethi and Gerald L. Thompson. *Optimal Control Theory: Applications to Management Science and Economics*. Kluwer Academic Publishers, Boston, 2 edition, 2000.

[111] Matthew Sobel. Optimal operation of queues. In A.B. Clarke, editor, *Mathematical Methods in Queueing Theory*, volume 98 of *Lecture Notes in Economics and Mathematical Systems*, pages 232–261. Springer-Verlag, 1974.

[112] A. Soltis. The Nights the Lights Went Out. *New York Post*, 2001. November 16.

[113] CNN Staff and Associated Press. Millions left without power in West. *Cable News Network*, 1996. August 10.

[114] S. Stidham and N.U. Prabhu. Optimal control of queueing systems. In A.B. Clarke, editor, *Mathematical Methods in Queueing Theory*, volume 98 of *Lecture Notes in Economics and Mathematical Systems*, pages 263–294. Springer-Verlag, 1974.

[115] Shaler Stidham. Optimal control of admission to a queueing system. *IEEE Transactions on Automatic Control*, AC-30(8):705–713, August 1985.

[116] T. Sweeny and C. Moozakis. MCI Frame Net Melts Down. *InternetWeek*, 1999. August 12.

[117] H.M. Taylor and S. Karlin. *An Introduction to Stochastic Models*. Academic Press, San Diego, 3 edition, 1998.

[118] S.V. Tembe and R.W. Wolff. The optimal order of service in tandem queues. *Operations Research*, 22:824–832, 1974.

[119] M. E. Thyfault. AT&T Experiences Frame Relay Outage. *Infowar.Com & Interpact, Inc.*, 1998. April 28.

[120] Donald M. Topkis. Minimizing a submodular function on a lattice. *Operations Research*, 26(2):305–321, 1978.

[121] Pantelis Tsoucas and Jean Walrand. On the interchangeability and stochastic ordering of $\cdot$/M/1 queues in tandem. *Advances in Applied Probability*, 19:515–520, 1987.

[122] S. Walker. Rail Logjam Leaves Farmers, Christmas Retailers Stranded. *The Christian Science Monitor*, 1997. November 5.

[123] J. Walrand. *An Introduction to Queueing Networks*. Prentice Hall, Englewood Cliffs, NJ, 1988.

[124] J. Walrand and P. Variaya. Sojourn times and the overtaking condition in Jacksonian networks. *Advances in Applied Probability*, 12:1000–1018, 1980.

[125] H. Wang and J. S. Thorp. Optimal Locations for Protection System Enhancement: A simulation of Cascading Outages. *IEEE Trans. on Power Delivery*, 16(4), Sep 2001.

[126] D. Watts. A Simple Model of Fads and Cascading Failures. Technical Report SFI Working Paper SFI-00-12-062, The Santa Fe Institute, 2000.

[127] Richard R. Weber. On the optimal assignment of customers to parallel servers. *Journal of Applied Probability*, 15:406–413, 1978.

[128] Richard R. Weber. The interchangeability of $\cdot$/M/1 queues in series. *Journal of Applied Probability*, 16:690–695, 1979.

[129] J. Westbrook. Load balancing for response time. In *Proc. of the 3rd Annual European Symposium on Algorithms (ESA'95)*, 1995.

[130] W. Willinger, R. Govindan, S. Jamin, V. Paxson, and S. Shenker. Scaling phenomena in the Internet: Critically examining criticality. *Proc. Nat. Acad. Sci.*, 99, suppl. 1:2573–2580, 2002.

[131] P. Winkler. Optimality and greed in dynamic allocation. Technical report, Bell Labs, July 1999.

[132] Wayne Winston. Optimality of the shortest line discipline. *Journal of Applied Probability*, 14:181–189, 1977.

[133] Uri Yechiali. On optimal balking rules and toll charges in the GI/M/1 queueing process. *Operations Research*, 19(2):349–370, March-April 1971.

[134] Uri Yechiali. Customers' optimal joining rules for the GI/M/s queue. *Management Science*, 18(7):434–443, March 1972.

[135] S. Zacks and M. Yadin. Analytic characterization of the optimal control of a queueing system. *Journal of Applied Probability*, 7:617–633, 1970.