Robust and Secure Routing For Queuing Networks and Internet of Vehicles

Qian Xie
qianxie@nyu.edu

July 10, 2020

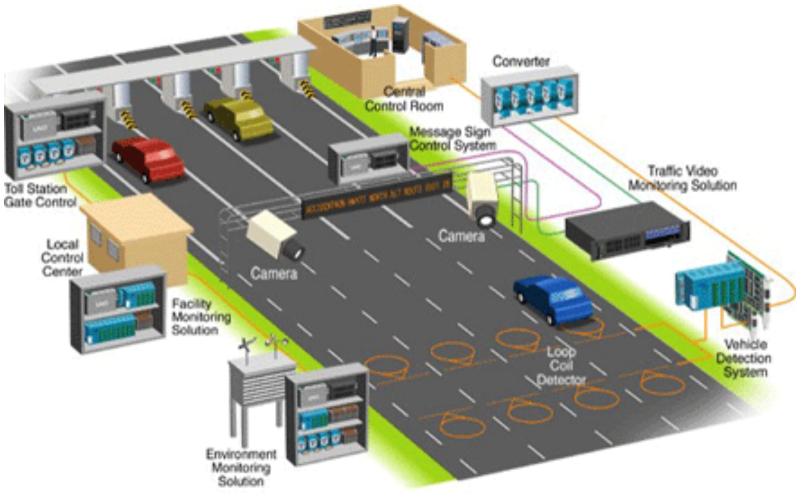
NYU | TANDON SCHOOL OF ENGINEERING

Outline

- Background of queuing model
- Robust routing for network systems
- Application to district routing
- Secure routing for parallel queues

Queuing model

- What it captures: queuing due to random arrival and/or random service time
- What it not captures: demand & capacity fluctuations
- Study topics: routing, sequencing, service rate control, admission control
- Applications: transportation, manufacturing networks (production lines), communication/computer networks



Robust routing for network systems

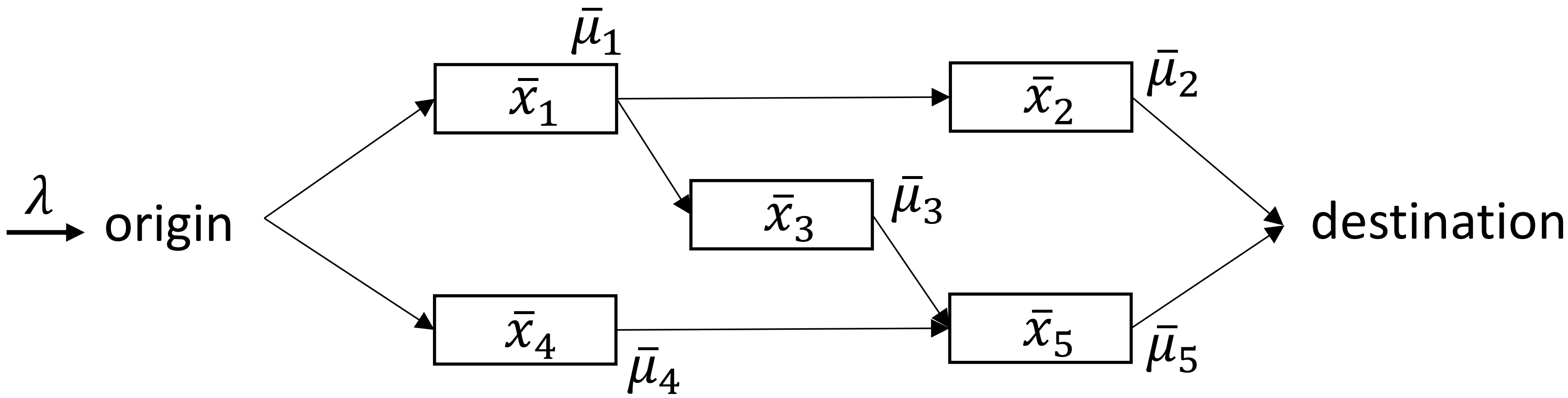
- In practical settings, model data may be
  - unavailable
  - hard to estimate
  - vary over time
- Suppose that we know the topology of a network, but do not know the demand and supply/capacity

Learning-based vs. robust control

- How to make decisions in an unknown environment?
- Solution 1: learn the environment from observation
  - learning-based adaptive control
  - efficient & smart
  - requires sufficient data
  - vulnerable to unhealthy data
- Solution 2: independent of environment parameters
  - robust control
  - easy & robust
  - guarantee stability but not efficiency
  - resist modeling error and/or non-stationary environment
- Solution 2 motivates model-based independent control

Formulation

- Multi-class Jackson queuing network with with Poisson arrivals & exponential service times
- Multiple origins, multiple destinations, acyclic
- Real-time OD-specific queue sizes can be observed
- Control actions: routing, sequencing, and holding
- Arrival and service rates unknown

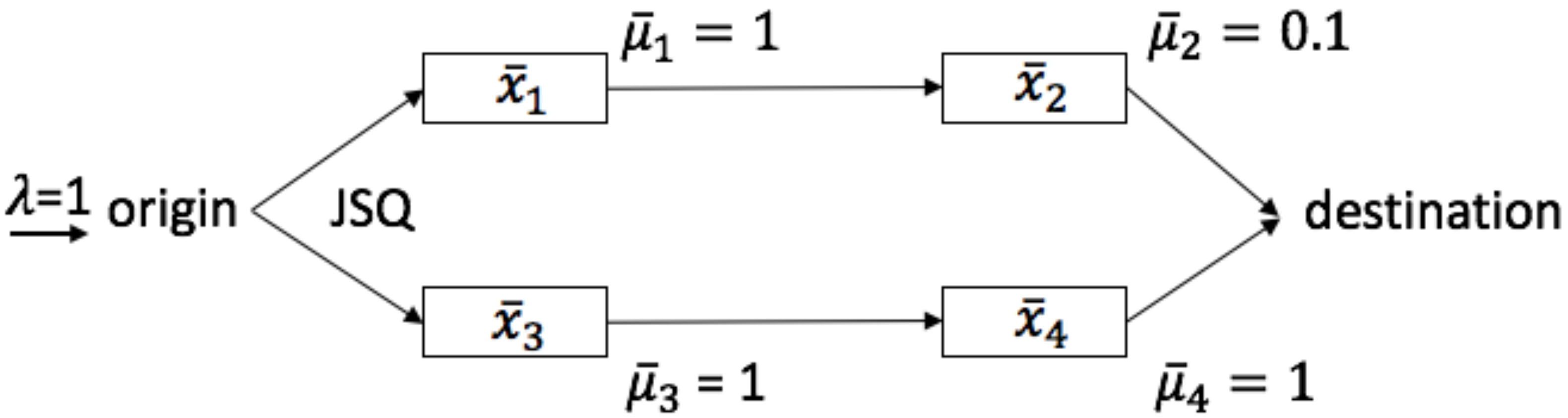


Join the shortest queue

- Simple case: parallel queues
- Intuitive routing policy: join the shortest queue (JSQ)
  - route the arrival to the shortest queue
  - ties are broken uniformly at random
- Standard results:
  - System is stable if and only if arrival rate < total service rate
  - Optimal for symmetric servers
- MDI: no info about arrival/service rates are needed
- Throughput-maximizing: if demand < capacity, then system is stable

JSQ fails for networks

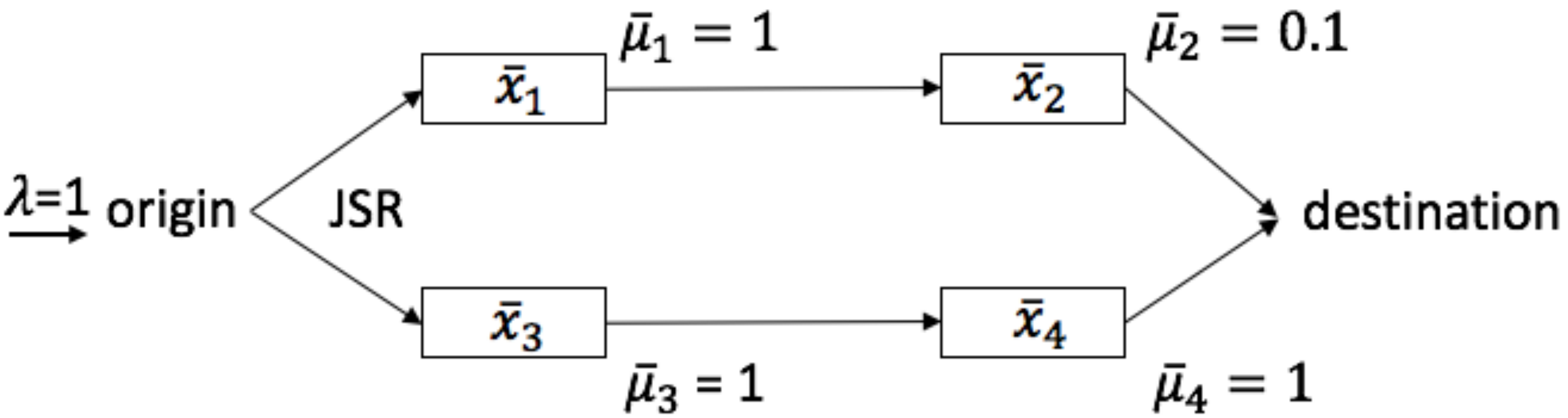
- Can we extend JSQ to networks? No!



- By symmetry & Burke's theorem, departure process from servers 1 & 3 are both Poisson of rate 0.5
- However, 0.5 exceeds the service rate of server 2 (0.1)
- Thus, the network is unstable!

Solution: join the shortest route

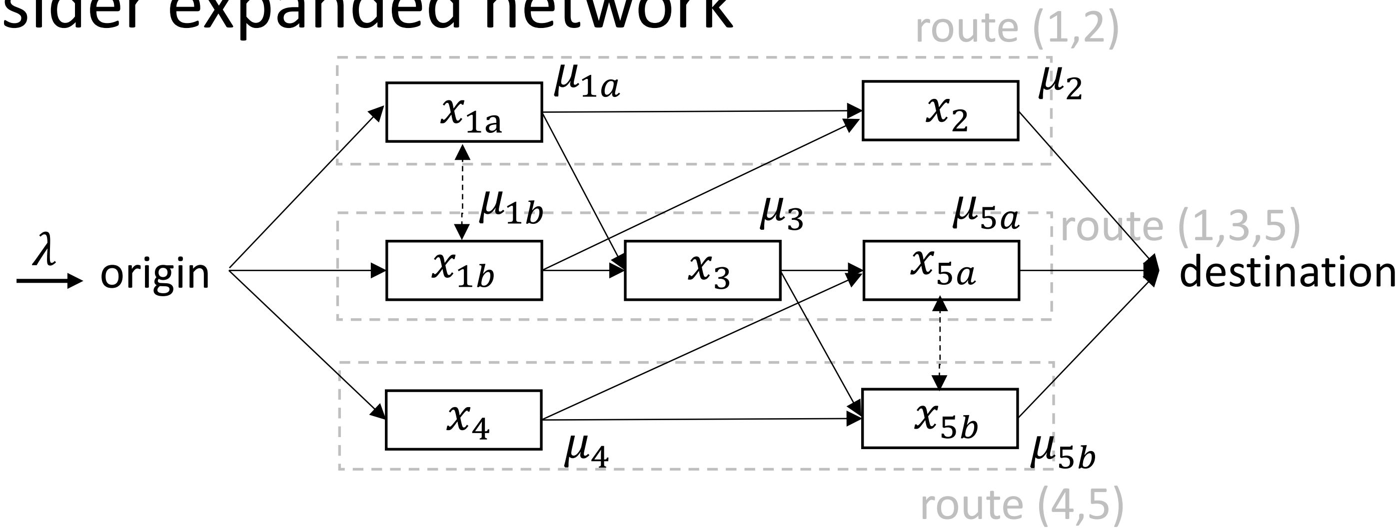
- Why JSQ fails?
  - Server 2 will be congested, but such information is not used at the



- To fix this, consider the total queue sizes on each route:
  - Join queue 1 if $\overline{x}_1 + \overline{x}_2 < \overline{x}_3 + \overline{x}_4$
  - Join queue 3 if $\overline{x}_1 + \overline{x}_2 > \overline{x}_3 + \overline{x}_4$
  - Ties broken uniformly at random
- Improve JSQ to JSR

How about more complex networks?

- What if the network is not parallel/serial?
- Consider expanded network



- Then the previous route-sum is not easy to extend.
- We consider an alternative:
- $y_1 = \max\{x_{1a}, \frac{1}{2}(x_{1a} + x_2)\}$, $y_2 = \max\{x_{1b}, \frac{1}{2}(x_{1b} + x_3), \frac{1}{3}(x_{1b} + x_3 + x_{5a})\}$, $y_3 = \max\{x_4, \frac{1}{2}(x_4 + x_{5b})\}$

Multi-class centralized control

- Join the shortest "route": $\min_k y_k$
- This applies to multi-class (multi-OD) traffic
- Centralized control: requires global information
- JSR is model-data independent
- Joint work with Li Jin (submitted to IEEE-TCNS)

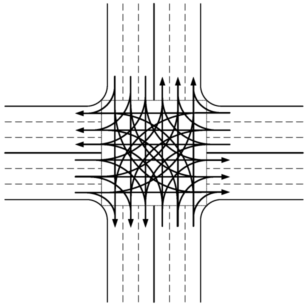
Single-class decentralized control

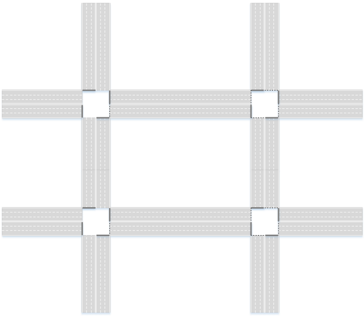
- How about decentralized setting?
  - The decision at each server is based on the local traffic information
- Why JSQ does not work for networks?
  - Congestion info cannot propagate to upstream servers
- Solution: artificial holding to propagate such info
  - keep upstream queue size > downstream queue size
  - e.g. subserver 1b is not allowed to discharge if $x_{1b} \leq x_3$
- JSQ with artificial spillback!
- Joint work with Li Jin (submitted to IEEE-TCNS)

Application to district routing

Find routes for all CAVs in a district

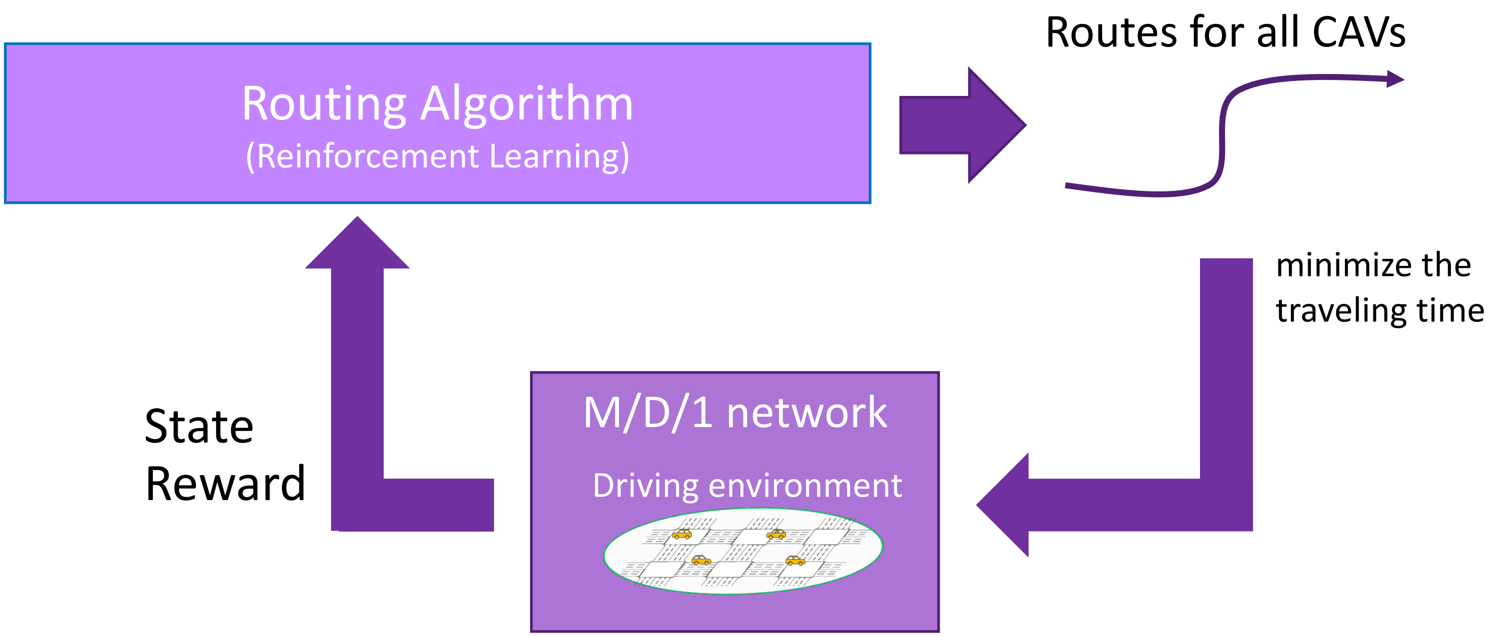
1. **Objective**: minimize the average traveling time of CAVs
2. **Actions**: assigning routes to CAVs
3. **Constraints**:
   a. Physical constraints/sequencing in the driving environment (AIM, CAV moving…)
   b. CAVs has their own source and destination





Training of RL

- Joint work with NYU ECE High Speed Networking Lab

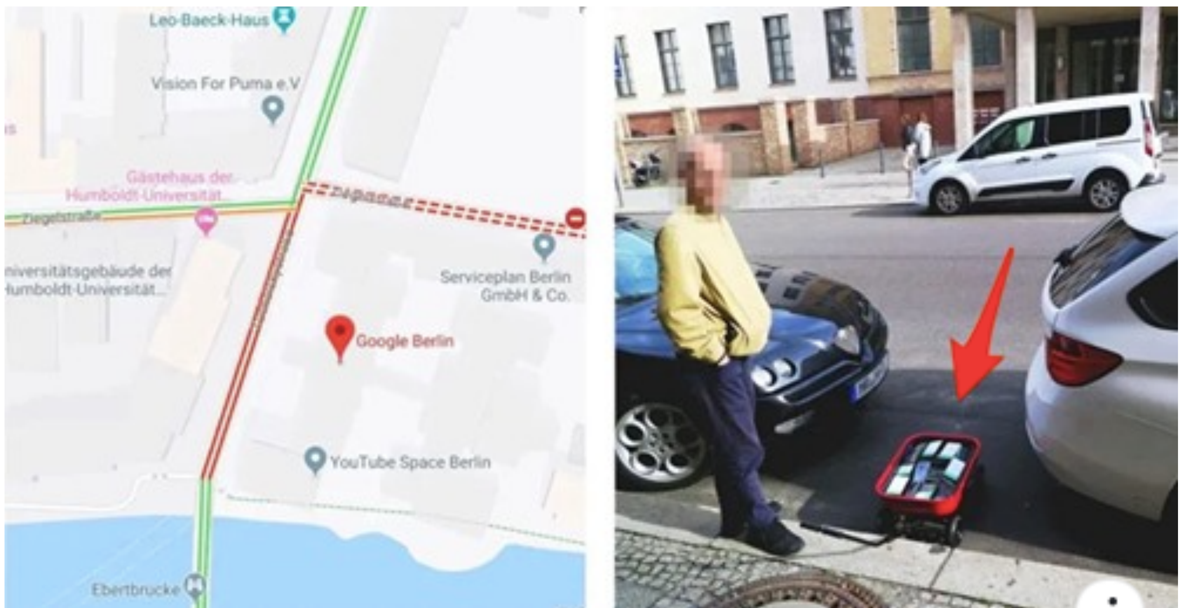


Security risks in cyber-physical systems

- Cyber-physical systems rely on data flowing through the network
- Cyber components are vulnerable to malicious attacks that bring security risks
- How does data quality/integrity impact performance?
- How cyber security vulnerabilities impact physical system?

Malicious behaviors in IoV

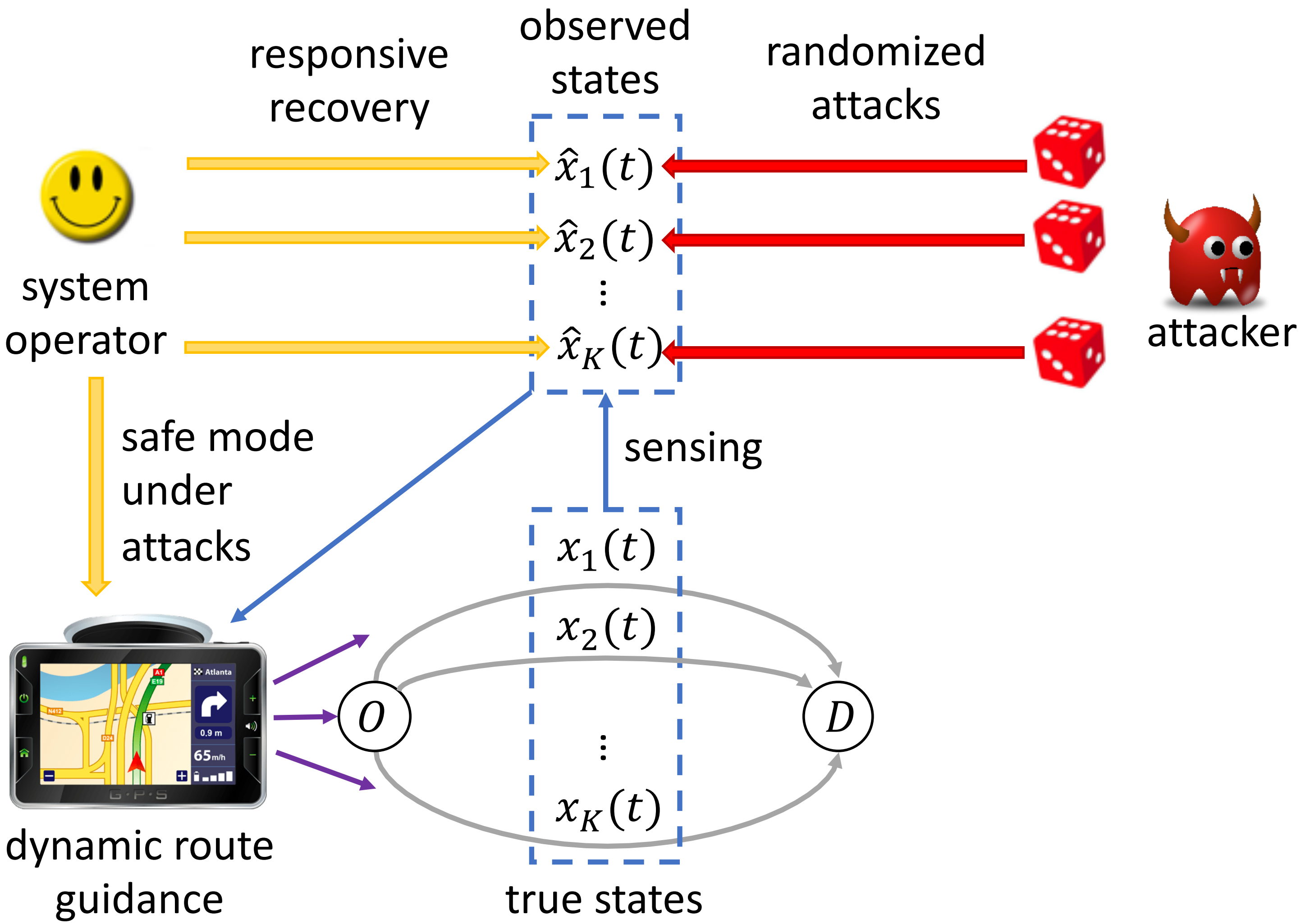
- In the Internet of Vehicles (IoV), vehicles typically make decisions based on real-time routing guidance services
- The info provided by such services can be faulty, and the misled travelers may suffer extra travel times



BUSINESSINSIDER.COM

An artist wheeled 99 smartphones around in a wagon to create fake traffic jams on Google Maps

Security vulnerabilities in ITSs



Research questions

Modeling & analysis

- How to model stochastic & recurrent attacks?
- How to quantify attacker's incentive?
- How to quantify the impact due to attacks?
- How to evaluate security risk?

Resource allocation

- How to allocate security resources, including redundant components, diagnosis mechanisms, etc.?

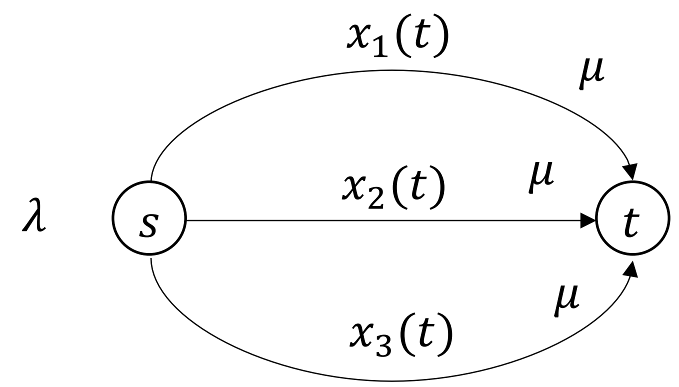
Control design

- How to design traffic control strategies that are less sensitive to various types of attacks?

Queuing model

Basic model

- Poisson arrivals of rate λ
- Parallel queuing servers with service rate μ
- State: vector of queues



$$X(t) = [X_1(t), X_2(t), \dots, X_K(t)]$$

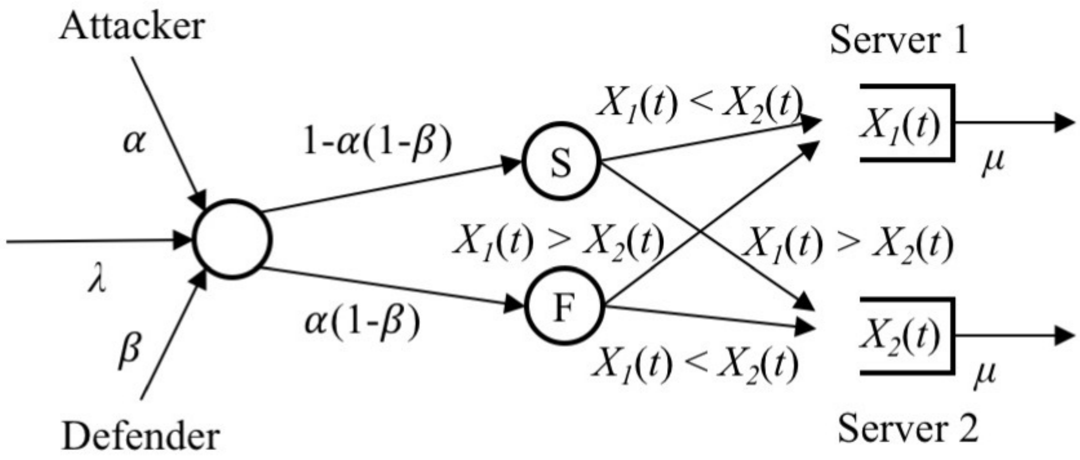
- Dynamic routing: optimal control strategy to route jobs (e.g. vehicles, components, data packets)
- Provably optimal routing policy: send-to-shortest-queue [Ephremides, Varaiya & Walrand 80]
- Note: implementing the optimal routing policy requires perfect observation of system state $X(t)$
- If observation imperfect, then closed-loop can be worse than open-loop (e.g. round robin or Bernoulli routing)

Failure (attacker) model

- Denial-of-service (DoS):
  - Attacker compromise sensing
  - Operator loses observation temporarily
  - With constant probability a, a job does not go to the shortest queue (e.g. join-a-random-queue)
- Spoofing:
  - Attacker modifies sensing
  - Operator makes decision according to manipulated sensing
  - With state-dependent probability $\alpha(x)$, an attacker manipulates the routing (e.g. send-to-longest-queue)
- Objective: balance queuing cost and attacking cost

Defender model

- Decision making:
  - With probability $\beta(x)$, the system operator (defender) secures the routing (i.e. ensuring correct routing)
- Objective: balance queuing cost and defending cost
- Routing is compromised if and only if attacked & not defended
  - i.e. $\alpha(x) = 1$ & $\beta(x) = 0$ or $\alpha(x)\big(1 - \beta(x)\big) = 1$



Defending strategy (constant DoS probability)

Theorem 3. Consider a two-queue system with a constant DoS probability. The optimal defending strategy $\beta^*(x)$ has the following properties:

- Defender either defends or does not defend (no probabilistic defense), i.e. $\beta^*(x) \in \{0,1\}$
- No need to defend ($\beta^* = 0$) when $x_1 = x_2$
- Fixing $x_1 + x_2$, defend for larger $|x_1 - x_2|$
$$|x_1 - x_2| \uparrow \Rightarrow \beta^*(x) \uparrow$$
- Fixing $|x_1 - x_2|$, defend for smaller $x_1 + x_2$
$$x_1 + x_2 \uparrow \Rightarrow \beta^*(x) \uparrow$$

Proof idea: analyze properties of cumulative discounted cost using Hamiltonian Jacobian equation and induction on value iteration.

Security game

Infinite-horizon, dynamic, two-player zero-sum stochastic game

Markovian, state-dependent policies

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

$$\alpha^*(x) = \text{argmax}_\alpha \, V_A^*(x, \beta^*),$$
$$\beta^*(x) = \text{argmin}_\beta \, V_B^*(x, \alpha^*).$$

The value of the attacker/defender is $V_A^*(x, \beta^*)$ / $V_B^*(x, \alpha^*)$. In particular, (α^*, β^*) is a Markovian perfect equilibrium.

Remark. According to Shapley's extension on minimax theorem,

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

Question. Existence of MPE? (Countable infinite state space!)

Joint work with Zhengyuan Zhou (NYU Stern) and Li Jin