ZUbers against ZLyfts Apocalypse: An Analysis Framework for DoS Attacks on Mobility-as-a-Service Systems

Chenyang Yuan * Jérôme Thai * Alexandre M. Bayen
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA
{chenyang.yuan, jerome.thai, bayen}@berkeley.edu

ABSTRACT

The vulnerability of Mobility-as-a-Service (MaaS) systems to Denial-of-Service (DoS) attacks is studied. We use a queuing-theoretical framework to model the re-dispatch process used by operators to maintain a high service availability, as well as potential cyber-attacks on this process. It encompasses a customer arrival rate model at different sections of an urban area to pick up vehicles to travel within the network. Expanding this re-balance model, we analyze DoS cyber-attacks of MaaS systems by controlling a fraction of the cars maliciously through fake reservations (so called *Zombies*) placed in the system (similar to the computer science field where a *Zombie* is a computer that a remote attacker has accessed for malicious purpose). The attacker can then use the block-coordinate descent algorithm proposed in the present work to derive optimal strategies to minimize the efficiency of the MaaS system, thereby allowing us to quantify the economic loss of such systems under attack. The technique is shown to work well and enables us to arbitrarily deplete taxi availabilities based on the attacker's choice and the *radius* of attacks, which is demonstrated by drawing a "Cal" logo in Manhattan. Finally, a cost-benefit analysis from 75 million taxi trips shows diminishing returns for the attacker and that countermeasures raising the attack cost to more than $15 protect MaaS systems in NYC from *Zombies*.

1. INTRODUCTION

In recent years, the rapid expansion of Mobility-as-a-Service (MaaS) systems such as ride-sharing services and (electric) car rental programs triggered research on the optimal management of such systems: optimal dispatch [9], optimal fleet size [5], general design [18], and optimal re-balancing algorithms [21] to achieve the current level of performance of MaaS systems. The necessary central coordination of the dispatch of vehicles can be seen as re-balancing the network and it can be done manually as commonly done by taxi companies with human dispatchers, by apps such as taxi hailing apps, or by incentivization from the two-sided markets formed by ride-sharing companies such as Uber or Lyft.

However, the *cyber-physical* nature of MaaS systems makes them vulnerable to *physical attacks* from malicious drivers, *cyber attacks* on the hailing apps, and *economic attacks* by controlling the two-sided markets. These DoS attacks can re-route a fraction of vehicles to reduce the time usage of the network and decrease the profits for the MaaS company. Hence, the security of this type of cyber-physical systems has gained a lot of attention recently [2]. While there have been research on the security of general networks [14, 22], with applications to power systems [17, 20], communication systems [1, 19], and freeway control [15], this work is amongst the first to study their security against cyber-attacks on MaaS systems, and understand the extent of attacks that can be performed on them.

Expanding an established framework on MaaS systems, attacks can be seen as malicious agents controlling the vehicles of the MaaS system, which we will refer to as *Zombie* passengers. When they are serviced by real cars, e.g. Uber or Lyft, these cars become *Zombified*, *i.e.* a *ZUber* or a *ZLyft* (similar attacks involving one company calling and canceling vehicles of the other have happened in the past [16]).The term *Zombie* is used following computer science terminology for a computer that has been compromised remotely by a hacker to launch DoS attacks. Our main contributions in this article include: (i) a framework for the study of cyber-security in MaaS systems encompassing different features, *e.g.* attack *budget* and *radius*, (ii) designing a block-coordinate descent algorithm to optimize attacks,

*These two authors contributed equally.

Type	rate	routing	contribution
customer	ϕ_i	α_{ij}	MAS model [5]
balancer	ψ_i	β_{ij}	re-balancing [21]
Zombie	ν_i	κ_{ij}	**cyber-security**

Table 1: Summary of Models
Different types of passenger with their arrival rates, routing probabilities, and the authors who introduced them.

(iii) a case study in NYC showing the extent of damage we can do with our attacks using a model generated from 75 million real taxi trips. This framework will ultimately be usable to compute the optimal attack price point of an attacker, hence helping cab companies to adjust their cancellation fees to protect themselves against such attacks.

2. A QUEUING-THEORETICAL MODEL

We consider a MaaS system in an urban area divided into N small sections (typically spanning 2 or 3 city blocks) indexed by $i \in \mathcal{S}$. We assume that M vehicles provide service to customers between pairs of sections $(i,j) \in \mathcal{S}\times\mathcal{S}$. Next we describe previous models for customer travel and balancing mechanisms on the network. Finally, we introduce our model for *Zombies*. Table 1 summarizes these three models.

2.1 Model Description

Customer model: Customers arrive at each section i following a time-invariant Poisson process with rate $\phi_i > 0$. Upon arrival at a section i, a customer chooses to go to section $j \neq i$ with probability $\alpha_{ij} \geq 0$, where $\sum_{j\in\mathcal{S}} \alpha_{ij} = 1$ and $\alpha_{ii} = 0$ for all $i \in \mathcal{S}$. Furthermore, if a vehicle is not available at a section upon arrival of a customer, the customer leaves without service (*i.e.* customers do not queue). The model also assumes that there is sufficient capacity for vehicle to queue for passengers, as is often the case of pickup locations or taxi stations. The travel times for different passengers traveling from section i to section j constitute an independently and identically distributed (i.i.d.) sequence of exponentially distributed random variables with mean $T_{ij} > 0$. This model was used in [5] to describe a vehicle rental company as a queuing network.

Re-balancing process: In any MaaS systems, there is a need for re-balancing to account for uneven demand. A re-balancing vehicle is one traveling to a destination without customers to fulfill the demand at its destination. The process has been studied extensively [9, 11, 21] and we use the framework of [21] to model it with *balancers* driving these re-balancing vehicles. This paradigm is analogous to the MaaS company "spoofing" its own drivers for re-balancing purposes. In [21], each section i generates balancers according to a Poisson process with rate $\psi_i \geq 0$ and routes these balancers to section $j \neq i$ with probability β_{ij}, where $\sum_{j\in\mathcal{S}} \beta_{ij} = 1$ and $\beta_{ii} = 0$ for all $i \in \mathcal{S}$. The re-balancing process is assumed to be independent from the customer arrival process. The model also supposes that the balancer is lost if there is no car at the section upon its generation.

Cyber-security: We extend the re-balancing work of [21] for the purpose of cyber-security analysis. We assume the attacker can generate malicious agents or *Zombies* at each section i following a Poisson process with rate $\nu_i \geq 0$ and route them to section $j \neq i$ with probability $\kappa_{ij} \geq 0$, where $\sum_{j\in\mathcal{S}} \kappa_{ij} = 1$ and $\kappa_{ii} = 0$ for all $i \in \mathcal{S}$. We assume that the re-balancing policy does not detect the attacks and its parameters ψ_i and β_{ij} only depend on the customers' demand ϕ_i and α_{ij}. We also define the *radius* r of an attack, which is the furthest (Manhattan or ℓ_1) distance that *a Zombie* can be routed through. This captures the fact that the attacker has a weaker control over the network than customers. For example, if the attacker targets a ride-sharing company by making a call and then canceling, only nearby vehicles will be dispatched and affected. Hence we define $\mathcal{E}$ the set of pairs $(i,j) \in \mathcal{S}\times\mathcal{S}$ such that routing is allowed from i to j. In other words, denoting $\mathbf{1}_A$ the indicator function of event A, we have the constraints

$$\mathbf{1}_{\{(i,j)\notin\mathcal{E}\}}\kappa_{ij} = 0 \quad \forall\, i,j \tag{1}$$

2.2 Comments on the Model

Although travel times are in general not exponentially distributed, their distribution does not affect the predictive accuracy of similar queuing networks [7]. The customers' routing probabilities α_{ij} reasonably constitute an irreducible Markov chain for dense environments, and we do not consider congestion, even though it negatively affects the efficiency of the network and the effect of re-balancing.

Note that the "passenger loss" assumption in the model where passengers not willing to wait (they leave the section immediately when there are no taxis available) is accurate in numerous US markets. This framework is a good setting for analyzing the benefits and vulnerability of MaaS systems: (i) with high service availability (the median wait time for an Uber in major U.S. cities in 2014 was under 4 min [12]), and (ii) competing against other MaaS or alternate transportation systems (particularly in dense cities where the waiting time is critical).

The passenger loss model is particularly relevant in an adversarial setting in which attacks aim at reducing service availability to incur passenger loss and encourage passengers to use a rival system, similar to the reported DoS attacks between Uber and Lyft [16]. From an an-

alytical perspective, the passenger loss model considerably simplifies our model because customer arrivals at a section is equivalent to a virtual service to the vehicles currently queuing (and available) at the section.

The re-balancing and attacks are respectively modeled as balancers and *Zombies* following the same process as customers (with passenger loss), but independently and with different arrival rates and routing probabilities, thus allowing to combine the customer demand, the re-balancing process, and the attacks into a single queuing network. In our case, the loss of balancers and *Zombies* describe processes that encourage a re-allocation of vehicles to sections but does not enforce it.

Another critical assumption is that there is no attacker-defender game (see [3]) since the re-balancing only aims at high service availability given customers' demand, and does not try to defend from possible attacks. An interesting extension would be the analysis of a one-stage game in which the balancers moves first with the knowledge of the *Zombies*' best response.

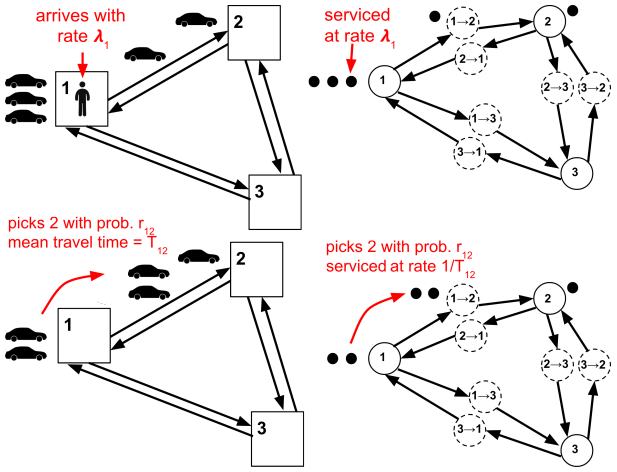

Figure 1: Illustration on a three section network. On the left, a passenger arrives at section 1 and picks a car to go to section 2. The equivalent Jackson network is shown on the right side.

3. NETWORK ANALYSIS

Following [5] and [21], the model described above can be cast into a closed Jackson network, which we now present with a cyber-attack extension.

3.1 Jackson Network Model

In the present work, we combine the customer, balancer, and *Zombie* processes. From the superposition of independent Poisson processes, the total arrival process of all three types of passengers is Poisson with rate

$$\lambda_i = \phi_i + \psi_i + \nu_i \tag{2}$$

where ϕ_i, ψ_i, and ν_i respectively represent the arrival rates of customers, balancers, and *Zombies*. A generalized passenger that arrives will either be classified as one of the three classes with respective probabilities ϕ_i/λ_i, ψ_i/λ_i, and ν_i/λ_i. The routing probability $r_{ij} := \mathbb{P}(i \to j)$ of a generalized passenger arriving at section i to select a destination j is then given by

$$r_{ij} = \sum_{\text{class}} \mathbb{P}(i \to j \,|\, \text{class})\, \mathbb{P}(\text{class}) \tag{3}$$

With α_{ij}, β_{ij}, and κ_{ij} being the routing probabilities associated to each class, we have (with λ_i given by (2)):

$$r_{ij} = \alpha_{ij}\frac{\phi_i}{\lambda_i} + \beta_{ij}\frac{\psi_i}{\lambda_i} + \kappa_{ij}\frac{\nu_i}{\lambda_i} \tag{4}$$

Sections are modeled as single-server (SS) nodes (or "section" nodes) and the route between two sections as infinite-server (IS) nodes (or "route" nodes). When a generalized passenger arrives at a non-empty section, a vehicle departs from that node to move to a route node that connect the origin to the destination selected by that passenger. After spending an exponentially distributed amount of time at the route node (the travel-time), the vehicle moves to the destination section node (see Figure 1).

From a queuing perspective, if vehicles are present at section i, they are processed with service rate λ_i given by (2), and are routed to the IS (route) node between sections i and j with probability r_{ij} given by (4). Then vehicles at an IS node between sections i and j are processed in parallel (*i.e.* assuming infinite capacity roads with no congestion effects) with service rate $1/T_{ij}$ each and move to SS node i with probability 1. Hence, the MAS system is modeled as a closed Jackson network with respect to the vehicles with vehicle service rate $\mu_n(x_n)$ at a generalized node n given by

$$\mu_n(x_n) = \begin{cases} \lambda_i & \text{if } n = \text{section } i \\ x_n/T_{ij} & \text{if } n = \text{route } i \to j \end{cases} \tag{5}$$

where $x_n \in \{0, 1, \cdots, M\}$ is the number of vehicles at node n (and M the number of vehicles in the network). Note that μ_n only depends on x_n on a route node. The routing probability $p_{nn'}$ from node n to node n' is

$$p_{nn'} = \begin{cases} r_{ij} & \text{if } n = \text{section } i,\ n' = \text{route } i \to j \\ 1 & \text{if } n = \text{route } i \to j,\ n' = \text{section } j \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

3.2 Asymptotic Behavior and Fairness

A quantity of interest is the availability, which is defined as the percentage of customers who find a vehicle available at a section upon arrival. Mathematically, it is

$\mathcal{S}$	Set of SS (section) nodes, $\lvert\mathcal{S}\rvert = N$
M	Fleet size of the MaaS system
T_{ij}	Mean travel time from $i \in \mathcal{S}$ to $j \in \mathcal{S}$
ϕ_i, α_{ij}	Customer arrival rate and routing matrix
ψ_i, β_{ij}	Balancer arrival rate and routing matrix
ν_i, κ_{ij}	*Zombie* arrival rate and routing prob.
$A_i(M)$	Prob. of $i \in \mathcal{S}$ of having ≥ 1 vehicle
a_i	asymptotic availability, $\gamma_i/\max_{j\in\mathcal{S}}\gamma_j$
$\mathbf{1}_A$	indicator function of condition A

Table 2: Summary of notations

given by the following steady-state probability (see [8]):

$$A_i(M) := \mathbb{P}(X_i \geq 1) = \frac{\gamma_i G(M-1)}{G(M)} \tag{7}$$

where the random variable X_i represent the queue length at section $i \in \mathcal{S}$. Note that the quantity $G(M)$ above is the normalization factor associated to the equilibrium state distribution of the queue lengths $\{X_i\}_{i\in\mathcal{S}}$ provided by the Gordon-Newell theorem [6]. The computation of $G(M)$ is very expensive with complexity that grows as $\binom{|\mathcal{N}|+M-1}{|\mathcal{N}|}$, where $|\mathcal{N}|$ is the cardinality of $\mathcal{N}$ (*i.e.*, the number of nodes in the network), so that $|\mathcal{N}| = N^2$. Hence, we want to obtain performance metrics without computing explicitly the quantity $G(M)$, *e.g.* by studying the asymptotic behavior of the network when the fleet size M goes to infinity. The following result from [13] gives the asymptotic availability at a SS node i:

$$a_i := \lim_{M\to\infty} A_i(M) = \frac{\gamma_i}{\max_{j\in\mathcal{S}}\gamma_j} \tag{8}$$

where $\max_{j\in\mathcal{S}}\gamma_j$ is the highest relative utilization. Hence, when M approaches infinity, sections with the highest relative utilization can have availability arbitrarily close to 1, while other sections have availability strictly less than 1, since in this case $\gamma_i < \max_{j\in\mathcal{S}}\gamma_j$).

To cancel this effect, Zhang and Pavone [21] designed a re-balancing policy with balancer arrival rates ψ_i and routing β_{ij} that maintain fairness in the network, *i.e.* $\gamma_i = \gamma_j$ for all $i, j \in \mathcal{S}$. When M is goes to infinity, this means that the availability of all sections goes to 1 since $\gamma_i = \max_{j\in\mathcal{S}}\gamma_j$ for all $i \in \mathcal{S}$. In addition to imposing fairness, they minimize the number of re-balancing vehicles given by the quantity $\sum_{i,j\in\mathcal{S}} T_{ij}\beta_{ij}\psi_i$.

4. OPTIMAL ATTACK PROBLEM

The contributions of the present article encompass the objectives of an attacker into an optimization framework, which we solve very efficiently.

4.1 Maximizing Passenger Loss

If the MaaS company gets a constant amount per ride, the attacker wants to maximize customer loss, *i.e.* minimize the customers picking a vehicle:

$$\min \sum_{i\in\mathcal{S}} \phi_i A_i(M) \tag{9}$$

If the MaaS system gets an amount that is proportional to the length of the ride, a more harmful objective is

$$\min \sum_{i,j\in\mathcal{S}} \phi_i \alpha_{ij} T_{ij} A_i(M) \tag{10}$$

hence the total time usage for the customers is minimized. Both objectives have general form

$$\min \sum_{i\in\mathcal{S}} w_i A_i(M) \tag{11}$$

where $w_i > 0$ are some user-defined arbitrary weights. To avoid computing $G(M)$ due to the complexity, the availabilities $A_i(M)$ are normalized with $\max_{j\in\mathcal{S}}\gamma_j$ and consequently study the availability $A_i(M)$ when the fleet size M goes to ∞ (see (8))

$$\min \sum_{i\in\mathcal{S}} w_i \frac{\gamma_i}{\max_{j\in\mathcal{S}}\gamma_j} = \min \sum_{i\in\mathcal{S}} w_i a_i \tag{12}$$

Finally, there must be one $i \in \mathcal{S}$ such that $a_i = 1$, hence the objective is equivalent to finding the index k such that a_k is set to 1 and minimizing over the remaining quantities $\{a_i\}_{i\neq k}$

$$\min_{k\in\mathcal{S}} \left\{ w_k \cdot 1 + \min_{\{a_i\}_{i\neq k}} \sum_{i\neq k} w_i a_i \right\} \tag{13}$$

Hence, we can solve $|\mathcal{S}| = N$ programs and select the one with the minimum objective value.

4.2 Attack Budget

The most important constraints are the traffic equations of the Jackson network. Using Lemmas 4.1 and 4.2 in [21], they can be written in terms of SS (section) nodes and asymptotic utilization a_i

$$(\phi_i+\psi_i+\nu_i)a_i = \sum_{j\in\mathcal{S}} (\alpha_{ji}\phi_j + \beta_{ji}\psi_j + \kappa_{ji}\nu_j)a_j, \ \forall\, i \tag{14}$$

Let $k \in \mathcal{S}$ such that $a_k = 1$, then the constraint is

$$\phi_k + \psi_k + \nu_k = \sum_{j\in\mathcal{S}} (\alpha_{jk}\phi_j + \beta_{jk}\psi_j + \kappa_{jk}\nu_j)a_j \tag{15}$$

Note that the constraint (15) is redundant since summing the constraints (14) for $i \neq k$ (with $a_k = 1$) gives (15). Furthermore, the attacker injects *Zombies* with arrival rates ν_i and routing matrix κ_{ij} to achieve (13). With no restriction on the attack rates, setting $\nu_i = \nu > 0$ for all $i \neq k$ and routing all the *Zombies* to

section k with probability 1 gives, using (15)

$$\sum_{j\neq k} a_j \leq (\phi_k + \psi_k + \nu_k)/\nu \to 0 \quad \text{when } \nu \to +\infty$$

Then the positive utilizations a_i go to 0 for all $i \neq k$ and the problem is reduced to $\min_{k\in\mathcal{S}} w_k$. A more realistic problem is setting a limited budget b for the attacks

$$\sum_{i\in\mathcal{S}} \nu_i \leq b \tag{16}$$

4.3 Formulation

We suppose the customers' and balancers' demands are given, and define their combined rate and routing probabilities as

$$\varphi_i := \phi_i + \psi_i \tag{17}$$
$$\delta_{ij} := (\alpha_{ij}\phi_i + \beta_{ij}\psi_i)/(\phi_i + \psi_i) \tag{18}$$

and so the combined routing probabilities r_{ij} of the customers, balancers, and *Zombies* given in (4) can be expressed as follows

$$r_{ij} = \frac{\delta_{ji}\varphi_j + \kappa_{ji}\nu_j}{\varphi_i + \nu_i} \quad \forall\, i,\, j \in \mathcal{S} \tag{19}$$

Given $k \in \mathcal{S}$ such that $a_k = 1$, the *Optimal Attack Problem* (OAP) consists in manipulating the *Zombie* arrival rates ν_i and routing κ_{ij} probabilities such that:

$$\min_{\kappa_{ij},\nu_i,a_i} \sum_{i\neq k} w_i a_i + \frac{p}{2}\sum_i \nu_i^2 \tag{20}$$

$$\text{s.t. } a_i = \sum_{j\in\mathcal{S}} \frac{\delta_{ji}\varphi_j + \kappa_{ji}\nu_j}{\varphi_i + \nu_i} a_j \quad \forall\, i \in \mathcal{S}\setminus\{k\} \tag{21}$$

$$\kappa_{ij} \geq 0,\ \sum_j \kappa_{ij} = 1, \quad \mathbf{1}_{\{(i,j)\notin\mathcal{E}\}}\kappa_{ij} = 0 \tag{22}$$

$$\nu_i \geq 0,\ \sum_i \nu_i \leq b \tag{23}$$

Note that we have included a ℓ_2-regularization term $\frac{p}{2}\sum_i \nu_i^2$ in our objective. It compensates for the budget constraint (23) which is known to encourage sparsity [4] in the entries ν_i. Sparsity allocates most of the budget b to a few sections i, resulting in very high attack rates, which is unrealistic in practice because of limitations inherent to the physical nature of the MaaS, *e.g.* bounded fleet size, limited capacity of streets. Hence, the ℓ_2-regularization captures these physical limitations.

We have also included the a_i in the decision variables since they vary. In fact, the a_i are function of κ_{ij}, ν_i and can be written directly as $a_i(\kappa, \nu)$.

Lemma 1. *For any attack strategies ν_i and κ_{ij}:*

$$a_i > 0 \text{ for all } i \in \mathcal{S} \tag{24}$$
$$a_i \text{ is uniquely defined for all } i \in \mathcal{S} \tag{25}$$

Name	Fix	Vary	Minimize
Attack Routing Pb.	ν_i	a_i, κ_{ij}	$\sum_i w_i a_i$
Min. Attack Pb.	a_i	κ_{ij}, ν_i	$\sum_i \nu_i^2$
Attack Rate Pb.	κ_{ij}	ν_i, a_i	$\sum_i w_i a_i + \frac{p}{2}\sum_i \nu_i^2$

Table 3: Summary of three-step algorithm

Proof. By assumption, the probabilities α_{ij} constitute an irreducible Markov chain. By equation (4), the probabilities r_{ij} lead to an irreducible Markov chain as well. The $\{a_i\}_i$ vector satisfying equations (21) is proportional to the steady state distribution for the transition probabilities $\{r_{ij}\}_{ij}$ and by the Perron-Frobenius theorem, it is positive [10]. Finally, the constraint $a_k = 1$ completely fixes the vector $\{a_i\}_i$. ∎

4.4 Block-coordinate Descent

The above optimization program is non-convex and is difficult to solve numerically. Specifically, the vector $\{a_i\}_{i\in\mathcal{S}}$ is a function of κ_{ij}, ν_i from Lemma 1, hence the gradient of the objective is given by

$$\sum_{l\neq k} w_i(\{\partial_{\nu_i} a_l\}_i\,,\, \{\partial_{\kappa_{ij}} a_l\}_{i,j}) + p(\{\nu_i\}_i,\, \{0\}_{i,j}) \tag{26}$$

where each the partial derivative of a_i satisfies a set of $N-1$ linear equations obtained by differentiating the balance constraints (21). Hence, computing the gradient prohibitively requires to solve N^2 linear programs of dimension $N-1$ by differentiating the constraints (21). The total complexity for computing the gradient is $(N^2-N)^2 \geq (N-1)^4$. Our of our main contributions is the design of a tractable block-coordinate descent algorithm to solve the above problem, where each sub-problem is summarized in Table 3. We pose the *Minimum Attack Problem* (MAP) and the *Attack Routing Problem* (ARoP) and show that they can be re-formulated as a quadratic and a linear program respectively with N^2 non-negative variables and N constraints. Using an efficient solver, CPLEX, we solve the MAP and ARoP efficiently. The *Attack Rate Problem* (ARaP) has N variables which are $\{\nu_i\}_{i\in\mathcal{S}}$ and can be solved efficiently using a projected gradient descent algorithm. The gradient computation requires solving N linear programs of dimension $N-1$, hence an $O(N^3)$ complexity that is tractable. We also note that the ARoP, MAP, and ARaP can be interpreted as specific attack scenarios in their own right.

5. A TAXONOMY OF ATTACKS

In this section, our contribution is proposing a taxonomy of attack scenarios described by the ARoP, MAP, and ARaP (in addition to the OAP). In each scenario,

the attacker is given a fixed allocation of either the availabilities a_i, the attack rates ν_i, or the attack routing κ_{ij}, and he wants to allocate the two other types of "recources" optimally to harm the system. We also describe how each one of these programs fits into the proposed block-coordinate descent algorithm.

5.1 Attack Routing Problem (ARoP)

In this scenario, the attacker can only inject attacks with fixed rates. For example, the attacker has placed devices at different sections $i \in \mathcal{S}$ that remotely spoof the hailing apps of nearby vehicles, to send them to specific locations. Hence, given ν_i, the attacker wants to optimize the routing to achieve objective (13). This is the *Attack Routing Problem* (ARoP), which can be re-formulated as a Linear Program from this lemma

Theorem 1. *Let us consider the following linear program (LARoP)*

$$\min_{y_{ij}} \sum_{ij} w_i y_{ij} \tag{27}$$

$$s.t. \sum_{j\neq i} (\lambda_i y_{ij} - \nu_j y_{ji}) = \sum_{j>l} \delta_{ji}\varphi_j y_{jl} \quad \forall\, i \neq k \tag{28}$$

$$y_{ij} \geq 0, \quad \sum_{j\neq k} y_{kj} = 1 \tag{29}$$

Let $y^\star_{ij}$ be an optimal solution to LARoP. Then, an optimal solution of the ARoP is

$$a_i = \sum_{j\neq i} y^\star_{ij} \tag{30}$$

$$\kappa_{ij} = y^\star_{ij}/a_i \tag{31}$$

Proof. We can obtain LARoP from the OAP by fixing ν_i and making the change of variables $y_{ij} := \kappa_{ij} a_i$ to equations (20) – (22). ∎

We decrease the $\sum_{i\neq k} w_i a_i$ part of the objective of the OAP with respect to a_i, κ_{ij} by solving the above program efficiently with CPLEX, as part of our block-coordinate descent algorithm.

5.2 Attack Rate Problem (ARaP)

In this scenario, the attacker hacks the apps of the vehicles to display "ghost" demands at specific sections i. With fixed routing κ_{ij}, the attack rates ν_i are chosen to achieve objective (20). The *Attack Rate Problem* (ARaP) consists in optimizing the OAP with respect to the rates ν_i for all i and the asymptotic availabilities a_i for $i \neq k$, while the routing of attacks κ_{ij} are fixed. Since the sum $\sum_{i\neq k} w_i a_i$ is a function of the ν_i, we compute the Jacobian matrix of the vector $\{a_i\}_{i\neq k}$, which is given by the following:

Lemma 2. *The Jacobian matrix $(\partial a_i/\partial \nu_j)_{i\neq k, j\in\mathcal{S}}$ of dimension $(N-1)\times N$ has columns $x_j \in \mathbb{R}^{N-1}$ for $j \in \mathcal{S}$ that satisfy*

$$(D - M)\, x_j = v_j \quad \forall\, j \in \mathcal{S} \tag{32}$$

where D is a diagonal matrix with entries $\{\varphi_i + \nu_i\}_{i\neq k}$, $M = \{\phi_j \delta_{ji} + \nu_j \kappa_{ji}\}_{i\neq k, j\neq k}$, and $v_j \in \mathbb{R}^{N-1}$ for $j \in \mathcal{S}$ are vectors with entries $\{a_j(\kappa_{ji} - \mathbf{1}_{\{i=j\}})\}_{i\neq k}$ where $\mathbf{1}_A$ is the indicator function of event A.

Solving the above N systems of $N-1$ linear equations gives the Jacobian of $\{a_i\}_{i\neq k}$. Hence we can solve the ARaP with the projected gradient descent algorithm, where g is the gradient of the objective:

$$\{\nu_i\}_{i\in\mathcal{S}} := \Pi\left(\{\nu_i\}_{i\in\mathcal{S}} - t\, g\right) \tag{33}$$

$$g := \sum_{i\neq k} \left(\partial a_i/\partial \nu_j\right)_{j\in\mathcal{S}} + p(\nu_j)_{j\in\mathcal{S}} \tag{34}$$

where $t > 0$ is the step size and Π is the projection onto the ℓ_1-ball of radius b, *i.e.* $\{x \in \mathbb{R}^{\mathcal{S}}_{\geq 0} : \sum_{i\in\mathcal{S}} x_i \leq b\}$. We use the $O(N \log N)$ implementation described in [4]. We use a step size decreasing in $1/\sqrt{n}$ where n is the number of iterations and complement it with a simple line search to have a lower objective at each iteration.

5.3 Minimum Attack Problem (MAP)

We consider a scenario in which the attacker wants to achieve target availabilities a_i at each station in the network with the minimum quadratic cost of attacks $\frac{1}{2}\sum_i \nu_i^2$. Given a target of fixed positive availabilities a_i for each section, the attacker instead minimizes the cost of such attacks. The *Minimum Attack Problem* (MAP) can be formulated as follows

$$\min_{\kappa_{ij},\nu_i} \frac{1}{2}\sum_i \nu_i^2 \tag{35}$$

$$\text{s.t. } a_i = \sum_{j\in\mathcal{S}} \frac{\delta_{ji}\varphi_j + \kappa_{ji}\nu_j}{\varphi_i + \nu_i} a_j \quad \forall\, i \in \mathcal{S}\setminus\{k\} \tag{36}$$

$$\kappa_{ij} \geq 0,\ \sum_j \kappa_{ij} = 1, \quad \mathbf{1}_{\{(i,j)\notin\mathcal{E}\}}\kappa_{ij} = 0 \tag{37}$$

$$\nu_i \geq 0 \quad \forall\, i \in \mathcal{S} \tag{38}$$

The constraints can be formulated as flow constraints

Theorem 2. *Let us define*

$$s_i := a_i\varphi_i - \sum_{j\neq i} a_j \delta_{ji}\varphi_j \quad \forall\, i \in \mathcal{S} \tag{39}$$

and consider the following Quadratic Program

$$\min_{x_{ij}} \sum_i \frac{1}{2a_i^2}\left(\sum_j x_{ij}\right)^2 \tag{40}$$

$$s.t. \sum_{j\neq i}(x_{ji} - x_{ij}) = s_i \qquad \forall\, i \in \mathcal{S} \tag{41}$$

$$x_{ij} \geq 0 \quad \mathbf{1}_{\{(i,j)\notin\mathcal{E}\}} x_{ij} = 0 \qquad \forall\, i, j \in \mathcal{S} \tag{42}$$

This is always feasible. Let $x^\star_{ij}$ be an optimal solution to it. Then, an optimal solution to the MAP is:

$$\nu_i = \sum_{j \neq i} x^\star_{ij}/a_i \tag{43}$$

$$\kappa_{ij} = \begin{cases} x^\star_{ij}/(\nu_i a_i) & \text{if } \nu_i > 0 \\ 1/\sum_j \mathbf{1}_{\{(i,j)\in\mathcal{E}\}} & \text{otherwise} \end{cases} \tag{44}$$

Proof. We apply the following change of variables

$$x_{ij} := \nu_i \kappa_{ij} a_i \quad \forall\, i, j \tag{45}$$

which converts the MAP into the above program with $\{s_i\}_{i\in\mathcal{S}}$ given by (39) and $\nu_i = \sum_{j\neq i} x_{ij}/a_i$ as a result of the change of variable. Note that x_{ij} can be interpreted as the rate of attack from station i to j. This problem is feasible because the capacity on each edge is unbounded and the source flows sum to 0:

$$\sum_i s_i = \sum_i a_i \varphi_i - \sum_{i,j\neq i} a_j \delta_{ji} \varphi_i = 0 \tag{46}$$

Therefore, we can find the minimal-cost attacks that achieve any arbitrary availabilities. ∎

Within the proposed block-coordinate descent framework, we add the budget constraint (23) to the MAP using the solution of the previous step as initial solution, and solve it efficiently using CPLEX. Note that the objective of the above program can be generalized to any convex function, and a linear objective results in a *min-cost-flow problem* (MCFP). This reduction to a MCFP was shown in [21] for the purpose of re-balancing vehicles with an objective minimizing the number of re-balancing trips

$$\min_{\psi_i,\, \beta_{ij}} \sum_{i,j} \psi_i T_{ij} \beta_{ij} \tag{47}$$

where ψ_i, β_{ij} are the *balancers* arrival rates and routing probabilities respectively. In our case, the MAP step of our algorithm redistributes the highest attack rates among sections, thus avoiding numerical corner cases associated to the sparsity promoting constraint (23).

6. APPLICATION TO NYC TAXI SYSTEM

We illustrate the results on a data set comprising 1.1 billion trips gathered since Yellow Taxis' digital trip record began, from January 2009 to June 2015. We infer the parameters for the queueing theoretic model described in (5) and (6) from these trips. This data is provided by the New York City Taxi and Limousine Commission and includes pickup and drop-off locations, trip time, and fare for each trip.

We divide midtown and downtown Manhattan into 531 square sections using a grid that is aligned with the road layout of Manhattan, each section serving a region 1-2-block wide. Only grid squares with a significant arrival rate are considered. This results in a Jackson network with more than 282,000 nodes when road nodes between each pair of sections are included. Then we use all 75 million trips originating during the weekday evening peak period from 5-7pm to infer the parameters for the queueing model ϕ_i and T_{ij} via the sample mean, and α_{ij} via Laplacian smoothing. In the process, we checked that ϕ_i closely follows a Poisson distribution, validating our assumption about customer arrivals. The customer arrival rate is about 10,600 per hour and there are about 2,500 taxis in the network in this time period.

6.1 Experiment 1: Controlling Availabilities

In this experiment, given any arbitrary set of availabilities a_i, we find the minimal cost of attacks needed to control the network such that the resulting availabilities match the provided ones. To illustrate this we show that we can create arbitrary availability patterns in the city, in particular the "Cal" logo, see Figure 2a. An optimal control is obtained by solving the MAP (i.e. the second step in our block-coordinate descent algorithm to solve the OAP), but with a linear objective $\min \sum_i \nu_i$, resulting in a linear program. We proceed with the simulation by first balancing the network (assuming that real MaaS have high service availability) using the methodology of [21], *i.e.* solving the MAP with the availabilities uniformly equal to 1 and with a linear objective of the form (47). This yields a total rate of 2,200 re-balancing vehicles per hour. We then compute the attack strategy on the balanced network. When the attacking radius is unlimited, injecting 800 Zombies per hour achieves the availability pattern encoded in the "Cal" logo.

Next, we decrease the radius of attacks by limiting the routing from a section i to values between 1 and 15 in terms of Manhattan distance (or ℓ_1 with a section block as a unit of length), and find the minimum attacks rate needed to create the "Cal" logo, as illustrated by Figures 2. Figures 2b-c show the optimal attack strategies in terms of the rates and routing directions, and Figure 2d the minimal attack rates as a decreasing function of the radius, see Figure 2 for more details.

6.2 Experiment 2: Minimizing Availabilities

In this experiment, we solve the OAP on the Jackson network model learned from the NYC taxi trips data. To avoid numerical difficulties related to the very large disparity in customer arrival rates at the different sections (the squares close to Grand Central station having customer arrival rates of 200 vehicles per hour while the sections along East river have one customer arrival every four hours on average between 5 and 7pm), we cluster adjacent blocks together and aggregate the arrival rates such that the minimum arrival rate at a section is 30 customers per hour, resulting in a re-

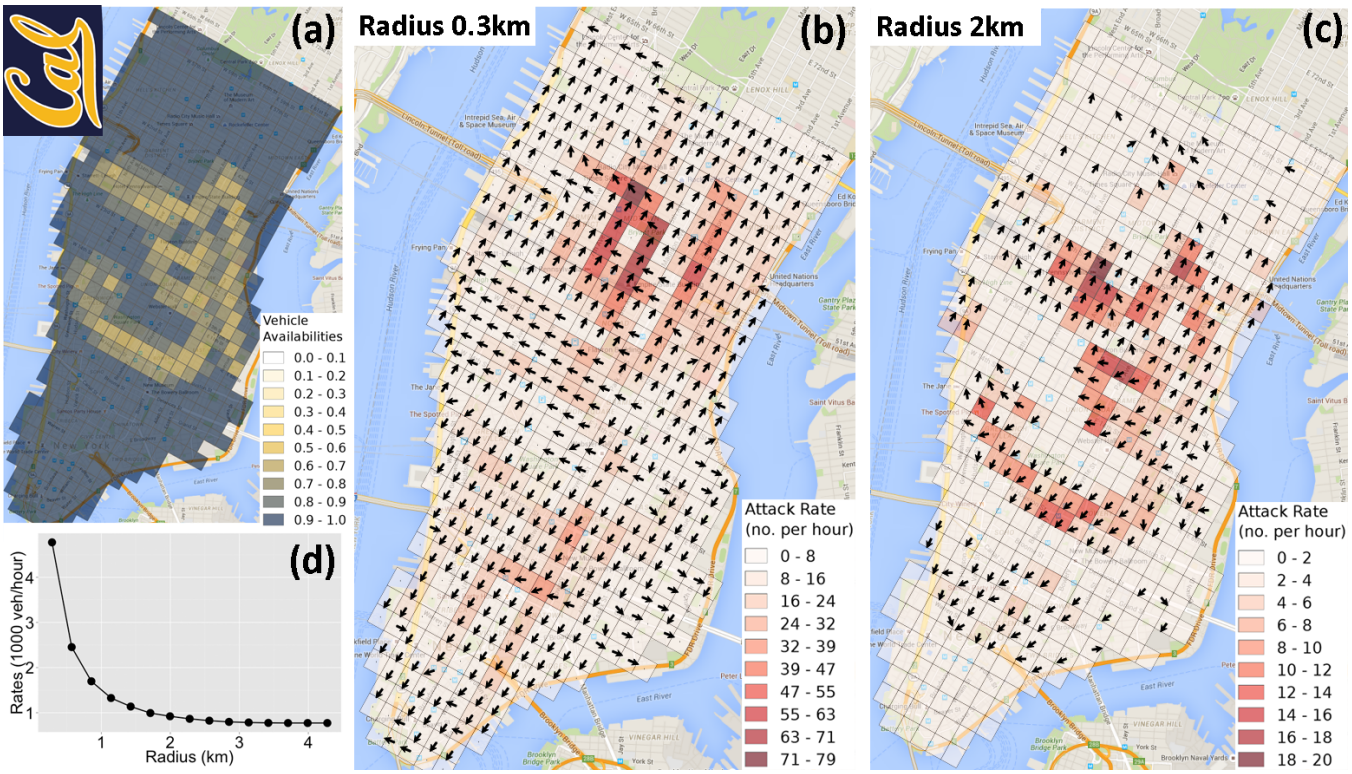

Figure 2: Effect of Radius of Attacks

(a): Target availability pattern following a pixelated version of the "Cal" logo. (b) and (c): Best attack policy to achieve the target with maximum ℓ_1-radius of 0.3km (1 block) and 2km (7 blocks) respectively: each arrow shows the direction of the κ_{ij}-weighted barycenter of the destination sections j from an origin i, and the color of each square encodes the attack rate. (d): Total attack rate per hour needed to achieve the specified availabilities as a function of radius. We can see that if we limit the radius of attacks to one block, as in (b), vehicles are routed through many intermediate stations, whereas in (c), increasing the radius allows the attacker to remove cars from regions with low availabilities (yellow in (a)) and send them directly to the borders of Manhattan. Hence, limiting the attack radius to small values greatly hinders the attacks' effectiveness, and increasing the radius past 1-2 km does result in diminishing returns.

duction to 331 blocks. We then balance the network (due to the dispatching of vehicles in response to uneven demand) and apply the proposed block-coordinate descent algorithm for solving the OAP with an objective (10) minimizing the customer time usage in the network. The different steps are summarized in Algorithm 1.

Algorithm 1 Algorithm for solving the AOP.

choose arbitrary section $k \in \mathcal{S}$.
initialize ν_i and κ_{ij}
while stopping criteria not satisfied:
    update a_i, κ_{ij} via ARoP with ν_i fixed.
    update ν_i, κ_{ij} via MAP with a_i fixed.
    update a_i, ν_i via ARaP with κ_{ij} fixed.
return a_i, ν_i, κ_{ij}

We do not set a limit on the radius of attacks and apply the descent method for values of the budget b of attack rate in $\{100, 500, 1000, 1500, 2000, 2500, 3000, 5000, 7000, 10000\}$ with the parameter p controlling the weight of the ℓ_2-regularization equal to 0.1 for $b \leq 1000$ and 0.01 otherwise. The total customer and *balancer* arrival rates remains unchanged on the reduced network, with 10,600 and 2,200 vehicles per hour respectively, hence the budget of attacks accounts for values from 0.8% to 44% of the total rate (all three types of passengers). Initializing with uniform *Zombies* arrival rate throughout the network and uniform distributions for the routing probabilities, Algorithm 1 gives an attack strategy that consistently send *waves of Zombies* to several spots around the center of Manhattan, see Figure 3a and b. In equilibrium, these target regions have high availabilities while the rest of Manhattan has very low availabilities. We can also see that the attacks are concentrated on one

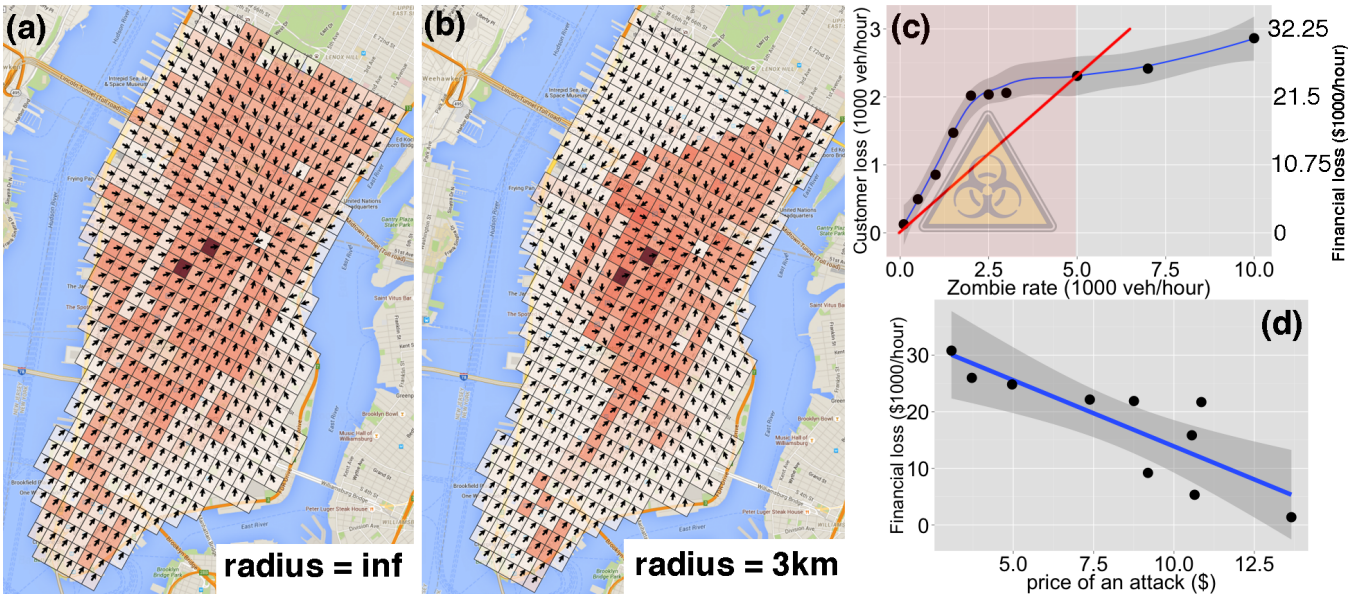

Figure 3: Optimal Attack Rates and Routing

(a) and (b): The attack rates and routing probabilities for a total budget of 2000 *Zombies* per hour are showed in the same style as in Figure 2, with an unlimited radius and 3km (9 squares) radius respectively. (c): Passenger/financial loss as a function of attacks from 10 simulations of the Jackson network (each one associated to a given budget and a strategy computed from the OAP). The vertical scale on the left shows the rate of passenger loss and the one on the right the financial loss assuming that a passenger spends $10.75 on an average. The red line denotes the price of attack (assuming $5/unit) against the budget. If 100% of the loss is gained by the attacker (from stealing customers), then the red region is financially beneficial for the attacker. The red line shows that an attack costing $5/unit (its slope) incurs a maximum loss of $22,500/hour for the MaaS system. (d): Maximum financial loss for the MaaS system as a function of the cost of one unit of attack, obtained from (c). A cost of attack above $15 protects the system.

spot if the radius can be high, while the attacks concentrate on several different spots around Manhattan for a low radius.

6.3 Experiment 3: Network Simulation

Solving for the attack rates using the OAP gives very low objective values, with a loss of customer time usage from 60% to 100%. This surprising efficiency is in fact the asymptotic behavior of the system under attacks, where most of the vehicles are blocked in the center region because the balancer process does not re-dispatch the vehicles in other parts of the network in reaction to the attacks. To account for the transient state since coordinated attacks can be seen as a *malicious shock* propagating throughout the network, we run a simulation of the Jackson network used for our model to study the effectiveness of our attack strategy, with 2500 taxis (average number of taxis in the area at the time of the day used for our parameter inference). We start from a closed network in equilibrium and introduce attacks. For each queue in the simulation, customers, balancers and *Zombies* arrive with our specified rates, and are lost when there are no vehicles in the queue. We then record the number of customers lost for one hour and subtract from this the base rate of loss when the network is balanced. This difference is the passenger lost from our attacks. Figure 3c and 3d show the results of our analysis. Assuming that the cost of conducting an attack is $5 (the cost of canceling an Uber ride) and the gain of the attacker is $10.75 (the average cost of a ride in the area estimated from our data-set), Figure 3c shows that it is not economical to attack with more than 5000 *Zombies* per hour. From this, we can deduce that a cost of attack greater than $15 protects the MaaS sytem against attacks.

7. CONCLUSION

In this paper, we propose and implement a framework to analyze DoS attacks on MaaS systems. We propose a tractable block-coordinate descent algorithm to compute attack strategies which are applied to the network in Manhattan.

This framework can also be extended to more general settings, such as helping MaaS companies control their networks if they can predict demand, to intentionally unbalance their network in anticipation of high demand in certain areas. In addition to this, MaaS companies can use this model to defend themselves against such attacks, such as setting an appropriate price for canceling rides.

This also opens up exciting avenues of future work. We have largely ignored congestion effects on the network, one extension is to include these effects and design attacks that create congestion in critical parts of the city. We also assumed that the MaaS company does not respond to the attacks, and it would be interesting to model this as a attacker-defender game.

Acknowledgments

The authors would like to thank Cathy Wu and Ramtin Pedarsani for insightful discussions on the problem.

8. REFERENCES

[1] S. Bohacek, J. P. Hespanha, J. Lee, C. Lim, and K. Obraczka. Enhancing security via stochastic routing. In *Proc. of the 11th IEEE Int. Conf. on Comput. Communications and Networks*, May 2002.

[2] A. A. Cardenas, S. Amin, and S. Sastry. Secure control: Towards survivable cyber-physical systems. *International Conference on Distributed Computing Systems*, 2008.

[3] L. A. Cox. Game Theory and Risk Analysis. *Risk Analysis*, 29, 2009.

[4] J. Duchi, S. Gould, and D. Koller. Projected subgradient methods for learning sparse gaussians. *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, 2008.

[5] D. K. George and C. H. Xia. Fleet-sizing and service availability for a vehicle rental system via closed queueing networks. *European Journal of Operational Research*, 211(1):198–207, 2011.

[6] W. J. Gordon and G. F. Newell. Closed Queuing Systems with Exponential Servers. *Operations Research*, 15, 1967.

[7] R. C. Larson and A. R. Odoni. *Urban operations research*. 1981.

[8] S. S. Lavenberg. *Computer performance modeling handbook*. Elsevier, 1983.

[9] Z. Liao. Real-time taxi dispatching using Global Positioning Systems. *ACM*, 46:81–83, 2003.

[10] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, 2000.

[11] F. Miao, S. Lin, S. Munir, J. A. Stankovic, H. Huang, D. Zhang, T. He, and G. J. Pappas. Taxi dispatch with real-time sensing data in metropolitan areas ï£¡ a receding horizon control approach. *International Conference on Cyber-Physical Systems*, pages 100–109, 2015.

[12] P. Mosendz and H. Sender. EXCLUSIVE: Here's How Long It Takes to Get an Uber in U.S. Cities. *Newsweek*, December 4 2014.

[13] R. R. Muntz and J. W. Wong. Asymptotic properties of closed queueing network models. *8th Annual Princeton Conference on Information Sciences and Systems*, pages 348–352, 1974.

[14] T. R. Nudell and A. Chakrabortty. Ensuring Localizability of Node Attacks in Consensus Networks via Feedback Graph Design. *American Control Conference*, 2014.

[15] J. Reilly, S. Martin, M. Payer, and A. Bayen. On Cybersecurity of Freeway Control Systems: Analysis of Coordinated Ramp Metering Attacks. *Transportation Research, Part B*, 2014.

[16] S. Rodriguez. Lyft vs. Uber: The battle between the ridesharing rivals intensifies. *Los Angeles Times*, September 26 2015.

[17] K. C. Sou, H. Sandberg, and K. H. Johansson. Detection and Identification of Data Attacks in Power System. *2012 American Control Conference*, 2012.

[18] K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton, and M. Pavone. Toward a Systematic Approach to the Design and Evaluation of Automated Mobility-on-Demand Systems. *in Gereon Meyer, Sven Beiker (editors). Road Vehicle Automation, (Lecture Notes in Mobility), Springer*, 2014.

[19] A. Teixeira, D. Perez, H. Sandberg, and K. H. Johansson. Attack Models and Scenarios for Networked Control Systems. *High Confidence Networked Systems*, 2012.

[20] J. Weimer, S. Kar, and K. H. Johansson. Distributed Detection and Isolation of Topology Attacks in Power Networks. *HiCoNS*, 2012.

[21] R. Zhang and M. Pavone. Control of robotic mobility-on-demand systems: a queueing-theoretical perspective. *International Journal of Robotics Research*, 2015.

[22] B. Zhu, A. Joseph, and S. Sastry. Taxonomy of Cyber Attacks on SCADA Systems. *Proceedings of CPSCom 2011: The 4th IEEE International Conference on Cyber, Physical and Social Computing*, 2011.