ROBUST AND SECURE FEEDBACK CONTROL FOR QUEUEING NETWORKS

THESIS

Submitted in Partial Fulfillment

of the Requirements for

the Degree of

MASTER OF SCIENCE (Transportation Planning and Engineering)

at the

NEW YORK UNIVERSITY
TANDON SCHOOL OF ENGINEERING

by

Qian Xie

September 2021

ROBUST AND SECURE FEEDBACK CONTROL FOR QUEUING NETWORKS

THESIS

Submitted in Partial Fulfillment

of the Requirements for

the Degree of

MASTER OF SCIENCE (Transportation Planning and Engineering)

at the

NEW YORK UNIVERSITY
TANDON SCHOOL OF ENGINEERING

by

Qian Xie

September 2021

Approved:

Advisor Signature

August 25, 2021

Date

Department Chair Signature

August 26, 2021

Date

University ID: N12332500

Net ID: qx463

Vita

Qian Xie was born in Guangzhou, China. She obtained her Bachelor's degree in Computer Science from the Institute for Interdisciplinary Information Sciences, Tsinghua University, in Beijing, China. She received research assistantship from NYU Tandon School of Engineering and C2SMART University Transportation Center.

Acknowledgements

I would like to thank my research advisor Li Jin, thesis advisor Joseph Chow, and academic advisor Elena Prassas for their support and guidance. I would also like to thank my group fellows Yu Tang, Haoran Su, and Xi Xiong for insightful discussions.

Qian Xie, New York University Tandon School of Engineering
August 25, 2021

ABSTRACT

ROBUST AND SECURE FEEDBACK CONTROL FOR QUEUEING NETWORKS

by

Qian Xie

Advisor: Prof. Joseph Y.J. Chow, Ph.D.

Co-Advisor: Prof. Li Jin, Ph.D.

Submitted in Partial Fulfillment of the Requirements for

the Degree of Master of Science (Transportation Planning and Engineering)

September 2021

Feedback control is commonly used in a variety of queueing networks, including transportation networks, production lines, supply chains, and communication networks. Most existing works base on the full knowledge of model data, the perfect observation of the traffic states, and the perfect implementation of the control. However, in practical settings, such data may not always be available and accurate. Therefore, this thesis fills the research gap on 1) the analysis of queueing network with data loss/errors; 2) the design of robust and secure feedback control that resists non-stationary environments and/or security risks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Queueing model is a model that quantifies the queuing and delays due to random arrival times and/or random service times but not captures demand and capacity fluctuations. Specifically, a queue is a waiting line and a queueing network is a network of servers and queues. The jobs (customers, vehicles, packets, etc.) arrive at random. They join the waiting queues when the servers are busy. Queueing models have been widely applied to many engineering systems such as transportation network, manufacturing system (production line), communication nework, and computer network. Examples include security checkpoint, toll booth, airport landing, ride-sharing, retail counter, and call center.

Feedback control, or closed-loop control, is a class of strategy that uses feedback from the output at the input to reduce errors and improve stability. Typical feedback control on queueing networks include routing, sequencing, service rate control, and admission control. Routing refers to allocating jobs to downstream servers while sequencing refers to selecting a job from the waiting queue to serve, e.g., first come first serve (FCFS). Service rate control determines the power of the servers. Admission control holds or rejects jobs to avoid congestion propagation. In this thesis, I consider all above feedback control in Chapter 2 when we analyze and design model-data independent control for Jackson queueing networks. Then in Chapter 3 and Chapter 4, I just focus on dynamic routing for parallel queues (servers).

Stability and optimization are two important topics associated with queueing networks with feedback control. However, unlike simple queueing systems (e.g., M/M/1), it is hard to compute the steady-state probabilities in general. Therefore, we need some useful tools. For stability, or mean boundedness, we can use Lyapunov function. For optimization, we can use theory of optimal control or Hamilton-Jacobi-Bellman (HJB) equation. In Chapter 4, both two topics are discussed, while in Chapter 2 and Chapter 3, I only focus on the stability issues.

Feedback control on queueing network relies on data collection and transmission via connected sensing and actuating components. However, in practical settings, such data may not always be available and accurate. Besides, the lack of robust-by-design and secure-by-design features make the components susceptible to data loss and data errors. One real-world instance is the Internet of Vehicles (IoV), where vehicles not only communicate with each other but also with pedestrians and infrastructures. The vehicles typically make decisions based on the real-time routing guidance services such as Google Maps and Waze. Such services heavily depend on traffic data but also face risks of random malfunctions and malicious attacks [1,67]. It has been reported that an artist used phones to create a phantom traffic jam on navigation apps [38]. We can expect that in the near future, hackers can spoof traffic sensor data or create phantom traffic jams in navigation apps for selfish or malicious intent (e.g., leading other vehicles to take a different road). Under such circumstances, the information provided by such services can be faulty, and the misled travelers may suffer extra travel times.

To the best of my knowledge, most existing analysis and design approaches base on the full knowledge of model data (i.e., arrival rates and service rates) [5], the perfect observation of the traffic states (i.e., queueing lengths), and the prefect implementation of the control [21,23]. There are research gaps on the control design for networks with partial knowledge of model data and imperfect state observation/control implementation. Nevertheless, the impact of random and strategic sensing faults has not been well understood; practical fault-tolerant mechanisms have not been developed either. Meanwhile, trip advisory providers, transportation agencies, and the public are in general concerned with the security of IoV. To address such concerns, this thesis develops theoretical foundations and practical insights for building robust and secure queueing control for network systems including IoV. The modeling and analysis approach for sensing faults can also be extended to other real-world applications such as ride-sharing, public transit, and aircraft control.

The rest of this thesis is organized as follows: Chapter 2 targets on the design of stabilizing MDI (modal data-independent) control policies. Such control policies select control actions including routing, sequencing, and/or holding. Chapter 3 focuses on the analysis of stability conditions and guaranteed throughput of the parallel routes with faulty information of traffic states. Chapter 4 aims at the design of secure control for the parallel queues under the risks of faulty data and/or faulty routing.

Chapter 2

Stabilizing Model Data-Independent Control

This chapter is a joint work with Li Jin. Submitted to IEEE Transaction on Control of Network Systems. Published under Creative Commons CC By 4.0 License [88].

Classical queueing network control strategies typically rely on accurate knowledge of model data, i.e. arrival and service rates. However, such data are not always available and may be time-variant. To address this challenge, we consider a class of model data-independent (MDI) control policies that only rely on traffic state observation and network topology. Specifically, we focus on the MDI control policies that can stabilize multi-class Markovian queueing networks under centralized and decentralized policies. Control actions include routing, sequencing, and holding. By expanding the routes and constructing piecewise-linear test functions, we derive an easy-to-use criterion to check the stability of a multi-class network under a given MDI control policy. For stabilizable multi-class networks, we show that a centralized, stabilizing MDI control policy exists. For stabilizable single-class networks, we further show that a decentralized, stabilizing MDI control policy exists. In addition, for both settings, we construct explicit policies that attain maximal throughput and present numerical examples to illustrate the results.

2.1 Introduction

Control on multi-class queueing networks has been studied in numerous contexts of transportation, logistics, and communication systems [46,55,75,91]. Most existing analysis and design approaches rely on full knowledge of model data, i.e., arrival and service rates, to ensure stability and/or optimality [5]. However, in many practical settings, such data may be unavailable or hard to estimate, and may be varying over time. Such challenges motivate the idea of *model data-independent (MDI)* control policies. MDI control policies select control actions, including routing, sequencing, and/or holding, according to state observation and network topology but independent of arrival/service rates. Such policies are easy to implement and, if appropriately designed, can resist modeling error or non-stationary environment. However, the stability of general open multi-class queueing networks with centralized or decentralized MDI control policies has not been well studied.

In this chapter, we consider the stability of multi-class queueing networks with throughput-maximizing MDI control policies. Particularly, we focus on acyclic open queueing networks with Poisson arrivals and exponential service times. Jobs (customers) are classified according to their origin-destination (OD) information. Service rates are independent of job classes. A network is stabilizable if there exists a control policy that ensures positive Harris recurrence of the queueing process, whether the network is open-loop or closed-loop, centralized or decentralized [19]. By standard results on Jackson networks, stabilizability is equivalent to the existence of a (typically model data-dependent) stabilizing Bernoulli routing policy [30]. We assume that the class-specific arrival rates and the server-specific service rates are unknown to the controller. The main results are as follows:

1. An easy-to-use criterion to check the stability of a multi-class network under a given MDI control policy (Proposition 1).
2. For a multi-class network, a stabilizing centralized MDI control policy exists if and only if the network is stabilizable (Theorem 1).
3. For a single-class network, a stabilizing decentralized MDI control policy exists if and only if the network is stabilizable (Theorem 2).

Previous works on stability of queueing networks are typically based on full knowledge of model data [15,20,29,46,69,70,79]. So far, the best-studied MDI control policy is the join-the-shortest-queue (JSQ) routing policy for parallel queues [21,23,28,35,54,58,80,84] or simple networks [9], which requires only the queue lengths and does not rely on model data [14]. When and only when the network is stabilizable, i.e., the demand is less than capacity (service rate), the JSQ policy guarantees the stability of parallel queues/simple networks [9,27] and the optimality of homogeneous servers [21]. However, JSQ routing does not guarantee stability of more complex networks [14]. MDI routing for general networks has been numerically evaluated [50], but no structural results are available. Most studies on MDI routing for general networks are not aimed for stability [45,61,66,81]. In addition, decentralized dynamic routing has been considered for single origin-destination networks [33,68] but not in MDI settings.

To design stabilizing MDI control policies, we first develop a stability criterion (Proposition 1) based on route expansion for queueing networks and explicit construction of a piecewise-linear test function. The expanded network is essentially a parallel connection of all routes from the set of origins to the set of destinations. With this expansion, we use insights on the behavior of parallel queues and of tandem queues to construct the test function and derive the stability criterion. The test function can be used to obtain a smooth Lyapunov function verifying a negative drift condition. The piecewise-linear test function technique was proposed by Down and Meyn [19]; however, their implementation relies on linear programming formulations to determine parameters of the test function, which depends on model data. We extend this technique to the MDI setting using explicitly constructed test functions.

Based on the stability criterion, we design control policies in centralized and decentralized settings. First, for multi-class networks, we present a stabilizing centralized MDI control policy requiring dynamic routing and preemptive sequencing named JSR policy (Theorem 1). The control policy is obtained by minimizing the mean drift of the piecewise-linear test function, and the mean drift is guaranteed to be negative if and only if the network is stabilizable. The JSR policy, which is centralized and MDI, maximizes throughput among all control policies. Compared with other centralized policies, it does not require knowledge of model data, and compared with other MDI policies (e.g., JSQ), it guarantees stability for any stabilizable networks. Second, for single-class networks, we present a decentralized routing and holding policy that guarantees stability (Theorem 2). Such policies can also maximize the throughput since the stabilizability of the network implies that the throughput can be as large as close to the capacity. The results are closely related to the theory on the classical JSQ routing policy [14] and the decentralized max-pressure control policy [83].

The rest of this chapter is organized as follows. Section 2.2 defines the multi-class queueing network model. Section 2.3 presents the stability criterion based on route expansion and piecewise-linear test function. Section 2.4 and Section 2.5 consider the control design problem in centralized and decentralized settings respectively. Section 2.6 gives concluding remarks.

2.2 Multi-class queueing network

Consider an acyclic network of queueing servers with infinite buffer spaces. Let $\mathcal{N}$ be the set of *servers*. Each server n has an exponential *service rate* $\bar{\mu}_n$. The network has a set $\mathcal{S}$ of *origins* and a set $\mathcal{T}$ of *destinations*. jobs are classified according to their origins and destinations. That is, we can use an origin-destination (OD) pair $(S, T) \in \mathcal{C}$ to denote a *job class*, or simply *class*. For notational convenience, classes (OD pairs) are indexed by $c = (S_c, T_c)$. jobs of class c arrive at S_c according to a Poisson process of rate $\lambda_c \geq 0$. We assume that service rates are independent of job class.

The *topology* of the network is characterized by *routes* between origins and destinations. We use $|r|$ to denote the number of servers on route r. Let $\mathcal{R}_c$ be the set of routes between S_c and T_c, and define $\mathcal{R} = \bigcup_{c \in \mathcal{C}} \mathcal{R}_c$. Below is an example network to illustrate the notations.

Example 1. *Consider the Wheatstone bridge network in Fig. 2.1. Two classes of jobs arrive at S_1 (resp. S_2) with $\lambda_1 > 0$ (resp. $\lambda_2 > 0$). The set of servers is $\mathcal{N} = \{1, 2, \ldots, 5\}$ and the set of OD-specific routes are*

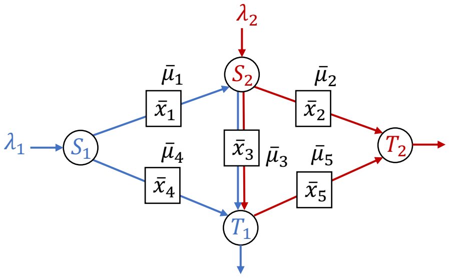


Figure 2.1: A two-class queueing network.

$$\mathcal{R}_1 = \{(1, 3), (4)\}, \ \mathcal{R}_2 = \{(2), (3, 5)\}.$$

The *state* of the network is defined as follows. Let $\bar{x} = [\bar{x}_n^c]_{n \in \mathcal{N}, c \in \mathcal{C}}$ be the vector of class-specific *job numbers*, where $\bar{x}_n^c$ is the number of jobs of class c in server n, either waiting or being served. Let $\bar{\mathcal{X}}$ be the space of $\bar{x}$. We use $\bar{X}(t)$ to denote the state of the queueing process at time t.

We consider three types of *control actions*, viz. routing, sequencing, and holding. All control actions are essentially Markovian (in terms of $\bar{x}$ plus additional auxiliary states) and are applied at the instant of *transitions*, which include the arrival of a job at an origin or the completion of service at a server. *Routing* refers to allocating an incoming job to a server downstream to the origin or allocating a job discharged by a server to another downstream server. *Sequencing* refers to selecting a job from the waiting queue to serve. The default sequencing policy is the first-come-first-serve (FCFS) policy. For the multi-class setting, we consider the preemptive-priority that can terminate an ongoing service and start serving jobs from another class, while the job with incomplete service is sent back to the queue.

Holding refers to holding a job that has completed its service in the server while blocking the other jobs in the queue from accessing the server.

Following [16], we say that a queueing network is *stable* if the queueing process is positive Harris recurrent. For details about the notion of positive Harris recurrence for queueing networks, see [15, 16, 19]. Finally, we say that the network is *stabilizable* if a stabilizing control exists. One can check the stabilizability using the following result:

Lemma 1. *An open acyclic queueing network is stabilizable if and only if there exists a vector* $[\xi_r]_{r\in\mathcal{R}}$ *such that*

$$\begin{aligned} &\xi_r \geq 0, \quad \forall r \in \mathcal{R}, \\ &\lambda_c = \sum_{r\in\mathcal{R}_c} \xi_r, \quad \forall c \in \mathcal{C}, \\ &\sum_{r\in\mathcal{R}:n\in r} \xi_r < \bar{\mu}_n, \quad \forall n \in \mathcal{N}. \end{aligned}$$

The proof and implementation are straightforward.

2.3 Stability criterion

In this section, we derive a stability criterion for multi-class networks under given control policies. The techniques that we use include the route expansion of the original network and the explicit construction of a piecewise-linear test function based on the network topology. In Section 2.3.1, we construct an expanded network based on the original network. In Section 2.3.2, we apply a piecewise-linear test function to the expanded network to obtain a stability criterion (Proposition 1) for both the expanded and the original networks.

2.3.1 Route expansion

For the convenience of constructing test function, we first introduce the route expansion. *Route expansion* refers to the construction of an *expanded network* based on the topology of *original network* (defined in Section 2.2). The high-level idea is to decompose the network into routes, and the specific procedures are:

1. Place all routes $\mathcal{R}$ in the original network in parallel.

2. Add two-way connections between duplicates of servers in the original network.

For example, Fig. 2.2 shows the expanded network constructed from the original network in Fig. 2.1.

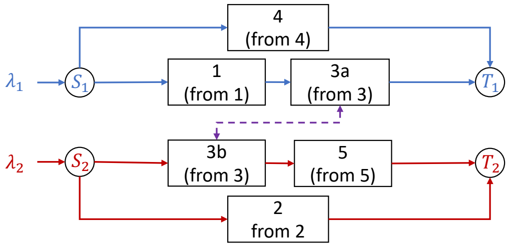


Figure 2.2: Route expansion of the network in Fig. 2.1.

We call "servers" in the expanded network as *subservers*, since they are obtained by duplicating actual servers in the original network. Subservers are indexed by k, $c_k \in \mathcal{C}$ is the class index, $r_k \in \mathcal{R}$ is the route index, and $i_k \in \{1, 2, \ldots, |r_k|\}$ is the numbering of subserver k on route r_k. We use $k \in r$ to refer to that subserver k is on route r. Let $\mathcal{K}$ be the set of all subservers and $\mathcal{K}_c$ be the set of subservers with $c_k = c$. We use $n_k \in \mathcal{N}$ to denote the actual server that corresponds to subserver k. In addition, let k_p (resp. k_s) denote the subserver immediately upstream (resp. downstream) to subserver k.

The *state* of the expanded network is $x = \{x_k^c; k \in \mathcal{K}, c \in \mathcal{C}\}$, denoting the vector of number of class-c jobs in subserver k. Let $x_k := \sum_{c\in\mathcal{C}} x_k^c, k \in \mathcal{K}$. The expanded state space is $\mathcal{X} = \mathbb{Z}_{\geq 0}^{|\mathcal{C}|\times|\mathcal{K}|}$. Note that the states of the expanded network and the states of the original network are related by

$$\bar{x}_n^c = \sum_{k \in \mathcal{K}: n_k = n} x_k^c, \quad k \in \mathcal{K}, \tag{2.1}$$

for each $n \in \mathcal{N}$.

The routing policy is characterized by $\pi : \mathcal{X} \to [0,1]^{|\mathcal{C}| \times |\mathcal{K}|^2}$, where $\pi_{k,k'}^c$ is the probability that a class-c job is routed from subserver k to subserver k'.

The holding policy is characterized by $\zeta : \mathcal{X} \to \{0,1\}^{|\mathcal{K}|}$, where ζ_k specifies whether subserver k is holding ($\zeta_k(x) = 0$) or not holding ($\zeta_k(x) = 1$) when the current state is x.

Two subservers k and k' are *duplicating* if $n_k = n_{k'}$. Note that the service rates of duplicating subservers are coupled in the sense that for each server $n \in \mathcal{N}$, at a given time, at most one subserver k such that $n_k = n$ can be actively serving jobs, or *active*. This can be modeled as an *imaginary service rate control policy* $\mu : \mathcal{X} \to \mathbb{R}^{|\mathcal{K}|}$ such that the service rate $\mu_k(x)$ of subserver k satisfies

$$\sum_{k: n_k = n} \mu_k(x) \le \bar{\mu}_n, \quad \forall x \in \mathcal{X}.$$

Such control policy is essentially equivalent to the class-based preemptive sequencing in the original network.

Note that $\{X(t) : t \ge 0\}$ is a Markov process, and the positive Harris recurrence refers to that there exists a unique invariant measure ν on $\mathcal{X}$ such that for every measurable set $D \subseteq \mathcal{X}$ with $\nu(D) > 0$ and for every initial condition $x \in \mathcal{X}$,

$$\Pr\{\tau_D < \infty | X(0) = x\} = 1,$$

where $\tau_D = \inf\{t \ge 0 : X(t) \in D\}$. Also, though $\{\bar{X}(t) : t \ge 0\}$ is not a Markov process, it will eventually converge to a steady state distribution.

The route expansion technique not only expands the network but also decomposes the state variables. Jobs can move along the expanded network using two transition mechanisms. One is *actual transition*, referring to moving a job from subserver k (or an origin) to its downstreamsubserver k_s (or a destination). The other is *imaginary transition* that moves a job from one subserver k to a duplicating subserver k' thereof, see imaginary switch in Section 2.5. Imaginary transitions always occur instantaneously. Note that an actual transition corresponds to a transition in the original network, while an imaginary transition does not; this is also revealed in (2.1).

One can always map a control action in the expanded network to the original network. However, an MDI control policy may not exist on the state space of the original network; we do need an expanded state space for MDI control. In addition, we allow imaginary control actions in the expanded network, including *imaginary service rate control* and *imaginary* *switch*; see Section 2.4 and Section 2.5. Such imaginary actions only make sense in the expanded network and do not correspond to actual service rate control or switch in the original network.

2.3.2 Stability of the expanded network

After introducing the expanded network and the mathematical definition of the control policy, the main question of this paper can be expressed in a formal way as follows.

Given an expanded queueing network and a control policy $\phi = (\pi, \mu, \zeta)$, how can we tell if the control policy ϕ is stabilizing, i.e., the expanded network is stable under ϕ?

The answer of this question will be given in Proposition 1. Before that, we need to introduce the test function technique first. As opposed to linear programming-based construction in [19], we provide an explicit construction, where parameters of the test function do not rely on solving any optimization problems. The high level idea is to identify the bottlenecks and their upstream subservers. Our construction is based on the route expansion described in the previous subsection.

1. For each class $c \in \mathcal{C}$ and expanded state $x \in \mathcal{X}$, define

   $$g_c(x) := \max_{\substack{K_c \subseteq \mathcal{K}_c: \\ \kappa \in K_c \Rightarrow \kappa_p \in \mathcal{K}_c}} \sum_{k \in K_c} a_k x_k,$$

   where $a_k \in (0,1)$ is a parameter.

2. Define a piecewise-linear *test function*

   $$V(x) := \max_{C \subseteq \mathcal{C}} \sum_{c \in C} b_c g_c(x),$$

   where $b_c \in (0,1)$ is a parameter.

We call $V(x)$ the test function rather than the Lyapunov function, since strictly speaking, a smooth Lyapunov function should be developed based on the piecewise-linear test function to verify the Foster-Lyapunov stability criterion. Down and Meyn [19] showed that as long as a piecewise-linear test function can be determined, one can always smooth it to obtain a qualified C^2 Lyapunov function.

Remark 1. *The test functions we proposed in this work are MDI. But generally speaking, they do not need to be MDI since it does not affect the control policies to be MDI.*

Definition 1 (Dominance)**.** *Consider state $x \in \mathcal{X}$.*

1. *We call C^* a set of* dominant *classes if*

   $$C^* \in \operatorname{argmax}_{C \subseteq \mathcal{C}} \sum_{c \in C} b_c g_c(x).$$

Each class $c \in C^*$ is a dominant *class.*

2. *We call K_c^* a set of* dominant *class-c subservers if*

$$K_c^* \in \underset{\substack{K_c \subseteq \mathcal{K}_c: \\ k \in K_c \Rightarrow k_p \in \mathcal{K}_c}}{\operatorname{argmax}} \sum_{k \in K_c} a_k x_k.$$

   Each subserver $k \in K_c^*$ is a dominant *class-c subserver.*

3. *A route $r \in \mathcal{R}_c$ is* dominant *if it includes dominant class-c subservers, i.e. there exists dominant class-c subserver $k \in K_c^*$ such that $k \in r$.*

   Let R_c be the set of dominant class-c routes.

4. *A subserver $b \in K_c^*$ is called a* bottleneck *if it is a dominant class-c subserver while its immediate downstream subserver $b_s \notin K_c^*$ is not.*

Remark 2. *A route or server is dominant if changes in its traffic state immediately affect the test function V.*

A *regime* X of the piecewise-linear test function is a subset of $\mathcal{X}$ such that there exist $C^X \subseteq \mathcal{C}$, $K^X = \bigcup_{c \in C^X} K_c^X \subseteq \mathcal{K}$, and $R^X = \bigcup_{c \in C^X} R_c^X \subseteq \mathcal{R}$ where C^X, K_c^X, R_c^X are dominant for each $x \in X$, i.e., the test function is linear over X. Let $\mathscr{X}$ be the set of regimes; note that $\bigcup_{X \in \mathscr{X}} X = \mathcal{X}$.

Definition 2 (Mean velocity and drift)**.** *Consider a multi-class network with state $x \in \mathcal{X}$ under an expanded control policy $\phi = (\pi, \mu, \zeta)$.*

1. *The* mean velocity *at state x is a function $v : \mathcal{X} \to \mathbb{R}^{|\mathcal{K}|}$ such that for each $k \in \mathcal{K}$,*

$$v_k(x) := \sum_{c \in \mathcal{C}} \lambda_c \pi^c_{S_c,k}(x) + \mu_{k_p}(x)\zeta_{k_p}(x) - \mu_k(x)\zeta_k(x).$$

   where $\pi^c_{S_c,k}$ is the probability that a class-c job is routed from origin S_c to subserver k, while μ_k and ζ_k are the controlled service rate and the holding status of the subserver k respectively.

2. *Given $X \in \mathscr{X}$ such that $x \in X$, the* mean drift over X *is given by*

$$D^X(x) := \sum_{c \in C^X} b_c \sum_{k \in K_c^X} a_k v_k(x).$$

Remark 3. *In our subsequent analysis, the mean drift $D^X(x)$ of the test function will play the role of infinitesimal generator applied to a Lyapunov function; see [19] for the connection between the test function and the Lyapunov function.*

The main result of this section is as follows:

Proposition 1. *Consider a multi-class network under the expanded control policy ϕ. Suppose there exist constants $M<\infty$, $\epsilon>0$, and $a_k, b_c \in (0,1)$ ($\forall\ c\in\mathcal{C},\ k\in K_c$), such that for each $X\in\mathscr{X}$ and each $x\in X$ where $|x|=\sum_{k\in\mathcal{K}} x_k > M$,*

$$\sum_{c\in C^X} b_c \sum_{k\in K_c^X} a_k v_k(x) \leq -\epsilon. \tag{2.2}$$

Then, the network is stable.

Proof. Consider the test function $V(x)$. By its definition, if $x\in X$, we have

$$V(x)=\sum_{c\in C^X} b_c \sum_{k\in K_c^X} a_k x_k.$$

The mean drift is given by

$$\begin{aligned} D^X(x) &= \sum_{c\in C^X} b_c \sum_{k\in K_c^X} a_k v_k(x) \\ &\overset{(2.2)}{\leq} -\gamma^{|C^X|-1}\epsilon \leq -\gamma^{|\mathcal{C}|-1}\epsilon, \quad x:|x|>M. \end{aligned}$$

One can then apply [19, Theorem 1] and [19, Lemma 5] to obtain the stability of the network. □

As a benchmark, the approach in [19, Theorem 1] requires solving linear programs to obtain parameters of the test functions in Proposition 1, while our approach explicitly constructs the parameters (see Section 2.4 and Section 2.5). Moreover, the proposed control, which is independent of model data, guarantees stability if and only if the network is stabilizable (see Theorem 1 and Theorem 2), while the approach in [19] relies on knowledge of model data.

2.4 Centralized control for multiple classes

In this section, we consider the "join-the-shortest-route (JSR)" policy (a joint routing and sequencing policy) for centralized control. The JSR policy is MDI and constructed based on the expanded network. We will show that it is stabilizing if and only if the network is stabilizable.

The test functions are constructed as follows.

1. For each class $c \in \mathcal{C}$, each route $r \in \mathcal{R}_c$, and each expanded state $x \in \mathcal{X}$, let

$$f_r(x) := \max_{k \in r} \alpha^{i_k - 1} \sum_{j: i_j \leq i_k} x_j,$$
$$g_c(x) := \max_{R_c \subseteq \mathcal{R}_c} \beta^{|R_c|-1} \sum_{r \in R_c} f_r(x),$$

   where $\alpha \in (0,1), \beta \in (0,1)$ are constant parameters.

2. The piecewise-linear *test function* is given by

$$V(x) := \max_{C \subseteq \mathcal{C}} \gamma^{|C|-1} \sum_{c \in C} g_c(x),$$

   where $\gamma \in (0,1)$ is a constant parameter.

Let the parameters be such that

$$\alpha = \beta \geq \frac{|\mathcal{R}|-1}{|\mathcal{R}|}, \quad \gamma \geq \frac{|\mathcal{C}|-1}{|\mathcal{C}|}, \tag{2.3}$$

and follow the notions of dominance accordingly (see Definition 1). Note that such MDI parameters α, β, γ always exist. The control that we consider in this subsection only depends on α, β, γ and is thus MDI. Specifically, we define the JSR policy as follows:

Definition 3 (Join-the-shortest-route (JSR) policy)**.**

1. *(Routing) At an origin S, an incoming job of class c is allocated to the route $r^* \in \mathcal{R}_c$ such that*

$$r^* \in \operatorname*{argmin}_{r \in \mathcal{R}_c} f_r(x).$$

   If there is only one minimum, then r^* must be a non-dominant route. Otherwise, let b^* be the bottleneck on route r^*. Then, an incoming job of class c is allocated to the route $r^* \in \mathcal{R}_c$ with the largest i_{b^*}, which is denoted by i_c. Further ties are randomly broken.

2. *(Imaginary service rate control) Let $\mathcal{K}_n$ be the set of subservers corresponding to server n and let $\mathcal{B}$ be the set of bottlenecks for a given x. Then, a subserver $k \in \mathcal{K}_n$ is activated if $k \in \mathcal{B}$. If multiple subservers are in $\mathcal{K}_n \cap \mathcal{B}$, then activate the subserver k^* such that*

$$k^* = \underset{k \in \mathcal{K}_n \cap \mathcal{B}}{\operatorname{argmin}} \{ i_{c_k} + |\mathcal{R}_{c_k}| \};$$

ties are randomly broken. This is to ensure that the bottlenecks are active to discharge jobs and only one of the duplicating subservers can be active.

The main result of this section is the following:

Theorem 1 (Stability of JSR policy)**.** *The JSR policy stabilizes a multi-class network if and only if the network is stabilizable.*

This theorem implies that the JSR policy is also throughput-maximizing, as long as the network is stabilizable, i.e., the demand is less than the total capacity. Note that the stabilizability can be easily checked using Lemma 1.

In the rest of this section, we apply Theorem 1 to study the stability of the Wheatstone bridge network under the JSR policy (Subsection 2.4.1) and then prove this theorem (Subsection 2.4.2).

2.4.1 Numerical Example

Consider the network in Fig. 2.1 and suppose that $\lambda_1 = \lambda_2 = \lambda = 1$ and $\bar{\mu}_n = \mu = 1$ for $n = 1, 2, 4, 5$ and $\bar{\mu}_3 = \frac{1}{4}$. This example is for illustrating the route expansion and the test function construction.

Note that under the above model parameters, the decentralized JSQ policy is destabilizing. To see this, $\bar{\mu}_1 = \bar{\mu}_4$ implies that on average, class-1 jobs are evenly distributed between server 1 and server 4. Thus, the average departure rate of class-1 jobs from server 1 is $\frac{1}{2}$, which exceeds the service rate of server 3. Therefore, the queue at server 3 is unstable. The main reason that the JSQ policy is destabilizing is the ignorance of downstream congestion. As $\bar{X}_3(t)$ gets large, a reasonable action is to allocate fewer class-1 jobs to server 1. However, the JSQ policy disallows such far-sighted decisions.

An alternative centralized stabilizing routing policy can be the following JSR policy:

1. A class-1 job arriving at S_1 is routed to server 1 if $\bar{X}_1^1(t) + \bar{X}_3^1(t) < \bar{X}_4^1(t)$, to server 4 if $\bar{X}_1^1(t) + \bar{X}_3^1(t) > \bar{X}_4^1(t)$, and randomly otherwise.

2. A class-2 job arriving at S_2 is routed to server 3 if $\bar{X}_3^2(t) + \bar{X}_5^2(t) < \bar{X}_2^2(t)$, to server 2 if $\bar{X}_3^2(t) + \bar{X}_5^2(t) > \bar{X}_2^2(t)$, and randomly otherwise.

3. The dominant class has a higher priority.

That is, when jobs are routed at S_1, the decision is based on not only the local state ($\bar{X}_1(t)$ and $\bar{X}_4(t)$), but also the state further downstream ($\bar{X}_3(t)$).

The expanded network is shown in Fig. 2.2. Each block in the figure represents a subserver. In particular, subservers 3a and 3b are decomposed from server 3; the other servers are remained. Solid arrows correspond to actual transitions in an original network, while dashed arrows correspond to imaginary transitions between duplicating subservers.

In the expanded network, a job can move along both solid and dashed arrows. The color of an arrow shows which class can move along it: blue means class (S_1, T_1), red means (S_2, T_2), and purple means both. For ease of presentation, we label (S_1, T_1) as class 1 and (S_2, T_2) as class 2. For example, a job of class (S_1, T_1) can visit subservers 4, 1, 3a, 3b and the destination T_1.

In the expanded network, the JSR policy works as follows.

1. A class-1 job arriving at S_1 is routed to subserver 4 if $X_4(t) < X_1(t) + X_{3a}(t)$, to subserver 1 if $X_4(t) > X_1(t) + X_{3a}(t)$, and randomly otherwise.

2. A class-2 job arriving at S_2 is routed to subserver 3b if $X_{3b}(t) + X_5(t) < X_2(t)$, to subserver 2 if $X_{3b}(t) + X_5(t) > X_2(t)$, and randomly otherwise.

3. If subserver 3a is dominant while subserver 3b is non-dominant, and server 3 is serving a class-2 job, then server 3 preempts the class-2 job being served in 3b to the class-1 job in 3a, and vice versa. If both subserver 3a and subserver 3b are dominant, then server 3 gives priority to class-2 job since the index of 3b is smaller.

By Theorem 1, the network can be stabilized by the JSR policy if and only if

$$\lambda_1 < 2,\ \lambda_2 < 2,\ \lambda_1 + \lambda_2 < \frac{9}{4}.$$

We use the following parameters for the test function:

$$\alpha = \beta = \gamma = \frac{3}{4},\ \epsilon = \left(\frac{3}{4}\right)^5.$$

One can verify that the above parameters satisfy (2.3) and Proposition 1 by considering the following cases:

1. Only one route is dominant. In this case, an incoming job is always allocated to a non-dominant route, leading to non-positive contribution to the mean drift:

$$D^X(x) \le -\gamma\beta\alpha\mu = -\left(\frac{3}{4}\right)^3 \le -\epsilon.$$

2. Two routes with different OD pairs are dominant. This case is analogous to the

previous case:

$$D^X(x) \leq -\gamma\beta\alpha\mu = -\Big(\frac{3}{4}\Big)^3 \leq -\epsilon.$$

3. Two routes with the same OD pair or more than two routes are dominant. In such cases, the mean drift satisfies

$$D^X(x) \leq \gamma\beta^3(\lambda - \mu - \alpha\mu) = -\Big(\frac{3}{4}\Big)^5 \leq -\epsilon.$$

Consequently, the network is stable under the MDI JSR policy.

2.4.2 Proof of Theorem 1

In this subsection, we will the sufficiency and the necessity respectively, based on the connection between the sign of the mean drift and the stabilizability of the network. When analyzing the mean drift, we consider two parts: external arrivals and internal transmission. We first show that any internal transmission does not positively contribute to the mean drift and then show that any positive contribution from external arrivals can always be compensated by internal transmissions.

Internal transmissions

Note that under the JSR policy, every job remains on the route assigned to the job when it enters the network. Hence, internal transmissions only occur between subservers on the same route.

Given x, consider an internal transmission from subserver k to subserver j; this implicitly requires $x_k \geq 1$. The definition of dominance ensures that if j is dominant, then so is k. Hence, we need to consider the following cases:

1. If k and j are both dominant, the transmission leads to zero contribution to the mean drift $D^X(x)$ for all X such that $x \in X$.

2. If k is dominant and j is non-dominant, the transmission leads to the following contribution to the mean drift:

$$-\alpha^{i_k-1}\mu_k(x) \leq 0.$$

Hence, internal transmissions never lead to positive contribution to the mean drift.

External arrivals

Given $x \neq 0$, consider a regime $X \in \mathscr{X}$ such that $x \in X$. For each $c \in \mathcal{C}$, the JSR policy ensures that if there exists a non-dominant route in $\mathcal{R}_c$, then an incoming job must be allocated to a non-dominant route in $\mathcal{R}_c$, leading to non-positive contribution to the mean drift. Hence, we only need to consider dominant classes c such that every route in $\mathcal{R}_c$ is dominant, i.e. $R_c^X = \mathcal{R}_c$. Recall that $C^* \subseteq \mathcal{C}$ is the set of dominant classes. The part of the mean drift associated with $c \in C^*$ satisfies

$$\begin{aligned} D_c^X(x) &\leq \gamma^{|C^*|-1}\beta^{|\mathcal{R}_c|-1}\Big(\alpha^{i_c-1}\lambda_c - \sum_{b\in\mathcal{B}^X:c_b=c}\alpha^{i_b-1}\mu_b(x)\Big) \\ &:= \gamma^{|C^*|-1}\Delta_c^X(x) \end{aligned}$$

over any regimes of the piecewise-linear test function, where i_c is given in Definition 3.

Lemma 2. *When $x \neq 0$, there is no empty bottleneck, i.e.*

$$x_b \geq 1. \tag{2.4}$$

Proof.

Since $x \neq 0$ and r_b is dominant, we have

$$\sum_{k:i_k\leq i_b} x_k > 0.$$

If $i_b = 1$, then the above inequality directly implies (2.4).

Now consider the case that $i_b \geq 2$. Since b is a bottleneck, we have

$$\alpha^{i_b-1}\sum_{k\in r_b:i_k\leq i_b} x_k \geq \alpha^{i_b-2}\sum_{k\in r_b:i_k\leq i_b-1} x_k,$$

which implies

$$x_b \geq (1-\alpha)\sum_{k:i_k\leq i_b} x_k > 0$$

and thus we have (2.4). □

Lemma 2 is to ensure that the bottlenecks are none-empty to discharge jobs and thus contribute negative terms to the drift.

Next, we show the sufficiency of Theorem 1. Based on the definition of the routing policy (see Definition 3), $\forall b \in \mathcal{B}^X$, we have $i_b \leq i_c$ when the incoming job is allocated to a dominant route. Then

$$\begin{aligned}\sum_{c\in C^*}\Delta_c^X(x) &\leq \sum_{c\in C^*}\beta^{|\mathcal{R}_c|-1}\alpha^{i_c-1}\Big(\lambda_c-\sum_{b\in\mathcal{B}^X:c_b=c}\mu_b(x)\Big)\\ &=\sum_{c\in C^*}\alpha^{i_c+|\mathcal{R}_c|-2}\Big(\lambda_c-\sum_{b\in\mathcal{B}^X:c_b=c}\mu_b(x)\Big)\\ &:=\sum_{c\in C^*}\alpha_c\beta_c\end{aligned}$$

Without loss of generality assume $C^*=\{1,2,\cdots,m\}$ and

$$i_1+|\mathcal{R}_1|\leq i_2+|\mathcal{R}_2|\leq\cdots\leq i_m+|\mathcal{R}_m|.$$

Then by using Abel transformation (summation by parts), the right hand side of the above inequality (abbr. RHS):

$$RHS=\sum_{i=1}^{m-1}(\alpha_i-\alpha_{i+1})\sum_{j=1}^{i}\beta_j+\alpha_m\sum_{j=1}^{m}\beta_m.$$

Based on the assumption, we have $\alpha_i\geq\alpha_{i+1}$ and

$$\begin{aligned}\sum_{j=1}^{i}\beta_j &= \sum_{j=1}^{i}\Big(\lambda_j-\sum_{b\in\mathcal{B}^X:c_b=j}\mu_b(x)\Big)\\ &=\sum_{j=1}^{i}\lambda_j-\sum_{n_b:b\in\mathcal{B}_i^X}\bar{\mu}_{n_b}\\ &=\sum_{j=1}^{i}\lambda_j-\sum_{n\in\mathcal{N}_i}\bar{\mu}_n\\ &<0,\end{aligned}$$

where $\mathcal{B}_i$ is the set of bottlenecks in the first i classes and $\mathcal{N}_i$ is the min-cut of the original network with the first i classes. Here we use the definition of the imaginary service rate control (see Definition 3) and Lemma 1.

Since $RHS<0$, we have $\sum_{c\in C^*}\Delta_c^X(x)<0$ and thus $D_c^X(x)<0$. Then by noting that internal transmissions lead to non-positive contributions to the mean drift, we have

$$D^X(x)\leq\sum_{c\in C^*}D_c^X(x)<0,$$

which implies stability. □

Finally, the necessity is apparent: if a network is not stabilizable, then there exists no MDI control that can stabilize the network.

2.5 Decentralized control for a single class

For a single-class network, we can drop the class index and use x_k to denote the number of jobs in subserver k. Note that such network has a single origin and a single destination. Again we can do route expansion on such network.

We consider a decentralized MDI control policy as follows.

Definition 4 (JSQ with artificial spillback)**.** *The JSQ with artificial spillback (JSQ-AS) policy is as follows:*

1. *(Routing) A discharged job is routed to the shortest downstream queue, with ties randomly broken.*

2. *(Holding) For each subserver k, any job which has finished the service will be held if and only if $X_{s_k}(t) \geq X_k(t)$.*

3. *(Imaginary switch) When a dominant subserver k is inactive while its non-dominant duplicate k' is active, and both are not in the holding status, then the job in k' is moved to (and discharged from) k after service, and then routed to the downstream of k (i.e., s_k).*

Note that under the holding policy, the process $\{X(t); t \geq 0\}$ admits an invariant set $\mathcal{Q} \subseteq \mathcal{X}$ given by

$$\mathcal{Q} := \{x \in \mathcal{X} : x_{s_k} \leq x_k, k \in \mathcal{K}\}. \tag{2.5}$$

Since we consider the long-time stability of the network, it suffices to consider the states in an invariant set. The above result indicates that in the invariant set $\mathcal{Q}$, the queue size of any subserver is upper-bounded by the queue size of its immediate upstream subserver.

The JSQ-AS policy is decentralized in the sense that control actions on subserver k only depend on local traffic information: the number of jobs in duplicate subservers $\{x_{k'} : n_k = n_{k'}\}$ and that in immediate downstream subservers $\{x_{s_{k'}} : n_k = n_{k'}\}$. A key characteristic of such policies is that congestion information can propagate through the network via the forced holding: if a subserver becomes congested (i.e. x_k gets large), the congestion will propagate to the upstream subservers in a cascading manner ("artificial spillback"). Importantly, such artificial spillback does not undermine throughput like the natural spillback caused by the limited buffer size. The reason is that though congestion can propagate, the queue size in any downstream subserver is not upper-bounded. Artificial spillback is the main difference between the JSQ-AS policy and the classic JSQ policies.

Note that though the JSQ-AS policy is constructed based on the expanded network, its actions can always be converted to the ones in the original network. Importantly, the decentralized control in the expanded network must also be decentralized in the original network. Also note that the imaginary switch has no impact on the original network or the test function.

The main result of this section is as follows:

Theorem 2 (Stability of JSQ-AS policy)**.** *For the route expansion of a single-class network, the JSQ-AS policy is stabilizing if and only if*

$$\lambda < \bar{\mu}^{mc}, \tag{2.6}$$

where $\bar{\mu}^{mc}$ is the min-cut service rate of the original network.

This theorem implies that JSQ-AS policy is also a throughput-maximizing policy since we allow any throughput that satisfies (2.6).

In the rest of this section, we apply Theorem 2 to study the stability of the Wheatstone bridge network under the JSQ-AS policy (Subsection 2.5.1) and then prove this theorem (Subsection 2.5.2).

2.5.1 Numerical Example

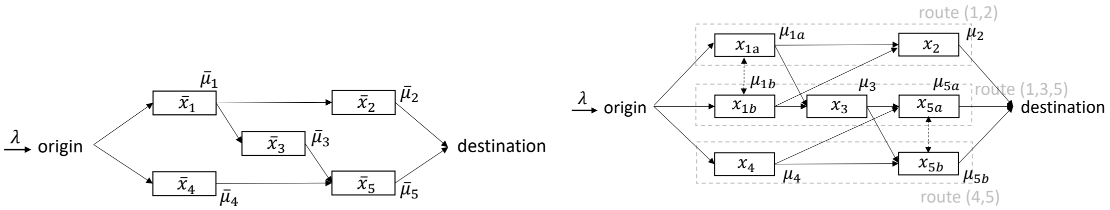


Figure 2.3: A single-class queueing network and its expanded network.

Consider the original network with route expansion in Fig. 2.3. Again suppose that $\lambda = 1$, $\bar{\mu}_n = \frac{3}{4}$ for $n = 1, 2, \cdots, 5$. Similarly, with the above parameters, the JSQ policy is destabilizing since the queue at server 5 is unstable. However, in the decentralized setting, the control actions can only depend on the local state, say the routing decision at the origin can be based on $\bar{X}_1(t)$ and $\bar{X}_4(t)$, but not $\bar{X}_5(t)$. A remedy is to introduce the holding policy (artificial spillback) to the JSQ policy so that the downstream congestion can be relieved and the local state can somehow reflect the states further downstream.

In the expanded network, server 1 is decomposed into subserver 1a and 1b, server 5 is decomposed into subserver 5a and 5b. The states in the original network and those in the expanded network satisfy $\bar{X}_1(t) = X_{1a}(t) + X_{1b}(t)$ and $\bar{X}_5(t) = X_{5a}(t) + X_{5b}(t)$. The initial states of the expanded network can be not unique. Say the initial queue size of server 1 is 2, then the initial queue sizes of subserver 1a and 1b can be 2, 0 or 1, 1 or 0, 2 respectively. Then the states are updated based on the model and our JSQ-AS policy. For example, the routing decision at the origin is based on $X_{1a}(t)$, $X_{1b}(t)$ and $X_4(t)$ rather than $\bar{X}_1(t)$ and $\bar{X}_4(t)$; a job which has just finished the service at server 3 will be held if $X_{5a}(t) \geq X_3(t)$, once released, it will be routed to the shorter downstream queue by comparing $X_{5a}(t)$ and $X_{5b}(t)$.

2.5.2 Proof of Theorem 2

This proof uses the connection between the stabilizability condition (2.6) and the sign of the mean drift. But before showing the mean drift is negative, we first present the explicit MDI piecewise-linear test function and several key lemmas that can help us analyze the mean drift.

The piecewise-linear test function is constructed as follows:

$$V(x) := \max_{\substack{K\subseteq\mathcal{K}:\\ \kappa\in K\Rightarrow p_\kappa\in\mathcal{K}}} \left\{\frac{1+(|K|-1)\delta}{|K|}\sum_{k\in K} x_k\right\},$$

where δ can be any small value such that $0<\delta<1$.

The following lemmas are useful in proving Theorem 2 where we consider the regime $X\subseteq\mathcal{Q}$ containing x.

Lemma 3. *A bottleneck can not be in the holding status.*

Proof. Otherwise, the bottleneck must have at least one downstream subserver. By (2.5), $x_{s_k}\geq x_b$. Since b is a bottleneck, we have

$$\frac{1+(|K^X|-2)\delta}{|K^X|-1}\sum_{k\neq b} x_k \leq \frac{1+(|K^X|-1)\delta}{|K^X|}\sum_{k\in K^X} x_k, \tag{2.7}$$

which implies

$$(1-\delta)\sum_{k\in K^X} x_k \leq |K^X|[1+(|K^X|-2)\delta]x_b. \tag{2.8}$$

Since $x_b\leq x_{s_k}$, we have

$$(1-\delta)\sum_{k\in K^X} x_k < |K^X|(1+|K^X|\delta)x_{s_k},$$

which is equivalent to

$$\frac{1+(|K^X|-1)\delta}{|K^X|}\sum_{k\in K^X} x_k < \frac{1+|K^X|\delta}{|K^X|+1}\Big(\sum_{k\in K^X} x_k + x_{s_k}\Big),$$

contradicting with the fact that subserver b is dominant and subserver s_k is non-dominant. □

Corollary 1. *Based on (2.8), we have $x_b>0$, i.e., any bottleneck b must be non-empty.*

This corollary and Lemma 3 ensure that all bottlenecks can discharge customers and contribute negative terms to the drift.

Lemma 4. *Let k_r^1 be the first subserver on route r, then either the route with the smallest $x_{k_r^1}$ is non-dominant or every route is dominant.*

Proof. If there is only one route, then that route must be dominant. Now assume there are at least two routes and route $\hat{r}$ has the smallest $x_{k_r^1}$, i.e. $\forall\ r \in \mathcal{R}$, $x_{k_{\hat{r}}^1} \leq x_{k_r^1}$. Suppose $k_{\hat{r}}^1 \in K^X$ and $\exists\ r \in \mathcal{R}$ s.t. $k_r^1 \notin K^X$. Note that by (2.5), $x_b \leq x_{k_{\hat{r}}^1} \leq x_{k_r^1}$, then from (2.7) we have

$$\frac{1+(|K^X|-1)\delta}{|K^X|}\sum_{k\in K^X} x_k < \frac{1+|K^X|\delta}{|K^X|+1}\Big(\sum_{k\in K^X} x_k + x_{k_r^1}\Big),$$

contradicting with our supposition. Therefore, either $\hat{r}$ is non-dominant or every route in $\mathcal{R}$ is dominant. □

Lemma 5. *If $x \in \mathcal{X}$ makes every route $r \in \mathcal{R}$ dominant, then we have*

$$\sum_{k\in\mathcal{B}^X} \mu_k(x) = \sum_{n:n=n_k,k\in\mathcal{B}^X} \bar{\mu}_n$$

Proof. Once there is an inactive bottleneck k and a non-dominant but active duplicate subserver k', the imaginary switch mechanism will move the job being served in k' to k and move one job in k to k'. This is allowed since both k and k' contain at least one job due to the fact that a job being served in k' and the bottleneck k must be non-empty. □

Similar to the proof of Theorem 1, we first analyze the internal transmissions and then the external arrivals.

Internal transmissions

In the proof of Theorem 1, we have already discussed the case where internal transmissions between subservers are on the same route. However, unlike the JSR policy, the JSQ-AS policy allows internal transmissions between subservers on different routes. Hence, we also need to consider the internal transmission from subserver k to subserver j where $r_k \neq r_j$.

The definition of dominance ensures that if k is non-dominant, so is s_k. According to the routing policy, $x_{s_k} \geq x_j$. Let ℓ be the first non-dominant subserver on route r_k and b be the bottleneck on route r_j. If j is non-dominant, then by (2.5), we have $x_\ell \geq x_{s_k} \geq x_j \geq x_b$. Now from (2.7) we can obtain

$$\frac{1+(|K^X|-1)\delta}{|K^X|}\sum_{k\in K^X} x_k < \frac{1+|K^X|\delta}{|K^X|+1}\Big(\sum_{k\in K^X} x_k + x_\ell\Big),$$

contradicting with the definition of dominant subservers.

Thus, it cannot be the case that k is non-dominant and j is dominant, which implies that any internal transmission does not positively contribute to the mean drift.

External arrivals

According to Lemma 4, if a non-dominant route exists, then the routing policy guarantees that an arriving job must be routed to the first subserver on a non-dominant route r; this leads to non-positive contribution to the mean drift. Otherwise, every route is dominant. Then for any $x \in \mathcal{Q}$ $(x \neq 0)$, the drift satisfies

$$\begin{aligned} D^X(x) &\overset{\text{Corollary 1}}{\leq} \frac{1+(|K^X|-1)\delta}{|K^X|}\Big(\lambda - \sum_{b\in\mathcal{B}^X} \mu_b(x)\zeta_b(x)\Big) \\ &\overset{(2.5)}{=} \frac{1+(|K^X|-1)\delta}{|K^X|}\Big(\lambda - \sum_{b\in\mathcal{B}^X} \mu_b(x)\Big) \\ &\overset{\text{Lemma 5}}{=} \frac{1+(|K^X|-1)\delta}{|K^X|}\Big(\lambda - \sum_{n:n=n_k,k\in\mathcal{B}^X} \bar{\mu}_n\Big) \\ &\overset{\text{Lemma 1}}{<} 0, \end{aligned}$$

which completes the proof. □

The JSQ-AS policy cannot be directly applied to multi-class network, because the imaginary switch mechanism may move a job to the subserver of a different class with a different destination. Although the imaginary service rate control in the JSR policy can be used for multiple classes, it needs global information such as the information of dominance and bottlenecks for the preemption, so it is not suitable for the decentralized setting. The design of a decentralized MDI control policy for multi-class network can be a future work.

2.6 Concluding remarks

This chapter studies the stability of open queueing networks under a class of model data-independent control policies. In addition, we derive an easy-to-use stability criterion based on route expansion of the network and explicit piecewise-linear test functions. With the stability criterion, we generalize the classical join-the-shortest-queue policy to ensure stability and attain maximum throughput under centralized/decentralized settings. Our analysis and design can also be applied to specific network control problems with stability issues.

For the future studies, one research direction can be the stabilizing decentralized MDI control on multi-class queueing network. The challenge lies in: 1) The imaginary service rate control used in the JSR approach needs global information (such as the information of dominance and bottlenecks) for the preemption, so it cannot be directly applied in the decentralized setting; 2) If the imaginary switch used in the JSQ-AS approach is applied to the multi-class network, then it may move a job to the subserver of a different class with a different destination. Thus, JSQ-AS is not suitable for multi-class network. However, we can still explore more clever approaches than the imaginary service rate control and the imaginary switch.

One application is the district routing, i.e., assigning routes to connected autonomous vehicles (CAV) in a district with the objective of minimizing the average travel time of CAVs. Each CAV has its own origin and destination. Each signal-free intersection can be viewed as a server. In the internet of vehicles, information (e.g., queueing length) can be shared among the intersections and vehicles. If there is a routing app or a system operator that knows the global information and uses it for the feedback control (routing, sequencing, and holding), we can consider it as a centralized setting. Otherwise, it is a decentralized setting. The queueing model can be used to simulate the driving environment and boost the reinforcement learning by replacing SUMO in the training part. This is a joint work with NYU ECE Highspeed Networking Lab.

Chapter 3

Resilience of Dynamic Routing with Sensing Faults

This chapter is a joint work with Li Jin. Published in conference version form at 2020 American Control Conference [87] and in report version form published at the NYU C2SMART Center [43].

Feedback dynamic routing is a commonly used control strategy in transportation systems. This class of control strategies relies on real-time information about the traffic state in each link. However, such information may not always be observable due to temporary sensing faults. In this chapter, we consider dynamic routing over two parallel links, where the sensing on each link is subject to recurrent and random faults. The faults occur and clear according to a finite-state Markov chain. When the sensing is faulty on a link, the traffic state on that link appears to be zero to the controller. Building on the theories of Markov processes and monotone dynamical systems, we derive lower and upper bounds for the resilience score, i.e., the guaranteed throughput of the network, in the face of sensing faults by establishing stability conditions for the network. We use these results to study how a variety of key parameters affect the resilience score of the network. The main conclusions are: (i) Sensing faults can reduce throughput and destabilize a nominally stable network; (ii) A higher failure rate does not necessarily reduce throughput, and there may exist a worst rate that minimizes throughput; (iii) Higher correlation between the failure probabilities of two links leads to greater throughput; (iv) A large difference in capacity between two links can result in a drop in throughput.

3.1 Introduction

The rapidly growing deployment of traffic sensing and vehicle-to-vehicle/infrastructure (V2V/ V2I) communications has enabled the concept of intelligent transportation system (ITS). In ITS, system operators and travelers have access to real-time traffic conditions and can thus make better decisions. Dynamic routing is a typical ITS capability, which is conducted via route guidance tools such as Google Maps and WAZE. System operators can also influence routing via tolling and instructions for traffic diversion, which also rely on real-time traffic conditions. A major challenge for dynamic routing in ITS is how to ensure system functionality and efficiency under a variety of sensing faults. Quality of sensing and communications significantly affects system performance. However, data health is a serious issue that system operators must face. On some highways, up to 30%-40% of loop sensors do not report accurate measurements [64, 82]; similar issue exists for camera sensors. Even though some routing guidance tools may have certain internal fault detection and correction actions, the benefits of such actions can be further evaluated. Moreover, without appropriate fault-tolerant mechanisms, feedback control algorithms may make decisions based on wrong information, and ITS may even perform worse than a comparable conventional transportation system. Therefore, ITS will not be well accepted by the public and transportation authorities unless the impact of sensing faults is adequately evaluated and addressed. However, such impact has not been well understood, and practically relevant fault-tolerant routing algorithms have not been developed.

In this chapter, we propose a novel model that synthesizes traffic flow dynamics and stochastic sensing faults. Based on this model, we evaluate the impact of faults on fault-unaware routing algorithm and derive practically relevant insights for designing fault-tolerant routing algorithms in ITS. We consider the routing problem over two parallel routes (links), as shown in Fig. 3.1 and Fig. 3.2.

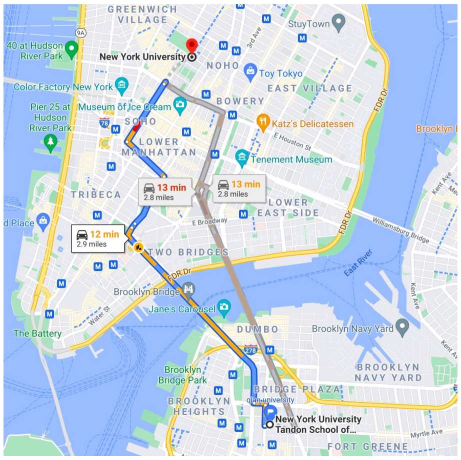


Figure 3.1: Selection over parallel routes

Our approach and results can be extended to more complex networks and a broader class of ITS control capabilities, such as ramp metering and speed limit control. We consider a stochastic model, since in practice it is not easy to deterministically predict when and where a sensing fault will occur. We show that this model leads to tractable analysis and insightful results for fault-tolerant design of ITS. We study the stability and guaranteed throughput of the network, which we consider as the resilience score. We also establish the link between the resilience score and key model parameters, including the number of fault-prone links and the average frequency and duration of faults.

Existing model-based traffic management approaches typically assume complete knowledge of the traffic condition [13,32,65,90], but feedback traffic management for ITS in the face of sensing faults has not been well studied. Como et al. [12] studied the resilience of distributed routing in the face of physical disruptions to link capacities in a dynamic flow network. Lygeros et al. [52] proposed a conceptual framework for fault-tolerant traffic management, but the concrete algorithms are still yet to be developed. A body of work on fault-tolerant control has been developed for a class of dynamical systems [7,62,93]. However, very limited results are available for recurrent and random faults. In addition, there exist some results on adaptive/learning-based fault-tolerant control with applications in electrical/mechanical/aerospace engineering [57,78,92], but these results are not directly applicable to ITS, nor do they explicitly consider stochastic sensing faults.

Our modeling approach is innovative in that we model the occurrence and clearance of sensing faults as a finite-state, continuous-time Markov process. If the sensing on a link is normal, travelers know the true traffic state (traffic density) on the link. If the sensing is faulty, the traffic state will appear to be zero to the travelers. Besides such denial-of-service, our modeling approach can also be extended to incorporate other forms of sensing faults, such as bias and distortion. We adopt the classical logit model [3] for routing; the essential principle of this model is that more traffic will go to a less congested link. When the sensing on a link is faulty, travelers may mistakenly consider a congested link to be uncongested. We show that such faulty information may affect the network's throughput. The discrete states of the Markov process are essentially modes for the flow dynamics, which govern the evolution of the continuous states. Hence, our model belongs to a class of stochastic processes called piecewise-deterministic Markov processes [4,17]. Similar models have been used for demand/capacity fluctuations [42,44]; this chapter extends the modeling approach to sensing faults.

A key step for resilience analysis is to determine the stability of the traffic densities under various combinations of parameters. We study the stability of the network based on the theory of continuous-time Markov processes [56]. We derive a necessary condition for stability by constructing a positively invariant set for the dynamic flow network. We derive a sufficient condition by considering a quadratic, switched Lyapunov function that verifies the Foster-Lyapunov drift condition. We exploit a special property of the flow dynamics, called cooperative dynamics [39,74], to derive an easy-to-check stability criterion, which states that the network is stable if there exists a queueing state such that the rate of change of the fastest growing queue averaged over the modes is negative.

Based on the stability analysis, we analyze the network's throughput (resilience score). We define throughput as the maximal inflow that the network can take while maintaining stable. As a baseline, we first study the behavior of the network if both links have the same flow functions. We perturb the baseline in multiple dimensions (probability and correlation of sensing faults on two links) and analyze how throughput can be affected. We also show that throughput reduces as the two link's asymmetry increases.

The main contributions of this chapter include (i) a novel stochastic model for sensing fault-prone transportation networks, (ii) easy-to-check stability conditions for the network, and (iii) resilience analysis under various settings. The rest of this chapter is organized as follows. In Section 3.2, we introduce the dynamic flow model with sensing faults. In Section 3.3, we establish the stability conditions. In Section 3.4, we study the resilience score under various scenarios. In Section 3.5, we summarize the conclusions and mention several future directions.

3.2 Dynamic flow model with sensing faults

Consider the two-link network in Fig. 3.2. Let $U_k(t)$ be the flow into link $k \in \{1, 2\}$ and $X_k(t)$ be the traffic density of link k at time t. The capacity of link k is $F_k \in [0, 1]$ where $F_1 + F_2 = 1$. The flow out of link k is $f_k(X_k(t))$, which is specified by the flow function

$$f_k(x_k) = F_k(1 - e^{-x_k}), \quad k = 1, 2. \tag{3.1}$$

The source node is subject to a constant demand $\eta \geq 0$, which is considered as a model parameter rather than a state or input variable in the subsequent analysis.

Travelers can observe the state $X(t)$. However, the observation is not always accurate. We consider the sensing on each link to be stochastically switching between a “good” and a “bad” mode. That is, we consider a set $\mathcal{S} = \{1, 2, 3, 4\}$ of *sensing fault modes*. The network switches between the two modes according to the Markov chain in Fig. 3.2. Each mode $s \in \mathcal{S}$ is characterized by a *fault mapping* $T_s : \mathbb{R}^2_{\geq 0} \to \mathbb{R}^2_{\geq 0}$

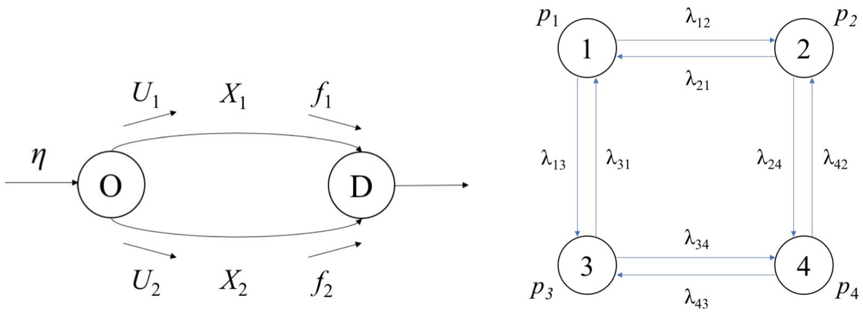


Figure 3.2: The two-link network and the Markov chain representing network switches among the sensing fault modes.

$$T_1(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \; T_2(x) = \begin{bmatrix} 0 \\ x_2 \end{bmatrix}, \; T_3(x) = \begin{bmatrix} x_1 \\ 0 \end{bmatrix}, \; T_4(x) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{3.2}$$

In mode s, the observed state is

$$\hat{x} = T_s(x).$$

At the source node, the demand η is distributed to each link according to a *routing policy* $\mu : \mathbb{R}^2_{\geq 0} \to \mathbb{R}^2_{\geq 0}$, which specifies the fraction of inflow that goes to each link according to the logit model

$$\mu_k(x) = \frac{e^{-\beta \hat{x}_k}}{\sum_{j=1}^{2} e^{-\beta \hat{x}_j}}, \quad k = 1, 2. \tag{3.3}$$

Note that the routing is based on the observed state rather than the true state.

For notational convenience, with a slight abuse of notation, we write

$$\mu(s, x) = \mu(T_s(x)). \tag{3.4}$$

That is, the routing policy can be viewed as a switching function $\mu : \mathcal{S} \times \mathbb{R}^2_{\geq 0} \to [0, 1]^2$ with a discrete argument $s \in \mathcal{S}$ and a continuous argument $x \in \mathbb{R}^2_{\geq 0}$. Finally, we emphasize that we consider η as a model parameter rather than a state or input variable in the subsequent analysis.

Then, we define the dynamics of the hybrid-state process $\{(S(t), X(t)); t > 0\}$ as follows. The discrete-state process $\{S(t); t > 0\}$ of the mode is a time-invariant finite-state Markov process that is independent of the continuous-state process $\{X(t); t > 0\}$ of the traffic densities. The state space of the finite-state Markov process is $\mathcal{S}$. The *transition rate* from mode s to mode s' is $\lambda_{s,s'}$. Without loss of generality, we assume that $\lambda_{s,s} = 0$ for all $s \in \mathcal{S}$ [76]. Hence, the discrete-state process evolves as follows:

$$\Pr\{S(t + \delta) = s' | S(t) = s\} = \lambda_{s,s'}\delta + \mathrm{o}(\delta), \quad \forall s' \neq s,\ \forall s \in \mathcal{S}.$$

where δ denotes infinitesimal time. We assume that the discrete-state process is ergodic [30] and admits a unique steady-state probability distribution $\{p_s; s \in \mathcal{S}\}$ satisfying

$$p_s \sum_{s' \neq s} \lambda_{s,s'} = \sum_{s' \neq s} p_{s'} \lambda_{s',s}, \quad \forall s \in \mathcal{S}, \tag{3.5a}$$

$$p_s \geq 0, \quad \forall s \in \mathcal{S}, \tag{3.5b}$$

$$\sum_{s \in \mathcal{S}} p_s = 1. \tag{3.5c}$$

The continuous-state process $\{X(t); t > 0\}$ is defined as follows. For any initial condition $S(0) = s$ and $X(0) = x$,

$$\frac{d}{dt} X_k(t) = \eta \mu_k\Big(S(t), X(t)\Big) - f_k\Big(X(t)\Big), \quad t \geq 0,\ k = 1, 2. \tag{3.6}$$

Note that the routing policy μ defined in (3.3)-(3.4) and the flow function f defined in (3.1) ensure that $X(t)$ is continuous in t. We can define the flow dynamics with a vector field $G : \mathcal{S} \times \mathbb{R}^2_{\geq 0} \to \mathbb{R}^2$ as follows:

$$G(s, x) := \eta \mu(s, x) - f(x). \tag{3.7}$$

The joint evolution of $S(t)$ and $X(t)$ is in fact a piecewise-deterministic Markov process and can be described compactly using an infinitesimal generator [4, 17]

$$\mathcal{L} g(s, x) = \Big(\eta \mu(s, x) - f(x)\Big)^T \nabla_x g(s, x) + \sum_{s' \in \mathcal{S}} \lambda_{s,s'}(g(s', x) - g(s, x)).$$

for any differentiable function g.

The network is stable if there exists $Z < \infty$ such that for any initial condition $(s, x) \in \mathcal{S} \times \mathbb{R}^2_{\geq 0}$

$$\limsup_{t\to\infty} \frac{1}{t} \int_{r=0}^{t} \mathrm{E}[|X(r)|]dr \leq Z. \tag{3.8}$$

This notion of stability follows a classical definition [16], some authors name it as “first-moment stable” [73]. The rest of this paper is devoted to establishing and analyzing the relation between the stability of the continuous-state process $\{X(t); t > 0\}$ and the demand η.

3.3 Stability analysis

The main result of this section is as follows.

Theorem 3. *Consider two parallel links with sensors switching between two modes as defined in section 3.2.*

1. *A necessary condition for stability is that*

$$\eta\Big(\frac{1}{e^{-\beta\underline{x}_2}+1}p_2+\frac{1}{2}p_4\Big)\leq F_1, \tag{3.9a}$$

$$\eta\Big(\frac{1}{e^{-\beta\underline{x}_1}+1}p_3+\frac{1}{2}p_4\Big)\leq F_2, \tag{3.9b}$$

$$\eta<1. \tag{3.9c}$$

where $\underline{x}_k$ is the solution to

$$\eta\frac{e^{-\beta\underline{x}_k}}{1+e^{-\beta\underline{x}_k}}=F_k(1-e^{-\underline{x}_k})$$

for $k=1,2$.

2. *A sufficient condition for stability is that there exists $\theta\in\mathbb{R}^2_{\geq0}$ such that*

$$\sum_{s=1}^{4}p_s\max_{k\in\{1,2\}}\Big\{\eta\frac{e^{-\beta T_{s,k}(\theta_k)}}{e^{-\beta T_{s,k}(\theta_2)}+e^{-\beta T_{s,k}(\theta_1)}}-F_k(1-e^{-\theta_k})\Big\}<0 \tag{3.10}$$

The rest of this section is devoted to the proof of the above result.

3.3.1 Proof of necessary condition

An apparent necessary condition for stability is

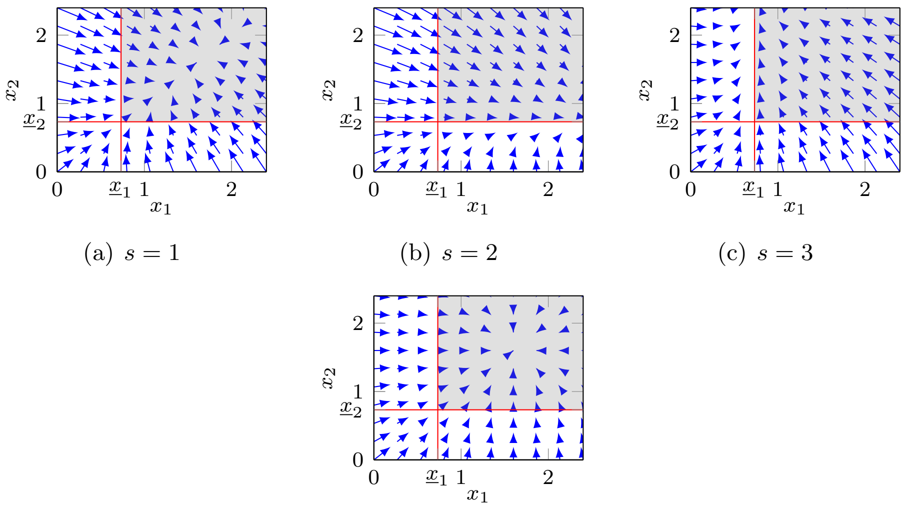
$$\eta<1. \tag{3.11}$$

If this does not hold, then the network is unstable even in the absence of sensing faults [41].

First, an invariant set of the process $\{X(t);t>0\}$ is $\mathcal{M}=[\underline{x}_1,\infty)\times[\underline{x}_2,\infty)$. To see this, note that for any $s\in\mathcal{S}$ and for any (x_1,x_2) such that $(x_1,x_2)\notin\mathcal{M}$, the vector G of time derivatives of the traffic densities has a non-zero component that points to the interior of the invariant set $\mathcal{M}$; see Figure 3.3.

Second, by ergodicity of the process $\{(S(t),X(t));t>0\}$ where $X(t)=\begin{bmatrix}X_1(t)\\X_2(t)\end{bmatrix}$, we have for $k\in\{1,2\}$,

$$X_k(t)=X_k(0)+\int_0^t\Big(u_k(\tau)-f_k(\tau)\Big)d\tau,$$



(a) $s = 1$

(b) $s = 2$

(c) $s = 3$

(d) $s = 4$

Figure 3.3: Illustration of the continuous state process and the invariant set $\mathcal{M}$. The arrows represent the vector field G defined in (3.7) for the four states.

where $u_k(\tau)$ and $f_k(\tau)$ are inflow and outflow of link k at time τ. Since $\lim_{t\to\infty}\frac{1}{t}X_k(0) = 0$ and $\lim_{t\to\infty}\frac{1}{t}X_k(t) = 0$ a.s., then

$$0 = \lim_{t\to\infty}\frac{1}{t}\left(\int_0^t \Big(u_k(\tau) - f_k(\tau)\Big)d\tau + X_k(0) - X_k(t)\right) = \lim_{t\to\infty}\frac{1}{t}\int_0^t \Big(u_k(\tau) - f_k(\tau)\Big)d\tau \quad \text{a.s.}$$

Note that $f_k(\tau) \le F_k$ for any $\tau \ge 0$ and $k \in \{1, 2\}$, hence

$$\lim_{t\to\infty}\frac{1}{t}\int_0^t u_k(\tau)d\tau = \lim_{t\to\infty}\frac{1}{t}\int_0^t f_k(\tau)d\tau \le \lim_{t\to\infty}\frac{1}{t}\int_0^t F_k d\tau = F_k. \tag{3.12}$$

According to the definition of steady-state probability,

$$\lim_{t\to\infty}\frac{1}{t}\int_0^t \mathbb{I}_{S(\tau)=s}d\tau = p_s, \quad \text{a.s.} \quad \forall s \in \mathcal{S}.$$

Combining with (3.12), we obtain

$$
\begin{aligned}
F_1 \geq& \lim_{t\to\infty}\frac{1}{t}\int_0^t u_1(\tau)d\tau = \lim_{t\to\infty}\frac{1}{t}\int_0^t \eta\mu_1(S(\tau),X(\tau))d\tau\\
=&\eta\lim_{t\to\infty}\frac{1}{t}\sum_{s=1}^{4}\int_0^t \mathbb{I}_{S(\tau)=s}\mu_1(S(\tau),X(\tau))d\tau\\
\geq&\eta\lim_{t\to\infty}\frac{1}{t}\Big(\int_0^t \mathbb{I}_{S(\tau)=1}0d\tau + \int_0^t \mathbb{I}_{S(\tau)=2}\frac{1}{1+e^{-\beta\underline{x_2}}}d\tau + \int_0^t \mathbb{I}_{S(\tau)=3}0d\tau + \int_0^t \mathbb{I}_{S(\tau)=4}\frac{1}{2}d\tau\Big)\\
=&\eta\Big(\frac{1}{1+e^{-\beta\underline{x_2}}}\lim_{t\to\infty}\frac{1}{t}\int_0^t \mathbb{I}_{S(\tau)=2}d\tau + \frac{1}{2}\lim_{t\to\infty}\frac{1}{t}\int_0^t \mathbb{I}_{S(\tau)=4}d\tau\Big)\\
=&\eta\Big(\frac{p_2}{1+e^{-\beta\underline{x_2}}} + \frac{p_4}{2}\Big),
\end{aligned}
$$

which gives (3.9a). We can prove (3.9b) in a similar way.

3.3.2 Proof of sufficient condition

Suppose that there exists a vector $\theta \in \mathbb{R}^2_{\geq 0}$ satisfying (3.10). Then, for the hybrid process $\{(S(t),X(t));t>0\}$, consider the Lyapunov function

$$V(s,x) = \frac{1}{2}\Big((x_1-\theta_1)_+ + (x_2-\theta_2)_+\Big)^2 + a_s\Big((x_1-\theta_1)_+ + (x_2-\theta_2)_+\Big) \tag{3.13}$$

where $(x_k-\theta_k)_+ = \max\{0, x_k-\theta_k\}$, $k=1,2$, and the coefficients a_s are given by

$$[a_1,a_2,a_3,a_4]^T = \begin{bmatrix} -\sum\limits_{i\neq 1}\lambda_{1i} & \lambda_{12} & \lambda_{13} & \lambda_{14}\\ \lambda_{21} & -\sum\limits_{i\neq 2}\lambda_{2i} & \lambda_{23} & \lambda_{24}\\ \lambda_{31} & \lambda_{32} & -\sum\limits_{i\neq 3}\lambda_{3i} & \lambda_{34}\\ 1 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \bar{G}-G(1,\theta)\\ \bar{G}-G(2,\theta)\\ \bar{G}-G(3,\theta)\\ 1 \end{bmatrix}$$

where G is defined in (3.7) and $\bar{G} = \sum_{s\in\mathcal{S}} p_s G(s,\theta)$. Based on the ergodicity assumption of the mode switching process, the matrix in the above must be invertible. This Lyapunov function is valid in that $V(s,x)\to\infty$ as $|x|\to\infty$ for all s. Define

$$\mathscr{D}_s = \max_{k\in\{1,2\}}\Big(\mu_k(s,\theta) - f_k(\theta_k)\Big), \quad s\in\mathcal{S}. \tag{3.14}$$

The Lyapunov function V essentially penalizes the quantity $(x-\theta)_+$, which can be viewed as a "derived state". Apparently, boundedness of $X(t)$ is equivalent to the boundedness of $(X(t)-\theta)_+$ Note that the dynamic equation of the derived state $(x-\theta)_+$ is slightly different from that of x:

$$\frac{d}{dt}(X_k(t)-\theta_k)_+ = D_k(S(t),X(t)) := \begin{cases} \mu_k(S(t),X(t)) - f_k(X(t) & X_k(t) > \theta_k, \\ (\mu_k(S(t),X(t)) - f_k(X(t))_+ & X_k(t) = \theta_k, \\ 0 & \text{otherwise,} \end{cases} \quad k = 1,2.$$

Applying the infinitesimal generator to the Lyapunov function, we obtain

$$\begin{aligned} \mathcal{L}V(s,x) =& \sum_{k=1}^{2}\sum_{j=1}^{2} D_j(s,x)(x_k-\theta_k)_+ + \sum_{s'\neq s}\Big(\lambda_{s,s'}(a_{s'}-a_s)\sum_{k=1}^{2}(x_k-\theta_k)_+\Big) + \sum_{k=1}^{2} a_{s,k}D_k(s,x) \\ =& \Big(\sum_{k=1}^{2} D_k(s,x) + \sum_{s'\neq s}\lambda_{s,s'}(a_{s'}-a_s)\Big)|(x_k-\theta_k)_+| + \sum_{k=1}^{2} a_{s,k}D_k(s,x) \end{aligned} \tag{3.15}$$

This proof establishes the stability of the process $\{(S(t),X(t)); t > 0\}$ by verifying that the Lyapunov function V as defined above satisfies the Foster-Lyapunov drift condition for stability [56]:

$$\mathcal{L}V(s,x) \le -c|x| + d \quad \forall (s,x) \in \mathcal{S} \times \mathbb{R}^2_{\ge 0} \tag{3.16}$$

for some $c > 0$ and $d < \infty$, where $|x|$ is the one-norm of x; this condition will imply (3.8). To proceed, we partition $\mathbb{R}^2_{\ge 0}$, the space of x, into two subsets:

$$\mathcal{X}_0 = \{x : 0 \le x \le \theta\},\ \mathcal{X}_1 = \mathcal{X}_0^C;$$

that is, $\mathcal{X}_0$ and $\mathcal{X}_1$ are the complement to each other in the space $\mathbb{R}^2_{\ge 0}$. In the rest of this proof, we first verify (3.16) over $\mathcal{X}_0$ and then over $\mathcal{X}_1$.

To verify (3.16) over $\mathcal{X}_0$, note that μ and f are bounded functions, so, for any $a_{s,k}$, there exists $d < \infty$ such that

$$d_1 \ge a_s \sum_{k=1}^{2} D_k(s,x) \quad \forall (s,x) \in \mathcal{S} \times \mathbb{R}^2_{\ge 0}. \tag{3.17}$$

In addition, $(x_k - \theta_k)_+ = 0$, $k = 1, 2, \ldots, K$ for all $x \in \mathcal{X}_0$; this and (3.15) imply

$$\mathcal{L}V(s,x) \le d_1. \tag{3.18}$$

Furthermore, for any $c > 0$, there exists $d_2 = c|\theta|$ such that $d_2 \ge c|x|$ for all $x \in \mathcal{X}_0$. Hence, letting $d = d_1 + d_2$, we have

$$\mathcal{L}V(s,x) \le -c|x| + d \quad \forall (s,x) \in \mathcal{S} \times \mathcal{X}_0. \tag{3.19}$$

To verify (3.16) over $\mathcal{X}_1$, we further decompose $\mathcal{X}_1$ into the following subsets:

$$\begin{aligned}
\mathcal{X}_1^1 &= \{x \in \mathcal{X}_1 : x_1 \geq \theta_1, x_2 < \theta_2\},\\
\mathcal{X}_1^2 &= \{x \in \mathcal{X}_1 : x_1 < \theta_1, x_2 \geq \theta_2\},\\
\mathcal{X}_1^3 &= \{x \in \mathcal{X}_1 : x_1 \geq \theta_1, x_2 \geq \theta_2\}.
\end{aligned}$$

For each $x \in \mathcal{X}_1^1$, we have

$$\begin{aligned}
\mathcal{L}V(s,x) &= \Big(D_1(s,x) + \sum_{s' \neq s} \lambda_{s,s'}(a_{s'} - a_s)\Big)|(x-\theta)_+| + a_s \sum_{k=1}^{2} D_k(s,x)\\
&\leq \left(\Big(\mu_1(s,x) - f_1(x_1)\Big) + \sum_{s' \neq s} \lambda_{s,s'}(a_{s'} - a_s)\right)|(x-\theta)_+| + d_1\\
&\leq \Big(\mathscr{D}_s + \sum_{s' \neq s} \lambda_{s,s'}(a_{s'} - a_s)\Big)|(x-\theta)_+| + d_1 \qquad (3.20)
\end{aligned}$$

From the definition of a_s, we have

$$\mathscr{D}_s + \sum_{s' \neq s} \lambda_{s,s'}(a_{s'} - a_s) = \frac{1}{4} \sum_{s' \in \mathcal{S}} p_{s'} \mathscr{D}_{s'}$$

The above and (3.20) imply

$$\mathcal{L}V(s,x) \leq \frac{1}{4}\left(\sum_{s' \in \mathcal{S}} p_{s'} \mathscr{D}_{s'}\right)|x| + d, \quad x \in \mathcal{X}_1^1.$$

Let $c := -\frac{1}{4}\sum_{s' \in \mathcal{S}} p_{s'} \mathscr{D}_{s'}$. From (3.10), we have $c > 0$. Hence, we have

$$\mathcal{L}V(s,x) \leq -c|x| + d, \quad \forall (s,x) \in \mathcal{X}_1^1.$$

Analogously, we can show

$$\mathcal{L}V(s,x) \leq -c|x| + d, \quad \forall (s,x) \in \mathcal{X}_1^2 \cup \mathcal{X}_1^3,$$

and hence

$$\mathcal{L}V(s,x) \leq -c|x| + d, \quad \forall (s,x) \in \mathcal{X}_1,$$

The above and (3.19) imply the drift condition (3.16), which completes the proof.

3.4 Resilience analysis

In this section, we study the resilience score, i.e. the guaranteed throughput (the supremum of η that maintains stability), under various scenarios. We first consider two symmetric links and focus on the impact of transition rates of the discrete state (Section 3.4.1). Then, we study how the throughput varies with the asymmetry of the links (Section 3.4.2).

3.4.1 Impact of transition rates

If the two links are homogeneous in the sense that they have same flow functions $f_1 = f_2$, we have the main result of this section as follows:

Proposition 2. *For the homogeneous network, the resilience score η^*, i.e. the guaranteed throughput has a lower bound of*

$$\eta^* \geq \frac{1}{1+p_2+p_3}. \tag{3.21}$$

Proof: The lower bound results from the sufficient condition in Theorem 3.

The homogeneity implies that $F_1 = F_2 = 1/2$ and $\theta_1 = \theta_2$. Now (3.10) means that there exists $\theta_1 \in \mathbb{R}_{\geq 0}$ such that

$$\Big(\frac{1}{2}(p_1+p_4) + \frac{1}{1+e^{-\beta\theta_1}}(p_2+p_3)\Big)\eta < \frac{1}{2}(1-e^{-\theta_1}),$$

that is,

$$\Big(1 + \frac{1-e^{-\beta\theta_1}}{1+e^{-\beta\theta_1}}(p_2+p_3)\Big)\eta < 1-e^{-\theta_1}. \tag{3.22}$$

Let $z = e^{-\theta_1} \in (0,1]$, then (3.22) can be expressed as there exists $z \in (0,1]$ such that

$$\Big(1 + \frac{1-z^\beta}{1+z^\beta}(p_2+p_3)\Big)\eta < 1-z,$$

that is,

$$z^{\beta+1} - \Big(1-(1-p_2-p_3)\eta\Big)z^\beta + z - \Big(1-(1+p_2+p_3)\eta\Big) < 0. \tag{3.23}$$

Let $g(z)$ be the left-hand side of (3.23). Since $g(z)$ is monotonically increasing on (0,1] (proof is provided in Appendices), η should satisfy

$$g(0) = (1+p_2+p_3)\eta - 1 < 0,$$

or

$$\eta < \frac{1}{1+p_2+p_3},$$

which gives the lower bound.

□

Table 3.1: Nominal model parameters.

Parameter	Notation	Nominal value
Link 1 capacity	F_1	0.5
Link 2 capacity	F_2	0.5
Routing sensitivity to congestion	β	1

Next, we discuss how characteristics of link failures (specifically, link failure rate and link failure correlation) affect the resilience score. Table 3.1 lists the nominal values considered in this subsection.

Link failure rate: Suppose that the health of each link is independent of the other link. Furthermore, suppose that the failure rates of both links are identical, denoted as p, then

$$\begin{aligned} p_2 + p_4 &= p = p_3 + p_4, \\ \underline{\eta}^* &= \frac{1}{1+p_2+p_3} = \frac{1}{1+2p(1-p)}. \end{aligned}$$

When the link failure rate is either 0 or 1, the two-link network becomes open-loop, the lower bound can naturally be 1. The lower bound reaches minimum when the link failure rate is 0.5; see Figure 3.4.

Link failure correlation: Suppose that the health of each link is correlated with the other link while the failure rates of both links are still identical. Denote the correlation as ρ, then

$$\begin{aligned} \rho &= \frac{p_4 - (p_2+p_4)(p_3+p_4)}{\sqrt{p_2 p_3}} = \frac{p - p_2 - p^2}{p}, \\ \underline{\eta}^* &= \frac{1}{1+p_2+p_3} = \frac{1}{1+2p(1-p-\rho)}. \end{aligned}$$

As the link failure correlation increases from $-p$ to $1-p$, the lower bound increases from $\frac{1}{1+2p}$ to 1. When the failure of the two links are strongly (positively) correlated, the two-link network also turns to be open-loop and hence the lower bound reaches 1; see Figure 3.4.

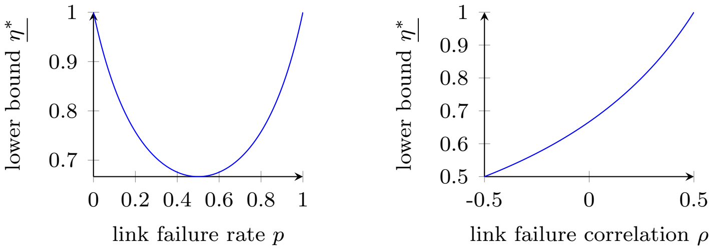


Figure 3.4: Impact of link failure probability ($\rho = 0$) and link failure correlation ($p = 0.5$) on the lower bound of resilience score

3.4.2 Impact of heterogeneous link capacities

Now we relax the assumption of symmetric links and allow $F_1 \neq F_2$. Without loss of generality, we assume that $F_1 \geq F_2$. Instead, we will consider symmetric failure rate, i.e. $p_2 = p_3$. The following result links the resilience score to $|F_1 - F_2|$, which quantifies the asymmetry of links:

Proposition 3. *Suppose that $p_2 = p_3$ and $F_1 \geq F_2$. Then, the resilience score has a lower bound of*

$$\eta^* \geq \min\left\{\frac{1-(F_1-F_2)}{1-p_1}, \frac{1-p_4(F_1-F_2)}{1+2p_2}\right\}. \tag{3.24}$$

Proof: Let $y = e^{-\theta_1}$, $z = e^{-\theta_2}$, $\rho = \frac{y^\beta - z^\beta}{y^\beta + z^\beta}$. (3.10) implies that there exists $y, z \in (0, 1]$ such that

$$\begin{aligned} &p_1 \max\left\{\frac{\eta y^\beta}{y^\beta + z^\beta} - F_1(1-y), \frac{\eta z^\beta}{y^\beta + z^\beta} - F_2(1-z)\right\} \\ &+p_2 \max\left\{\frac{\eta}{1+z^\beta} - F_1(1-y), \frac{\eta z^\beta}{1+z^\beta} - F_2(1-z)\right\} \\ &+p_3 \max\left\{\frac{\eta y^\beta}{y^\beta+1} - F_1(1-y), \frac{\eta}{y^\beta+1} - F_2(1-z)\right\} \\ &+p_4 \max\left\{\frac{\eta}{2} - F_1(1-y), \frac{\eta}{2} - F_2(1-z)\right\} \leq 0 \end{aligned} \tag{3.25}$$

If $\frac{1}{2-p_1} < F_1 - F_2 \leq 1$, when

$$\eta \leq \frac{1-(F_1-F_2)}{1-p_1},$$

there exists $y \leq 1 - \frac{\eta + F_2}{F_1}$ such that (3.25) holds.

If $0 \leq F_1 - F_2 \leq \frac{1}{2-p_1}$, when

$$F_1 - F_2 \leq \eta \leq \frac{1-(1-p_1-2p_2)(F_1-F_2)}{1+2p_2},$$

there exists y, z satisfying $\rho(F_1 - F_2) \geq F_1(1-y) - F_2(1-z) \geq 0$ and $\rho < \frac{F_1 - F_2}{\eta}$ such that (3.25) holds and when

$$\eta < F_1 - F_2,$$

there exists $y \leq 1 - \frac{\eta + F_2}{F_1}$ such that (3.25) holds.

Therefore,

$$\eta^* \geq \begin{cases} \frac{1-(F_1-F_2)}{1-p_1}, & \frac{1}{2-p_1} < F_1 - F_2 \leq 1 \\ \frac{1-(1-p_1-2p_2)(F_1-F_2)}{1+2p_2}, & 0 \leq F_1 - F_2 \leq \frac{1}{2-p_1} \end{cases}$$

The details of the proof are provided in Appendices.

□

Now we are ready to discuss how link capacity difference affects the resilience score.

When $F_1 = F_2$, the lower bound is $\frac{1}{1+2p_2}$, in consistence with our lower bound in subsection 3.4.1, and the upper bound is 1 (note that when $\sqrt{2}\max\{p_2, p_3\} + p_4 \leq 1$, we can derive

$$\eta < 1$$

from the necessary condition).

As $F_1 - F_2$ increases, the lower bound gradually drops and after certain point, it drops faster to 0 while the upper bound remains 1 for a while and then drops to 0. It means that when the difference between two link capacities gets larger, one link starts getting more congested than the other, then the system can be less stable.

When $F_1 \to 1$, $F_2 \to 0$, the network has weak resilience to the sensing faults and the resilience score tends to be zero.

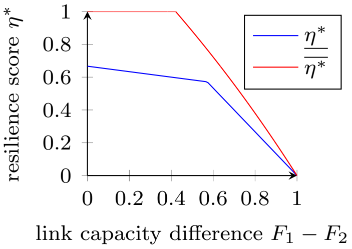


Figure 3.5: Impact of link capacity difference on the lower and upper bound of resilience score ($p_1 = p_2 = p_3 = p_4 = 1/4$)

3.5 Concluding remarks

In this chapter, we propose a two-link dynamic flow model with sensing faults to study the stability conditions and guaranteed throughput of the network. Based on this model, we are able to derive lower and upper bounds of the resilience score and analyze the impact of transition rates and heterogeneous link capacities on them.

This work can be extended in several directions [79]. First, we can consider a more general network, say single-origin-single-destination acyclic network with a time-invariant inflow at the origin, rather than a simple two parallel link network. Second, various forms of flow functions that are relevant to different applications including road traffic, production line, and data packets can be assumed in the model. Third, the logit model (dynamic routing) can be replaced with other control laws such as ramp metering and max-pressure control. Last, several types of fault modes that capture cyber-physical disruptions (e.g. accidents are physical disruptions) can also be discussed.

The results can be validated using real-world highway traffic data. For example, we can conduct a simulation of sensing faults and feedback ramp control for traffic flow on Interstate I210 near Los Angeles using PeMS data [40].

This work also provides an implementable approach to designing resilient smart highway systems with fault-tolerant traffic control. Specific use cases include route guidance and ramp control.

Appendices

The monotonicity of $g(z)$ in subsection 3.4.1

The first derivative and the second derivative of $g(z)$ are

$$g'(z) = (\beta+1)z^{\beta} - \Big(1-(1-p_2-p_3)\eta\Big)\beta z^{\beta-1} + 1,$$

$$g''(z) = \beta z^{\beta-2}h(z),$$

where $h(z) = (\beta+1)z - \Big(1-(1-p_2-p_3)\eta\Big)(\beta-1)$.

If $0<\beta\le 1$, then $h(z) > h(0) = \Big(1-(1-p_2-p_3)\eta\Big)(1-\beta) \ge 0$, $g''(z) = \beta z^{\beta-2}h(z) > 0$. Hence $g'(z)$ is monotonically increasing on (0,1]. Since $g'(z) > g'(0) = 1$, $g(z)$ is also monotonically increasing on (0,1].

If $\beta > 1$, let $z_0 = \Big(1-(1-p_2-p_3)\eta)\Big)\frac{\beta-1}{\beta+1}$, then $h(z) < 0$ on $(0, z_0)$ and $h(z) > 0$ on $(z_0, 1]$. Since $g''(z)$ has same sign as $h(z)$, $g'(z) \ge g'(z_0) = 1 > 0$. Therefore, $g(z)$ is monotonically increasing on (0,1].

Detailed proof of Proposition 3

If $\frac{1}{2-p_1} < F_1 - F_2 \le 1$, then assume

$$\eta \le \frac{1-(F_1-F_2)}{1-p_1},$$

let $y \le 1 - \frac{\eta+F_2}{F_1}$ (note that $\eta \le \frac{1-(F_1-F_2)}{1-p_1} < F_1 - F_2$ means y exists), we have

$$\frac{\eta(1-z^{\beta})}{1+z^{\beta}} < \eta \le F_1(1-y) - F_2 < F_1(1-y) - F_2(1-z).$$

Now (3.25) can be expressed as

$$\begin{aligned} &p_1\Big(\frac{\eta z^{\beta}}{y^{\beta}+z^{\beta}} - F_2(1-z)\Big) + p_2\Big(\frac{\eta z^{\beta}}{1+z^{\beta}} - F_2(1-z)\Big) + \\ &p_3\Big(\frac{\eta}{y^{\beta}+1} - F_2(1-z)\Big) + p_4\Big(\frac{\eta}{2} - F_2(1-z)\Big) \le 0, \end{aligned} \tag{3.26}$$

that is,

$$\frac{1}{2}\Big(1 - \frac{y^{\beta}-z^{\beta}}{y^{\beta}+z^{\beta}}p_1 + (\frac{1-y^{\beta}}{1+y^{\beta}} - \frac{1-z^{\beta}}{1+z^{\beta}})p_2\Big)\eta - F_2(1-z) \le 0.$$

Fix y and note that when $z = 0$,

$$\begin{aligned}\text{LHS} &= \frac{1}{2}\Big(1 - p_1 - \frac{2y^\beta}{1+y^\beta}\Big)\eta - F_2 \\ &< \frac{1}{2}(1-p_1)\eta - F_2 \\ &= \frac{1}{2} - \frac{1}{2}(F_1 - F_2) - F_2 \\ &= 0,\end{aligned}$$

then intermediate value theorem implies that there exists z such that LHS ≤ 0.

If $0 \leq F_1 - F_2 \leq \frac{1}{2-p_1}$, first assume

$$F_1 - F_2 \leq \eta \leq \frac{1-(1-p_1-2p_2)(F_1-F_2)}{1+2p_2},$$

let y, z satisfies $\rho(F_1 - F_2) \geq F_1(1-y) - F_2(1-z)$ (fix y and note that when $z = 0$, $\rho(F_1 - F_2) + F_2(1-z) = (F_1 - F_2) + F_2 = F_1 > F_1(1-y)$, intermediate value theorem implies that such z exists) and $F_1(1-y) - F_2(1-z) \geq 0$ (let $y \leq 1 - \frac{F_2}{F_1}(1-z)$) and $\rho < \frac{F_1-F_2}{\eta}$ (since $\frac{F_1-F_2}{\eta} \leq 1$, such ρ exists), we have

$$\rho\eta \geq \rho(F_1 - F_2) \geq F_1(1-y) - F_2(1-z).$$

Now (3.25) can be expressed as

$$\begin{aligned} p_1\Big(\frac{\eta y^\beta}{y^\beta + z^\beta} - F_1(1-y)\Big) + p_2\Big(\frac{\eta}{1+z^\beta} - F_1(1-y)\Big) + \\ p_3\Big(\frac{\eta}{y^\beta + 1} - F_2(1-z)\Big) + p_4\Big(\frac{\eta}{2} - F_2(1-z)\Big) \leq 0,\end{aligned}$$

that is,

$$\frac{1}{2}\Big(1 + \rho p_1 + (\frac{1-z^\beta}{1+z^\beta} + \frac{1-y^\beta}{1+y^\beta})p_2\Big)\eta - (p_1+p_2)F_1(1-y) - (1-p_1-p_2)F_2(1-z) \leq 0.$$

Fix ρ and note that when $z = 0$ (and hence $y = 0$),

$$\begin{aligned}\text{LHS} =&\frac{1}{2}\rho\eta p_1 + \frac{1}{2}(1+2p_2)\eta - (p_1+p_2)(F_1 - F_2) - F_2 \\ <&\frac{1}{2}(F_1 - F_2)p_1 + \frac{1}{2}\Big(1 - (1-p_1-2p_2)(F_1-F_2)\Big) - (p_1+p_2)(F_1-F_2) - F_2 \\ =&0,\end{aligned}$$

then intermediate value theorem implies that there exists z (and y) such that LHS ≤ 0.

Next assume $\eta < F_1 - F_2$, let $y \leq 1 - \frac{\eta + F_2}{F_1}$ (note that $\eta < F_1 - F_2$ means such y exists), thus (3.25) can also be expressed as (3.26). Note that $\eta < F_1 - F_2 \leq \frac{1-(F_1-F_2)}{1-p_1}$, we can use similar proof as the case $\frac{1}{2-p_1} < F_1 - F_2 \leq 1$.

Chapter 4

Strategic Defense against Reliability and Security Failures

This chapter is a joint work with Li Jin. To be submitted to IEEE Transactions on automatic Control [89].

This chapter aims to study the strategies and resilience of dynamic routing for intelligent transportation system and other engineering systems such as production lines and communication networks. Such network systems rely on connected sensing and actuating devices for computation, communication, and coordination. However, the lack of secure-by-design features makes them susceptible to random sensing faults and malicious spoofing attacks. This motivates the design of feedback control strategies that are both cost-efficient and reliable under such reliability and security failures. We consider a parallel multi-server queuing system and model it as a Markovian decision process. A system operator protects the routing guidance for incoming jobs dynamically based on the system status (queue lengths). We find out that the optimal decisions of the operator are threshold-based. While in the security setting, we extend the model to an attacker-defender game. The attacker manipulates the routing guidance so that the jobs are wrongly allocated to servers. Both attacking and defending induce technological costs. Hence, each player has to balance the technological cost and queuing cost. The equilibria regimes and the best responses also have threshold-based properties. For both settings, we study the dynamics and derive the sufficient stability conditions of the queuing system. Besides theoretical insights, we also present solution algorithms and numerical analysis that can support practical applications.

4.1 Introduction

The operation of a feedback-controlled network system relies heavily on data collection and transmission that are vulnerable to random malfunctions and malicious attacks. For example, in the internet of vehicles, researchers have found that traffic sensors and traffic lights can be easily intruded and manipulated [11, 31]; wired/wireless communication between connected vehicles are also in the threat of various forms of attacks [1, 67]. Similar security risks also exist in production lines [49] and communication networks [18]. However, such security risks have not been well studied in conjunction with the dynamics of network systems, which is typically modeled as queueing processes. Moreover, since the vulnerable sensing and actuating components are physically distributed, it is hard to predict when a component will be attacked, and which component will be attacked. It can be seen that it is economically infeasible and technically unnecessary to completely prevent reliability and security failures. Yet, it is crucial to understand the risk levels under the two scenarios and to design strategic defense mechanisms for the queueing system.

In this chapter, we develop novel models and methods to evaluate the reliability/security risks of dynamic routing and to design an efficient deployment of protecting/defending resources. We consider a system of parallel servers and queues with dynamic routing subject to faults due to random malfunctions or malicious attacks. The system operator (defender) protects the routing guidance for incoming jobs dynamically based on the system status (queue lengths). The attacker manipulates the routing guidance so that the jobs are wrongly allocated to servers. Attacking and defending both induce technological costs, so the defender has to balance the technological cost and queueing cost. Our goal is to quantify the efficiency loss (in terms of queueing delays) due to such failures and design cost-efficient feedback control strategies. We characterize the structures of the defending strategies and develop algorithms that efficiently compute the strategies. We also demonstrate our approach via a series of computational examples. The proposed methods are relevant to the resilient design of intelligent transportation systems, production lines, and communication networks.

Specifically, we consider a homogeneous Poisson arrival process of jobs and n parallel exponential servers with identical service rates. If both sensing and actuating are normal, the system operator allocates incoming jobs to the shortest queue; if the queues are equal, the job is routed randomly to each server with equal or non-equal probabilities. We focus on two scenarios of failures:

1. Reliability: The routing is faulty with a constant probability. When a fault occurs, a job is randomly allocated to one of the n servers; otherwise the job is allocated to the shortest queue. The system operator deploys security resources to control the probability of faults. Deploying security resources induces a technological cost on the defender, and the cost is identical for all jobs. The defender aims to balance the

efficiency loss due to the faults and the technological cost to deploy security resource.

2. Security: A malicious attacker is able to modify the routing instruction with a randomly generated one. The defender is able to defend individual jobs to ensure correct routing. Both attack and defense induce technological costs. The attacker's (resp. defender's) decision is the probability of attacking (resp. defending) the routing of each customer. The attacker (resp. defender) is interested in balancing the long-time-average network-wide queueing cost minus the attacking cost (resp. plus the defending cost). We assume that both players use Markovian strategies; i.e. the probabilities of attacking and defending only depend on the state of the queueing system.

Numerous results have been developed for parallel queueing systems without sensing/ actuating faults [21,23,26,27,36,37,59]. Although some of these results provide hints for our problem, they do not directly apply to the setting with failures. Parallel queueing systems have been studied with delayed [47], erroneous [6], or decentralized information [60], which provides insights for our purpose. Previous work typically relies on characterization or approximation of the steady-state distribution of the queueing state; however, this analysis approach is hard to be synthesized with reliability failure and security game models. In addition, it is hard to study the steady-state distribution of queueing systems with state-dependent transition rates.

To address this challenge, we use a Lyapunov function-based approach to study the stability (i.e. boundedness) of the queueing system and to obtain upper bounds for the mean number of jobs in the system. This approach has been applied to queueing systems in settings different from that in this chapter [16,22,46]. Importantly, we use this approach to study the queueing dynamics under state-dependent defending strategies. Using an upper bound for queueing cost derived from the Foster-Lyapunov criterion [56], we formulate a design problem for security resource deployment. We also formulate a dynamic programming (DP) to compute the optimal defending strategy. Using a numerical example, we show that the DP algorithm gives a solution that is consistent with our theoretical conclusion.

Next, we characterize the equilibria of the attacker-defender game. Game theory is a powerful tool for security risks analysis that has been extensively used in various engineering systems [24, 53, 85]. Game theoretic approaches have been applied to studying security of routing in transportation [48, 86] and communications [8, 34]. However, to the best of our knowledge, the security risk of feedback routing policies has not been well studied from a perspective combining game theory and queueing theory, which is essential for capturing the interaction between the queueing dynamics and the players' decisions. For open-loop attacking and defending strategies, we quantitatively characterize the security risk (in terms of attack-induced queueing delay and technological cost for defense) in various scenarios. We show that the game has multiple regimes for equilibria dependent on the technological costs of attacking and of defending as well as the demand. A key finding is that the attacker would either attack no jobs or attack all jobs. When the attacking cost is high, the attacker may have no incentive to attack any jobs; consequently, the defender does not need to defend any jobs. When the attacking cost is low, the attacker will attack every job; in this case, the defender's behavior will depend on the defending cost. The regimes also depend on the arrival rate of jobs: for higher arrival rates, the attacker has a higher incentive to attack, and the defender has a higher incentive to defend. For closed-loop strategies, we again use the Lyapunov function-based approach to derive an upper bound for the queueing cost resulting from the attacker-defender game. In particular, we show that the defender has a higher incentive to defend if the difference between the longest and the shortest queues is larger. We also develop an algorithm that computes the equilibria of the game and quantifies the security risk.

4.2 Parallel queueing system and failure models

4.2.1 Queueing model

Consider a parallel queueing system. Jobs arrive according to a Poisson process of rate λ. Each server serves jobs at an exponential rate of μ. We use $X(t) = \begin{bmatrix} X_1(t) & X_2(t) & \cdots & X_n(t) \end{bmatrix}^T$ to denote the number of jobs, either waiting or being served, in the n servers, respectively. The state space of the parallel queueing system is $\mathbb{Z}^n_{\geq 0}$.

Without any failures, any incoming job is allocated to the shortest queue. If there are multiple shortest queues, then the job is randomly allocated to one of them with equal probabilities.

4.2.2 Reliability failures

Suppose that when a job arrives at the system, its allocation is correct with probability $(1 - a)$ and is faulty with probability $a \in [0, 1]$. If the allocation is correct, the job joins the shortest queue. If the allocation is faulty, then the job joins a random queue; the probability of joining the ith queue is p_i where $\sum_{i=1}^{n} p_i = 1$. Fig. 4.1 illustrates the routing in the presence of reliability failures.

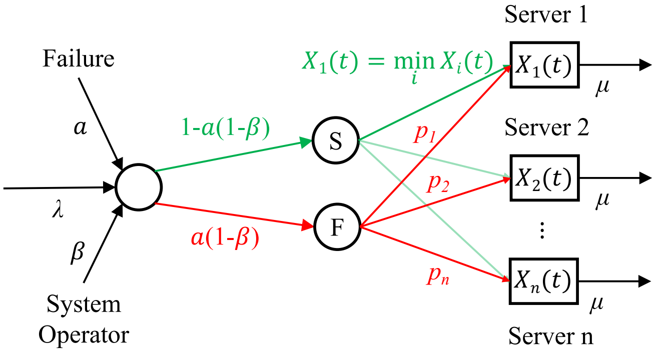


Figure 4.1: A n-queue system with shorter-queue routing under reliability failures.

The system operator can deploy additional resources to ensure correct routing. The probability of protecting is a state-dependent Markovian policy $\beta : \mathbb{Z}^n_{\geq 0} \to [0, 1]$, which is selected by the system operator. Protecting a job induces a one-time cost of c_b on the system operator.

The objective of the system operator is to balance the queueing cost and the protecting cost. We formulate this problem as an infinite-horizon continuous-time Markov decision process.

The system operator aims to minimize the expected cumulative discounted cost $J(x)$:

$$J^*(x) = \min_{\beta} J(x, \beta) = \min_{\beta} \mathbb{E}\Big[\int_0^\infty e^{-\rho t} C(X(t)) dt \Big| X(0) = x \Big],$$

where ρ is the discounted factor and $C : \mathbb{Z}^n_{\geq 0} \to \mathbb{R}$ is the immediate cost defined as

$$C(\xi) = |\xi|+c_b\beta(\xi).$$

4.2.3 Security failures

Suppose that a malicious attacker is able to compromise the system operator (defender)'s dynamic routing. When a job arrives and is being allocated, the attacker is able to modify the instruction sent by the operator so that the job is mistakenly allocated to a non-shortest queue. If the attacker attacks, she needs to select the queue that the job joins. Since we only consider Markovian strategies, it is apparent that the attacker's best action is to allocate the job to the longest queue. Attacks have no impact when the queues are equal. Each job is attacked with a state-dependent probability $\alpha : \mathbb{Z}^n_{\geq 0} \to [0, 1]$, where $\alpha(x)$ is selected by the attacker. Fig. 4.2 illustrates the routing in the presence of reliability failures.

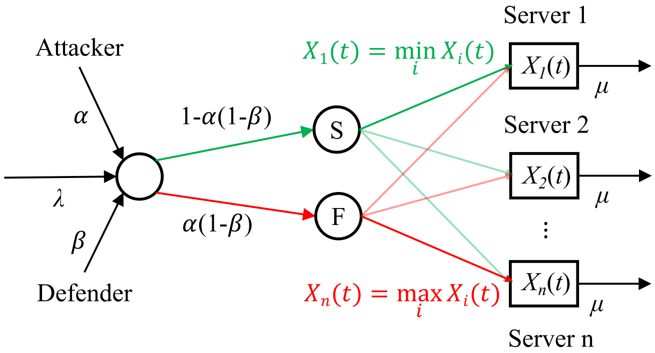


Figure 4.2: A n-queue system with shorter-queue routing under security failures.

The defender model is essentially the same as that in the reliability setting. The only difference is that in the security setting, the defender knows that she is playing a security game with the strategic attacker.

We formulate the interaction between the attacker and the defender as an infinite-horizon stochastic game with Markovian strategies.

The attacker aims to maximize the expected cumulative discounted reward $V(x, \alpha, \beta)$ given the defender's Markovian strategy β:

$$V^*_A(x, \beta) = \max_{\alpha} V(x, \alpha, \beta) = \max_{\alpha} \mathbb{E}\Big[\int_0^{\infty} e^{-\rho t} R(X(t))dt \Big| X(0) = x\Big],$$

where $R : \mathbb{Z}^n_{\geq 0} \to \mathbb{R}$ is the immediate reward defined as

$$R(\xi) = |\xi|+c_b\beta(\xi) - c_a\alpha(\xi).$$

Similarly, the defender aims to minimize the expected cumulative discounted loss given the attacker's Markovian strategy α:

$$V_B^*(x, \alpha) = \min_{\beta} V(x, \alpha, \beta).$$

4.3 Protection against random faults

In this section, we consider the design of the system operator's state-dependent protecting policy from two aspects: stability and optimality.

It is well known that a parallel n-queue system is stable if and only if the demand is less than the total capacity, i.e. $\lambda < n\mu$. In the following result, we will see that random faults can destabilize the system.

Proposition 4. *The unprotected n-queue system with faulty probability a is stable if and only if*

$$\lambda < n\mu, \tag{4.1a}$$

$$ap_{max}\lambda < \mu. \tag{4.1b}$$

Furthermore, when the system is stable, the number of jobs is upper-bounded by

$$\bar{X} := \limsup_{t\to\infty} \sum_{s=0}^{t} \mathbb{E}[f(X(s))] \leq \frac{\lambda + n\mu}{2\Big(\mu - \max(ap_{\max}, 1/n)\lambda\Big)}. \tag{4.2}$$

Now consider the n-queue system under protecting policy. We say that the n-queue system is stabilizable if a stabilizing policy exists.

Theorem 4. *Consider the n-queue system subject to faults. The routing of a job is faulty with probability a. The system operator protects each job with a state-dependent probability $\beta : \mathbb{Z}^n_{\geq 0} \to [0,1]$. Then, a stabilizable n-queue system is stable if for any $x \in \mathbb{Z}^n_{\geq 0}$ such that x is not a diagonal vector, we have*

$$\beta(x) > 1 - \frac{\mu|x|-\lambda x_{\min}}{a\lambda\Big(\sum\limits_{i=1}^{n} p_i x_i - x_{\min}\Big)}, \tag{4.3}$$

where $x_{\min} = \min_i x_i$ and $|x|= \sum_{i=1}^{n} x_i$. Furthermore, under the above condition, the number of jobs is upper-bounded by

$$\bar{X} \leq \frac{\lambda + n\mu}{2c}, \tag{4.4}$$

where $c = \min\limits_{x\succ\mathbf{0}}\{\mu - \lambda x_{\min}/|x|-a(1-\beta(x))\lambda(\sum_{i=1}^{n} p_i x_i - x_{\min})/|x|\}$.

Next, we study the structure of the optimal defending policy for the dynamic routing problem.

The Hamiltonian-Jacobi-Bellman equation (derived from Kolmogorov equation) of the dynamic programming can be written as [10]

$$0 = \min_{\beta}\{|x|+c_b\beta(x) - \rho J^*(x) + \mathcal{L}^{\beta} J^*(x)\},$$

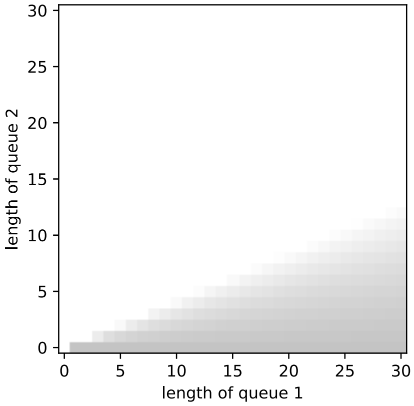


Figure 4.3: The characterization of the threshold of the stabilizing protecting policy for a two-queue system.

where $\mathcal{L}^\beta$ is the infinitesimal generator under control policy β. That is,

$$\begin{aligned}(\rho+\lambda+n\mu)J^*(x) =& \min_\beta \Big\{ |x| + c_b\beta(x) + \mu\sum_i J^*((x-e_i)^+) + \lambda \min_j J^*(x+e_j) \\ &+ (1-\beta(x))a\lambda\Big(\sum_i p_i J^*(x+e_i) - \min_j J^*(x+e_j)\Big)\Big\} \end{aligned} \tag{4.5}$$

Here $+(-)e_i$ means adding (subtracting) 1 from i-th element.

Definition 5. *The optimal defending policy is defined as*

$$\beta^* = \operatorname*{argmin}_\beta J(x,\beta).$$

Remark 4. *When $a=0$, $\beta^*\equiv 0$ and when $x_1=x_2=\cdots=x_n$, $\beta^*(x)=0$.*

Lemma 6. *The optimal defending policy is bang-bang, taking $\beta^*(x)=0$ or $\beta^*(x)=1$.*

Proof. The expression to be minimized in the right-hand side of (4.5) is linear in $\beta(x)$, so the minimum is reached at the endpoints, that is, 0 or 1.

Therefore, the defending policy is deterministic at each state x, either defend ($b=1$) or not to defend ($b=0$). Now the HJB equation turns into

$$\begin{aligned}(\rho+\lambda+n\mu)J^*(x) =& \min_{b\in\{0,1\}} \Big\{ |x| + c_b b + \mu\sum_i J^*((x-e_i)^+) + \lambda \min_j J^*(x+e_j) \\ &+ (1-b)a\lambda\Big(\sum_i p_i J^*(x+e_i) - \min_j J^*(x+e_j)\Big)\Big\} \\ \overset{def}{=}& \min_{b\in\{0,1\}} \Big\{ c(x,b) + \sum_{x'} q(x'|x,b)J^*(x')\Big\}\end{aligned}$$

Using the uniformization trick [51,63], we have

$$J^*(x) \overset{def}{=} \min_b \Big\{ \tilde{c}(x,b) + \gamma \sum_{x'} p(x'|x,b) J^*(x') \Big\}, \tag{4.6}$$

where $\Lambda = \lambda + n\mu$, $\gamma = \Lambda/(\rho+\Lambda)$, $\tilde{c}(x,b) = c(x,b)/(\rho+\Lambda)$ and $p(x'|x,b) = q(x'|x,b)/\Lambda$. Without loss of generality, we assume $\rho + \Lambda = 1$ in the following.

We can set up a value iteration form of the HJB equation as

$$J^{(k+1)}(x) = \min_b \Big\{ \tilde{c}(x,b) + \gamma \sum_{x'} p(x'|x,b) J^{(k)}(x') \Big\}. \tag{4.7}$$

The main theorem of this section is given below.

Theorem 5. *The optimal defending policy $\beta^* : \mathbb{Z}^n_{\geq 0} \to [0,1]$ is a threshold policy characterized by n non-intersecting monotonically non-decreasing threshold functions. Specifically, in each polyhedron $\mathscr{X}_m = \{x \in \mathbb{Z}^n_{\geq 0} \mid x_i \geq x_m, \forall 1 \leq i \leq n\}$ $(m = 1, 2, \cdots, n)$, $\beta^*(x)$ is monotonically non-decreasing in x_i $(i \neq m)$ when other variables are fixed and monotonically non-increasing in x_m when other variables are fixed. See Figure 4.4.*

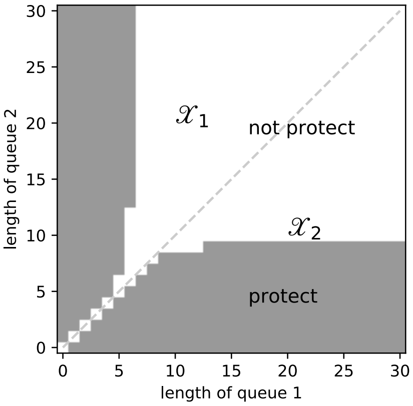


Figure 4.4: The characterization of the optimal protecting policy for a two-queue system.

Based on Theorem 5, the key findings are: the defender is more likely to defend when (1) the queue lengths are ”unbalanced”; (2) queues are close to empty.

4.3.1 Proof of stability criteria

Consider the quadratic Lyapunov function

$$W(x) = \frac{1}{2}\sum_{i=1}^{n} x_i^2.$$

For the unprotected case, by applying infinitesimal generator, we have

$$\begin{aligned}\mathcal{L}W(x) =& a\lambda\frac{1}{2}\sum_{i=1}^{n} p_i\Big((x_i+1)^2 - x_i^2\Big) + (1-a)\lambda\frac{1}{2}\Big((x_{\min}+1)^2 - x_{\min}^2\Big)\\ &+ \mu\frac{1}{2}\sum_{i=1}^{n}\mathbb{I}_{x_i>0}\Big((x_i-1)^2 - x_i^2\Big)\\ =& a\lambda\sum_{i=1}^{n} p_i x_i + (1-a)\lambda x_{\min} - \mu\sum_{i=1}^{n} x_i + \frac{1}{2}\lambda + \frac{1}{2}\sum_{i=1}^{n}\mathbb{I}_{x_i>0}\mu.\end{aligned}$$

Note that

$$\mathcal{L}W(x) \le \Big(\max(ap_{\max}, 1/n)\lambda - \mu\Big)|x|+\frac{1}{2}(\lambda + n\mu).$$

Hence, by (4.1a)–(4.1b) there exists a constant $c = \mu - \max(ap_{\max}, 1/n)\lambda > 0$ and $d = \frac{1}{2}(\lambda + n\mu)$ such that

$$\mathcal{L}W(x) \le -c|x|+d, \quad \forall x \in \mathbb{Z}^n_{\ge 0}.$$

By [56, Theorem 4.3], the above implies (4.2) and thus stability.

For the protected case, by applying infinitesimal generator, we have

$$\begin{aligned}\mathcal{L}W(x) =& a(1-\beta(x))\lambda\frac{1}{2}\sum_{i=1}^{n} p_i\Big((x_i+1)^2 - x_i^2\Big) + \Big(1 - a(1-\beta(x))\Big)\lambda\frac{1}{2}\Big((x_{\min}+1)^2 - x_{\min}^2\Big)\\ &+ \mu\frac{1}{2}\sum_{i=1}^{n}\mathbb{I}_{x_i>0}\Big((x_i-1)^2 - x_i^2\Big)\\ =& a(1-\beta(x))\lambda\sum_{i=1}^{n} p_i x_i + \Big(1 - a(1-\beta(x))\Big)\lambda x_{\min} - \mu\sum_{i=1}^{n} x_i + \frac{1}{2}\lambda + \frac{1}{2}\sum_{i=1}^{n}\mathbb{I}_{x_i>0}\mu.\end{aligned}$$

Note that

$$\mathcal{L}W(x) \le a(1-\beta(x))\lambda\Big(\sum_{i=1}^{n} p_i x_i - x_{\min}\Big) + (\lambda x_{\min} - \mu|x|) + \frac{1}{2}(\lambda + n\mu)$$

Hence, by (4.3) there exists a constant $c = \min_{x\succ\mathbf{0}}\{\mu - \lambda x_{\min}/|x| - a(1-\beta(x))\lambda(\sum_{i=1}^{n} p_i x_i - x_{\min})/|x|\} > 0$ and $d = \frac{1}{2}(\lambda + n\mu)$ such that

$$\mathcal{L}W(x) \le -c|x|+d, \quad \forall x \in \mathbb{Z}^n_{\ge 0}.$$

By [56, Theorem 4.3], the above implies (4.4) and thus stability. □

4.3.2 Proof of Theorem 5

Proposition 5. *The optimal cost function $J^* : \mathbb{Z}^n_{\geq 0} \to \mathbb{R}$ has the following properties:*

(i) (symmetry) J^* is symmetric, i.e. $J^*(x) = J^*(\sigma x)$ where σx is a permutation of x.

(ii) (monotonicity) J^* is non-decreasing, i.e. $J^*(x) \geq J^*(y)$ if $x_i \geq y_i$ for all i $(1 \leq i \leq n)$.

(iii) (convexity) J^* is convex in each variable, i.e. $J^*(x + e_i) - J^*(x) \leq J^*(x + 2e_i) - J^*(x + e_i)$.

(iv) (Schur convexity) J^* is Schur convex, i.e. $J^*(x + e_i) \geq J^*(x + e_j)$ if $x_i \geq x_j$.

(v) (supermodularity) J^* is supermodular, i.e. $J^*(x + e_i + e_j) + J^*(x) \geq J^*(x + e_i) + J^*(x + e_j)$.

Since we need the symmetry and Schur convexity in the proof of Theorem 5, we provide the proofs of them and omit the proofs of other properties.

Proof of symmetry. Note that for any x,

- $|\sigma x| = |x|$,
- $\{\sigma((x - e_1)^+), \cdots, \sigma((x - e_n)^+)\}$ is a permutation of $\{(x - e_1)^+, \cdots, (x - e_n)^+\}$,
- $\{\sigma(x + e_1), \cdots, \sigma(x + e_n)\}$ is a permutation of $\{x + e_1, \cdots, x + e_n\}$,

then by (4.6) we can conclude that $J^*(x) = J^*(\sigma x)$. □

Proof of Schur convexity. We will use induction to prove $x_i \geq x_j \Rightarrow J^{(k)}(x+e_i) \geq J^{(k)}(x+e_j)$ for any x, k.

Base Step. It is easy to verify that $J^{(0)} = 0$, $J^{(1)}(x) = |x|$ and $J^{(2)}(x) = (1 + \Lambda)|x| + \lambda - \mu \sum_i \mathbb{I}_{x_i > 0}$. Then we have $x_i \geq x_j \Rightarrow J^{(2)}(x + e_i) \geq J^{(2)}(x + e_j)$ for any x. Note that the inequality is strict for some x, say $(1, 0, \cdots, 0)$. The reason we start the base step from $k = 2$ is to avoid reaching trivial conclusions, say all inequalities are basically equalities. We will use similar base steps in the proofs of Theorem 5 and Theorem 7.

Induction Step. According to the value iteration (4.7),

$$\begin{aligned}J^{(k+1)}(x+e_i)-J^{(k+1)}(x+e_j) =&\mu\sum_{l=1}^{n}[J^{(k)}((x+e_i-e_l)^+)-J^{(k)}((x+e_j-e_l)^+)]\\&+\lambda\Big[\min_l J^{(k)}(x+e_i+e_l)-\min_l J^{(k)}(x+e_j+e_l)\Big]\\&+\min\Big\{c_b, a\lambda\Big[\sum_{l=1}^{n}p_l J^{(k)}(x+e_i+e_l)-\min_l J^{(k)}(x+e_i+e_l)\Big]\Big\}\\&-\min\Big\{c_b, a\lambda\Big[\sum_{l=1}^{n}p_l J^{(k)}(x+e_j+e_l)-\min_l J^{(k)}(x+e_j+e_l)\Big]\Big\}.\end{aligned}$$

Note that based on the induction hypothesis, when $x_i \geq x_j$, for any l we have

$$J^{(k)}((x-e_l)^+ + e_i) \geq J^{(k)}((x-e_l)^+ + e_j),$$

$$J^{(k)}(x+e_i+e_l) \geq J^{(k)}(x+e_j+e_l),$$

and thus

$$J^{(k)}((x+e_i-e_l)^+) \geq J^{(k)}((x+e_j-e_l)^+),$$

$$\min_l J^{(k)}(x+e_i+e_l) \geq \min_l J^{(k)}(x+e_j+e_l).$$

Then we can conclude that

$$J^{(k+1)}(x+e_i) \geq J^{(k+1)}(x+e_j).$$

Therefore, the Schur convexity $x_i \geq x_j \Rightarrow J^*(x+e_i) \geq J^*(x+e_j)$ always holds. □

Proof of Theorem 5. Let $m = \operatorname{argmin}_i x_i$. To demonstrate the existence of the threshold policy, we will show that

$$\begin{aligned}&\beta^*(x+e_i) \geq \beta^*(x) \quad (\forall i \neq m)\\&\beta^*(x+e_m) \leq \beta^*(x).\end{aligned} \tag{4.8}$$

Because of Schur convexity, $J^*(x+e_i) \geq J^*(x+e_m)$ $(\forall i \neq m)$. We can rewrite (4.6) as

$$J^*(x) = \min_{b\in\{0,1\}}\Big\{|x|+c_b b+\mu\sum_{i=1}^{n}J^*((x-e_i)^+)+\lambda J^*(x+e_m)+(1-b)a\lambda\Big[\sum_{i=1}^{n}p_i J^*(x+e_i)-J^*(x+e_m)\Big]\Big\}.$$

Let $\Delta(x) = \sum\limits_{i=1}^{n} p_i J^*(x+e_i) - J^*(x+e_m)$, then (4.8) is essentially

$$\begin{aligned} &\Delta(x+e_i) \geq \Delta(x) \quad (\forall i \neq m) \\ &\Delta(x+e_m) \leq \Delta(x). \end{aligned} \tag{4.9}$$

We will use induction based on value iteration to prove (4.9), that is, let $\Delta^{(k)}(x) = \sum\limits_{i=1}^{n} p_i J^{(k)}(x+e_i) - J^{(k)}(x+e_m)$, it is sufficient to show

$$\begin{aligned} &\Delta^{(k)}(x+e_i) \geq \Delta^{(k)}(x) \quad (\forall i \neq m) \\ &\Delta^{(k)}(x+e_m) \leq \Delta^{(k)}(x), \end{aligned} \tag{4.10}$$

for all k.

Induction step. According to the value iteration (4.7), we have $\forall j \neq m$,

$$\begin{aligned} \Delta^{(k+1)}(x+e_j) - \Delta^{(k+1)}(x) =& \mu \sum_{i=1}^{n} [\Delta^{(k)}((x+e_j-e_i)^+) - \Delta^{(k)}((x-e_i)^+)] \\ &+ \lambda[\Delta^{(k)}(x+e_j+e_m) - \Delta^{(k)}(x+e_m)] \\ &+ \sum_{i=1}^{n} p_i \min\left\{c_b, a\lambda\Delta^{(k)}(x+e_j+e_i)\right\} - \min\left\{c_b, a\lambda\Delta^{(k)}(x+e_j+e_m)\right\} \\ &- \sum_{i=1}^{n} p_i \min\left\{c_b, a\lambda\Delta^{(k)}(x+e_i)\right\} + \min\left\{c_b, a\lambda\Delta^{(k)}(x+e_m)\right\}, \end{aligned}$$

and

$$\begin{aligned} \Delta^{(k+1)}(x+e_m) - \Delta^{(k+1)}(x) =& \mu \sum_{i=1}^{n} [\Delta^{(k)}((x+e_m-e_i)^+) - \Delta^{(k)}((x-e_i)^+)] \\ &+ \lambda[\Delta^{(k)}(x+2e_m) - \Delta^{(k)}(x+e_m)] \\ &+ \sum_{i=1}^{n} p_i \min\left\{c_b, a\lambda\Delta^{(k)}(x+e_m+e_i)\right\} - \min\left\{c_b, a\lambda\Delta^{(k)}(x+2e_m)\right\} \\ &- \sum_{i=1}^{n} p_i \min\left\{c_b, a\lambda\Delta^{(k)}(x+e_i)\right\} + \min\left\{c_b, a\lambda\Delta^{(k)}(x+e_m)\right\}, \end{aligned}$$

Note that based on the induction hypothesis, we have $\forall j \neq m$,

$$\Delta^{(k)}((x+e_j-e_i)^+) \geq \Delta^{(k)}((x-e_i)^+) \geq \Delta^{(k)}((x+e_m-e_i)^+),$$

$$\Delta^{(k)}(x+e_j+e_i) \geq \Delta^{(k)}(x+e_i) \geq \Delta^{(k)}(x+e_m+e_i),$$

$$\Delta^{(k)}(x+e_j+e_m) \geq \Delta^{(k)}(x+e_m) \geq \Delta^{(k)}(x+2e_m).$$

Then we can conclude that

$$\Delta^{(k+1)}(x+e_j) \geq \Delta^{(k+1)}(x) \quad (\forall j \neq m)$$

$$\Delta^{(k+1)}(x) \leq \Delta^{(k+1)}(x+e_m).$$

Thus the existence of an optimal threshold policy is established. □

4.3.3 Numerical Analysis

In this subsection, we introduce an algorithm that can estimate the optimal state-dependent protecting policy β^*, then we use it to conduct numerical analysis on 1) the relationship between the incentive to defend and key parameters; 2) the comparison between the optimal policy and two naive open-loop policies.

We call the solution algorithm truncated policy iteration (see Algorithm 1), it is adapted from the classic policy iteration algorithm [77] and based on the dynamic programming (4.7).

The incentive to defend is non-decreasing in the failure probability a, non-increasing in the technology cost c_b and non-decreasing in the throughput λ. See Figure 4.5.

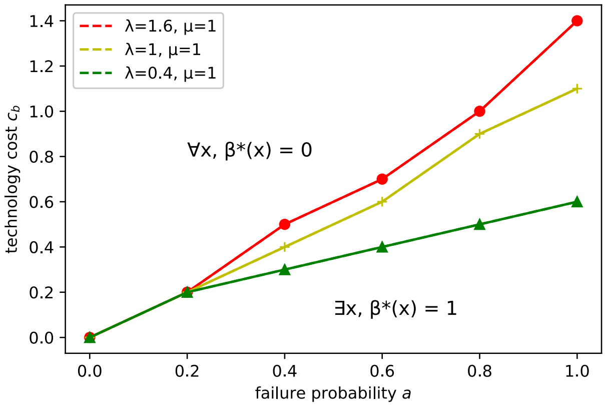


Figure 4.5: The relationship between β^* and a, c_b, λ (fixing $\mu = 1$, $n = 2$).

The optimal closed-loop protecting policy β^* can significantly reduces the security risk, compared to the open-loop policies. See Figure 4.6. The simulation results are based on the average of 20 episodes, each with 10000s.

Algorithm 1 Truncated policy iteration for estimating $\beta \approx \beta^*$ (continuing)

```
Algorithm parameters: small $\epsilon > 0$
Initialize array $J \in \mathbb{R}$ and $\beta \in \{0, 1\}$ arbitrarily (e.g. $J(x) = 0$, $\beta(x) = 0$, for all $x \in \mathcal{X} = \{0, 1, 2, \cdots, B\}^n$
repeat
    repeat
        $\Delta \leftarrow 0$
        foreach $x \in \mathcal{X}$ do
            $v \leftarrow J(x)$
            $c \leftarrow |x| + c_b \beta(x)$
            $J(x) \leftarrow c + \sum_{x'} p(x'|x, b) J(x')$
            $\Delta \leftarrow \max(\Delta, |v - J(x, b)|)$
        end
    until $\Delta < \epsilon$;
    $stable \leftarrow True$
    foreach $x \in \mathcal{X}$ do
        old-action$\leftarrow \beta(x)$
        if $a\lambda\Big(\sum_i p_i J^*(x + e_i) - \min_j J^*(x + e_j)\Big) < c_b$ then
            $\beta(x) = 0$
        end
        else
            $\beta(x) = 1$
        end
        if $old\text{-}action \neq \pi(x)$ then
            $stable \leftarrow False$
        end
    end
until $stable = True$;
Output a deterministic policy $\beta \approx \beta^*$
```

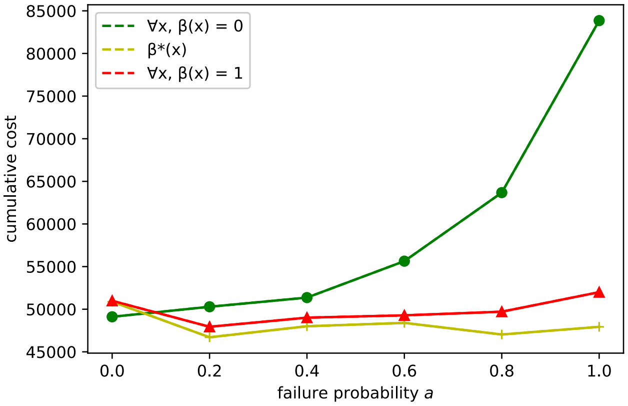


Figure 4.6: The comparison of cumulative cost between open-loop policies and the optimal closed-loop policy ($\lambda = 0.4, \mu = 0.25, c_b = 0.005$).

4.4 Defense against strategic attacks

For the state-dependent attacking and defending strategies, we derive the following property for the stability of the n-server system and for any equilibrium:

Theorem 6. *Consider the n-server system subject to attacks. The attacker (resp. defender) follows a Markovian strategy $\alpha : \mathbb{Z}^n_{\geq 0} \to [0, 1]$ (resp. $\beta : \mathbb{Z}^n_{\geq 0} \to [0, 1]$). Then, the n-server system is stable if there exists a compact set $\mathcal{X}_0 = [0, \theta]^n$ such that for any $x \in \mathbb{Z}^n_{\geq 0}$, when x is not a diagonal vector, we have*

$$\alpha(x)(1 - \beta(x)) < \frac{\mu - \lambda x_{\min}/|x|}{\lambda(x_{\max} - x_{\min})/|x|}, \tag{4.11}$$

where $x_{\max} = \max_i x_i$. Furthermore, any equilibrium (α^*, β^*) must satisfy the above, and the number of jobs is upper-bounded by

$$\bar{X} \leq \frac{\lambda + n\mu}{2c}, \tag{4.12}$$

where $c = \min_{x \succ \mathbf{0}} \{\mu - \lambda x_{\min}/|x| - \alpha(x)(1 - \beta(x))\lambda(x_{\max} - x_{\min})/|x|\}$.

Next, we discuss the equilibria of the stochastic security game.

Definition 6. *The optimal attacking (resp. defending) strategy α^* (resp. β^*) satisfies that for each state $x \in \mathbb{Z}^n_{\geq 0}$,*

$$\alpha^*(x) = \operatorname*{argmax}_{\alpha} V^*_A(x, \beta^*), \ \ \beta^*(x) = \operatorname*{argmin}_{\beta} V^*_B(x, \alpha^*).$$

The value of the attacker (defender) is $V^*_A(x, \beta^*)$ (resp. $V^*_B(x, \alpha^*)$). In particular, (α^*, β^*) is a Markovian perfect equilibrium.

Proposition 6. *The Markovian perfect equilibrium of this two-person non-cooperative stochastic security game always exists.*

Proof. Note that the state space $\mathbb{Z}^n_{\geq 0}$ is countable and the action space $[0, 1]$ is compact. By [25], the total-discounted return equilibrium policy exists. □

According to Shapley's extension on minimax theorem for stochastic game [71],

$$V^*_B(x, \alpha^*) = V^*_A(x, \beta^*) = V^*(x).$$

Similar to the derivation of (4.6), by assuming $\rho + \lambda + n\mu = 1$ we get

$$\begin{aligned} V^*(x) = \max_{\alpha} \min_{\beta} \Big\{ &|x| + c_b\beta(x) - c_a\alpha(x) + \mu \sum_i V^*((x - e_i)^+) + \lambda \min_j V^*(x + e_j) \\ &+ \alpha(x)(1 - \beta(x))\lambda\Big(\max_j V^*(x + e_j) - \min_j V^*(x + e_j)\Big)\Big\}. \end{aligned} \tag{4.13}$$

The main theorem of this section is given below.

Theorem 7. *The stochastic security game has the following regimes of Markovian perfect equilibria* (α^*, β^*)*:*

- *Type I:* $(0, 0)$ *(low risk)*
- *Type II:* $(1, 0)$ *(medium risk)*
- *Type III:* $(c_b/\delta^*, 1 - c_a/\delta^*)$ *(high risk) where* $\delta^*(x) = \lambda(\max_j V^*(x+e_j) - \min_j V^*(x+e_j))$

Furthermore, Type I and Type II regimes are characterized by $n(n-1)$ *non-intersecting symmetric monotonically non-decreasing threshold functions; Type II and Type III regimes are characterized by other* $n(n-1)$ *non-intersecting symmetric monotonically non-decreasing threshold functions (see Figure 4.7 and Figure 4.8).*

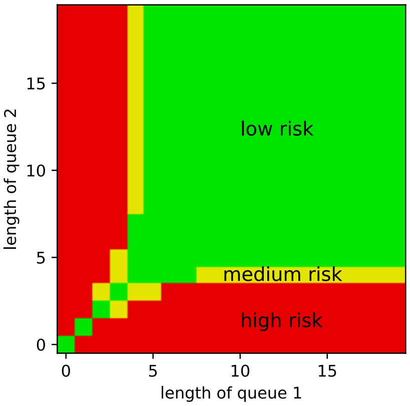


Figure 4.7: The equilibria regimes of the stochastic security game for a two-queue system.

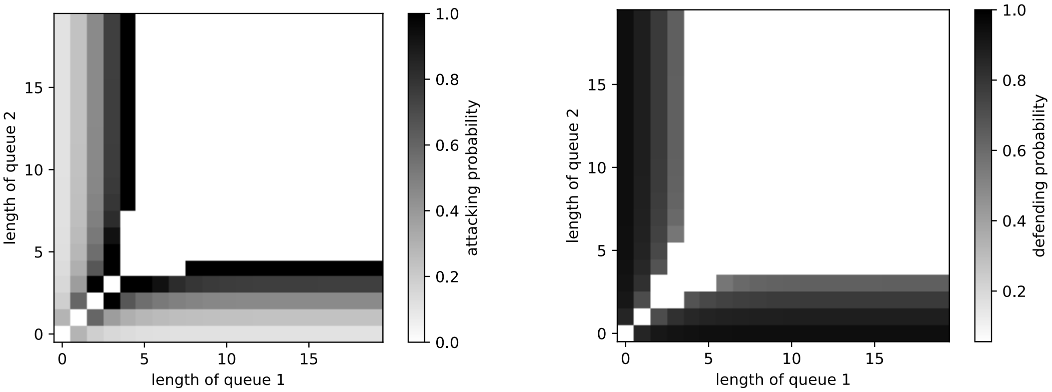


Figure 4.8: The optimal attacking and defending strategies for a two-queue system.

4.4.1 Proof of stability results

Consider the quadratic Lyapunov function

$$W(x) = \frac{1}{2}\sum_{i=1}^{n} x_i^2.$$

Applying infinitesimal generator, we have

$$\mathcal{L}W(x) = \alpha(x)(1-\beta(x))\lambda x_{\max} + \Big(1-\alpha(x)(1-\beta(x))\Big)\lambda x_{\min} - \sum_{i=1}^{n}\mu x_i + \frac{1}{2}\lambda + \frac{1}{2}\sum_{i=1}^{n}\mathbb{I}_{x_i>0}\mu.$$

Note that

$$\mathcal{L}W(x) \le \alpha(x)(1-\beta(x))\lambda(x_{\max}-x_{\min}) + \lambda x_{\min} - \mu|x| + \frac{1}{2}(\lambda+n\mu).$$

Hence, by (4.11) we have $c = \min_x\{\mu - \lambda x_{\min}/|x| - \alpha(x)(1-\beta(x))\lambda(x_{\max}-x_{\min})/|x|\} > 0$ and $d = \frac{1}{2}(\lambda+n\mu)$ such that

$$\mathcal{L}W(x) \le -c|x| + d, \quad \forall x \in \mathbb{Z}^n_{\ge 0}.$$

By [56, Theorem 4.3], the above implies (4.12) and thus stability. □

4.4.2 Proof of Theorem 7

Proposition 7. *The value function $V^*: \mathbb{Z}^n_{\ge 0} \to \mathbb{R}$ has the following properties:*

(i) V^* is symmetric, i.e. $V^*(x) = V^*(\sigma x)$ where σx is a permutation of x.

(ii) V^* is non-decreasing, i.e. $V^*(x) \ge V^*(y)$ if $x_i \ge y_i$ for all i $(1 \le i \le n)$.

(iii) V^* is convex in each variable, i.e. $V^*(x+e_i) - V^*(x) \le V^*(x+2e_i) - V^*(x+e_i)$.

(iv) V^* is Schur convex, i.e. $V^*(x+e_i) \ge V^*(x+e_j)$ if $x_i \ge x_j$.

(v) V^* is supermodular, i.e. $V^*(x+e_i+e_j) + V^*(x) \ge V^*(x+e_i) + V^*(x+e_j)$.

The proof should be similar to the proof of Proposition 5, so we omit it here.

Proof of Theorem 7. Based on the symmetry, without loss of generality, we only need to consider the case when $x_1 = \max_i x_i, x_n = \min_i x_i$. Because of Schur convexity, $V(x+e_1) = \max_j V(x+e_j)$, $V(x+e_n) = \min_j V(x+e_j)$. We can rewrite (4.13) as

$$V^*(x) = \max_\alpha \min_\beta \Big\{ |x| + c_b\beta(x) - c_a\alpha(x) + \mu\sum_i V^*((x-e_i)^+) + \lambda V^*(x+e_n) \\ + \alpha(x)(1-\beta(x))\lambda\Big(V^*(x+e_1) - V^*(x+e_n)\Big)\Big\}. \tag{4.14}$$

Let $\mathcal{D}(x) = V^*(x + e_1) - V^*(x + e_n)$. To demonstrate the existence of threshold functions, we will show that the type of the equilibrium is monotonically non-decreasing in x_1 when other variables are fixed and monotonically non-increasing in x_n when other variables are fixed, that is,

$$\mathcal{D}(x + e_1) \geq \mathcal{D}(x), \quad \mathcal{D}(x + e_n) \leq \mathcal{D}(x). \tag{4.15}$$

We will use induction based on value iteration to prove, that is, let $\mathcal{D}^{(k)}(x) = V^{(k)}(x+e_1) - V^{(k)}(x + e_n)$, it is sufficient to show

$$\mathcal{D}^{(k)}(x + e_1) \geq \mathcal{D}^{(k)}(x), \quad \mathcal{D}^{(k)}(x + e_n) \leq \mathcal{D}^{(k)}(x). \tag{4.16}$$

Induction step. According to the value iteration form of (4.14), we have

$$\begin{aligned}
\mathcal{D}^{(k+1)}(x + e_1) - \mathcal{D}^{(k+1)}(x) =& \mu[\mathcal{D}^{(k)}(x) - \mathcal{D}^{(k)}((x - e_1)^+)] \\
&+ \mu[\mathcal{D}^{(k)}((x + e_1 - e_n)^+) - \mathcal{D}^{(k)}((x - e_n)^+)] \\
&+ \lambda[\mathcal{D}^{(k)}(x + e_1 + e_n) - \mathcal{D}^{(k)}(x + e_n)] \\
&+ \max\left\{0, \min\left\{\lambda\mathcal{D}^{(k)}(x + 2e_1) - c_a, c_b - \frac{c_a c_b}{\lambda\mathcal{D}^{(k)}(x + 2e_1)}\right\}\right. \\
&- \max\left\{0, \min\left\{\lambda\mathcal{D}^{(k)}(x + e_1 + e_n) - c_a, c_b - \frac{c_a c_b}{\lambda\mathcal{D}^{(k)}(x + e_1 + e_n)}\right\}\right. \\
&- \max\left\{0, \min\left\{\lambda\mathcal{D}^{(k)}(x + e_1) - c_a, c_b - \frac{c_a c_b}{\lambda\mathcal{D}^{(k)}(x + e_1)}\right\}\right. \\
&+ \max\left\{0, \min\left\{\lambda\mathcal{D}^{(k)}(x + e_n) - c_a, c_b - \frac{c_a c_b}{\lambda\mathcal{D}^{(k)}(x + e_n)}\right\}\right..
\end{aligned}$$

Note that based on the induction hypothesis, we have

$$\mathcal{D}^{(k)}((x + e_1 - e_n)^+) \geq \mathcal{D}^{(k)}((x - e_n)^+) \geq \mathcal{D}^{(k)}(x) \geq \mathcal{D}^{(k)}((x - e_1)^+),$$

$$\mathcal{D}^{(k)}(x + 2e_1) \geq \mathcal{D}^{(k)}(x + e_1) \geq \mathcal{D}^{(k)}(x + e_1 + e_n) \geq \mathcal{D}^{(k)}(x + e_n).$$

Then we can conclude that $\mathcal{D}^{(k+1)}(x+e_1) \geq \mathcal{D}^{(k+1)}(x)$ and prove $\mathcal{D}^{(k+1)}(x) \leq \mathcal{D}^{(k+1)}(x+e_n)$ in a similar way. Thus the existence of the threshold functions is established. The derivation of the equilibria regimes is given in the following subsection. □

4.4.3 Equilibrium analysis

Based on the Bellman equation (4.13), we develop an algorithm adapted from Shapley's algorithm [2, 71] to compute the minimax value and minimax equilibrium strategy. See Algorithm 2.

Algorithm 2 Shapley's algorithm for estimating $V \approx V^*$, $\beta \approx \beta^*$, $\alpha \approx \alpha^*$ (continuing)

```
Set $V(x) = 0$ for all $x \in \mathcal{X}$
repeat
    $\Delta \leftarrow 0$
    foreach $x \in \mathcal{X}$ do
        $v \leftarrow V(x)$
        Build auxiliary matrix game $M(x, V)$
        Compute the value $val(M)$ by using Shapley-Snow method
        $V(x) \leftarrow val(M)$
        $\Delta \leftarrow |v - V(x)|$
    end
until $\Delta < \epsilon$;
foreach $x \in \mathcal{X}$ do
    Build auxiliary matrix game $M(x, V)$
    Compute $\alpha$ and $\beta$ from $M(x, V)$ by using Shapley-Snow method
end
```

Here the auxiliary matrix game $M(x, V)$ is

$$\Big(|x| + \mu \sum_i V((x - e_i)^+) + \lambda \min_j V(x + e_j)\Big) \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & c_b \\ -c_a + \delta & -c_a + c_b \end{bmatrix}$$

and the value $val(M)$ given by Shapley-Snow method [72] is

$$val(M) = \begin{cases} |x| + \mu \sum\limits_i V((x - e_i)^+) + \lambda \min_j V(x + e_j), & \delta < c_a \\ |x| - c_a + \mu \sum\limits_i V((x - e_i)^+) + \lambda \max_j V(x + e_j), & c_a \leq \delta < c_b \\ |x| + c_b + \mu \sum\limits_i V((x - e_i)^+) + \lambda \min_j V(x + e_j) - c_a c_b / \delta, & d \geq \max\{c_a, c_b\} \end{cases}$$

where $\delta = \lambda(\max_j V(x + e_j) - \min_j V(x + e_j))$.

Remark 5. *The equilibrium (α^*, β^*) and the value V^* are in the following three cases depending on the relationship between $\delta^* = \lambda(\max_j V^*(x + e_j) - \min_j V^*(x + e_j))$ and c_a, c_b.*

- $\delta^* < c_a \Rightarrow (\alpha^*, \beta^*) = (0, 0)$
- $c_a \leq \delta^* < c_b \Rightarrow (\alpha^*, \beta^*) = (1, 0)$
- $\delta^* \geq \max\{c_a, c_b\} \Rightarrow (\alpha^*, \beta^*) = (c_b/\delta^*, 1 - c_a/\delta^*)$

4.5 Concluding Remarks

In this chapter, we analyze the reliability/security risk of feedback-controlled queueing systems and propose advice for strategic defense. We consider a system of parallel servers and queues with dynamic routing subject to reliability and/or security failures. For the reliability setting, we formulate it as an infinite-horizon Markov decision problem. We derive sufficient conditions for stability under state-dependent defending strategies. By formulating the infinite-horizon dynamic programming, we prove the threshold-based characteristic of the system operator's optimal protecting policy. We also use the truncated policy iteration to compute the policy. For the security setting, we formulate it as an attacker-defender game. The attacker selects the probability of modifying a job's allocation while the defender selects the probability of defending a job's allocation. Both attacking and defending induce technological costs, so the defender has to balance the technological cost and queuing cost. We characterize the equilibria regimes and apply Shapley's algorithm to compute the state-dependent strategies for both players. We also present numerical analysis to illustrate our proposed models and methods.

This work can be further extended to general networks. Another research direction of future studies is the algorithm design. Specifically, the truncated policy iteration and the Shapley's algorithm have the limitations that 1) they can only run for finite state space rather than countable infinite state space; 2) they need a relatively large space for storage. Such limitations motive the design of more efficient (in both time and space) deep reinforcement learning algorithms. It may be useful to utilize the threshold-based properties and policy/value function optimization in the design.

This work also provides the basis for allocating recovery resources and designing reliable failure-tolerant routing algorithms. The results can also support the design of secure transportation and logistics systems. Specific applications include app-based routing, signal-free intersection control, and packet routing.

Bibliography

[1] Mohammed Saeed Al-Kahtani. Survey on security attacks in vehicular ad hoc networks (vanets). In *2012 6th international conference on signal processing and communication systems*, pages 1–9. IEEE, 2012.

[2] Tansu Alpcan and Tamer Başar. *Network security: A decision and game-theoretic approach*. Cambridge University Press, 2010.

[3] Moshe E Ben-Akiva, Steven R Lerman, and Steven R Lerman. *Discrete choice analysis: theory and application to travel demand*, volume 9. MIT press, 1985.

[4] Michel Benaïm, Stéphane Le Borgne, Florent Malrieu, and Pierre-André Zitt. Qualitative properties of certain piecewise deterministic markov processes. In *Annales de l'IHP Probabilités et statistiques*, volume 51, pages 1040–1075, 2015.

[5] Dimitris Bertsimas, Ioannis Ch Paschalidis, and John N Tsitsiklis. Optimization of multiclass queueing networks: Polyhedral and nonlinear characterizations of achievable performance. *The Annals of Applied Probability*, pages 43–75, 1994.

[6] Frederick J Beutler and Demosthenis Teneketzis. Routing in queueing networks under imperfect information: Stochastic dominance and thresholds. *Stochastics: An International Journal of Probability and Stochastic Processes*, 26(2):81–100, 1989.

[7] Mogens Blanke, Michel Kinnaert, Jan Lunze, Marcel Staroswiecki, and J Schröder. *Diagnosis and fault-tolerant control*, volume 2. Springer, 2006.

[8] Stephan Bohacek, Joao P Hespanha, Katia Obraczka, Junsoo Lee, and Chansook Lim. Enhancing security via stochastic routing. In *Proceedings. Eleventh International Conference on Computer Communications and Networks*, pages 58–62. IEEE, 2002.

[9] Maury Bramson et al. Stability of join the shortest queue networks. *The Annals of Applied Probability*, 21(4):1568–1625, 2011.

[10] Fwu-Ranq Chang. *Stochastic optimization in continuous time*. Cambridge University Press, 2004.

[11] Qi Alfred Chen, Yucheng Yin, Yiheng Feng, Z Morley Mao, and Henry X Liu. Exposing congestion attack on emerging connected vehicle based traffic signal control. In *NDSS*, 2018.

[12] Giacomo Como, Ketan Savla, Daron Acemoglu, Munther A Dahleh, and Emilio Frazzoli. Robust distributed routing in dynamical networks—part i: Locally responsive policies and weak resilience. *IEEE Transactions on Automatic Control*, 58(2):317–332, 2012.

[13] Samuel Coogan and Murat Arcak. A compartmental model for traffic networks and its dynamical behavior. *IEEE Transactions on Automatic Control*, 60(10):2698–2703, 2015.

[14] JG Dai, John J Hasenbein, and Bara Kim. Stability of join-the-shortest-queue networks. *Queueing Systems*, 57(4):129–145, 2007.

[15] Jim G Dai. On positive Harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. *The Annals of Applied Probability*, pages 49–77, 1995.

[16] Jim G Dai and Sean P Meyn. Stability and convergence of moments for multiclass queueing networks via fluid limit models. *IEEE Transactions on Automatic Control*, 40(11):1889–1904, 1995.

[17] Mark HA Davis. Piecewise-deterministic markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(3):353–376, 1984.

[18] Hongmei Deng, Wei Li, and Dharma P Agrawal. Routing security in wireless ad hoc networks. *IEEE Communications magazine*, 40(10):70–75, 2002.

[19] D Down and Sean P Meyn. Piecewise linear test functions for stability and instability of queueing networks. *Queueing Systems*, 27(3-4):205–226, 1997.

[20] Parijat Dube and Rahul Jain. Bertrand games between multi-class queues. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 8588–8593. IEEE, 2009.

[21] Anthony Ephremides, P Varaiya, and Jean Walrand. A simple dynamic routing problem. *IEEE transactions on Automatic Control*, 25(4):690–693, 1980.

[22] Atilla Eryilmaz and Rayadurgam Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *IEEE/ACM Transactions on Networking (TON)*, 15(6):1333–1344, 2007.

[23] Patrick Eschenfeldt and David Gamarnik. Join the shortest queue with many servers. the heavy-traffic asymptotics. *Mathematics of Operations Research*, 43(3):867–886, 2018.

[24] S Rasoul Etesami and Tamer Başar. Dynamic games in cyber-physical security: An overview. *Dynamic Games and Applications*, pages 1–30, 2019.

[25] Awi Federgruen. On n-person stochastic games by denumerable state space. *Advances in Applied Probability*, 10(2):452–471, 1978.

[26] L Flatto and HP McKean. Two queues in parallel. *Communications on pure and applied mathematics*, 30(2):255–263, 1977.

[27] Robert D Foley and David R McDonald. Join the shortest queue: stability and exact asymptotics. *The Annals of Applied Probability*, 11(3):569–607, 2001.

[28] G Foschini and JACK Salz. A basic dynamic routing problem and diffusion. *IEEE Transactions on Communications*, 26(3):320–327, 1978.

[29] Serguei Foss and Natalia Chernova. On the stability of a partially accessible multi-station queue with state-dependent routing. *Queueing Systems*, 29(1):55–73, 1998.

[30] Robert G Gallager. *Stochastic processes: theory for applications*. Cambridge University Press, 2013.

[31] Branden Ghena, William Beyer, Allen Hillaker, Jonathan Pevarnek, and J Alex Halderman. Green lights forever: Analyzing the security of traffic infrastructure. In *8th {USENIX} Workshop on Offensive Technologies ({WOOT} 14)*, 2014.

[32] Gabriel Gomes and Roberto Horowitz. Optimal freeway ramp metering using the asymmetric cell transmission model. *Transportation Research Part C: Emerging Technologies*, 14(4):244–262, 2006.

[33] Jean Gregoire, Xiangjun Qian, Emilio Frazzoli, Arnaud De La Fortelle, and Tichakorn Wongpiromsarn. Capacity-aware backpressure traffic signal control. *IEEE Transactions on Control of Network Systems*, 2(2):164–173, 2014.

[34] Hang Guo, Xingwei Wang, Hui Cheng, and Min Huang. A routing defense mechanism using evolutionary game theory for delay tolerant networks. *Applied Soft Computing*, 38:469–476, 2016.

[35] Varun Gupta, Mor Harchol Balter, Karl Sigman, and Ward Whitt. Analysis of join-the-shortest-queue routing for web server farms. *Performance Evaluation*, 64(9-12):1062–1081, 2007.

[36] Bruce Hajek. Optimal control of two interacting service stations. *IEEE transactions on automatic control*, 29(6):491–499, 1984.

[37] Shlomo Halfin. The shortest queue problem. *Journal of Applied Probability*, 22(4):865–878, 1985.

[38] Alex Hern. Berlin artist uses 99 phones to trick google into traffic jam alert. *theguardian.com Available at: https://www.theguardian.com/technology/2020/feb/03/berlin-artist-uses-99-phones-trick-google-maps-traffic-jam-alert*, 2020.

[39] Morris W Hirsch. Systems of differential equations that are competitive or cooperative ii: Convergence almost everywhere. *SIAM Journal on Mathematical Analysis*, 16(3):423–439, 1985.

[40] Zhanfeng Jia, Chao Chen, Ben Coifman, and Pravin Varaiya. The pems algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*, pages 536–541. IEEE, 2001.

[41] Li Jin and Saurabh Amin. Stability of fluid queueing systems with parallel servers and stochastic capacities. *IEEE Transactions on Automatic Control*, 63(11):3948–3955, 2018.

[42] Li Jin and Saurabh Amin. Analyzing a tandem fluid queueing model with stochastic capacity and spillback. In *986th Transportation Research Board Annual Meeting*. TRB, 2019.

[43] Li Jin, Chen Feng, Qian Xie, and Xuchu Xu. Design of resilient smart highway systems with data-driven monitoring from networked cameras. 2020.

[44] Li Jin and Yining Wen. Behavior and management of stochastic multiple-origin-destination traffic flows sharing a common link. In *58th IEEE Conference on Decision and Control*. IEEE, 2019.

[45] FP Kelly and CN Laws. Dynamic routing in open queueing networks: Brownian models, cut constraints and resource pooling. *Queueing systems*, 13(1-3):47–86, 1993.

[46] PR Kumar and Sean P Meyn. Stability of queueing networks and scheduling policies. *IEEE Transactions on Automatic Control*, 40(2):251–260, 1995.

[47] Joy Kuri and Anurag Kumar. Optimal control of arrivals to queues with delayed queue length information. *IEEE Transactions on Automatic Control*, 40(8):1444–1450, 1995.

[48] Aron Laszka, Waseem Abbas, Yevgeniy Vorobeychik, and Xenofon Koutsoukos. Detection and mitigation of attacks on transportation networks as a multi-stage security game. *Computers & Security*, 87:101576, 2019.

[49] Edward A Lee. Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369. IEEE, 2008.

[50] Xiang Ling, Mao-Bin Hu, Rui Jiang, and Qing-Song Wu. Global dynamic routing for scale-free networks. *Physical Review E*, 81(1):016113, 2010.

[51] Steven A Lippman. Applying a new device in the optimization of exponential queuing systems. *Operations Research*, 23(4):687–710, 1975.

[52] John Lygeros, Datta N Godbole, and Mireille Broucke. A fault tolerant control architecture for automated highway systems. *IEEE Transactions on Control Systems Technology*, 8(2):205–219, 2000.

[53] Mohammad Hossein Manshaei, Quanyan Zhu, Tansu Alpcan, Tamer Bacşar, and Jean-Pierre Hubaux. Game theory meets network security and privacy. *ACM Computing Surveys (CSUR)*, 45(3):25, 2013.

[54] Saied Mehdian, Zhengyuan Zhou, and Nicholas Bambos. Join-the-shortest-queue scheduling with delay. In *2017 American Control Conference (ACC)*, pages 1747–1752. IEEE, 2017.

[55] Sean P Meyn. Sequencing and routing in multiclass queueing networks part i: Feedback regulation. *SIAM Journal on Control and Optimization*, 40(3):741–776, 2001.

[56] Sean P Meyn and Richard L Tweedie. Stability of markovian processes iii: Foster–lyapunov criteria for continuous-time processes. *Advances in Applied Probability*, 25(3):518–548, 1993.

[57] Prashant Mhaskar, Adiwinata Gani, Nael H El-Farra, Charles McFall, Panagiotis D Christofides, and James F Davis. Integrated fault-detection and fault-tolerant control of process systems. *AIChE Journal*, 52(6):2129–2148, 2006.

[58] Arpan Mukhopadhyay and Ravi R Mazumdar. Analysis of randomized join-the-shortest-queue (jsq) schemes in large heterogeneous processor-sharing systems. *IEEE Transactions on Control of Network Systems*, 3(2):116–126, 2015.

[59] Randolph D Nelson and Thomas K Philips. *An approximation to the response time for shortest queue routing*, volume 17. ACM, 1989.

[60] Yi Ouyang and Demosthenis Teneketzis. Signaling for decentralized routing in a queueing network. *Annals of Operations Research*, pages 1–39, 2015.

[61] Christos H Papadimitriou and John N Tsitsiklis. The complexity of optimal queueing network control. In *Proceedings of IEEE 9th Annual Conference on Structure in Complexity Theory*, pages 318–322. IEEE, 1994.

[62] Ron J Patton. Fault-tolerant control: the 1997 situation. *IFAC Proceedings Volumes*, 30(18):1029–1051, 1997.

[63] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[64] Ram Rajagopal, XuanLong Nguyen, Sinem Coleri Ergen, and Pravin Varaiya. Distributed online simultaneous fault detection for multiple sensors. In *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 133–144. IEEE Computer Society, 2008.

[65] Jack Reilly, Samitha Samaranayake, Maria Laura Delle Monache, Walid Krichene, Paola Goatin, and Alexandre M Bayen. Adjoint-based optimization on a network of discretized scalar conservation laws with applications to coordinated ramp metering. *Journal of optimization theory and applications*, 167(2):733–760, 2015.

[66] Fengyuan Ren, Tao He, Sajal K Das, and Chuang Lin. Traffic-aware dynamic routing to alleviate congestion in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(9):1585–1599, 2011.

[67] Fatih Sakiz and Sevil Sen. A survey of attacks and detection mechanisms on intelligent transportation systems: Vanets and iov. *Ad Hoc Networks*, 61:33–50, 2017.

[68] P Sarachik and U Ozguner. On decentralized dynamic routing for congested traffic networks. *IEEE Transactions on Automatic Control*, 27(6):1233–1238, 1982.

[69] Yunus Sarikaya, Tansu Alpcan, and Ozgur Ercetin. Dynamic pricing and queue stability in wireless random access games. *IEEE Journal of Selected Topics in Signal Processing*, 6(2):140–150, 2011.

[70] Ketan Savla and Emilio Frazzoli. A dynamical queue approach to intelligent task management for human operators. *Proceedings of the IEEE*, 100(3):672–686, 2011.

[71] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

[72] Lloyd S Shapley and RN Snow. Basic solutions of discrete games. *Contributions to the Theory of Games*, 1:27–35, 1952.

[73] Peng Shi and Fanbiao Li. A survey on markovian jump systems: modeling and design. *International Journal of Control, Automation and Systems*, 13(1):1–16, 2015.

[74] Hal L Smith. *Monotone Dynamical Systems: An Introduction to the Theory of Competitive and Cooperative Systems*. Number 41. American Mathematical Soc., 2008.

[75] Stephen L Smith, Marco Pavone, Francesco Bullo, and Emilio Frazzoli. Dynamic vehicle routing with priority classes of stochastic demands. *SIAM Journal on Control and Optimization*, 48(5):3224–3245, 2010.

[76] Gilbert Strang, Gilbert Strang, Gilbert Strang, and Gilbert Strang. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

[77] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[78] Xidong Tang, Gang Tao, and Suresh M Joshi. Adaptive actuator failure compensation for nonlinear mimo systems with an aircraft control application. *Automatica*, 43(11):1869–1883, 2007.

[79] Yu Tang and Li Jin. Analysis and control of dynamic flow networks subject to stochastic cyber-physical disruptions. *arXiv preprint arXiv:2004.00159*, 2020.

[80] Yu Tang, Yining Wen, and Li Jin. Security risk analysis of the shorter-queue routing policy for two symmetric servers. In *2020 American Control Conference (ACC)*, pages 5090–5095. IEEE, 2020.

[81] Don Towsley. Queuing network models with state-dependent routing. *Journal of the ACM (JACM)*, 27(2):323–337, 1980.

[82] JWC Van Lint, SP Hoogendoorn, and Henk J van Zuylen. Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research Part C: Emerging Technologies*, 13(5-6):347–369, 2005.

[83] Pravin Varaiya. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36:177–195, 2013.

[84] Nikita Dmitrievna Vvedenskaya, Roland L'vovich Dobrushin, and Fridrikh Izrailevich Karpelevich. Queueing system with selection of the shortest of two queues: An asymptotic approach. *Problemy Peredachi Informatsii*, 32(1):20–34, 1996.

[85] Manxi Wu and Saurabh Amin. Securing infrastructure facilities: When does proactive defense help? *Dynamic Games and Applications*, pages 1–42, 2018.

[86] Manxi Wu, Li Jin, Saurabh Amin, and Patrick Jaillet. Signaling game-based misbehavior inspection in v2i-enabled highway operations. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2728–2734. IEEE, 2018.

[87] Qian Xie and Li Jin. Resilience of dynamic routing in the face of recurrent and random sensing faults. In *2020 American Control Conference (ACC)*, pages 1173–1178. IEEE, 2020.

[88] Qian Xie and Li Jin. Stabilizing queuing networks with model data-independent control. *arXiv preprint arXiv:2011.11788*, 2020.

[89] Qian Xie and Li Jin. Strategic defense of feedback-controlled parallel servers against reliability and security failures. *Working Paper*, 2021.

[90] Huan Yu and Miroslav Krstic. Traffic congestion control for aw–rascle–zhang model. *Automatica*, 100:38–51, 2019.

[91] Rick Zhang, Federico Rossi, and Marco Pavone. Analysis, control, and evaluation of mobility-on-demand systems: a queueing-theoretical approach. *IEEE Transactions on Control of Network Systems*, 6(1):115–126, 2018.

[92] Xiaodong Zhang, Thomas Parisini, and Marios M Polycarpou. Adaptive fault-tolerant control of nonlinear uncertain systems: an information-based diagnostic approach. *IEEE Transactions on automatic Control*, 49(8):1259–1274, 2004.

[93] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual reviews in control*, 32(2):229–252, 2008.