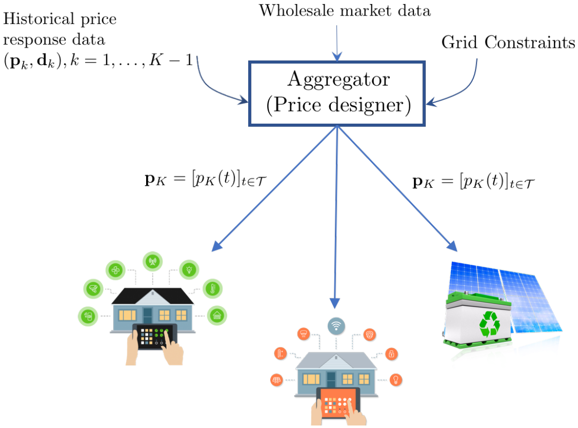
UC-Lab Center for Distribution System Cybersecurity

Mahnoosh Alizadeh and Ramtin Pedarsani, UCSB

March 2019

Centralized real-time pricing algorithms

- The main part of my talk was focused on learning the price response of customers through bandit models:



Real-time pricing based on multi-armed bandits

Goal: keeping the grid safe while learning price response

Aggregator's problem

$$\min_{\mathbf{p}_t} \sum_{t=1}^{T} g(\boldsymbol{\ell}^\star(\mathbf{p_t}), \mathbf{d_t})$$

s.t. dist-flow constraints

where

$$\boldsymbol{\ell}^\star(\mathbf{p}_t) = \sum_{i=1}^{Q} a_i(\boldsymbol{\theta}, \mathbf{p}_t)\boldsymbol{\ell}_i^\star(\mathbf{p}_t)$$

Real-time pricing based on multi-armed bandits

Goal: keeping the grid safe while learning price response

Aggregator's problem

$$\min_{\mathbf{p}_t} \sum_{t=1}^{T} g(\boldsymbol{\ell}^\star(\mathbf{p_t}), \mathbf{d_t})$$

s.t. dist-flow constraints

where

$$\boldsymbol{\ell}^\star(\mathbf{p}_t) = \sum_{i=1}^{Q} a_i(\boldsymbol{\theta}, \mathbf{p}_t)\boldsymbol{\ell}_i^\star(\mathbf{p}_t)$$

We have considered both Thompson Sampling and UCB based solutions and we are making progress on providing **performance guarantees**

Other questions we will ask this year

TS provides statistical models of load response to different prices $\rightarrow$ Natural applications in outlier detection

Other questions we will ask this year

TS provides statistical models of load response to different prices $\rightarrow$ Natural applications in outlier detection

- Can we flag potentially compromised agents and exclude their response from our learning algorithm?
- Can we send further price signals to verify that the user is compromised? $\rightarrow$ Security-aware learning

Other questions we will ask this year

TS provides statistical models of load response to different prices $\rightarrow$ Natural applications in outlier detection

- Can we flag potentially compromised agents and exclude their response from our learning algorithm?
- Can we send further price signals to verify that the user is compromised? $\rightarrow$ Security-aware learning

Unlike distributed algorithms, the only feedback from RTP users is physical (load) and the cost of attacks will be on physical reliability

Other questions we will ask this year

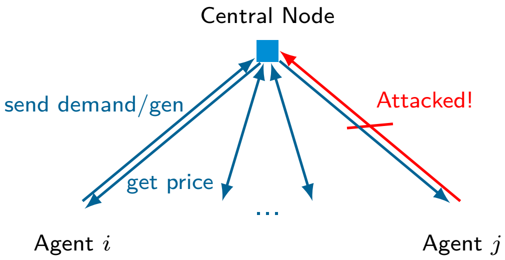
TS provides statistical models of load response to different prices $\rightarrow$ Natural applications in outlier detection

- Can we flag potentially compromised agents and exclude their response from our learning algorithm?
- Can we send further price signals to verify that the user is compromised? $\rightarrow$ Security-aware learning

Unlike distributed algorithms, the only feedback from RTP users is physical (load) and the cost of attacks will be on physical reliability

- We will study the physical costs of adverserial response from a limited set of loads (depends on location in network, appliance type)
- Certain appliance clusters may be affected all at the same time (say someone hacks Chargepoint)

Attack-resilient distributed demand response algorithms

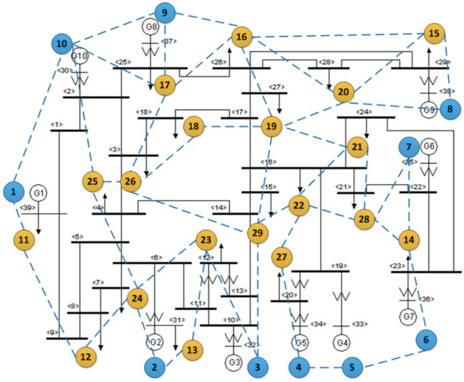


We will continue our analysis of robustified distributed resource allocation algorithms and its specific applications in demand response:

- Finalize our results and then improve our guarantees
- What do our guarantees look like in distribution systems?

Robust decentralized demand response algorithm

- Numerical study in power networks.
- Other attack models: node and link failures



Center-free Algorithm: Node failures

- We consider a Byzantine attack model.
- We model the measurement of a node under attack as

$$\mathbf{z}_i(t) = \mathbf{x}^* + \mathbf{a}_i(t),$$

where $\mathbf{a}_i(t)$ is the disturbance induced by the attacker.

- Bounded perturbation: $\|a_i(t)\|_p \leq \delta$.
- Analyzing the tradeoff of optimization performance and ratio of attacked nodes.

Center-free Algorithm: Link failures

- We consider a time-varying attack model on links.
- The communication link is not jammed anymore, but completely broken.
- Once a link is broken, it will be established again after finite amount of time (few iterations).
- This attack results in random and time-varying connectivity graph.
- Goal: Design a robust algorithm and analyze the performance.