C2SMART
CONNECTED CITIES WITH SMART TRANSPORTATION

A USDOT University Transportation Center

New York University

Rutgers University

University of Washington

The University of Texas at El Paso

City College of New York

Securing Intelligent Transportation Systems against Spoofing Attacks

April 2021

Securing Intelligent Transportation Systems against Spoofing Attacks

Li Jin
New York University Tandon School of Engineering
ORC-ID: 0000-0002-5282-2327

Qian Xie
New York University Tandon School of Engineering
ORC-ID: 0000-0002-3513-2674

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

Acknowledgements

We appreciate the support from the US Department of Transportation, the NYU Tandon School of Engineering faculty startup funds, and the NYU Undergraduate Summer Research Internship program. Undergraduate students including Dorothy Ng and Eliot C. Brown also contributed to this project. We also appreciate the assistance from Shri Iyer, Joseph C. Williams, and John Petinos.

Executive Summary

Modern transportation networks are subject to malicious spoofing attacks, i.e. injection of falsified, misleading data. In this project, we study the reliability/security risk of feedback-controlled queuing systems and propose methods for strategic defense. We consider a system of parallel servers and queues with dynamic routing subject to reliability and/or security failures. In the reliability setting, we study the impact of faulty routing (i.e. an incoming job not being allocated to the shortest queue) on the queuing cost. We show that the system is stable if and only if the probability of wrongly allocating jobs to a server is less than the ratio between the server's service rate and the total arrival rate. We derive sufficient condition for stability under state-dependent defending strategies, characterize structure of the optimal strategy, and develop a dynamic programming algorithm to compute the strategy. For the security setting, we formulate an attacker-defender game that characterizes possible security failure scenarios. The attacker selects the probability of modifying a job's allocation, and the defender selects the probability of defending a job's allocation. Both attacking and defending induce technological costs. For state-independent strategies, we show that the regimes are qualitatively different for low and high demands. For state-dependent strategies, we characterize the equilibria structure and propose an algorithm that numerically computes the security risk. We also present computational examples to illustrate the proposed models and methods. In addition, we study the behavior of transportation networks under strategic data spoofing and propose diagnosis and secure routing strategies. We consider a multi-class Jackson network with Bernoulli routing. The system operator (SO) decides the routing probabilities based on the traffic demand and service rates. An adversary is able to create non-existing, phantom traffic demand of any class. Phantom demand does not affect service of real traffic but can mislead the SO's routing decisions. We use a non-zero-sum sequential game to model the interaction between the players. We first characterize the attacker's decision and study network resiliency (in terms of throughput and queuing cost) in the absence of diagnosis and secure routing. Then, we design a diagnosis scheme to fully or partially recover the spoofed data. The diagnosis result further leads to a routing scheme optimizing the worst-case scenario. We also present a running example to illustrate the main concepts and results.

Table of Contents

List of Figures

Section 1: Introduction

Modern transportation networks are increasingly connected and autonomous, thanks to the introduction of real-time sensing and autonomous decision-making capabilities. However, the cyber components enabling connectivity and autonomy are subject to persistent security threats. A major threat is spoofing attacks via injection of falsified information. For transportation networks, a typical form of spoofing attack is to inject non-existent (i.e. ``phantom'') traffic into routing/navigation tools, and real incidents have been reported. Such attacks can create fake congestion, mislead routing decisions, and lead to real congestion or even system breakdown (traffic jam). Although such security risk has recently become well-known in the cyber-physical systems (CPS) community, the transportation community still lacks tools to systematically evaluate and respond to the risk.

Dynamic routing is a classical control strategy applicable to a variety of engineering systems, including transportation [1], production lines [2], and communications [3]. The idea of dynamic routing is that a job (e.g. a vehicle, a part, or a data packet) is allocated to a server with a shorter queue, which has been proved to be optimal if the system operator has perfect observation of the system states and perfect implementation of the policy [4]. Such sensing and actuating typically rely on cyber components connected via wired or wireless communications. Although connectivity can significantly improve throughput and reduce delay, it is vulnerable to random component malfunctions and malicious remote attacks and thus brings reliability and security risks. In intelligent transportation systems, researchers have shown that traffic sensors and traffic lights can be easily intruded and manipulated [5, 6]. Similar security risks also exist in production lines [7] and communication networks [8]. However, such risk has not been well modeled and studied in conjunction with the dynamics of the engineering systems, which is typically modeled as queuing processes.

In this project, we develop novel models and methods to evaluate the reliability/security risk of dynamic routing and to design an efficient deployment of protecting resources. We consider a parallel queuing system with a routing mechanism that is subject to faults due to hardware malfunctions or malicious attacks. The proposed approach quantifies the efficiency loss (in terms of queuing delay) due to reliability and/or security failures that occur randomly. We study both open-loop and closed-loop (i.e. queuing state-dependent) defending strategies that restrict the reliability/security risk while maintaining an acceptable budget. 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 resilient design of intelligent transportation systems, production lines, and communications.

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 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. A defender is able to deploy security resource to control the probability of faults. Deploying security resource 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 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 queuing 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 queuing system.

Numerous results have been developed for parallel queuing systems without sensing/ actuating faults [9, 4, 10, 11, 12, 13, 14]. Although some of these results provide hints for our problem, they do not directly apply to the setting with failures. Parallel queuing systems have been studied with delayed [15], erroneous [16], or decentralized information [17], which provides insights for our purpose. Previous work typically relies on characterization or approximation of the steady-state distribution of the queuing 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 queuing 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 queuing system and to obtain upper bounds for the mean number of jobs in the system. This approach has been applied to queuing systems in settings different from that in this paper [18, 19, 20]. Importantly, we use this approach to study the queuing dynamics under state-dependent defending strategies. Using an upper bound for queuing cost derived from the Foster-Lyapunov criterion [21], 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.

In addition, we study the resiliency of a class of transportation networks with respect to spoofing attacks. We consider a single-origin-single-destination (SOSD) open Jackson network with open-loop Bernoulli routing. Spoofing attacks can mislead the system operator's (SO's) routing decisions. We use a

queuing-theoretic approach to quantify the travel cost and a game-theoretic approach to characterize the interaction between the attacker and the SO.

There is extensive work on security of generic CPS, but very limited results are available for transportation systems. For generic CPS, a typical class of methods are game-theoretic approaches; these approaches usually do not involve physical dynamics of CPS. There are recent results on secure control for linear systems, but transportation systems are mostly nonlinear. Another class of methods is discrete-event systems (DES), which are more suitable for microscopic logical analysis rather than macroscopic quantitative/monetized analysis. The major challenge to bridge the gap between generic CPS security theory and transportation applications is the mapping from generic formulation to concrete transportation models and metrics. Among the limited transportation-specific results, Reilly et al. studied the potential for attackers to create specific congestion patterns via coordinated manipulation of ramp controllers. Wu et al. considered the impact of malicious electronic toll deception on system efficiency. Laszka et al. considered a scenario where an attacker can directly override traffic signaling and proposed heuristics for computing detection and mitigation strategies. However, the above results do not consider ``phantom traffic'' attacks and do not quantify key performance metrics. There also exists work on non-strategic network failures, which provides insights for the strategic setting in this paper.

We consider a Stackelberg leader-follower game over an SOSD Jackson network. The SO determines the Bernoulli routing probabilities according to the observed traffic demand at the origin and the service rate of each node. This decision is open-loop and cannot be adjusted in response to real-time traffic state. The attacker can inject a certain amount of phantom traffic at the origin and can arbitrarily route the phantom traffic over the network. Phantom traffic does not consume service capabilities of servers. The SO cannot differentiate the actual and phantom demand at the origin, so the SO will make decisions based on the sum thereof. In the security game, the SO moves first with the objective to minimize the worst-case queuing cost. The attacker moves second with the objective to maximize the queuing cost. We use a Wheatstone bridge network to illustrate the major insights about the secure routing problem. In addition, 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 [22, 23, 24]. Game theoretic approaches have been applied to studying security of routing in transportation [25, 26] and communications [27, 28]. 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 queuing theory, which is essential for capturing the interaction between the queuing dynamics and the players' decisions. For open-loop attacking and defending strategies, we quantitatively characterize the security risk (in terms of attack-induced queuing 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 queuing 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.

Section 2: Model setup

Subsection 2.1 Queuing model

Consider a parallel queuing system. Jobs arrive according to a Poisson process of rate λ. Each server serves jobs at an exponential rate of μ. We use $X(t) = [X_1(t), X_2(t), \dots, X_K(t)]^{\mathrm{T}}$ to denote the number of jobs, either waiting or being served, in the n servers, respectively.

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.

Subsection 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 $1/n$. Fig. 1 illustrates the routing in the presence of reliability failures.

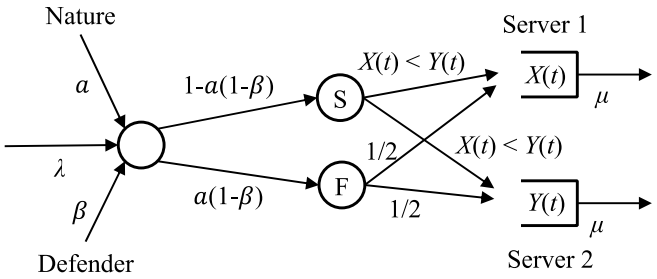

Figure 1: Two-queue System with Shorter-queue Routing under Reliability Failures.

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

The objective of the defender is to balance the queuing cost and the defending cost. We formulate this problem as an infinite-horizon continuous-time Markov decision process.

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

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

where ρ is the discounted factor and C is the immediate cost defined as:

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

Subsection 2.3 Security failures

Suppose that a malicious attacker can compromise the operator's dynamic routing. When a job arrives and is being allocated, the attacker can 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 α, where $\alpha(x)$ is selected by the attacker. Fig. 2 illustrates the routing in the presence of reliability failures.

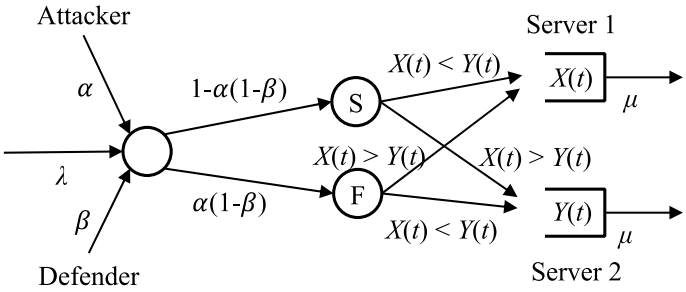

Figure 2: Two-queue System with Shorter-queue Routing under Security Failures.

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:

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

where $R: \mathbb{Z}_{\geq 0}^n \to \mathbb{R}$ 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:

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

Section 3: Robust routing

Subsection 3.1 Stability criteria

Proposition 1. The unprotected n-queue system is stable if and only if

$$\lambda < n\mu. \tag{1}$$

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

$$\bar{X} \coloneqq limsup_{t\to\leftarrow} \sum_{s=0}^{t} \mathbb{E}\big[f\big(X(s)\big)\big] \leq \frac{\lambda + n\mu}{2(\mu - \lambda/n)}.$$

Theorem 1. 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 β. Then n-queue system is stable if there exists a compact set $\chi_0 = [0,\theta]^n$ such that for any $x \in \chi_0^c$, when $x_{min}/|x| < 1/n$,

$$\beta(x) > 1 - \frac{\mu - \lambda x_{min}/|x|}{a\lambda\left(\frac{1}{n} - \frac{x_{min}}{|x|}\right)}, \tag{1}$$

where $x_{min} = \min_i x_i$ and $|x| = \sum_{i=1}^n x_i$. Furthermore, any equilibrium (α^*, β^*) must satisfy the above, and the number of jobs is upper bounded by

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

where $c = \min_x \mu - \lambda x_{min}/|x| - a\big(1-\beta(x)\big)\lambda(1/n - x_{min}/|x|)$.

Subsection 3.2 Optimal dynamic routing

The Hamiltonian-Jacobi-Bellman equation (derived from Kolmogorov equation) of the dynamic programming can be written as [29]

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

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

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

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

Definition 1. *The optimal defending policy is defined as*

$$\beta^* = argmin_\beta J(x,\beta).$$

Remark 1. *When* $x_1 = x_2 = \cdots = x_n$, $\beta^*(x) = 0$.

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

$$\begin{aligned}(\rho + \lambda + n\mu)J^*(x) \\ = min_{b\in 0,1} \Big\{ |x| + c_b b + \mu \sum_i J^*((x-e_i)^+) + \lambda min_j J^*\big(x+e_j\big) \\ + (1-b)\frac{a}{n}\lambda \sum_i \Big(J^*(x+e_i) - min_j J^*\big(x+e_j\big)\Big)\Big\}\end{aligned} \tag{3}$$

Based on the uniformization trick, we can assume $\rho + \lambda + n\mu = 1$ without loss of generality.

The main theorem of this section is given below.

Theorem 2. The optimal defending policy β^* is a threshold policy characterized by n non-intersecting symmetric monotonically non-decreasing threshold functions (see Figure 3).

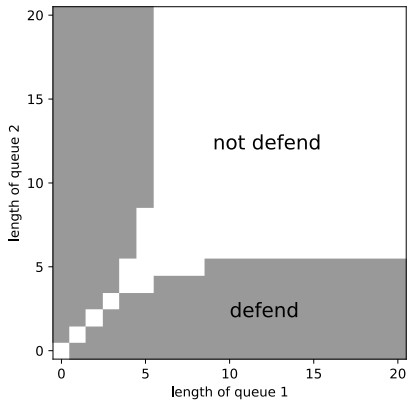

Figure 3: Two-queue System with Shorter-queue Routing under Security Failures.

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

Subsection 3.3 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 the infinitesimal generator, we have

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

We can relax it to

$$\mathcal{L}W(x) \le \left(\frac{\lambda}{n} - \mu\right)|x| + \frac{1}{2}(\lambda + n\mu).$$

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

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

By [21, Theorem 4.3], the above implies (3) and thus stability.

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

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

Again, we can relax it to:

$$\mathcal{L}W(x) = a\big(1-\beta(x)\big)\lambda\left(\frac{|x|}{n}-x_{min}\right) + (\lambda x_{min} - \mu|x|) + \frac{1}{2}(\lambda+n\mu).$$

Hence, by (2) there exists a constant $c = \min_x \mu - \lambda x_{min}/|x| - a\big(1-\beta(x)\big)\lambda(1/n - x_{min}/|x|) > 0$ and $d = \frac{1}{2}(\lambda+n\mu)$ such that:

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

By [21, Theorem 4.3], the above implies (3) and thus stability.

Subsection 3.3 Proof of stability criteria

Based on the symmetry, without loss of generality, we only need to consider the case when $x_n = \min_i x_i$. To demonstrate the existence of the threshold policy, we will show that $\beta^*(x)$ is monotonically non-decreasing in x_i $(i<n)$ when other variables are fixed and monotonically non-increasing in x_n when other variables are fixed, that is,

$$\begin{aligned}\beta^*(x+e_i) &\ge \beta^*(x) \quad (\forall i<n)\\ \beta^*(x+e_n) &\le \beta^*(x).\end{aligned} \tag{4}$$

Because of Schur convexity, $J^*(x+e_i) \ge J^*(x+e_n)(\forall\, i<n)$. We can rewrite (3) as

$$J^*(x) = \min_{b\in 0,1}\Bigg\{|x| + c_b b + \mu\sum_{i=1}^{n}J^*((x-e_i)^+) + \lambda J^*(x+e_n) + (1-b)\frac{a}{n}\lambda\left[\sum_{i=1}^{n}J^*(x+e_i) - nJ^*(x+e_n)\right]\Bigg\}.$$

Let $\Delta(x) = \sum_{i=1}^{n}J^*(x+e_i) - nJ^*(x+e_n)$, then (4) is essentially

$$\begin{aligned}\Delta(x+e_i) &\geq \Delta(x) \quad (\forall i < n)\\ \Delta(x+e_n) &\leq \Delta(x).\end{aligned} \tag{5}$$

We will use induction based on value iteration to prove (5), that is, let $\Delta^{(k)}(x) = \sum_{i=1}^{n} J^{(k)}(x+e_i) - nJ^{(k)}(x+e_n)$, it is sufficient to show

$$\begin{aligned}\Delta^{(k)}(x+e_i) &\geq \Delta^{(k)}(x) \quad (\forall i < n)\\ \Delta^{(k)}(x+e_n) &\leq \Delta^{(k)}(x),\end{aligned} \tag{6}$$

for all k.

Base step. Initially, with $J^{(0)} = 0$, we have $\Delta^{(0)} = 0$, then (6) holds trivially because it consists of two equalities. We can run more iterations, say $J^{(1)}(x) = |x|$ and $J^{(2)}(x) = (1+\lambda+n\mu)|x| + \lambda - \mu\sum_i \mathbb{I}_{x_i>0}$, until (6) includes some strict inequalities and holds non-trivially.

Induction step. Based on the induction hypothesis, we have $\forall j < n$,

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

$$\Delta^{(k)}\left(x+e_j+e_i\right) \geq \Delta^{(k)}(x+e_i) \geq \Delta^{(k)}(x+e_n+e_i),$$

$$\Delta^{(k)}\left(x+e_j+e_n\right) \geq \Delta^{(k)}(x+e_n) \geq \Delta^{(k)}(x+2e_n).$$

Then according to the value iteration form of (3), we can conclude

$$\Delta^{(k+1)}\left(x+e_j\right) \geq \Delta^{(k+1)}(x) \quad (\forall j < n)$$

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

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

Section 4: Secure routing

Subsection 4.1 Stability results

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

Theorem 3. *Consider the n-queue system subject to attacks. The attacker (resp. defender) follows a Markovian strategy $\alpha: \mathbb{Z}_{\geq 0}^n \to [0,1]$ (resp. $\beta: \mathbb{Z}_{\geq 0}^n \to [0,1]$). Then n-queue system is stable if there exists a compact set $\chi_0 = [0,\theta]^n$ such that for any $x \in \chi_0^c$, when $x_{min} \neq x_{max}$,*

$$\alpha(x)\big(1-\beta(x)\big) < \frac{\mu - \lambda x_{min}/|x|}{\lambda(x_{max}-x_{min})/|x|}, \tag{7}$$

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*

$$\bar{X} \leq \frac{\lambda + n\mu}{2c}, \tag{8}$$

where $c = \min_x\{\mu - \lambda x_{min}/|x| - \alpha(x)\big(1-\beta(x)\big)\lambda(x_{max}-x_{min})/|x|\}$.

Subsection 4.2 Stochastic security game

Definition 2. *The optimal attacking (resp. defending) strategy* α^* *(resp.* β^**) satisfies that for each state* $x \in \mathbb{Z}_{\geq 0}^n$,

$$\alpha^*(x) = argmax_\alpha V_A^*(x, \beta^*),\ \beta^*(x) = argmin_\beta V_B^*(x, \alpha^*).$$

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

proof. Note that the state space $\mathbb{Z}_{\geq 0}^n$ is countable and the action space $[0,1]$ is compact. By [32], the total-discounted return equilibrium policy exists.

According to Shapley's extension on minimax theorem for stochastic game [33],

$$V_B^*(x, \alpha^*) = V_A^*(x, \beta^*) = V^*(x).$$

Again by using the uniformization trick and assuming $\rho + \lambda + n\mu = 1$ we get

$$V^*(x) = \max_\alpha \min_\beta \Big\{ |x| + c_b\beta(x) - c_a\alpha(x) + \mu \sum_i V^*((x-e_i)^+) + \lambda min_j V^*\big(x+e_j\big) + \alpha(x)\big(1-\beta(x)\big)\lambda\Big(max_j V^*\big(x+e_j\big) - min_j V^*\big(x+e_j\big)\Big)\Big\}. \tag{9}$$

The main theorem of this section is given below.

Theorem 4. *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* $\delta^*(x) = \lambda\left(\max_j V^*(x + e_j) - \min_j V^*(x + e_j)\right)$

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 and Figure 5).*

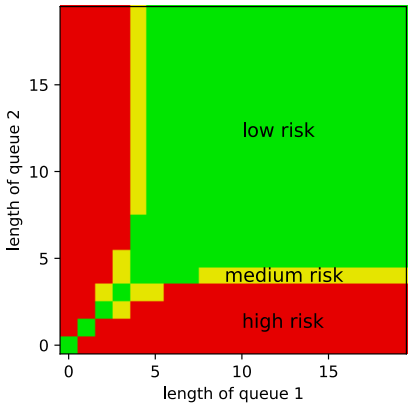

Figure 4: The Equilibria Regimes of the Stochastic Security Game for a Two-queue System.

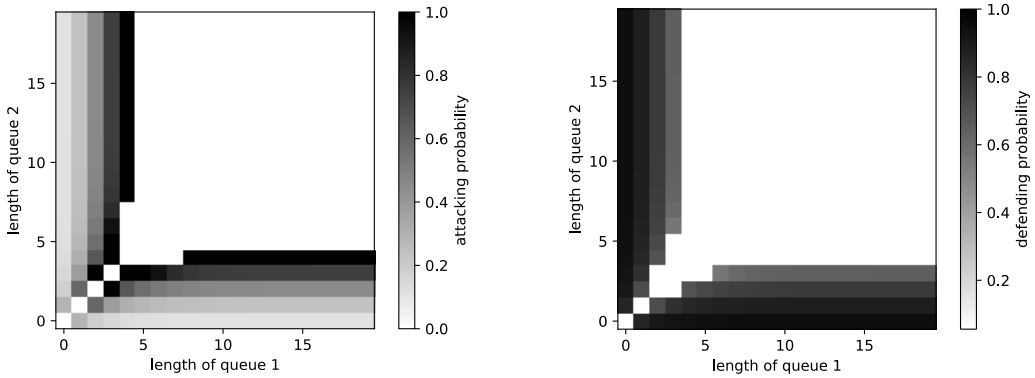

Figure 5: The Optimal Attacking and Defending Strategies for a Two-queue System.

Subsection 4.3 Proof of stability of results

Consider the quadratic Lyapunov function

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

By applying infinitesimal generator, we have

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

Again we can relax it to

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

Hence, by (2) there exists a constant $c = \min_x \mu - \lambda x_{min}/|x| - a\big(1-\beta(x)\big)\lambda(1/n - x_{min}/|x|) > 0$ and $d = \frac{1}{2}(\lambda + n\mu)$ such that

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

By [21, Theorem 4.3], the above implies (8) and thus stability.

Subsection 4.4 Proof of Theorem 4

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, $V(x+e_1) = \max_j V\big(x+e_j\big)$, $V(x+e_n) = \min_j V\big(x+e_j\big)$. We can rewrite (9) as

$$V^*(x) = \max_{\alpha}\min_{\beta}\Big\{|x| + c_b\beta(x) - c_a\alpha(x) + \mu\sum_i V^*((x-e_i)^+) + \lambda V^*(x+e_n)$$
$$+\alpha(x)\big(1-\beta(x)\big)\lambda\big(V^*(x+e_1) - V^*(x+e_n)\big)\Big\}. \qquad (10)$$

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,

$$\mathcal{D}(x+e_1) \ge \mathcal{D}(x), \quad \mathcal{D}(x+e_n) \le \mathcal{D}(x).$$

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

$$\mathcal{D}^{(k)}(x+e_1) \ge \mathcal{D}^{(k)}(x), \quad \mathcal{D}^{(k)}(x+e_n) \le \mathcal{D}^{(k)}(x).$$

Base step. Similar to the base step in the proof of Theorem 2.

Induction step. 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 according to the value iteration form of (10), we can conclude $\mathcal{D}^{(k+1)}(x+e_1) \geq \mathcal{D}^{(k+1)}(x)$ 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.

Section 5: Strategic secure routing

Subsection 5.1: Modeling and formulation

Consider an open acyclic Jackson network with a set of nodes $\mathcal{N}$ and a set of links $\mathcal{E}$; see Figure 6 for an example.

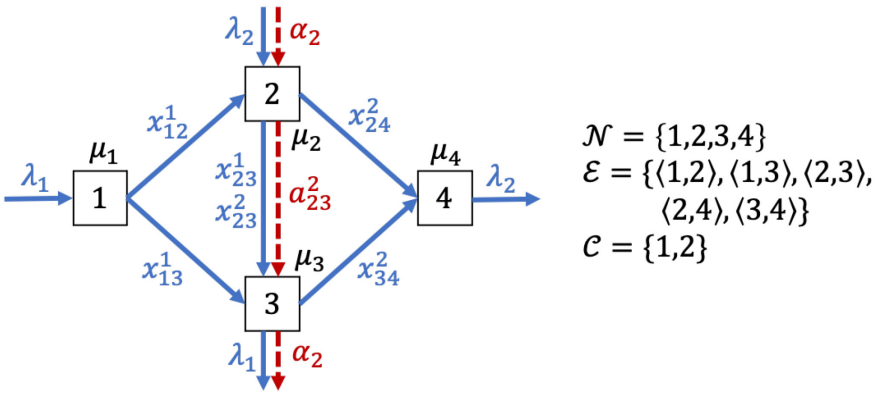

Figure 6: A Multi-class Network.

Each job is assigned an origin-destination (OD) pair (o_c, d_c), where o_c is the index of the origin, d_c is the index of the destination, and c is the *class* of the job. We assume that a job's class only determines its OD pair but does not affect the service time. Let $\mathcal{C}$ be the set of classes. Class-c jobs arrive at node o_c according to a Poisson process of rate λ_c . Let $\mathcal{N}_c$ be the set of nodes from which node d_c is accessible.

Each link contains a server indexed by $\langle i,j\rangle$, where i (resp. j) is the node upstream (resp. downstream) to the server. We assume Bernoulli routing; i.e., at each node i, a class-c job is routed to a downstream server $\langle i,j\rangle$ with probability p_{ij}^c. Equivalently, we can describe the routing using the average flow x_{ij}^c, which should satisfy the following constraints:

$$\sum_{j:\langle j,i\rangle\in\mathcal{E}} b_{ij}^c + \mathbb{I}_{i=o_c}\lambda^c = \sum_{k:\langle i,k\rangle\in\mathcal{E}} x_{ik}^c \quad , \quad i\in\mathcal{N}, c\in\mathcal{C}, \tag{1a}$$

$$\sum_{c\in\mathcal{C}} x_{ij} < \mu_j,\ \langle i,j\rangle\in\mathcal{E}, \tag{1b}$$

$$x_{ij}^c \geq 0,\ \langle i,j\rangle\in\mathcal{E}, c\in\mathcal{C}. \tag{1c}$$

Thus, one can obtain the routing probabilities by

$$p_{ij}^c = \frac{x_{ij}^c}{\sum_{j:\langle i,k\rangle\in\mathcal{E}} x_{ik}^c},\ c\in\mathcal{C}, \langle i,j\rangle\in\mathcal{E}. \tag{2}$$

Let $\mathcal{X}$ be the set of x satisfying the above constraints. The network is *stabilizable* if $\mathcal{X}\neq\emptyset$. By standard results from queuing theory [1], the *queuing cost* is given by

$$q(x) = \begin{cases} \sum_{j:\langle i,k\rangle\in\mathcal{E}} \frac{\sum_{c\in\mathcal{C}} x_{ij}^c}{\mu_{ij} - \sum_{c\in\mathcal{C}} x_{ij}^c} & x\in\mathcal{X} \\ \infty & \text{otherwise.} \end{cases}$$

Next, we specify how the attacker and the SO jointly influence the flow x_{ij}^c and thus the queuing cost $q(x)$.

An external adversary (attacker) can create *phantom demand* into the system. Phantom demand spoofs the SO's knowledge about the traffic demands but does not affect the service of actual jobs. That is, for each class $c\in\mathcal{C}$, the attacker can create class-c phantom jobs at rate $a^c\geq 0$ via node o_c. Hence, the attacker's action is fully characterized by vector $a\in\mathbb{R}_{\geq 0}^{|\mathcal{C}|}$. The attacker's action is constrained by its budget $\|a\|_1\leq\bar{a}$.

The SO's action is to select the Bernoulli routing probabilities at each node based on the demand pattern. This is equivalent to specify the vector $b=\left[b_{ij}^c\right]_{c\in\mathcal{C},\langle i,j\rangle\in\mathcal{E}}$, where b_{ij}^c is the class-c flow through link (server) $\langle i,j\rangle$. We assume that the SO cannot distinguish between actual and phantom jobs. Thus, the SO only sees a demand of

$$\widehat{\lambda^c} = \lambda^c + a^c$$

at node o_c for $c \in \mathcal{C}$ and selects the routing probabilities accordingly. Hence, the SO's action $b \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|\times|\mathcal{C}|}$ satisfies

$$\sum_{j:\langle j,i\rangle\in\mathcal{E}} b_{ij}^c + \mathbb{I}_{i=o_c}\lambda^c + \mathbb{I}_{i=o_c}a^c = \sum_{k:\langle i,k\rangle\in\mathcal{E}} b_{ik}^c,\ i \in \mathcal{N}, c \in \mathcal{C}, \quad (3a)$$

$$\sum_{c\in\mathcal{C}} b_{ij} < \mu_j,\ \langle i,j\rangle \in \mathcal{E}, \quad (3b)$$

$$b_{ij}^c \geq 0,\ \langle i,j\rangle \in \mathcal{E}, c \in \mathcal{C}. \quad (3c)$$

Then, the routing probabilities are given by

$$p_{ij}^c = \frac{b_{ij}^c}{\sum_{j:\langle i,k\rangle\in\mathcal{E}} b_{ik}^c},\ c \in \mathcal{C}, \langle i,j\rangle \in \mathcal{E}. \quad (4)$$

Note that the SO's action b is in general not equal to the average flow x defined in Section 1.1. To differentiate, we call x the *actual flow* and call b the *observed flow*. By (1)-(4), the actual and observed flows are related by

$$x_{ij}^c(a,b) = \frac{\lambda^c}{\lambda^c+a^c} b_{ij}^c,\ \langle i,j\rangle \in \mathcal{E}, c \in \mathcal{C}. \quad (5)$$

Consequently, the queuing cost q also depends on the attacker's and the SO's actions as well as the demand vector λ. Hence, we use the notation $q(a,b|\lambda)$ to emphasize such dependency.

We model the interaction between the attacker and the SO as a non-zero-sum sequential game, where the attacker moves first.

Stage I: Attacker moves. The attacker assumes that the SO would fully trust the observed demand $\hat{\lambda} = \lambda + a$ and optimize the routing accordingly. That is, the attacker assumes that the SO would determine the flows b by solving the following optimization problem:

$$\min_{b\in\mathcal{X}(\lambda+a)} q(0,b|\lambda+a). \quad (6)$$

In other words, the attacker believes that the SO is "naïve". Let $\mathcal{B}^*$ be the set of optimal solutions to the above problem. The attacker's objective is to maximize the queuing cost conditional on the above assumptions. That is, the attacker determines its action a by solving

$$\max_{a:\|a\|_1\leq\bar{a}} \min_{b\in\mathcal{B}^*} q(a,b|\lambda).$$

The attacker's move ends here. We assume that the attacker is unaware of any diagnosis or secure routing actions for the SO.

Stage II: SO moves. The SO first diagnoses the demand data and then route the traffic accordingly. We assume the following for the SO.

1. The SO knows the attacker's budget $\bar{a}$.
2. The SO knows that the attacker believes the SO's naivety in the sense of (6).
3. The SO knows the attacker's objective specified by (7).

Based on the above assumptions, the SO applies some diagnosis tool to the observed demand $\hat{\lambda}$ and obtains a set of possible true values $\mathcal{L}(\hat{\lambda})$ for the actual demand λ. Then, the SO optimizes the worst-case queuing cost as follows.

$$\min_{b\in\left(\cap_{\nu\in\mathcal{L}(\hat{\lambda})}\mathcal{X}(\nu)\right)} \max_{\nu\in\mathcal{L}(\hat{\lambda})} q(0,b|\nu).$$

That is, the SO's final action b must stabilize the network for all possible values for the demand. If $\left(\cap_{\nu\in\mathcal{L}(\hat{\lambda})}\ \mathcal{X}(\nu)\right)\neq\emptyset$, the network is said to be *securely stabilizable*, which means that the SO can guarantee network stability regardless of the attacker's action. If $\left(\cap_{\nu\in\mathcal{L}(\hat{\lambda})}\ \mathcal{X}(\nu)\right)$, the network is said to be *not securely stabilizable*, which means that the SO **does not know** whether stability is guaranteed or not.

Remark 1 *"Not securely stabilizable" is in general not equivalent to "unstabilizable". Note that a network's stabilizability depends on the actual demand pattern. If the SO does not know actual demand pattern exactly, then the SO may not know the network's stabilizability.*

Subsection 5.2: Attacker's move

In this section, we study the attacker's action and analyze a network's security risk in the absence of diagnosis and secure routing.

Specifically, we study the following two questions.

1. Stability: when can the attacker make the network appear to be unstabilizable to the SO?

2. Optimality: if the network must appear to be stabilizable to the SO, how will the attacker optimize its action?

In general, the more budget the attacker has, the more likely that the network will appear to be unstabilizable to the SO. We can quantify the ``stabilizability'' via a threshold \bar a^* such that the

attacker can make the network appear to be unstabilizable if and only if \bar a\ge \bar a^*. A formal definition of \bar a^* is as follows.

The destabilizing budget $\bar{a}^*$ of a network with demand λ is given by

$$\bar{a}^* = \max_x \sum_c a^c$$

s.t. $\sum_{\{j:<j,i>\in E\}} b_{ij}^c + \sum_{c\in C} I_{i=o_c}(\lambda^c + a^c) = \sum_{k:<i,k>\in E} b_{ik}^c,\ i \in N, c \in C$

$\sum_{\{c\in C\}} b_{ij}^c \le \mu_{ij}, < i,j >\in E,$

$$b_{ij}^c \ge 0,$$

$< i,j >\in E, c \in C.$

In the above optimization, the parameters are λ^c and μ_{ij}, and the decision variable b_{ij}^c and a^c. Since $0 \le b_{ij}^c \le \mu_{ij}$ and $\lambda^c > 0$, it is apparent that a^c are bounded. Hence, the above linear programming must have an optimal solution, and $\bar{a}^*$ must exist. The use of $\bar{a}^*$ is to determine whether the attacker's decision-making problem has an optimal solution: if $a \ge \bar{a}^*$, then there exists an action for the attacker such that the queuing cost is infinite, and thus the naïve SO's decision making has no optimal solution.

Since α^* is given by a linear programming, it can be efficiently computed by the simplex method. Furthermore, we can derive ``closed-form'' (i.e. not requiring solving LPs) bounds thereon. For each class c, we can identify a set of routes R_c that start from o_c and ends at d_c. We use $i \in r$ to denote that node i is on route r. For each class c, we can also identify a set of cuts K_c that separates the origin o_c and the destination d_c. Then, we can compute an upper bound on the throughput score as follows:

Proposition Consider an acyclic Jackson network $< N, E >$ with actual demand β and service rate μ. Then, the network's resiliency score α^* is bounded by

$$\min_{<i,j>\in E}\left(\mu_{ij} - \sum_{c\in C}\frac{|\{r \in R_c: < i,j >\in r|}{|R_c|}\lambda^c\right)_+ \le \alpha^* \le \min_c \min_K \left(\sum_{<i,j>\in K} \mu_{ij} - \lambda^c\right).$$

where $(.)_+$ denotes the positive part of a number and |.| denotes the cardinality of a set.

Proof.

To obtain the lower bound, consider the action b for the SO:

$$b_{ij}^c = \frac{|\{r \in R_c: < i,j >\in r\}|}{|R_c|}(\lambda^c + a^c), < i,j >\in E.$$

`

That is, the SO attempts to distribute class-c traffic evenly over the routes R_c available to this class. Then, if the capacity constraints were dropped, the actual flow can be obtained from

$$x_{ij}^c(a,b) = \frac{|r \in R_c: < i,j > \in r|}{|R_c|}\lambda^c.$$

Hence, the total flow through link $< i,j >$ is

$$x_{ij} = \sum_c \frac{|\{r \in R_c: < i,j > \in r\}|}{|R_c|}\lambda^c.$$

Hence, if $\alpha < \mu_{ij} - x_{ij}$ for each $< i,j > \in E$, the attacker cannot destabilize the network; this leads to the lower bound.

The upper bound essentially results from the classical max-flow-min-cut theorem. For any class c with origin o_c and destination d_c, the min-cut capacity is given by

$$K_{\min}^c = \min_{K \in K_c} \sum_{<i,j> \in K} \mu_{ij}.$$

Hence, if $\bar{a} + \lambda^c \geq K_{min}^c$, the attacker can let $a^c = \bar{a}$and mislead the SO to believe that the network cannot accommodate the class-c demand. The above arguments maintain true if the influence of other classes is also considered. (QED)

The example below illustrates the above results. Consider the Wheatstone-bridge network in the figure below. Suppose that all servers have a service rate of 1 and that $\lambda^1 = \lambda^2$. The total demand is$|\lambda|_1 = \lambda^1 + \lambda^2$, i.e. the 1-norm of the vector λ. The figure shows how the bounds and the exact resiliency score varies as the total demand increases.

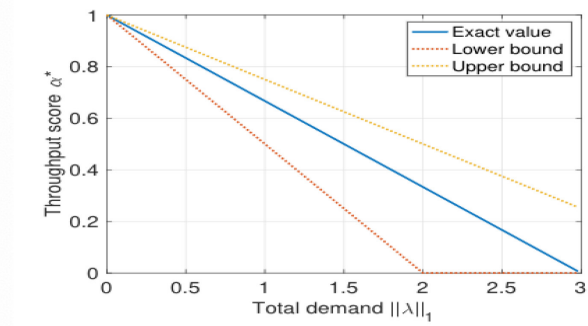

Figure 7: Explicit Bounds and Exact Value for Destabilizing Budget of Wheatstone-bridge Network.

One can check that the network is stabilizable if and only if $|\lambda|_1 < 3$. (Note that we force $\lambda^1 = \lambda^2 = |\lambda|_1/2$) Hence, the throughput score vanishes as $|\lambda|_1$ approaches 0. In this particular example, since the demands at the two origins are set equal, the throughput score happens to be linear in the total demand. Such linearity is not guaranteed in general.

Remark. The bounds are tight for single-class networks with homogeneous parallel servers.

There are two incentives for attacking resource allocation, which constitute the cost function.

One is to mislead the SO to route more resources to the sideways (rather than the shared path). This is realized because the queueing cost grows rapidly as more flow are allocated to it. As the observed inflow increases, the SO tend to route more to the sideways. Consider an extreme case for digestion where the system is close its limit. The SO will approximately route 1/3 to the central path and the 2/3 to the sideways, while SO tend to route a larger proportion to the central path when the observed in flow is small. We conjecture that the optimal strategy of this attacking incentive is to allocate resources equally to two nodes.

The other attacking incentive is to distort the sharing on the shared path. If one side presents more observed inflow, that side tend to share a larger portion on the shared path (i.e., the other side will route more to the sideway). Intuitively, the optimal strategy is to allocate all the resources to a single node.

As a result, the convexity of the objective function is inconclusive. When $\bar{a}$ is relatively large (comparing to λ), the first incentive is more influential, while the second is dominating otherwise. The figure below generated by simulation illustrates the idea.

The first set of figures are examples where the first incentive is more influential. The parameters adopted in this simulation is $\mu = 1, \bar{a} = 1, \lambda_2 = 0.2$. The real flow of the first node λ_1 are $0.1, 0.15, 0.2$, respectively from left to right. It can be seen that the cost function presents no convexity. There is no clear conclusion which node should be attacked in this case.

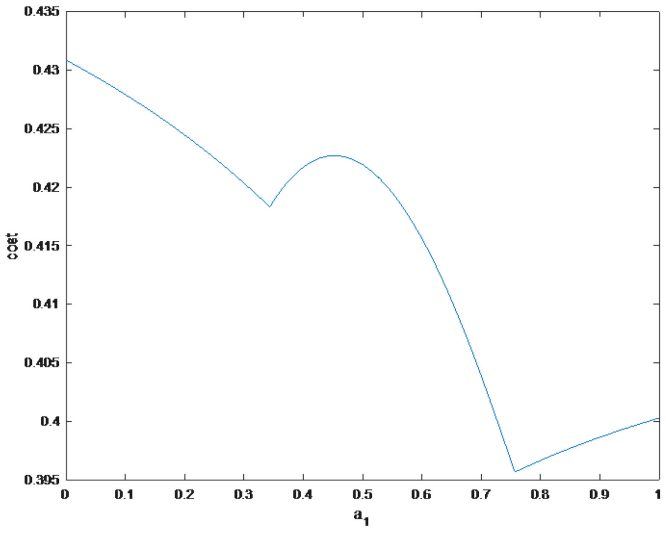

Figure 8a: How the Cost Varies with the Attacker's Action ($\mu = 1, \bar{a} = 1, \lambda_1 = 0.1, \lambda_2 = 0.2$).

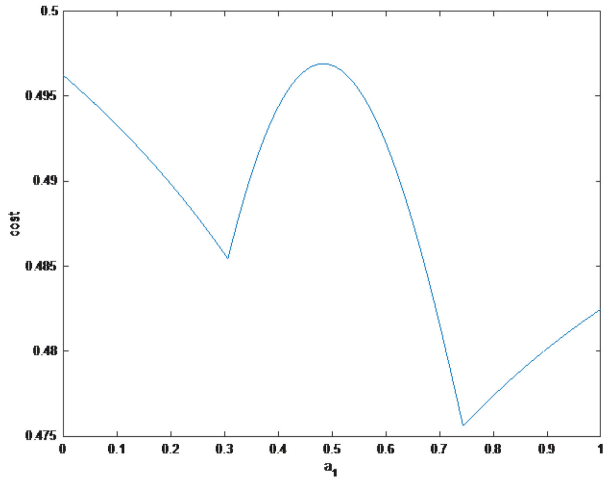

Figure 8b: How the Cost Varies with the Attacker's Action ($\mu = 1, \bar{a} = 1, \lambda_1 = 0.15, \lambda_2 = 0.2$).

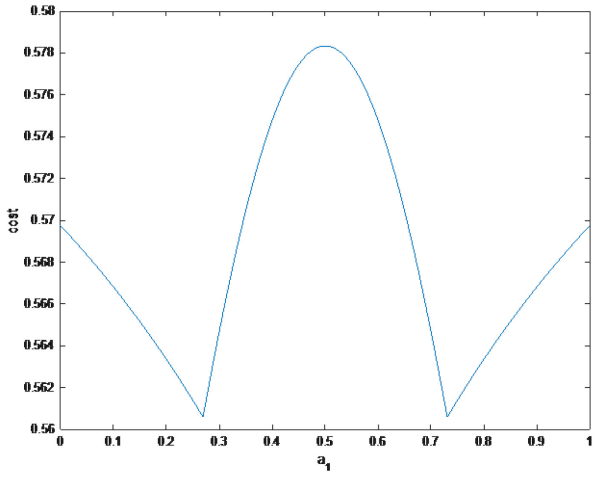

Figure 8c: How the Cost Varies with the Attacker's Action ($\mu = 1, \bar{a} = 1, \lambda_1 = 0.2, \lambda_2 = 0.2$).

The second set of figures are examples where the second incentive began to gain impact. The parameters adopted in this simulation is $\mu = 1, \bar{a} = 1, \lambda_2 = 0.5$. The real flow of the first node λ_1 are $0.3, 0.4, 0.5$, respectively from left to right. It can be seen that the cost function is quasi-convex, though not convex. The optimal attacking strategy is thus to allocate all resources to a single node (the less node).

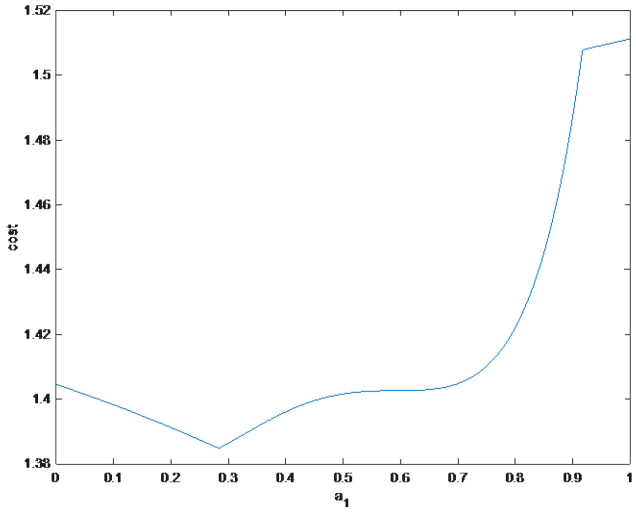

Figure 9a: How the Cost Varies with the Attacker's Action ($\mu = 1, \bar{a} = 1, \lambda_1 = 0.3, \lambda_2 = 0.5$).

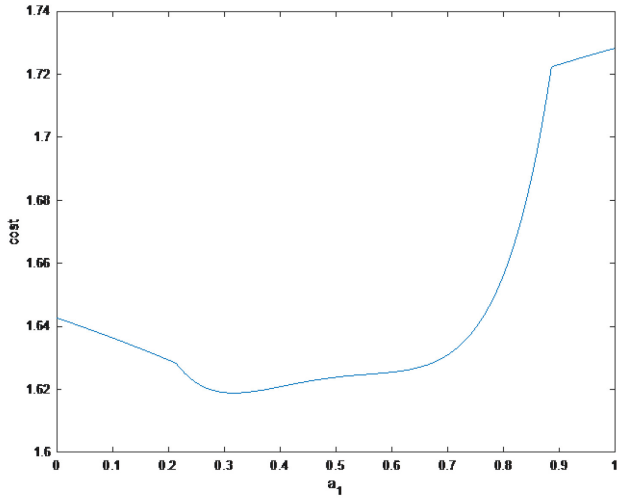

Figure 9b: How the Cost Varies with the Attacker's Action ($\mu = 1, \bar{a} = 1, \lambda_1 = 0.4, \lambda_2 = 0.5$).

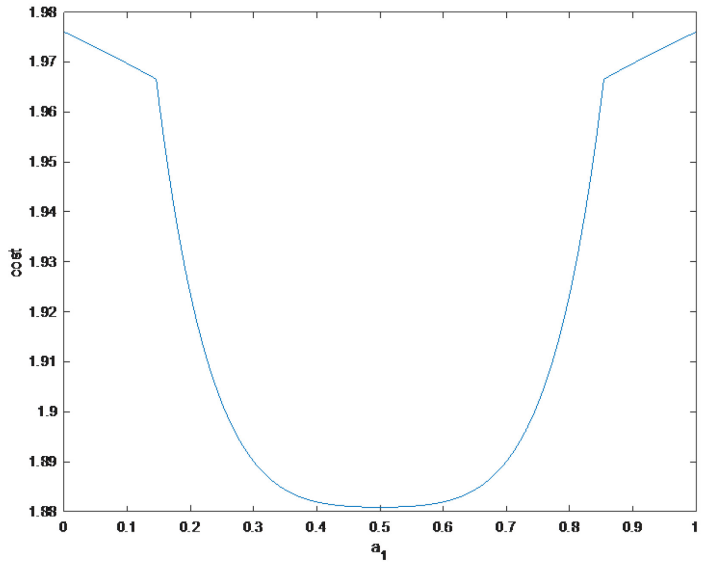

Figure 9c: How the Cost Varies with the Attacker's Action ($\mu = 1, \bar{a} = 1, \lambda_1 = 0.5, \lambda_2 = 0.5$).

Finally, when $\bar{a}$ is small relative to λ, the seocnd impact is dominating. The parameters adopted in this simulation is $\mu = 1, \bar{a} = 0.2, \lambda_2 = 0.5$. The real flow of the first node λ_1 are $0.3, 0.4, 0.5$, respectively from left to right. It can be seen that the cost function is convex. The optimal attacking strategy is still to allocate all resources to a single node (the less node).

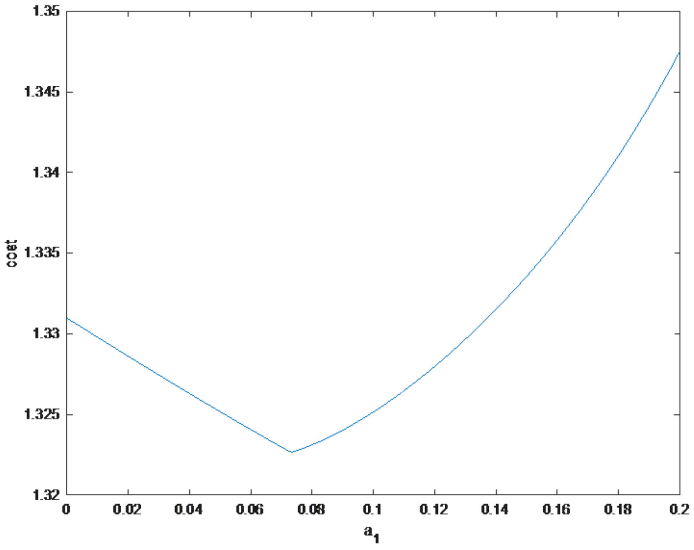

Figure 10a: How the Cost Varies with the Attacker's Action ($\mu = 1, \bar{a} = 0.2, \lambda_1 = 0.3, \lambda_2 = 0.5$).

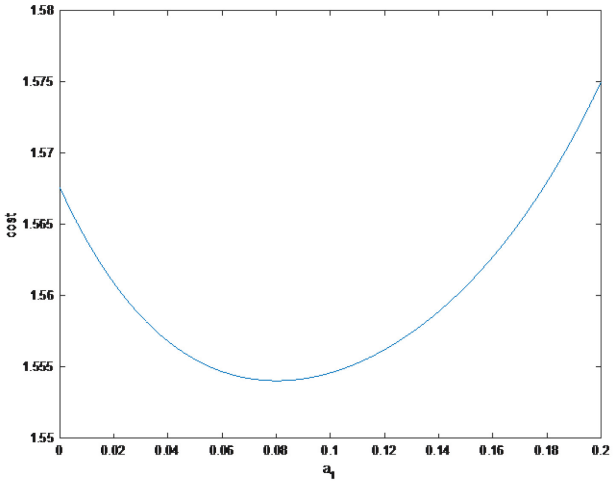

Figure 10b: How the Cost Varies with the Attacker's Action ($\mu = 1, \bar{a} = 0.2, \lambda_1 = 0.4, \lambda_2 = 0.5$).

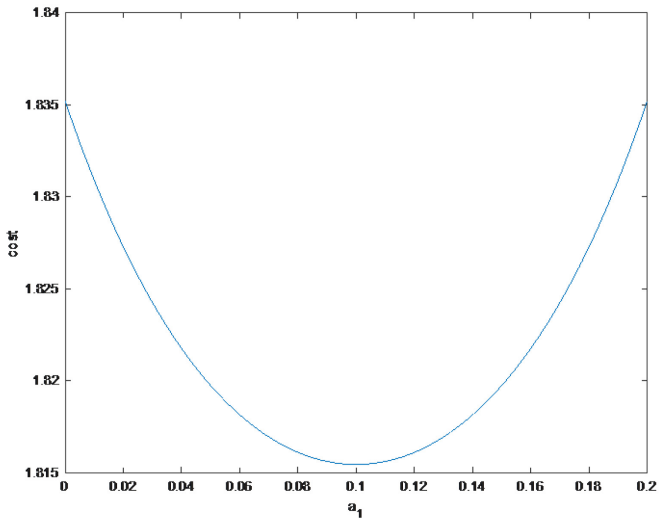

Figure 10c: How the Cost Varies with the Attacker's Action ($\mu = 1, \bar{a} = 0.2, \lambda_1 = 0.5, \lambda_2 = 0.5$).

Section 6: Conclusions

In this project, we studied the theory of diagnosis and secure routing for transportation networks subject to a rather broad class of random or strategic disruptions. We use queuing networks to model the behavior of transportation systems. We model reliability and security failures as misleading information and/or inappropriate decisions. Occurrence of reliability failures is modeled as switches of the state observation mapping or actuation policy, which are driven by finite-state Markov chains. Security failures are modeled as games between an attacker and a defender, both being strategic. For open-loop attacking and defending strategies, we quantitatively characterize the security risk (in terms of attack-induced queuing 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 queuing 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. Finally, we studied the impact of malicious modification of network boundary condition. We presented preliminary results on the structure of attacker and defender actions as well as security risk estimation.

References

[1] Y.-C. Hung and G. Michailidis, "Optimal routing for electric vehicle service systems,"European Journal of Operational Research, vol. 247, no. 2, pp. 515–524, 2015.

[2] G.-J. van Houtum, I. J. Adan, J. Wessels, and W. H. Zijm, "Performance analysisof parallel identical machines with a generalized shortest queue arrival mechanism,"OR-Spektrum, vol. 23, no. 3, pp. 411–427, 2001.

[3] V. Gupta, M. H. Balter, K. Sigman, and W. Whitt, "Analysis of join-the-shortest-queue routing for web server farms," Performance Evaluation, vol. 64, no. 9-12, pp. 1062–1081, 2007.

[4] A. Ephremides, P. Varaiya, and J. Walrand, "A simple dynamic routing problem," IEEE transactions on Automatic Control, vol. 25, no. 4, pp. 690–693, 1980.

[5] B. Ghena, W. Beyer, A. Hillaker, J. Pevarnek, and J. A. Halderman, "Green lights forever: Analyzing the security of traffic infrastructure," in 8th USENIX Workshop on Offensive Technologies (WOOT 14), 2014.

[6] Q. A. Chen, Y. Yin, Y. Feng, Z. M. Mao, and H. X. Liu, "Exposing congestion attack on emerging connected vehicle based traffic signal control." in NDSS, 2018.

[7] E. A. Lee, “Cyber physical systems: Design challenges,” in 2008 11th IEEE Interna- tional Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC). IEEE, 2008, pp. 363–369.

[8] H. Deng, W. Li, and D. P. Agrawal, “Routing security in wireless ad hoc networks,” IEEE Communications magazine, vol. 40, no. 10, pp. 70–75, 2002.

[9] L. Flatto and H. McKean, “Two queues in parallel,” Communications on pure and applied mathematics, vol. 30, no. 2, pp. 255–263, 1977.

[10] S. Halfin, “The shortest queue problem,” Journal of Applied Probability, vol. 22, no. 4, pp. 865–878, 1985.

[11] R. D. Nelson and T. K. Philips, An approximation to the response time for shortest queue routing. ACM, 1989, vol. 17, no. 1.

[12] R. D. Foley and D. R. McDonald, “Join the shortest queue: stability and exact asymp- totics,” The Annals of Applied Probability, vol. 11, no. 3, pp. 569–607, 2001.

[13] P. Eschenfeldt and D. Gamarnik, “Join the shortest queue with many servers. the heavy- traffic asymptotics,” Mathematics of Operations Research, vol. 43, no. 3, pp. 867–886, 2018.

[14] B. Hajek, “Optimal control of two interacting service stations,” IEEE transactions on automatic control, vol. 29, no. 6, pp. 491–499, 1984.

[15] J. Kuri and A. Kumar, “Optimal control of arrivals to queues with delayed queue length information,” IEEE Transactions on Automatic Control, vol. 40, no. 8, pp. 1444–1450, 1995.

[16] F. J. Beutler and D. Teneketzis, “Routing in queueing networks under imperfect infor- mation: Stochastic dominance and thresholds,” Stochastics: An International Journal of Probability and Stochastic Processes, vol. 26, no. 2, pp. 81–100, 1989.

[17] Y. Ouyang and D. Teneketzis, “Signaling for decentralized routing in a queueing net- work,” Annals of Operations Research, pp. 1–39, 2015.

[18] P. Kumar and S. P. Meyn, “Stability of queueing networks and scheduling policies,” IEEE Transactions on Automatic Control, vol. 40, no. 2, pp. 251–260, 1995.

[19] J. G. Dai and S. P. Meyn, “Stability and convergence of moments for multiclass queueing networks via fluid limit models,” IEEE Transactions on Automatic Control, vol. 40, no. 11, pp. 1889–1904, 1995.

[20] A. Eryilmaz and R. Srikant, “Fair resource allocation in wireless networks using queue- length-based scheduling and congestion control,” IEEE/ACM Transactions on Network- ing (TON), vol. 15, no. 6, pp. 1333–1344, 2007.

[21] S. P. Meyn and R. L. Tweedie, “Stability of markovian processes iii: Foster–lyapunov criteria for continuous-time processes,” Advances in Applied Probability, vol. 25, no. 3, pp. 518–548, 1993.

[22] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bac¸sar, and J.-P. Hubaux, “Game theory meets network security and privacy,” ACM Computing Surveys (CSUR), vol. 45, no. 3, p. 25, 2013.

[23] M. Wu and S. Amin, “Securing infrastructure facilities: When does proactive defense help?” Dynamic Games and Applications, pp. 1–42, 2018.

[24] S. R. Etesami and T. Ba¸sar, “Dynamic games in cyber-physical security: An overview,” Dynamic Games and Applications, pp. 1–30, 2019.

[25] M. Wu, L. Jin, S. Amin, and P. Jaillet, “Signaling game-based misbehavior inspection in v2i-enabled highway operations,” in 2018 IEEE Conference on Decision and Control

(CDC). IEEE, 2018, pp. 2728–2734.

[26] A. Laszka, W. Abbas, Y. Vorobeychik, and X. Koutsoukos, “Detection and mitigation of attacks on transportation networks as a multi-stage security game,” Computers & Security, vol. 87, p. 101576, 2019.

[27] S. Bohacek, J. P. Hespanha, K. Obraczka, J. Lee, and C. Lim, “Enhancing security via stochastic routing,” in Proceedings. Eleventh International Conference on Computer Communications and Networks. IEEE, 2002, pp. 58–62.

[28] H. Guo, X. Wang, H. Cheng, and M. Huang, “A routing defense mechanism using evolutionary game theory for delay tolerant networks,” Applied Soft Computing, vol. 38, pp. 469–476, 2016.

[29] F.-R. Chang, Stochastic optimization in continuous time. Cambridge University Press, 2004.

[30] S. A. Lippman, “Applying a new device in the optimization of exponential queuing systems,” Operations Research, vol. 23, no. 4, pp. 687–710, 1975.

[31] M. L. Puterman, Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons, 2014.

[32] A. Federgruen, “On n-person stochastic games by denumerable state space,” Advances in Applied Probability, vol. 10, no. 2, pp. 452–471, 1978.

[33] L. S. Shapley, “Stochastic games,” Proceedings of the national academy of sciences, vol. 39, no. 10, pp. 1095–1100, 1953.

[34] T. Alpcan and T. Ba¸sar, Network security: A decision and game-theoretic approach. Cambridge University Press, 2010.

[35] L. S. Shapley and R. Snow, “Basic solutions of discrete games,” Contributions to the Theory of Games, vol. 1, pp. 27–35, 1952.