Resiliency of Mobility-as-a-Service Systems to Denial-of-Service Attacks

Jérôme Thai[1] and Chenyang Yuan[1] and Alexandre M. Bayen[1,2]

Abstract— **Mobility-as-a-Service (MaaS) systems such as ride sharing services have expanded very quickly over the past years. However, the popularity of MaaS systems make them increasingly vulnerable to Denial-of-Service (DOS) attacks, in which attackers attempt to disrupt the system to make it unavailable to the customers. Expanding on an established queuing-theoretical model for MaaS systems, attacks are modeled as a malicious control of a fraction of vehicles in the network. We then formulate a stochastic control problem that maximizes the passenger loss in the network in steady state, and solve it as a sequence of linear and quadratic programs. Combined with a Jackson network simulation and an economic model of supply and demand for attacks, we quantify how raising the cost of attacks (via cancellation fees and higher level of security) removes economical incentives for DoS attacks. Calibrating the model on 1B taxi rides, we dynamically simulate a system under attack and estimate the passenger loss under different scenarios, such as arbitrarily depleting taxis or maximizing the passenger loss. Cost of attacks of $15 protects the MaaS system against DoS attacks. The contributions are thus a theoretical framework for the analysis of the network, and practical conclusions in terms of financial countermeasures to the attacks.**

I. Introduction

A. Motivation

Mobility-as-a-Service (MaaS) systems such as ride-sharing services and (electric) car rental programs have been expanding very quickly over the past years, *e.g.* Uber, Lyft, and Didi Kuaidi doing millions of rides a day [14]. Similarly, car-sharing programs are expanding quickly, such as Zipcar with more than 10,000 vehicles in the USA [33], along with City CarShare, and Car2Go. This revolution in Personal Urban Mobility [25] is accompanied with the growing population in dense cities with an estimate of 3B urbanites by 2050 [1]. Besides, the increased congestion of the road network will make car ownership no longer sustainable. Morgan Stanley's research shows that cars are driven just 4% of the time [22] while the average cost of car ownership is nearly $9000 a year [30]. For example, car ownership has dropped by 30% from 2001 to 2015 in London [31]. Instead, the population will increasingly rely on public transportation (bus usage has doubled in the same period) and MaaS systems.

Optimal management of MaaS systems: Since urban population will heavily depend on MaaS systems, research has become very active on their optimal management [20], [10], [36]. Dispatching or *re-balancing* is necessary to fulfill the uneven distribution of origins and destinations of the requested rides. It can be done manually 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. Besides, autonomous cars have arguably received a great deal of scientific attention, both Google and Tesla predicting that they will be available by 2020 [21], [8]. Hence we include fleets of autonomous vehicles as part of MaaS systems, and researchers demonstrate the sustainability of autonomous fleets, suggesting that 8000 rebalanced autonomous vehicles (70% of the size of NYC taxi fleet) can satisfy the taxi demand in Manhattan [36].

Vulnerability to Denial-of-Service attacks: As MaaS systems become ubiquitous, fleets of connected vehicles and their passengers will be increasingly vulnerable to Denial-of-Service (DoS) attacks where attackers disrupt the re-balancing of vehicles to make them unavailable to customers. Such attacks have already been reported: Uber claimed Lyft requested and canceled nearly 13,000 Uber rides and Lyft counted 5,560 canceled rides [19], [7], the goal being to steal each other's customers. Moreover, the vulnerability of self-driving cars to hacking is already a major concern. For example, General Motors created the new role of cybersecurity to protect the company's future autonomous vehicles [9]. Miller and Valasek suggested that it is possible to wirelessly control a fleet of 471,000 vehicles already on the road by exploiting a flaw in their Internet-connected feature [13]. Hence, our framework is also relevant for the impact analysis of DoS attacks on autonomous MaaS systems.

Cyber-security in transportation: The security of cyber-physical systems (and Internet of Things) have gained a lot of attention recently [3] because the consequences of cyber-attacks on them are not just financial, they could result in real-world and real-time physical problems. The vulnerability of transportation systems are real: two students hacked the traffic app Waze causing it to report a nonexistent traffic jam [34], a security researcher hacked traffic lights' sensors to trick their control systems into thinking that open roadways are congested [35]. Reilly et al. suggested different attack scenarios on Freeways via Coordinated Ramp Metering attacks [28].

B. Contributions and outline

To the best of our knowledge, we provide one of the first analysis frameworks for the financial impacts of DoS attacks on MaaS systems. Here are our contributions:

Detailed statistical methodology: Even though our model expands upon an established queueing-theoretical framework for the analysis of the sustainability benefits of MaaS systems, such as in [10], [36], we are among the first to provide

[1] Department of Electrical Engineering and Computer Sciences, University of California at Berkeley. {chenyang.yuan, jerome.thai, bayen}@berkeley.edu

[2] Department of Civil and Environmental Engineering, University of California at Berkeley.

a rigorous and detailed methodology for the learning and construction of our model. Starting from the representation of the taxi demand as a Poisson point process, we analyze the simplifying assumptions leading to the Jackson network model. This powerful mathematical framework enables to analyze the performance of networked systems at a macroscopic level. In general, queuing models have been widely used in transportation, computing, and telecommunication [23] and to design factories, shops, offices and hospitals [12].

How to attack in practice? We provide realistic scenarios of attacks on (autonomous) MaaS systems based on case studies of existing systems. Technically, it is possible to issue DoS attacks against Uber and Lyft with relatively low (material) costs, either by taking rides to make the service unavailable at the pick-up location, or cancelling rides. These rides can be made anonymous and cheap by purchasing short-lived phone numbers tied to human verification farms [32] and credit card numbers on black markets [5]. The possible attack of a fleet of connected vehicles would also be possible at a relatively low (material) price. As documented in [13], analyzing weaknesses in the vehicle's Internet-connected feature enables to gain access to the micro-controller and send commands to its physical parts. Assuming that all vehicles in the fleet have the same architecture, attacking a fleet only requires the analysis of one vehicle.

Modeling of the attacks: Attacks can be seen as malicious agents controlling the vehicles of the MaaS system, which we will refer to as *Zombie* passengers, following the computer science terminology *Zombie* for a computer that has been compromised remotely by a hacker to launch DoS attacks. Expanding an established framework in which the re-balanced MaaS system is cast into a queuing network where the city blocks in Manhattan can be seen as server nodes (or stations), and cars as packets moving between stations [36], one of our main contributions is to model the attacks as a stochastic process that controls a fraction of the packets (the cars) for malicious purpose. This malicious process is added to two stochastic processes introduced in [10], [36]: packets with customers (the taxi demand) learned from the taxi data provided by the NYC TLC, and a re-balancing process (the taxis being dispatched) to maintain high service availability in the network. Furthermore, to capture different types of attacks, 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. In the case of autonomous cars, the malicious behavior is more likely to be detected if the cars are controlled by the attacker for a long period of time. We also assume that the total rates of attacks is upper bounded by a budget b and study their impact with different values of b.

Impact of Large-scale attacks: Their effect is measured in *two steps*. 1) *a steady-state analysis* where we use the product-form stationary distribution to formulate an optimization program for an optimal *steady state* attack strategies that, e.g. maximize the customer loss or minimize the customer time usage of the system. Despite an intractable gradient computation ($O(N^4)$ complexity where N is the number of stations), we propose a block-coordinated descent algorithm in which each minimization block can be solved efficiently. 2) Then a *transient analysis* with a simulation of the network subject to the attack scheme computed in the previous step dynamically evaluates different metrics such as the increase in passenger loss or decrease in vehicle availabilities for one hour of attacks, see Figure 3.

Financial analysis: We propose a cost-benefit analysis to show the extent of damage that can be done with these attacks. Learning the queuing model from the taxi data provided by the NYC TLC, we show that raising the cost of attacks to \$15 is sufficient to deter rival companies from attacking via ride cancellations. Hence our framework will be usable to compute the optimal attack price-point of an attacker, hence helping cab companies to adjust the cost of attacking to protect themselves. The cost of attacks includes explicit costs (*e.g.* cancellation fees, hardware purchases), and hidden costs (*e.g.* probability of detection times the penalty).

II. Learning the Queueing model

We now introduce a discretization framework that can be used to study these systems in practice (and apply it to NYC).

A. A Poisson point process

We consider a bounded (geographical) region $R \subset \mathbb{R}^2$ and a time interval Ω in which a sequence of passenger rides $x_i = (t_i, o_i, d_i)$ for $i \in \mathbb{N}$ are requested, where t_i is the start time of the ride, $o_i \in R$ its origin, and $d_i \in R$ its destination. We model the sequence of ride requests as a Poisson point process $X = (X_t, X_o, X_d)$ in the *bounded* space $\Omega \times R \times R$ with an intensity function $\rho : \Omega \times R \times R \to \mathbb{R}_+$. For such a process, occurrences in a Lebesgue-measurable set $B \subseteq \Omega \times R \times R$ have locations that are independent and identically distributed (i.i.d.) in B with common density proportional to the intensity function ρ. Hence, a ride with origin o and start time t has its destination $d \in R$ distributed following:

$$P(X_d = d \,|\, o, t) = \frac{\rho(t, o, d)}{\int_{\{t\}\times\{o\}\times R} \rho} \tag{1}$$

by applying the above property with $B = \{t\} \times \{o\} \times R$.

For tractability, we discretize the region R into N tiles T_i indexed by $i \in \mathcal{S}$ and the time interval Ω into time windows of length Δt. Blocks are chosen small enough such that all trips end in a different block, and time intervals should be short so that the passenger demand can be assumed constant, see Figure 1 for an example of discretization in NYC. Then pickup requests with origin in tile T_i and in time window τ choose the destination tile T_j with probability

$$P(X_d \in T_j | X_t \in \tau, X_o \in T_i) = \frac{\int_{\tau\times T_i\times T_j} \rho}{\int_{\tau\times T_i\times R} \rho} \tag{2}$$

B. *Statistically learning the demand*

The pickup arrival rate in tile T and within time window τ follows a Poisson distribution with mean $\int_{\tau \times T \times R} \rho$. It is well-known that the sample mean is an unbiased minimum-variance estimator[1] (by achieving the Cramer-Rao lower bound), hence it is an efficient estimator of the Poisson process [15]. An example of sample mean computed for each tile in part of Manhattan is provided in Figure 1.

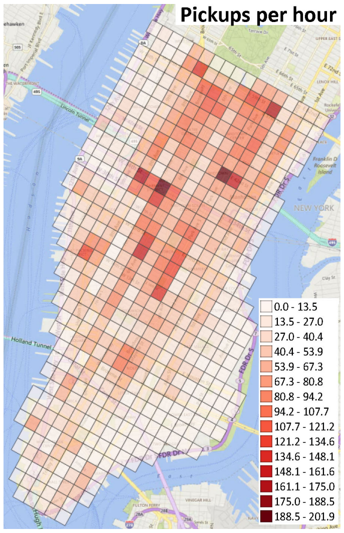

Fig. 1. **Average passenger arrival rates in Manhattan from January 2009 to June 2015 on weekdays from 5pm to 7pm, learned from a dataset of 1B taxi trips provided by the NYC TLC. The average pickup rate every 10min during weekdays is provided in our video: `https://www.youtube.com/watch?v=RwGttGlflsA`.**

From (2), the destination tiles T_j of a trip starting at tile T_i and in time interval τ follows a categorical distribution with probabilities denoted by p_{ij}^{τ}. The *maximum-a-posteriori* (MAP) of the parameters $\{p_{ij}^{\tau}\}_{j \in \mathcal{S}}$ is the mode of the posterior Dirichlet distribution

$$\text{MAP}(\{p_{ij}^{\tau}\}_{j \in \mathcal{S}} \,|\, \text{data}) = \frac{m_j + n_{ij}^{\tau}}{\sum_{k \in \mathcal{S}} m_k + n_{ik}^{\tau}} \tag{3}$$

where n_{ij}^{τ} is simply the number of trips starting at tile T_i in time interval τ and with destination T_j, and m_j are prior observations. Since we may not have any observations from the data,[2] we choose $m_k = 1$ for all k so that $m_k + n_{ik}^{\tau} > 0$. A possible improvement consists in choosing a prior distribution proportional to the destination arrival rates.

III. Queueing model

We now drop the superscript τ since we restrict our analysis to a specific time interval (5pm-7pm for the NYC case study). We have considered a MaaS system in an urban area divided into N tiles indexed by $i \in \mathcal{S}$. We assume that M vehicles provide service to customers between pairs of tiles $(i, j) \in \mathcal{S} \times \mathcal{S}$ and cast the MaaS system into a Jackson model. Since vehicles are 'processed' by a server in each tile, we will refer tiles as stations, which convey the fact that vehicles are queuing to be picked up by customers.

Type	rate	routing	contribution
customer	ϕ_i	α_{ij}	MAS model [10]
balancer	ψ_i	β_{ij}	re-balancing [36]
Zombie	ν_i	κ_{ij}	**cyber-security**

TABLE I
Different types of passenger with their arrival rates, routing probabilities, and the authors who introduced them.

A. *Three types of passengers*

We describe the model for vehicles picking up customers and re-balancing themselves in the network. Finally, we introduce our model for *Zombies*. Table I summarizes these three models. Section IV will justify our assumptions.

Customer model: Customers arrive at each tile i following a time-invariant Poisson process with rate $\phi_i > 0$. Upon arrival at a station i, a customer chooses to go to station $j \neq i$ 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 station 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 vehicles to queue for passengers, as is often the case of pickup locations or taxi stations. The travel times for different passengers traveling from station i to station 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 [10] 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 [20], [36] and we use the framework of [36] 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 [36], each station i generates balancers according to a Poisson process with rate $\psi_i \geq 0$ and routes these balancers to station $j \neq i$ 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 station upon its generation.

Cyber-security: We extend the re-balancing work of [36] for the purpose of cyber-security analysis. We assume the attacker can generate malicious agents or *Zombies* at each station i following a Poisson process with rate $\nu_i \geq 0$ and route them to station $j \neq i$ 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'

[1] Note that it is also a sufficient statistic for a Poisson distribution.

[2] In Figure 1, all observed trips starting at the edge of the region of study finish outside of it.

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. 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 of event A ($= 1$ if A is true, $= 0$ otherwise), we have the constraints

$$\mathbf{1}_{\{(i,j)\notin\mathcal{E}\}}\kappa_{ij} = 0 \quad \forall i, j \in \mathcal{S} \tag{4}$$

B. Jackson network model

Following [10] and [36], the model described above can be cast into a closed Jackson network, which we now present with a cyber-attack extension. 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

$$\lambda_i = \phi_i + \psi_i + \nu_i \tag{5}$$

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 station i to select a destination j is then given by $r_{ij} = \sum_{\text{class}} \mathbb{P}(i \to j \,|\, \text{class})\, \mathbb{P}(\text{class})$. With α_{ij}, β_{ij}, and κ_{ij} being the routing probabilities associated to each class, we have (with λ_i given by (5)):

$$r_{ij} = \alpha_{ij}\frac{\phi_i}{\lambda_i} + \beta_{ij}\frac{\psi_i}{\lambda_i} + \kappa_{ij}\frac{\nu_i}{\lambda_i} \tag{6}$$

Stations are modeled as single-server (SS) nodes (or "station" nodes) and the route between two stations as infinite-server (IS) nodes (or "route" nodes). When a generalized passenger arrives at a non-empty station, a vehicle departs from that node to move to a route node that connects 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 station node (see Figure 2).

From a queuing perspective, if vehicles are present at station i, they are processed with service rate λ_i given by (5), and are routed to the IS (route) node between stations i and j with probability r_{ij} given by (6). Then vehicles at an IS node between stations 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 j with probability 1. Hence, the MaaS 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

$$\mu_n(x_n) = \begin{cases} \lambda_i & \text{if } n = \text{station } i \\ x_n/T_{ij} & \text{if } n = \text{route } i \to j \end{cases} \tag{7}$$

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

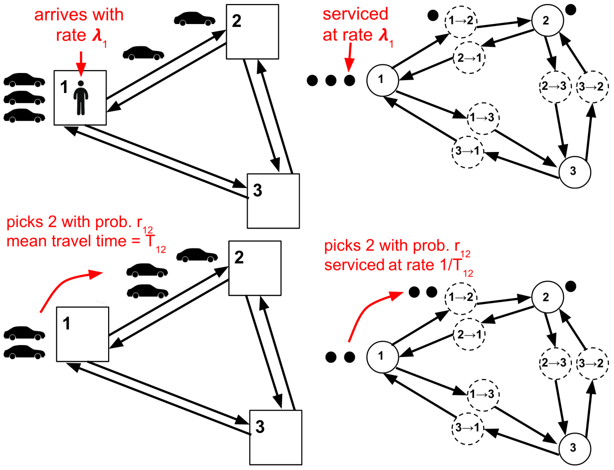
arrives with
rate λ_1
1
2
3
serviced
at rate λ_1
1
2
3
1→2
2→1
2→3
3→2
1→3
3→1
picks 2 with prob. r_{12}
mean travel time = T_{12}
picks 2 with prob. r_{12}
serviced at rate $1/T_{12}$

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

$$p_{nn'} = \begin{cases} r_{ij} & \text{if } n = \text{station } i,\ n' = \text{route } i \to j \\ 1 & \text{if } n = \text{route } i \to j,\ n' = \text{station } j \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

C. Asymptotic Behavior and Fairness

For Jackson networks, the throughput of vehicles π_n at a generalized node n satisfies $\pi_n = \sum_{n'} \pi_{n'} p_{n'n}$, and we can define the relative utilization at node n as $\gamma_n = \pi_n/\mu_n(1)$. If n is a station i, then $\gamma_i = \pi_i/\lambda_i$, *i.e.* vehicle throughput over passenger arrival rate. An important quantity is the availability $A_i(M)$, defined as the percentage of customers who find a vehicle available at a station upon arrival. It is given by the following steady-state probability (see [18]):

$$A_i(M) := \mathbb{P}(X_i \geq 1) = \frac{\gamma_i G(M-1)}{G(M)} \tag{9}$$

where X_i the queue length at station $i \in \mathcal{S}$. 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 [11]. 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 [27] gives the asymptotic availability at a SS node i:

$$a_i := \lim_{M\to\infty} A_i(M) = \frac{\gamma_i}{\max_{j\in\mathcal{S}} \gamma_j} \tag{10}$$

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2612828, IEEE Transactions on Control of Network Systems

IV. Comments on the model and methodology

A. Comments on the Jackson network assumptions

Exponential travel times: Although travel times are generally not exponentially distributed, the assumption does not affect the predictive accuracy of queuing networks [17].

Irreducibility: The customers' routing probabilities α_{ij} reasonably constitute an irreducible Markov chain for dense environments.

Passenger loss: Passengers not willing to wait (they leave the station immediately if no taxi is available) is accurate in numerous US markets: (i) with high service availability (the median wait time for an Uber in major U.S. cities in 2014 was under 4 min [26]), and (ii) in a competitive setting against other transportation systems (particularly in dense cities). 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. The loss model also considerably simplifies our model because customer arrivals at a station is equivalent to a virtual service to the vehicles currently queuing (and available) at the station.

Re-balancing and attack processes: The re-balancing and attacks are respectively modeled as balancers and *Zombies* similarly to the customer model (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 stations but does not enforce it. Besides, real-life re-balancing and attack processes are in general not stochastic. However, with large number of packets (our case study runs with 2500 taxis), the evolution of the stochastic processes tends to its fluid limit, thus approximating well its deterministic counterpart [16].

Local matching process: In our model, the matching only occurs locally between a nearby vehicle and a passenger. For carsharing, a vehicle at a station is matched to a passenger upon arrival. For a hailing app, there is no physical station. A 'station' represents instead a small area or tile (two city blocks in our case study), and a matching occurs when a requested vehicle picks up a passenger within the tile. If no vehicle is available at the station (or tile), the passenger gets impatient and leaves. Hence, the matching is only local and does not affect the (malicious) re-balancing process.

B. Independence of the three processes

Optimal static re-balancing strategy: The analysis is restricted 5-7pm weekday time period for the strong seasonality in the passenger demand, with only small variations within the time window. Instead of using an expansive real-time approach (with little optimality guarantees), we leverage the demand seasonality to efficiently compute (and predict) *optimal* strategies from the sample means of the rates and routing probabilities estimated from historical data (weekdays at 5-7pm). Hence, even though the parameters of the re-balancing process are a function of the historical passenger demand, both stochastic processes have constant parameters, and are thus independent. Hence, a real-time approach is only beneficial for large deviations from the historical means and the robustness of the static strategy can be assessed using estimated confidence intervals.

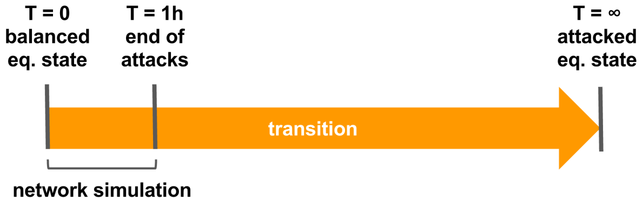

Fig. 3. **Steady-state analysis gives re-balancing and attack strategies for the balanced and attacked equilibrium states respectively ($T = 0$ and $T = \infty$). Since the attacked equilibrium state is not attainable in practice, a network simulation evaluates losses after 1h of attacks.**

The re-balancing does not respond to attacks: We assume that attacks are not detected by the system. This is a reasonable assumption if attackers directly compromise the re-balancing process, or if attacks occur for a short time period during which the system does not have time to respond. In fact, countering the attacks may require three high-latency steps: 1) detect with high confidence unusual deviations from the average queue lengths, 2) re-compute in real time efficient counter-measures, 3) re-dispatch vehicles after they completed their ongoing rides. And in our numerical experiment, we show that one hour of attack is sufficient to double the passenger losses.

C. Computing the impact of attacks

Step 1 - steady-state analysis: We use the product-form stationary distribution to formulate an optimization program for the optimal attack strategies following some attacker's objective. Despite the analytical benefit, this approach only optimizes for the *steady state*. Once attacks have started, the new equilibrium state may be reached after a long period of time. However, as discussed earlier, our framework only applies for a short time period, hence losses in equilibrium given by the analysis are likely to overestimate the losses over the 1-hour transient period. In addition, we assume that stations have sufficient capacity, but it is often optimal for attackers to send all vehicles to a single destination, which may overflow stations and breaks the assumption.

Step 2 - transient analysis: We simulate a Jackson network subject to the attack scheme computed in the previous step to dynamically evaluate the increase in passenger loss and decrease in vehicle availabilities for one hour of attacks. This second step resolves the limitations of the first step by computing transient losses, and not allowing stations to overflow, see Figure 3.

V. Steady-state analysis

The contributions in this section encompass the objectives of an attacker into an optimization framework. The steady-state strategy will serve as input for the network simulation.

A. 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 $\min \sum_{i \in \mathcal{S}} \phi_i A_i(M)$. If the MaaS system gets an amount that is proportional to the length of the ride, a more harmful objective is $\min \sum_{i,j \in \mathcal{S}} \phi_i \alpha_{ij} T_{ij} A_i(M)$ hence the total time usage for the customers is minimized.[3] Both objectives are of the 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 and we study the availability $A_i(M)$ when the fleet size M goes to ∞ (see (10))

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

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 $\{a_i\}_{i \neq k}$

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

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

B. Attack budget

The most important constraints are the traffic equations of the Jackson network. Using Lemmas 4.1 and 4.2 in [36], they can be written in terms of SS (station) 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 station k with probability 1 gives, using (15)

$$\phi_k + \psi_k + \nu_k = \sum_{j \neq k} (\alpha_{jk} \phi_j + \beta_{jk} \psi_j + \nu) a_j \geq \sum_{j \neq k} \nu a_j$$

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

Then the positive utilizations a_i go to 0 for all $i \neq k$ and the problem is reduced to $\min_{k \in \mathcal{S}} w_k$. Hence, a more realistic problem is setting a limited attack budget b: $\sum_{i \in \mathcal{S}} \nu_i \leq b$.

C. Formulation

Given the customers' and balancers' demands, we define their combined rate and routing probabilities as

$$\varphi_i := \phi_i + \psi_i \tag{16}$$

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

and so the combined routing probabilities r_{ij} of the customers, balancers, and *Zombies* given in (6) can be expressed as follows

$$r_{ij} = \frac{\delta_{ij} \varphi_i + \kappa_{ij} \nu_i}{\varphi_i + \nu_i} \quad \forall\, i,\, j \in \mathcal{S} \tag{18}$$

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:

$$\min_{\kappa_{ij}, \nu_i, a_i} \sum_{i \neq k} w_i a_i \tag{19}$$

$$\text{s.t. } a_i = \sum_{j \in \mathcal{S}} \frac{\delta_{ji} \varphi_j + \kappa_{ji} \nu_j}{\varphi_i + \nu_i} a_j \quad \forall\, i \in \mathcal{S} \setminus \{k\} \tag{20}$$

$$\kappa_{ij} \geq 0, \ \sum_j \kappa_{ij} = 1, \quad \mathbf{1}_{\{(i,j) \notin \mathcal{E}\}} \kappa_{ij} = 0 \tag{21}$$

$$\nu_i \geq 0, \ \sum_i \nu_i \leq b \tag{22}$$

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}:*

$$a_i > 0 \text{ for all } i \in \mathcal{S} \tag{23}$$

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

VI. Analytical Results in Steady-State

We first study a scenario in which the attacker aims at reducing the asymptotic availabilities at all but one station by a constant factor for a network in equilibrium. In this case, we show that the best strategy consists in routing all attacks to a single destination and we are able to derive analytical results for the rates of attacks.

A. Uniformly reducing availabilities

We consider a re-balancing network where the combined rate $\{\phi_i\}_i$ and routing probabilities $\{\delta_{ij}\}_{ij}$ of the real and re-balancing passengers are given, and we denote $\{a_i\}_{i \in \mathcal{S}}$ the resulting availabilities (before attacks). We consider a simple scenario in which the attacker reduces the availabilities at all stations by a constant factor, *i.e.* availability at station k is set to 1 and $\alpha \geq 1$ is maximized such that:

$$\tilde{a}_i = \begin{cases} 1 & \text{if } i = k \\ a_i/\alpha & \text{if } i \neq k \end{cases} \tag{25}$$

where $\tilde{a}_i$ are the availabilities resulting from the attacks. Now we propose and prove the optimality of an attack strategy that maximizes α.

[3] The distance D_{ij} between stations i and j can also be included in the objective since fares are usually a combination of the two.

Theorem 1. *Consider a (balanced) MaaS system with initial asymptotic availabilities $\{a_i\}_{i\in\mathcal{S}}$. If we are given a budget b for the attacks that is at least a certain amount:*

$$b \geq (1-a_k)\,\varphi_k \sum_{j\neq k} \delta_{kj}/a_j \tag{26}$$

Then the best attacks such that $\sum_i \nu_i \leq b$, resulting in station k having asymptotic availability equal to 1 and all other stations' availabilities decrease by the same factor $\alpha \geq 1$ can be achieved by the following policy:

$$\nu_i = \begin{cases} \frac{b\delta_{ki}}{a_i \sum_{j\neq k}\delta_{kj}/a_j} & \text{if } i \neq k \\ 0 & \text{if } i = k \end{cases} \tag{27}$$

$$\kappa_{ij} = \begin{cases} 1 & \text{if } i \neq k, j = k \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

We call it the "Single-Destination Attack Policy" (SDAP) since all attacks are routed to k. It results in:

$$\alpha = a_k + \frac{b}{\varphi_k \sum_{j\neq k}\delta_{kj}/a_j} \tag{29}$$

We make some comments on the *effectiveness* of attacks discussed presented in Theorem 1. Under the SDAP, $a_k = 1$ reduces condition (26) to $b \geq 0$, *i.e.* any budget leads to $\alpha \geq 1$. If $a_k < 1$, then $\alpha \geq 1$ requires a minimum positive budget given by (26). However, if $a_k < 1$ and (26) is not verified, then $\alpha < 1$ and re-normalizing so that we get valid asymptotic availabilities after attacks gives

$$\tilde{a}_i = \begin{cases} \alpha & \text{if } i = k \\ a_i & \text{if } i \neq k \end{cases} \tag{30}$$

where there exists $i \neq k$ such that $a_i = 1$. In this particular case, the attack only increases the asymptotic availability at station k while keeping other availabilities constant.

B. Case of balanced network under attacks

The result in Theorem 1 holds for MaaS systems with or without re-balancing passengers. If the MaaS is balanced, *i.e.* $a_i = 1$ for all $i \in \mathcal{S}$, then the SDAP reduces to

$$\nu_i = b\delta_{ki}\ \forall\, i \neq k, \quad \nu_k = 0 \tag{31}$$

$$\kappa_{ij} = \begin{cases} 1 & \text{if } i \neq k, j = k \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

resulting in $\tilde{a}_i = 1/\alpha$ for all $i \neq k$ and $\tilde{a}_k = 1$, with $\alpha = 1 + b/\varphi_k$. Hence, for a balanced network in equilibrium, the passenger loss incurred by this attack strategy when the fleet size approaches infinity is asymptotically

$$\sum_{i\in\mathcal{S}} w_i(a_i - \tilde{a}_i) = \sum_{i\neq k} w_i \frac{\alpha-1}{\alpha} = \frac{b}{\varphi_k + b}\sum_{i\neq k} w_i \tag{33}$$

We note that the attacks have great effects for small budgets, with incurred losses scaling linearly in b:

$$\sum_{i\in\mathcal{S}} w_i(a_i - \tilde{a}_i) \approx \frac{b}{\varphi_k}\sum_{i\neq k} w_i \quad \text{for} \quad b \ll \varphi_k \tag{34}$$

Hence, when routing the attacked vehicles to a single destination station k, it is best to pick a station k with low customer demand and low re-balancing rate $\varphi_k = \phi_k + \psi_k$ and small weight w_k. Concretely, an attack sending all the vehicles to a single station k aims at having an excess of supply at this station while depriving the rest of the network of vehicles. The quantity φ_k is the rate at which the vehicles are sent away from k from customer rides or re-dispatching, hence it is more effective to maliciously send vehicles in parts of the network with low activity.

C. Budget maximization as a prerequisite for optimality

We now show that all of the budget b has to be used for an attack to be optimal. While this result is intuitive and can be proved directly from the KKT conditions associated to the OAP, we present an alternate proof which gives additional insights on the OAP. Theorem 1 leads to the following result:

Theorem 2. *Equality $\sum_{i\in\mathcal{S}} \nu_i = b$ is a necessary condition for a solution of the OAP to be optimal.*

VII. Block-coordinate descent

In this section, we propose an algorithm to efficiently solve the OAP. Noting that first-order methods are not tractable because of the balance constraints, we propose a block-coordinate descent algorithm in which the three blocks can be solved very efficiently, two being *linear programs* (LP) with N^2 variables, and the third one a *quadratically constrained quadratic program* (QCQP) with N variables (N being the number of stations). We also add a small cost of attacking $p\sum_i \nu_i$ to the objective[4] such that objective becomes:

$$\min_{\kappa_{ij},\nu_i,a_i} \sum_{i\neq k} w_i a_i + p\sum_i \nu_i \tag{35}$$

The ℓ_1-regularization term is added for numerical reasons. Having a term in the objective that depends on the attack rates ν_i enables to pose the *Minimum Attack Problem* (MAP) for our block-coordinate descent algorithm, when the availabilities a_i are fixed. The MAP essentially computes a better re-allocation of the attacks (in terms of total rate minimization) to incur the same loss $\sum_i w_i a_i$ to the MaaS system. If the MAP computes a strictly better attack strategy, then necessarily $\sum_i \nu_i < b$, and from Theorem 2, the unused part of the budget can be used to increase the customer loss of the MaaS system, which is accomplished by the two other steps of the block-descent algorithm.

A. Non-tractable first-order methods

The OAP (19)-(22) is non-convex because the equality constraints in (20) are not linear, hence the well-known Lagrangian approach fail to provide sufficient conditions for optimality of a solution [2]. So one can only hope to find stationary points. In addition, first-order methods such as gradient descent algorithms are not tractable in practice. 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

[4] This can be seen as a ℓ_1-regularization term.

$\sum_{l\neq k} w_i \left\{\partial_{\kappa_{ij}} a_l\right\}_{(i,j)\in\mathcal{S}\times\mathcal{S}}$ where each partial derivative of a_i satisfies a set of $N-1$ linear equations obtained by differentiating the balance constraints (20). Hence, computing the gradient prohibitively requires to solve N^2 linear programs of dimension $N-1$ by differentiating the constraints (20). The total complexity for computing the gradient is $(N^2-N)^2 \geq (N-1)^4$, where N for a typical implementation of the model like in NYC is of the order of 500. One of our main contributions is the design of a tractable block-coordinate descent algorithm to solve the above problem. We pose the *Minimum Attack Problem* (MAP) and the *Attack Routing Problem* (ARoP) and show that they can be re-formulated as linear programs (LP) with N^2 non-negative variables and N constraints. We solve the MAP and ARoP efficiently with CPLEX. 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.

B. 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 stations $i\in\mathcal{S}$ 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:

THEOREM 3. *Let us consider the following linear program*

$$\min_{y_{ij}} \sum_{ij} w_i y_{ij} \tag{36}$$

$$\text{s.t.} \sum_{j\neq i}(\lambda_i y_{ij} - \nu_j y_{ji}) = \sum_{j>l} \delta_{ji}\varphi_j y_{jl} \quad \forall\, i\neq k \tag{37}$$

$$y_{ij}\geq 0, \quad \sum_{j\neq k} y_{kj} = 1 \tag{38}$$

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

$$a_i = \sum_{j\neq i} y^\star_{ij} \qquad \kappa_{ij} = y^\star_{ij}/a_i \tag{39}$$

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.

C. Attack Rate Problem (ARaP)

In this scenario, the attacker hacks the apps of the vehicles to display "ghost" demands at specific stations i. With fixed routing κ_{ij}, the attack rates ν_i are chosen to achieve objective (19). 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*

$$(D-M)\,x_j = v_j \quad \forall\, j\in\mathcal{S} \tag{40}$$

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:

$$\{\nu_i\}_{i\in\mathcal{S}} := \Pi\left(\{\nu_i\}_{i\in\mathcal{S}} - t\, g\right) \tag{41}$$

$$g := \sum_{i\neq k} (\partial a_i/\partial \nu_j)_{j\in\mathcal{S}} + p \tag{42}$$

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 [6]. 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.

$$t\leftarrow t/2 \quad \text{while} \quad f(\{\nu_i\}_{i\in\mathcal{S}} - t\,g) > f(\{\nu_i\}_{i\in\mathcal{S}}) \tag{43}$$

D. 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 cost of attacks $\sum_i \nu_i$. The *Minimum Attack Problem* (MAP) can be formulated as follows

$$\min_{\kappa_{ij},\nu_i} \sum_i \nu_i \tag{44}$$

$$\text{s.t. } a_i = \sum_{j\in\mathcal{S}} \frac{\delta_{ji}\varphi_j + \kappa_{ji}\nu_j}{\varphi_i+\nu_i} a_j \quad \forall\, i\in\mathcal{S}\setminus\{k\} \tag{45}$$

$$\kappa_{ij}\geq 0, \ \sum_j \kappa_{ij} = 1, \quad \mathbf{1}_{\{(i,j)\notin\mathcal{E}\}}\kappa_{ij} = 0 \tag{46}$$

$$\nu_i \geq 0 \quad \forall\, i\in\mathcal{S} \tag{47}$$

The constraints can be formulated as flow constraints

THEOREM 4. *Let us define*

$$s_i := a_i\varphi_i - \sum_{j\neq i} a_j\delta_{ji}\varphi_j \quad \forall\, i\in\mathcal{S} \tag{48}$$

and consider the following Linear Program

$$\min_{\{x_{ij}\}_{i\neq j}} \sum_{i,j\neq i} \frac{x_{ij}}{a_i} \tag{49}$$

$$\text{s.t.} \sum_{j\neq i}(x_{ji} - x_{ij}) = s_i \qquad \forall\, i\in\mathcal{S} \tag{50}$$

$$x_{ij}\geq 0 \quad \mathbf{1}_{\{(i,j)\notin\mathcal{E}\}} x_{ij} = 0 \qquad \forall\, i,j\in\mathcal{S} \tag{51}$$

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

$$\nu_i = \sum_{j \neq i} x^\star_{ij}/a_i \tag{52}$$

$$\kappa_{ij} = \begin{cases} x^\star_{ij}/(\nu_i a_i) & \textit{if } \nu_i > 0 \\ 1/\sum_j \mathbf{1}_{\{(i,j)\in\mathcal{E}\}} & \textit{otherwise} \end{cases} \tag{53}$$

Within the proposed block-coordinate descent framework, we add the budget constraint (22) 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 [36] for the purpose of re-balancing vehicles with an objective minimizing the number of re-balancing trips

$$\min_{\psi_i,\,\beta_{ij}} \sum_{i,j} \psi_i T_{ij} \beta_{ij} \tag{54}$$

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 stations, thus avoiding numerical corner cases associated to the sparsity promoting constraint (22).

Algorithm 1 Algorithm for solving the AOP.

```
choose arbitrary station $k \in \mathcal{S}$.
initialize $\nu_i$ and $\kappa_{ij}$
while stopping criteria not satisfied:
    update $a_i$, $\kappa_{ij}$ via Attack Routing Pb. (ARoP) with $\nu_i$ fixed.
    update $\nu_i$, $\kappa_{ij}$ via Min Attack Pb. (MAP) with $a_i$ fixed.
    update $a_i$, $\nu_i$ via Attack Rate Pb. (ARaP) with $\kappa_{ij}$ fixed.
return $a_i$, $\nu_i$, $\kappa_{ij}$
```

VIII. Quantifying countermeasures

We now study the economics of the resiliency of MaaS systems to DoS attacks and illustrate our results with a case study in Manhattan. In particular, we conduct a cost-benefit analysis and find that raising the expected cost of attacks to 1.5 times the gain for the attacker from incurring passenger loss protects MaaS systems from DoS attacks.

A. Data sources and methodology

For our case study in Manhattan, we choose tiles approximately of the size of two city blocks, which is a good trade-off between precision and tractability. Manhattan is divided into 531 tiles (see Figure 1), which gives a problem with $531^2 \approx 300,000$ decision variables that can be solved efficiently. The time windows are chosen to be one/two-hour long which is small enough to ignore time variability in the taxi demand. Using the 1.1 billion taxi trips from January 2009 to June 2015 provided by the NYC TLC, we extracted 75M passenger rides on all weekdays between 5pm and 7pm and we learned the customer demand ϕ_i, α_{ij} using the methodology presented in Section 2. The total customer arrival rate is about 10,600 per hour (see Figure 1) and there are about 2,500 taxis in the network in this time period.

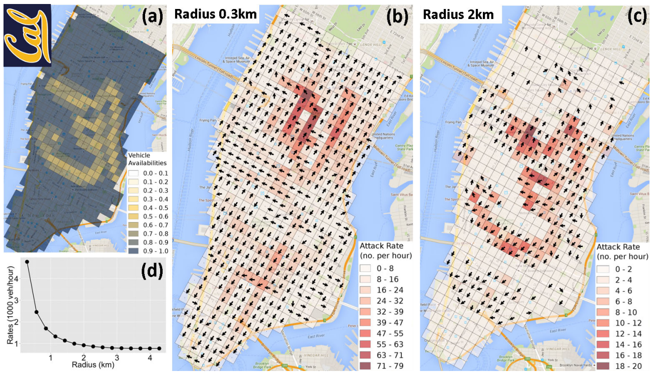

Fig. 4. **Effect of Radius of Attacks. (a): Target availability pattern following a pixelated version of the “Cal” logo. (b), (c): Best attack policy to achieve the target with maximum ℓ_1-radius of 0.3km (1 block) and 2km (7 blocks) respectively: each arrow shows the direction of the κ_{ij}-weighted barycenter of the destination stations 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. With small attack radius (b), vehicles are routed through many intermediate stations, whereas in (c), cars from regions with low availabilities are directly sent to the borders of Manhattan. Hence, limiting the attack radius greatly hinders the attacks’ effectiveness.**

We then solve the MAP with objective $\min \sum_{i,j} \phi_i T_{ij} \alpha_{ij}$ to estimate the optimal re-balancing process ψ_i, β_{ij}. Combining the customer demand and balancing process (assuming the system is balanced), the solution of the OAP provides an attack strategy that maximizes the passenger loss in the network. While the OAP is a useful framework for computing optimal attack strategies for a system in equilibrium, we also simulate a Jackson network with $N^2 \approx 300,000$ nodes, described in (7), (8), to dynamically estimate the passenger loss L incurred by the attacks during the first hour after the attacks have started.

B. Cost-benefit analysis

Following the methodology in [4], we propose an economic model of supply and demand for attacks. Assuming that attackers make rational decisions, we model a market of attacks in which the attacker wants to maintain a positive profit given by $\alpha L - \beta \sum_i \nu_i$ where αL is the gain for the attacker as a linear function the incurred passenger loss L and $\beta \sum_i \nu_i$ the cost of the attacks. Here, L are the transient losses which are obtained from network simulation in the transient analysis step, not to be confused with the objective in (12), see Section IV-C. The parameters α and β can be seen as a level of security, where the security increases if α is lower and β higher. Hence, given a level of security (α, β), attackers balance costs against benefits. We now provide some estimates of α, β.

Explicit cost of attacks: The explicit cost of attacks is generally very low. For instance, for ride-sharing services such as Uber or Lyft, pickup requests being cancelled using a real account cost \$5 per unit. The cost of a fake account is less than \$1 since both credit card numbers and phone numbers (tied to human verification farms) reportedly cost less than \$0.5 per unit [32], [5]. In addition, there is a *fixed cost*, *e.g.* the hardware required for generating the attacks. Following the attack on Waze [34], it is possible to emulate Android phones on a computer. Based on the following study [13], an attack on a fleet of Internet-connected autonomous vehicles requires the analysis of the hardware of one vehicle to be able to gain remote access to other vehicles of the fleet. Hence, the fixed cost of attacking MaaS systems is independent of the fleet size and the rate of attacks.

Hidden cost of attacks: The hidden costs are arguably much higher than the explicit costs. For current ride-sharing systems such as Uber and Lyft, suspicious (or malicious) accounts can be detected and blocked easily, along with its associated phone and credit card numbers. Buying phone and credit card numbers on the black markets has a risk of being caught by law enforcement agencies. These hidden costs can be modeled as $\beta^{\text{hidden}} = P(\text{detection}) \times \text{Penalty}$ *i.e.* a probability of being detected times the penalty of being caught. Hence, more efficient law enforcement and crimes detection can achieve a higher level of security by increasing $P(\text{detection})$ and the Penalty. It is worth noting that some taxi companies, *e.g.* Taxis G7 in France (`http://www.taxisg7.fr/`), does not require the creation of a PIN verified account to make a request, hence $P(\text{detection}) = 0$ and the only (explicit) cost is the call (\$.16/min). Hidden costs also include the working time necessary for designing DoS attacks. The cost of labor can be high and the number of hours necessary for designing an attack is an increasing function of the level of protection of cyber-physical systems against security breaches.

Gain for the attacker: Reasons for DoS attacks are multiple, *e.g.* extortion, blackmail, expression of anger and criticism, punishment (for refusing an extortion demand), see: `zeltser.com/reasons-for-denial-of-service-attacks/`. Because of the wide variety of motives, the benefits should be estimated case by case. In the case of anti-competition practice in two-sided networks (*e.g.* Uber and Lyft), the gains for DoS attacks can be enormous since successful platforms enjoy increasing returns to scale [29]. The high costs and high benefits of attacks on a large-scale MaaS system justifies the need of a business model for the attacker to make rational decisions.

C. Controlling availabilities

In this experiment, we find the minimal cost of attacks such that the resulting availabilities match an arbitrary set of availabilities a_i for $i \in \mathcal{S}$, such as the "Cal" logo, see Figure 4a. Assuming a balanced MaaS system, we first balance the network using the methodology of [36], *i.e.* solving the MAP (45)-(47) with the availabilities uniformly equal to 1 and with an objective that minimized the number of re-balancing vehicles (54). This yields a total rate of 2,200 re-balancing vehicles per hour. We then compute the attack strategy on the balanced network by solving the MAP for different attack radii. With unlimited attack radius, injecting only 800 Zombies per hour achieves the availability pattern encoded in the "Cal" logo. Assuming that a unit of attack is \$5 (current cancellation for a Uber/Lyft ride), only \$4000 per hour is sufficient to deplete the network following this pattern. With limited attack radius (routing only allowed between stations i and j within 15 blocks from each other in terms of Manhattan distance), a higher rate of attack is needed to reproduce the logo, see Figure 4d.

D. Minimizing availabilities

To avoid numerical difficulties related to the large disparities in customer arrival rates, we cluster adjacent blocks together such that the minimum aggregated arrival rate at a station is 30 customers per hour, resulting in a reduction to 331 blocks. We then balance the network and apply the proposed block-coordinate descent algorithm for solving the OAP with an objective minimizing the customer time usage in the network, *i.e.* (12) with $w_i = \sum_j \phi_i \alpha_{ij} T_{ij}$. The block-coordinate descent is given by Algorithm 1.

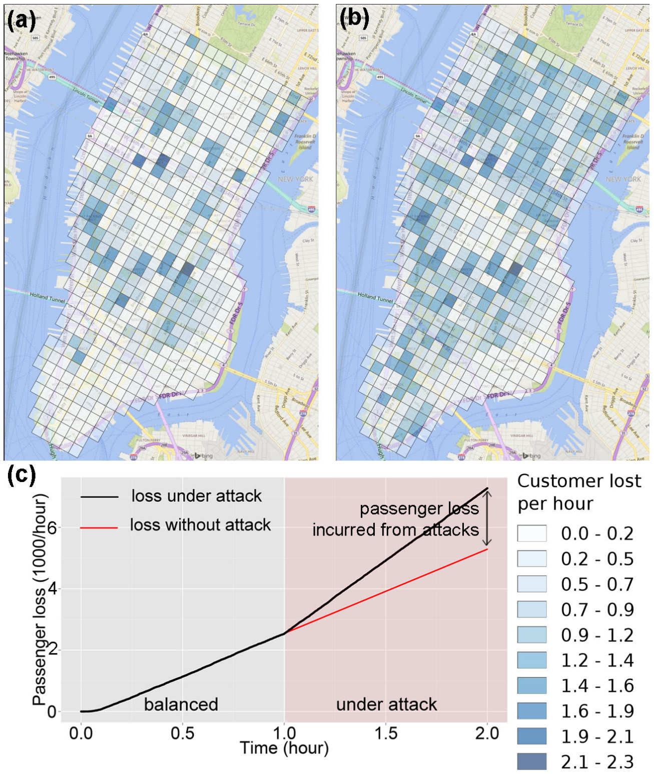

Fig. 5. **Network Simulation Results. A simulation is run with 2650 taxis in a Jackson network. After 1 hour of balancing, the network is attacked (following a strategy given by a solution to the OAP). The budget of attacks is 3000 requests per hour, corresponding to 19% of the total rate. The figure shows the passenger loss in log-scale per station over (a): 1 hour of balancing, (b): 1 hour of attacks. (c) shows the total number of customers lost over time. The total cumulative loss is slightly above 2000 passengers one hour after the start of the attacks.**

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 between 100 and 10,000 with the weight p of the ℓ_1-penalty 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 total attack rate accounts for 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, the OAP gives an attack strategy sending *Zombies* to several spots around the center of Manhattan, see Figure VIII-Ea and b. In equilibrium, these target regions have high availabilities while the rest of Manhattan has very low availabilities. These results are similar to the analytical ones in Section VI, where it was proved that the optimal attack strategy is one that sends all the vehicles in a single destination station (see Theorem 1).

E. 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 get blocked in the center region because the re-dispatch process does not send the vehicles in other parts of the network in reaction to the attacks. To account for the transient state, we run a simulation of the Jackson network used for our model with 2500 taxis (average number of taxis in the area at the time of the day used for our parameter inference). We record the number of customers lost for one hour and subtract from this the base rate of loss when the network is balanced. One run of a Jackson network simulation is presented in Figure 5 for a budget of 3000 attacks per hour. Slightly above 2000 passengers are lost after one hour of attacks. This gives the seventh sample point in Figure VIII-Ec. Figure VIII-Ec and VIII-Ed show the results of our analysis. Assuming that the cost of an attack is $5 (the cost of canceling an Uber/Lyft 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 VIII-Ec 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 system against attacks. This can be generalized to a cost of attacks being approximately 1.5 times higher than the gain from incurring passenger loss.

IX. Conclusions and future work

We described an analysis framework for quantifying the vulnerability to MaaS systems to DoS attacks. The Jackson network model enables to formulate a mathematical program for attack strategies that maximize the passenger loss in equilibrium. The strategy is then implemented on a network simulation to dynamically estimate the passenger loss incurred by the attacks. We then present a cost-benefit analysis applied to a case study in Manhattan. In the context of anti-competition practice, it is demonstrated that DoS attacks costing more than $15 per unit protects the MaaS system. The present work can be refined by, *e.g.*, designing an optimization program directly maximizing the transient losses over a short time horizon, relaxing the assumption of infinite capacity stations, and proposing countermeasures such as an anomaly detection algorithm.

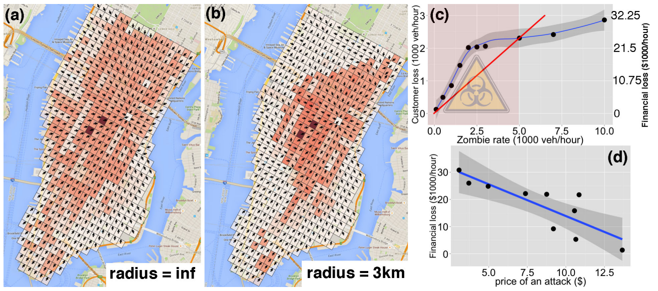

Fig. 6. **Optimal Attack Rates and Routing. (a) and (b): The attack rates and routing probabilities for a total budget of 2000 *Zombies* per hour are showed in the same style as in Figure 4, with an unlimited radius and 3km (9 squares) radius respectively. (c): Passenger/financial loss as a function 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, 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.**

Acknowledgments

The authors thank Cathy Wu, Ramtin Pedarsani, and Chao Zhang for insightful discussions on the problem.

References

[1] Urban life: Open-air computers. *The Economist*, 2012.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, March 8 2004.

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

[4] J. J. Cordes. An Overview of the Economics of Cybersecurity and Cybersecurity Policy. *CSPRI Report*, 2011.

[5] B. Dean. Your stolen credit card data is probably worth only 50 cents on the black market. *The Week*, 2015.

[6] 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.

[7] E. Fink. Uber's dirty tricks quantified: Rival counts 5,560 canceled rides. *CNN*, 2014.

[8] L. Gannes. Here's What It's Like to Go for a Ride in Google's Robot Car. *recode*, 2014.

[9] B. Geier. Car hacking: how big is the threat to self-driving cars? *Fortune*, 2014.

[10] 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.

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

[12] L. V. Green, P. J. Kolesar, and W. Whitt. Coping with time-varying demand when setting staffing requirements for a service system. *Production and Operations Management*, 16, 2007.

[13] A. Greenberg. Hackers remotely kill a jeep on the highway - with me in it. *Wired*, 2015.

[14] E. Huet. Uber's Global Expansion In Five Seconds. *Forbes*, 2014.

[15] R. W. Keener. *Theoretical Statistics*. Springer in Statistics, 2010.

[16] T. G. Kurtz. Limit theorems for sequences of jump markov processes approximating ordinary differential processes. *Journal of Applied Probability*, 8(2):344–356, 1971.

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

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

[19] R. Lawler. Uber Strikes Back, Claiming Lyft Drivers And Employees Canceled Nearly 13,000 Rides. *TechCrunch*, 2014.

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

[21] E. Mack. Elon Musk: Don't fall asleep at the wheel for another 5 years. *CNET*, 2014.

[22] J. McDuling. *Quartz*, 2014.

[23] D. A. Menascé, V. A. F. Almeida, and L. W. Dowdy. *Performance by Design: Computer Capacity Planning by Example*. Prentice Hall, 2004.

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

[25] W. J. Mitchell, C. E. Borroni-Bird, and L. D. Burns. *Reinventing the Automobile: Personal Urban Mobility for the 21st Century*. The MIT Press, 2010.

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

[27] 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.

[28] 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.

[29] J.-C. Rochet and J. Tirole. Platform Competition in Two-Sided Markets. *Journal of the European Economic Association*.

[30] C. Shunk. Average cost of car ownership rises to $8,946 per year. *Autoblog*, 2012.

[31] T. Slavin. Unless we stop driving cars, all other sustainable transport plans are pointless . *The Guardian*, 2015.

[32] K. Thomas, D. Latskiv, E. Bursztein, T. Pietraszek, C. Grier, and D. McCoy. Dialing Back Abuse on Phone Verified Accounts. *ACM CCS Conference*, 2014.

[33] N. Trejos. Zipcar expands fleet to more airports. *USA Today*, 2015.

[34] N. Tufnell. Students hack Waze, send in army of traffic bots. *Wired*, 2014.

[35] K. Zetter. Hackers can mess with traffic lights to jam roads and reroute cars. *Wired*, 2014.

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

Appendix

Proof of Lemma 1: By assumption, the probabilities α_{ij} constitute an irreducible Markov chain. By equation (6), the probabilities r_{ij} lead to an irreducible Markov chain as well. The $\{a_i\}_i$ vector satisfying equations (20) is proportional to the steady state distribution for the transition probabilities $\{r_{ij}\}_{ij}$ and by the Perron-Frobenius theorem, it is positive [24]. Finally, the constraint $a_k = 1$ completely fixes the vector $\{a_i\}_i$. □

Proof of Theorem 1: The balance equations before attacks are:

$$\sum_{j\neq i} a_j\varphi_j\delta_{ji} = a_i\varphi_i \quad \forall i \in \mathcal{S} \tag{55}$$

After attacks, the equations can be written as:

$$\sum_{j\neq i} \tilde{a}_j(\nu_j\kappa_{ji} + \varphi_j\delta ji) = \tilde{a}_i(\nu_i + \varphi_i) \quad \forall i \in \mathcal{S} \tag{56}$$

Given (25), the above equation at index k is:

$$\sum_{j\neq k} \frac{a_j}{\alpha}(\nu_j\kappa_{jk} + \varphi_j\delta jk) = \nu_k + \varphi_k \tag{57}$$

$$\frac{1}{\alpha} = \frac{\nu_k + \varphi_k}{\sum_{j\neq k} a_j(\nu_j\kappa_{jk} + \varphi_j\delta jk)} \tag{58}$$

We first maximize α with respect to the routing probabilities $\{\kappa_{ij}\}_{ij}$, which is clearly achieved when κ_{ij} satisfies the policy (28). As a result, equations (56) combined with (25) and (28) become:

$$\sum_{j\notin\{i,k\}} \frac{a_j}{\alpha}\varphi_j\delta ji + \varphi_k\delta_{ki} = \frac{a_i}{\alpha}(\nu_i + \varphi_i) \quad \forall i \neq k \tag{59}$$

Multiplying by α and subtracting (56) on both sides:

$$\varphi_k\delta_{ki}(\alpha - a_k) = a_i\nu_i \quad \forall i \neq k \tag{60}$$

$$\alpha = a_k + a_i\nu_i/(\varphi_k\delta_{ki}) \quad \forall i : \delta_{ki} > 0 \tag{61}$$

From (60), ν_i is proportional to δ_{ki}/a_i for all $i \neq k$, thus

$$\frac{\nu_i}{\sum_{i\neq k}\nu_i} = \frac{\delta_{ki}/a_i}{\sum_{j\neq k}\delta_{kj}/a_j} \quad \forall i \neq k \tag{62}$$

Plugging the above expression into (61)

$$\alpha = a_k + \frac{\sum_{i\neq k}\nu_i}{\varphi_k\sum_{j\neq k}\delta_{kj}/a_j} \tag{63}$$

Hence α is maximized when $\sum_{i\neq k}\nu_i = b$, setting $\{\nu_i\}_{i\in\mathcal{S}}$ to follow policy (27) (using (62)). We verify that the policy derived above is feasible given (55). Finally, we want $\alpha \geq 1$, which implies (26). □

Proof of Theorem 2: Suppose $b > 0$ (otherwise there is no attack). Let $(a_i, \nu_i, \kappa_{ij})$ be a feasible solution of the OAP such that $\sum_{i\in\mathcal{S}}\nu_i < b$. We show that it is not optimal. We combine the Zombies to the real and re-balancing passengers:

$$\tilde{\varphi}_i := \varphi_i + \nu_i \tag{64}$$

$$\tilde{\delta}_{ij} := (\delta_{ji}\varphi_j + \kappa_{ji}\nu_j)/(\varphi_i + \nu_i) \tag{65}$$

$$\tilde{b} := b - \sum_{i\in\mathcal{S}}\nu_i > 0 \tag{66}$$

Then applying policy the SDAP with $\tilde{\varphi}_i$, $\tilde{\delta}_{ij}$, $\tilde{b}$, a_i and k such that $a_k = 1$ decreases the a_i for $i \neq k$ by a factor $\alpha > 1$ (using (29) and the assumptions that $b, \varphi_k > 0$) Since the w_i's are positive by assumption and the a_i's are positive from Lemma 1, the objective decreases by a positive amount. Let us denote $\tilde{\nu}_i$ and $\tilde{\kappa}_{ij}$ the resulting attack policy. Then, the combination of (ν_i, κ_{ij}) and $(\tilde{\nu}_i, \tilde{\kappa}_{ij})$ given by $\tilde{\nu}_i + \nu_i$ and $(\tilde{\kappa}_{ji}\tilde{\nu}_j + \kappa_{ji}\nu_j)/(\tilde{\nu}_i + \nu_i)$ is still feasible for the OAP and decreases the objective by a positive amount. □

Proof of Theorem 4: The following change of variables

$$x_{ij} := \nu_i\kappa_{ij}a_i \quad \forall i, j \tag{67}$$

converts the MAP into the above program with $\{s_i\}_{i\in\mathcal{S}}$ given by (48) and $\nu_i = \sum_{j\neq i} x_{ij}/a_i$ as a result of the change of variable. This problem is feasible because the capacity on each edge is unbounded and the source flows sum to 0:

$$\sum_i s_i = \sum_i a_i\varphi_i - \sum_{i,j\neq i} a_j\delta_{ji}\varphi_i = 0 \tag{68}$$

Therefore, we can find the minimal-cost attacks that achieve any arbitrary availabilities. Finally, if $x^\star_{ij}$ minimizes $\sum_{i,j} x_{ij}/a_i$ then $\nu^\star_i$ given by (52) minimizes $\sum_i \nu_i$, and feasibility of $\kappa^\star_{ij}$ given by (53) can be checked, hence optimality of $\nu^\star_i$ and $\kappa^\star_{ij}$. □