Stability and Learning in Strategic Queueing Systems

Éva Tardos (Cornell)

Joint work with Jason Gaitonde (Cornell)

EC'20+EC'21

Math and Comp Sci of Market and Mechanism Design
Math Sciences Research Institute, Berkeley, Fall 2023

Organizers: Michal Feldman (Tel-Aviv University), Nicole Immorlica (Microsoft Research), LEAD Scott Kominers (Harvard Business School), Shengwu Li (Harvard University), Paul Milgrom (Stanford University), Alvin Roth (Stanford University), Tim Roughgarden (Stanford University), Eva Tardos (Cornell University)

Application deadline: December 1

In recent years, economists and computer scientists have collaborated with mathematicians, operations research experts, and practitioners to improve the design and operations of real-world marketplaces. Such work relies on robust feedback between theory and practice, inspiring new mathematics closely linked – and directly applicable – to market and mechanism design questions. This cross-disciplinary program seeks to expand the domains in which existing market design solutions can be applied; address foundational questions regarding our ways of developing and evaluating mechanisms; and build useful analytic frameworks for applying theory to practical marketplace design.

Motivation: Games and Price of Anarchy

Consider agents interacting in some game

- Main questions: What kinds of outcomes arise? How "good" are they?
- Price of Anarchy: worst-case gap between social welfare at Nash compared to social optimum

In many/most interactions are repeated interactions and participants use learning to figure out what works

- What do we mean by learning?
- What form of learning guarantees good outcome?
- What can we say about outcome of learning?

Example 1: traffic routing

- Traffic subject to congestion delays
- cars and packets follow shortest path
- Congestion game =cost (delay) depends only on congestion on edges

High Social Welfare: Price of Anarchy in Routing

Theorem (Roughgarden-T'02):

In any network with continuous, non-decreasing cost functions and very small users

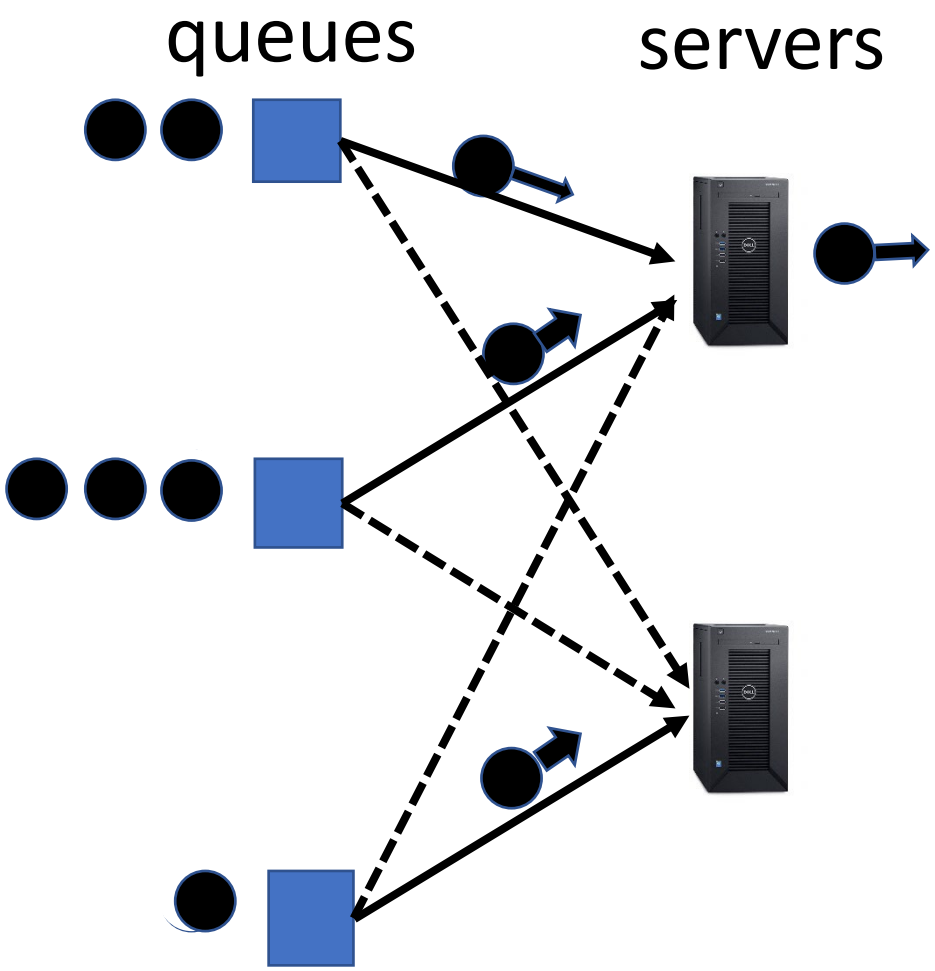
cost of Nash with rates r_i for all i	$\leq$	cost of opt with rates $2r_i$ for all i

Nash equilibrium: stable solution where no player had incentive to deviate.

Extra resource needed to guarantee good outcome at Nash

$$\text{Price of Anarchy} = \frac{\text{cost of worst Nash equilibrium}}{\text{"socially optimum" cost}}$$

Example 2: serving packets



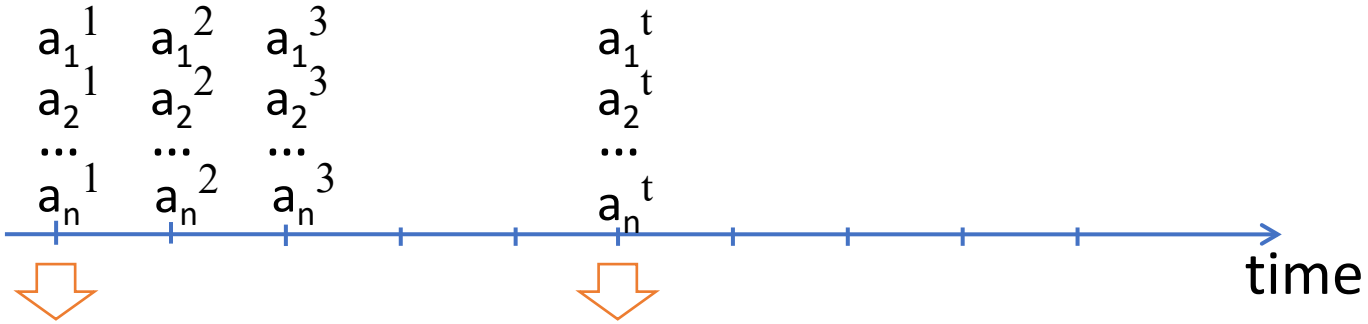
- Stream of packets that need serving
- servers have limited capacity

- Drop (or return) unsent packets, that need to get resend

Example 3: ad-auctions

- Repeatedly bidding for impression
- Limited by a budget for longer period

Learning outcome: No-regret without stability (Hannan consistency)



Simple behavioral assumption: no-regret learning:

do at least as well as any single action with hindsight

for each player i and each fixed action x:

$$\sum_t u_i\left(x, a_{-i}^t\right) \leq \sum_t u_i(a^t) + o(T)$$

Quality of Learning Outcomes: Price Anarchy No-regret/Hannan consistency as a behavioral assumption

Price of Anarchy [Koutsoupias-Papadimitriou'99]

$$PoA = \min_{a\ Nash} \frac{\sum_i u_i(a)}{Opt}$$

Assuming no-regret learners in stable game: [Blum, Hajiaghayi, Ligett, Roth'08, Roughgarden'09]

$$PoA = \lim_{T\to\infty} \frac{\sum_{t=1}^{T} \sum_i u_i(a^t)}{T\ Opt}$$

Extends to even to repeated games with **dynamically changing population** assuming no-regret learners : [Lykouris, Syrgkanis, T'16]

→ No regret is a natural and strong enough assumption on what learners achieve

Social Welfare of Learning Outcomes

Critical Assumption: new copy of the same game is repeated (no carryover effect between rounds other than through learning)

Is this reasonable?

Large population games: traffic routing

Morning rush-hour traffic

✔ No carryover effect (except through the learning of the agents)

Second-by-second packet traffic

✖ Packets take time to clear, dropped packets need to be resent in the next round

Price of Anarchy in Stateful Systems

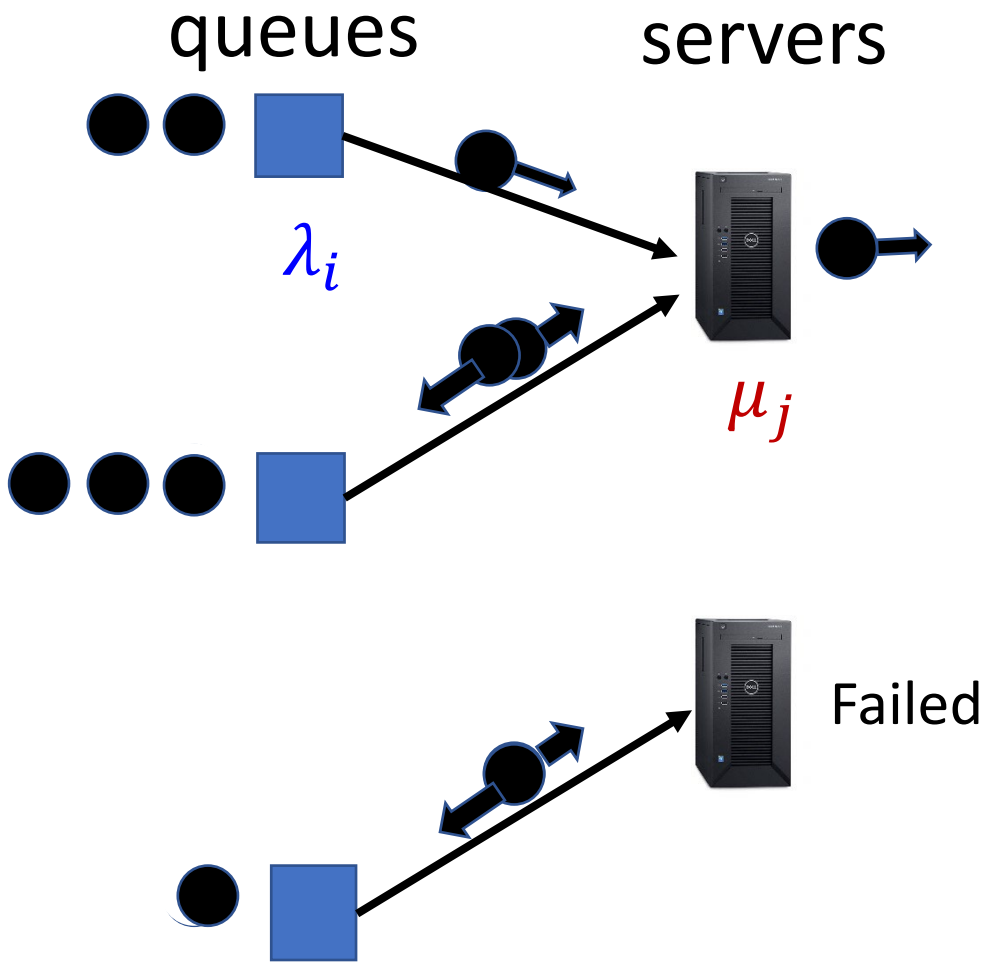
- Not as well understood: do PoA-style bounds still hold with dependence between games in each round?

Questions for queuing application:

- How much extra capacity ensures good system performance despite selfish users
- Is no-regret learning a good enough way to learn in presence of dependence between rounds

Simple Model of Queuing

- Queue i gets new packets with a Bernoulli process with rate λ_i
- Server j succeeds at serving a packet with probability μ_j
- Each time step: each queue can send **one** packet to one of the servers to try to get serviced
- **Server can process at most one packet and unserved packets get returned to queue**
- Servers attempt to serve oldest packet

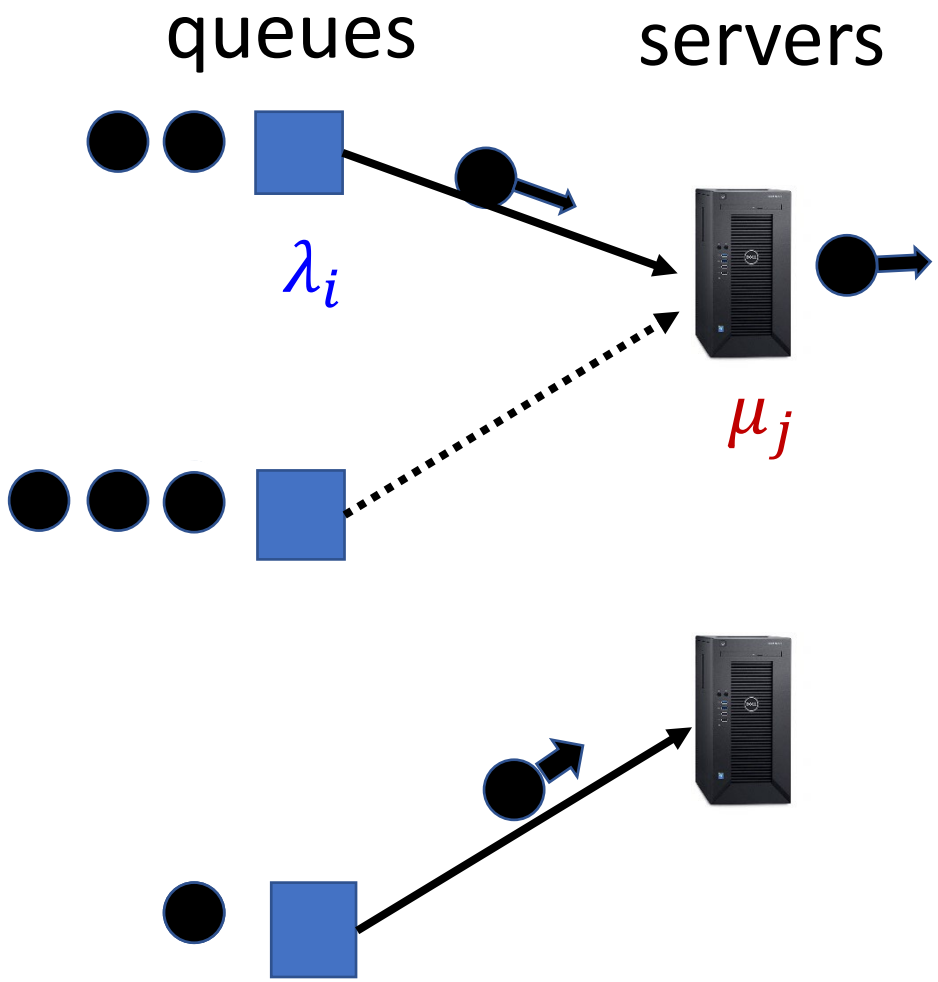


Baseline Measure: Coordinated Queues

Assume queues and servers are sorted:

$$1 > \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$$

$$1 \geq \mu_1 \geq \mu_2 \geq \cdots \geq \mu_m > 0$$



Theorem 0: necessary/sufficient condition for centralized stability: for all k,

$$\sum_{i=1}^{k} \lambda_i < \sum_{i=1}^{k} \mu_i$$

(Recall can only send one packet each round)

Selfish Queuing: Price of Anarchy

Theorem 1 [Gaitonde-T]: if queues use no-regret algorithms to select servers and for all k,

$$\sum_{i=1}^{k} \lambda_i < 0.5 \sum_{i=1}^{k} \mu_i$$

Then queue lengths/ages remain bounded in expectation.

Theorem 2 [Gaitonde-T]: If queues choose servers patiently, and for all k

$$\sum_{i=1}^{k} \lambda_i < 0.63 \sum_{i=1}^{k} \mu_i$$

then in every equilibrium, queue lengths/ages grow sublinearly. $0.63 = (e-1)/e$

Plan for rest of the time

- Outline of the proof of Theorem 1:
  why is no-regret a useful condition
- Limitation of no-regret:
  example showing no-regret can be too myopic
- Theorem 2 comments:
  worst case example
- Directions of further research

Theorem 1 Proof Idea (using no-regret)

Technical tool: [Pemantle, Rosenthal '04]: random process satisfying

i. Sufficiently regular
ii. Negative drift when large

remains bounded in expectation for all times

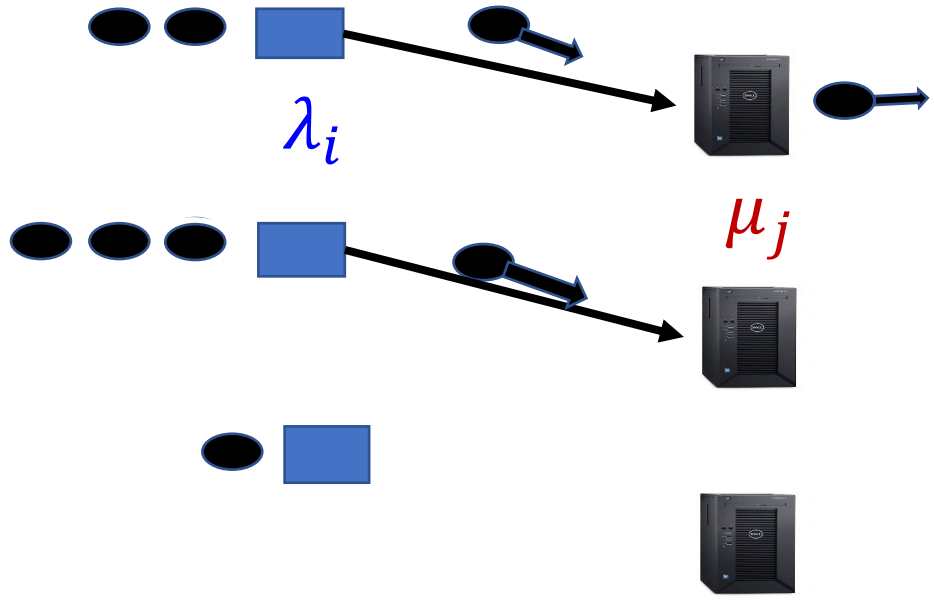
Use potential function $\sqrt{\Phi}$ where $\Phi \approx \sum_\tau \Phi_\tau$
with $\Phi_\tau =$ # packets aged τ or older in the system

Φ remains bounded $\rightarrow$ all queues remain bounded (in expectation)

- No-regret + factor 2 slack $\rightarrow$ negative drift when queues have large backup

Why Φ and How No-Regret Helps

- Look at queues with packets at least τ-old; they have priority
- Fix long window and look at best servers
- Either: i) many τ-old queues send there throughout window ➔ Φ_τ decreases by a lot



Why Φ and How No-Regret Helps

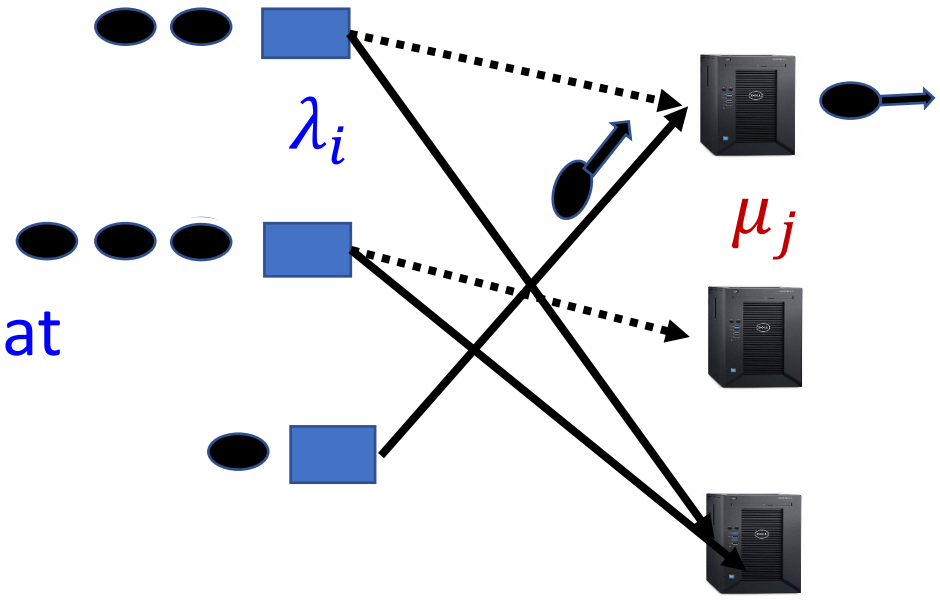
- Look at queues with packets at least τ-old; they have priority
- Fix long window and look at best/fastest servers
- Either: i) many τ-old queues send there throughout window ➔ decrease in queue size, OR

ii) they do not:

had priority there so no-regret kicks in:

Any queue with τ-old packets would have regret, unless it managed to get service for at least this much!

Apply at all thresholds τ simultaneously to get no-regret at all scales ➔ implies negative drift

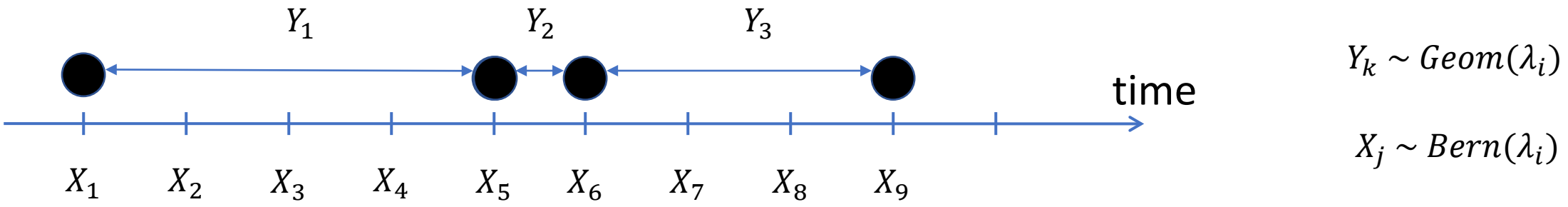


Extra Technical Details

- No-regret needs to hold with high-probability

  unlikely bad situations **will** happen, need to be able to recover

- Pemantle/Rosenthal needs "sufficiently regular" = bounded moments:
- use model with deferred decisions, study *ages* instead of *sizes:* age of oldest packet T_i^t in queue i



Extra Technical Details (3)

Potential: $\sqrt{\Phi}$ where $\Phi = \sum_\tau \Phi_\tau$
with $\Phi_\tau =$ # packets aged τ or older in the system

With deferred decision: $\boldsymbol{\Phi_\tau = \sum_i \lambda_i (T_i^t - \tau)}$
= expected # of packets age τ or older, given oldest in each queue

Contribution of one que to $\Phi = \sum_{\tau < T_i^t} \lambda_i \left(T_i^t - \tau\right) \approx \lambda_i (T_i^t)^2/2$

Clearing one packets decreases T_i^t by $Y \sim Geom(\lambda_i)$,
expected decrease is $1/\lambda_i$

Selfish Queuing: Price of Anarchy

Theorem 1 [Gaitonde-T '20]: if queues use no-regret algorithms to select servers and for all k,

$$\sum_{i=1}^{k} \lambda_i < 0.5 \sum_{i=1}^{k} \mu_i$$

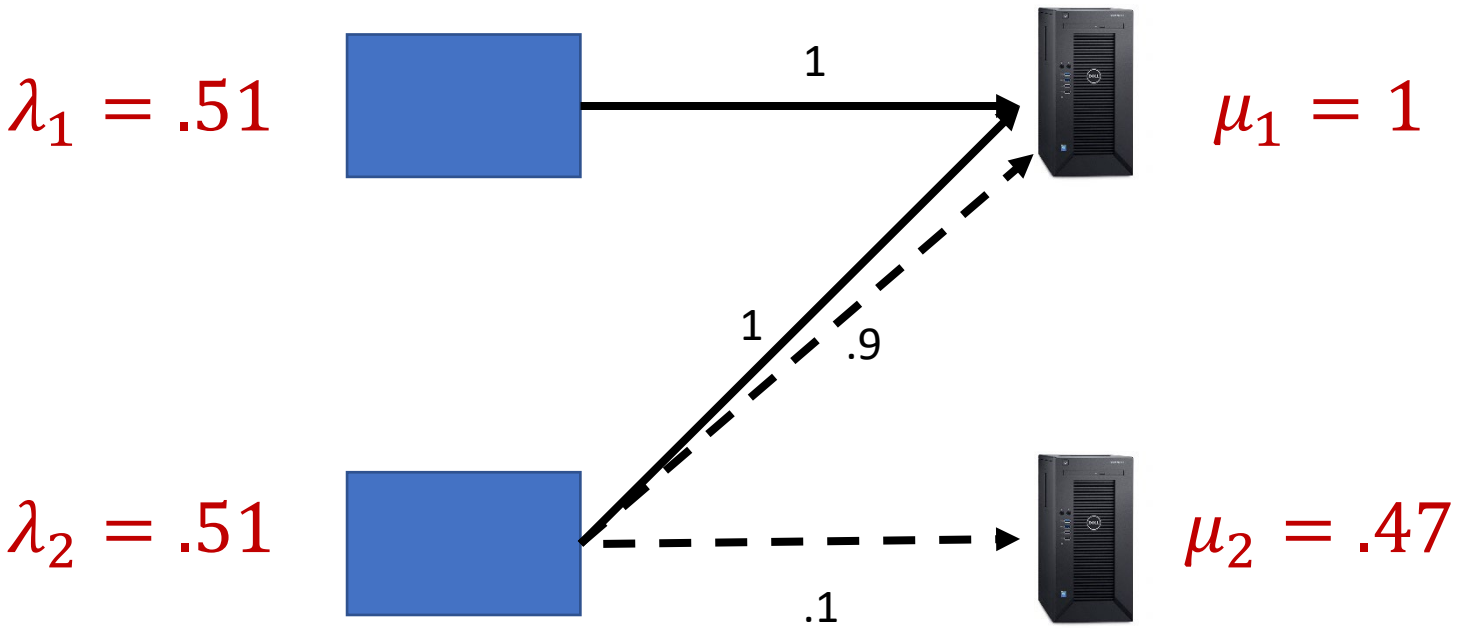
Then queue lengths/ages grow sublinearly, **and this bound is tight**.

Theorem 2 [Gaitonde-T'21]: If queues choose servers patiently, and for all k

$$\sum_{i=1}^{k} \lambda_i < 0.63 \sum_{i=1}^{k} \mu_i$$

then in every equilibrium, queue lengths/ages grow sublinearly. $0.63 = (e-1)/e$

No-regret is myopic: Example



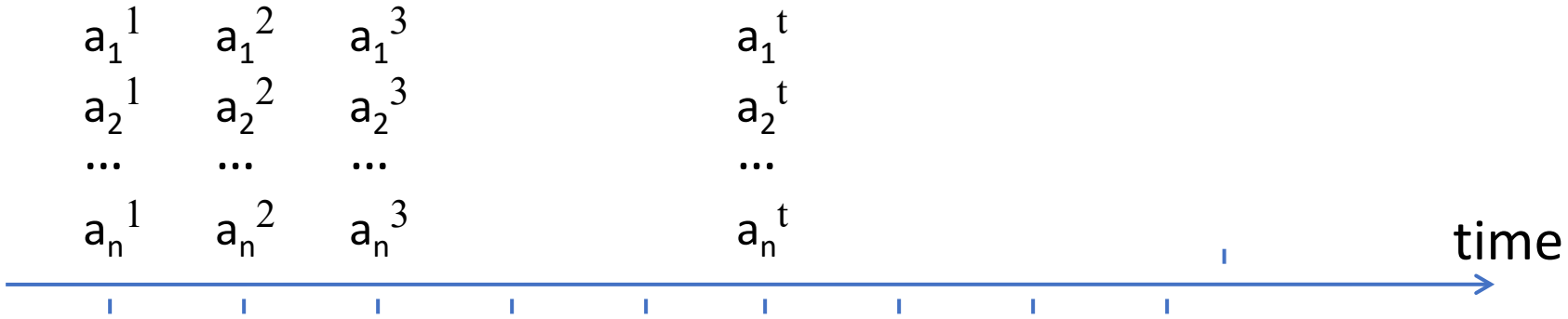
- Both sending to top server gets .5 rate and so has no-regret
- Age/split top server equally → linear growth
- Deviating gives regret

- **Moving to inferior server selfishly helps:**

  Why? Helps top queue clear, indirectly helping both queues clear!

But the 0.47 rate causes regret!

No-regret learning (Hannan consistency)



No regret: for any fixed action x:

regret

$$\sum_t u_i(a^t) \geq \sum_t u_i(x, a_{-i}^t) \; -error$$

Algorithmic idea for no-regret: explore-exploit

If outcome also depends on history:

$$\sum_t u_i(a^{1:t}) \geq \sum_t u_i((a_i^{1:t-1}, x), a_{-i}^{1:t}) - \mathrm{o(T)}$$

What's Going On?

- What we do: evaluate alternate outcome without considering long-term effect of the change

$$\sum_t u_i(a^{1:t}) \geq \sum_t u_i\left(\left(a_i^{1:t-1}, x\right), a_{-i}^{1:t}\right) - o(T)$$

Too myopic: not patient enough to see long term benefit of "bad" servers:

- What we may want (?):

$$\sum_t u_i(a^{1:t}) \geq \sum_t u_i\left(x^{1:t}, a_{-i}^{1:t}\right) - o(T)$$

- We study the patient queuing game with stationary strategies

Patient Queuing Game

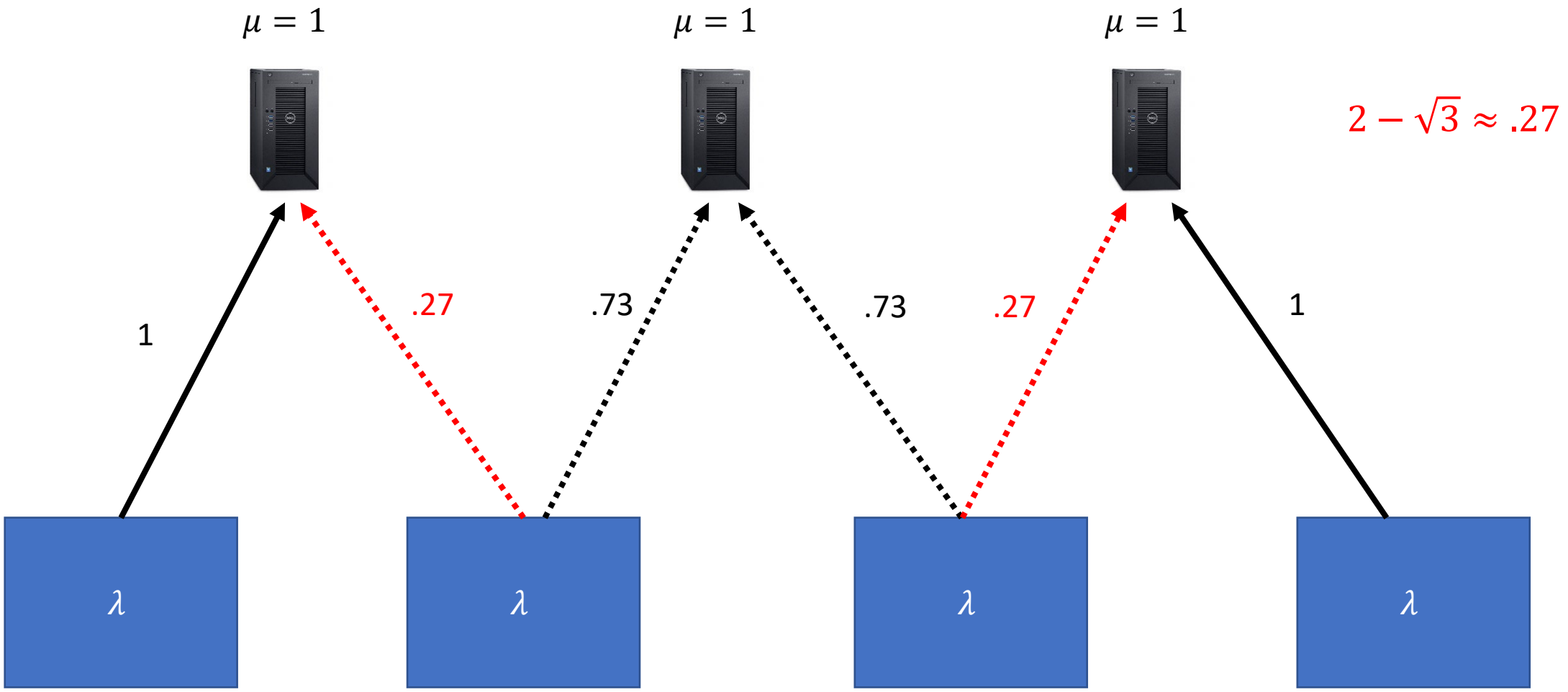
- Each queue i picks fixed randomization over servers, $p_i \in \Delta^{m-1}$ to be played in every round it has a packet
- Induces a Markov chain of queue ages
- Each queue i aims to minimize expected asymptotic aging rate, i.e.

$$\limsup_{T \to \infty} \frac{\mathbb{E}[Age\ of\ Queue\ i\ at\ Time\ T]}{T}$$

  when running Markov chain with randomizations $p_1, \ldots, p_n$

- [Gaitonde-T'21]: Nash equilibria of this game

Example of a Nash equilibrium



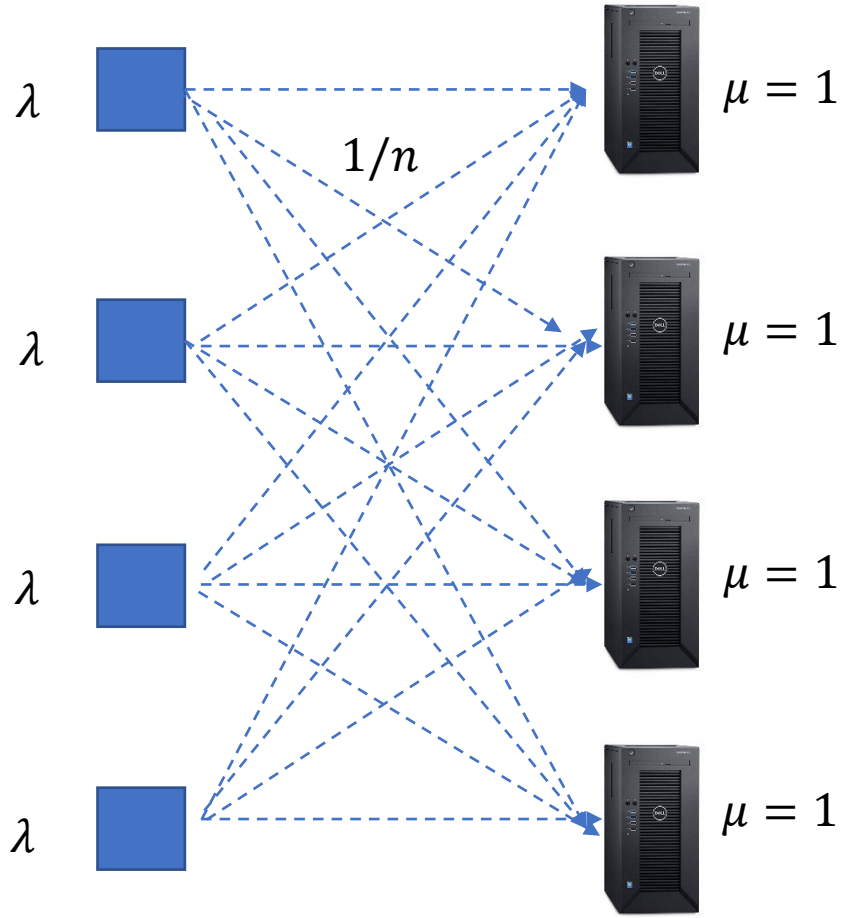
All ques clear if $\lambda \leq \frac{3-\left(2-\sqrt{3}\right)^2}{4} \approx .73$

Patient Queuing Game

- Immediate problems:
  - What are the asymptotic aging rates?
  - Why should there exist an equilibrium?
  - Price of anarchy?

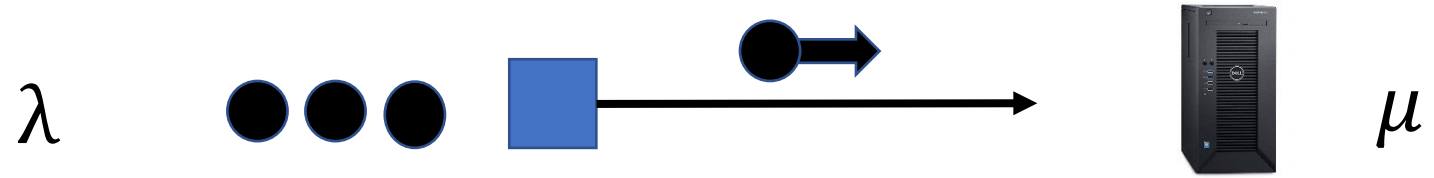
Price of Anarchy

- Worst-case (intuitively): n equal queues, n servers with rate 1, uniform mixing → worst case needs at least $\frac{e}{e-1}$ slack
- In general: fastest-aging queue cannot benefit from deviation at equilibrium, but not clear *why*
- *Queue incentives can come from many tight subsets!*



Long-Run Limits: One Queue

- Example: one queue, one server (no strategies, no competition)



- Ages by one deterministically
- Clears packet with probability μ
- Expected decrease in age $1/\lambda$ if server succeeds

→ Long-run aging rate should be

$$max\left\{0\,,1\,-\,\frac{\mu}{\lambda}\right\}$$

Long-Run Limits: General Case

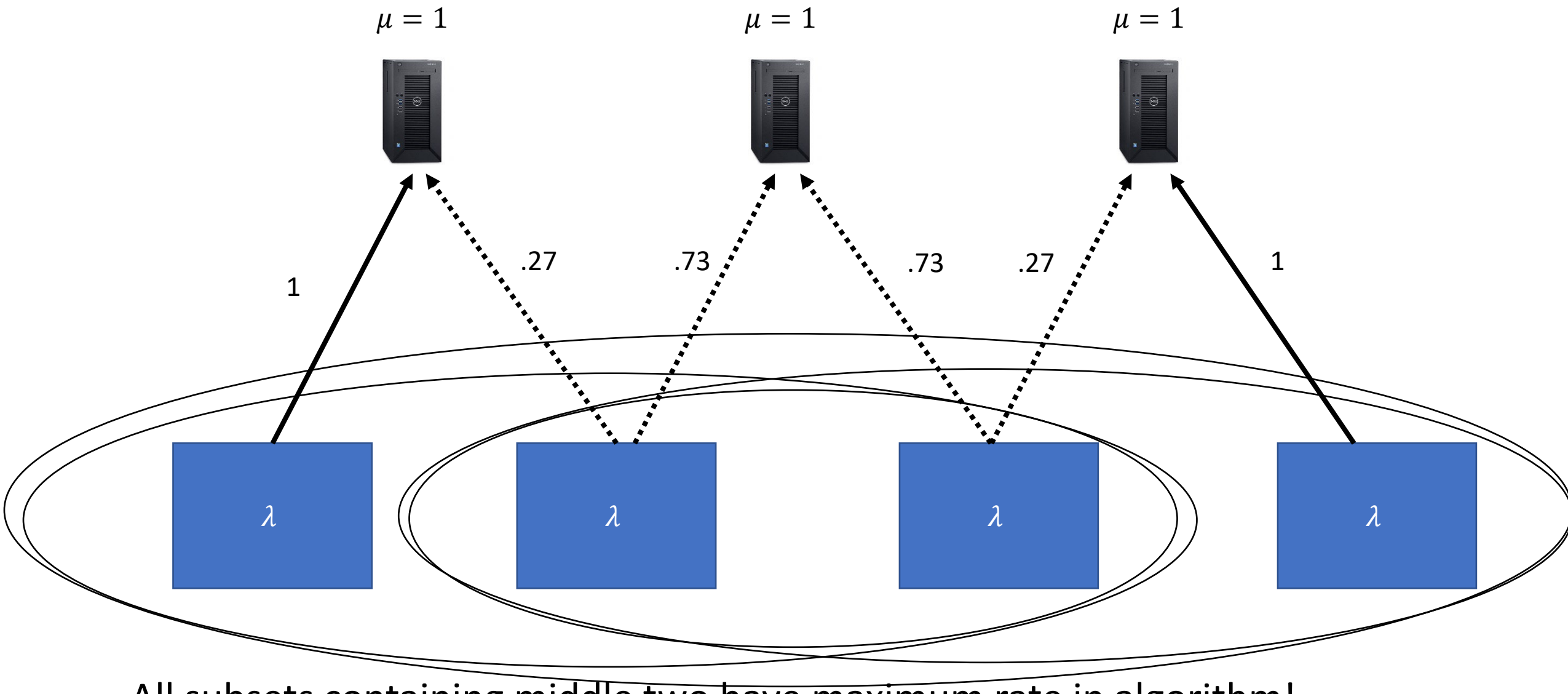
- Theorem [Gaitonde-T]: For fixed $p_1,\ldots, p_n$, queues almost surely cluster into groups with same long-run aging rate
- Fastest aging subset S_1 given by

$$S_1 = argmax_{S\subseteq[n]}\left[1 - \frac{\mu(S)}{\lambda(S)}\right]$$

where $\lambda(S) = \sum_{i\in S}\lambda_i$ = combined arrival rate of subset S,

$\mu(S) =$ expected # packets cleared by S when having priority

- Recurse on rest of queues by discounting servers by probability any queue in S_1 sends to server



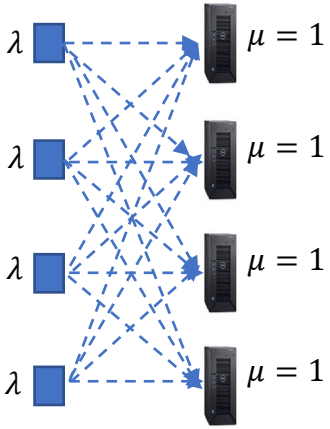
All subsets containing middle two have maximum rate in algorithm!

Structural Properties:

- set of subsets with maximum rate at each step is closed under union and intersection
- Aging rates continuous in $p_1,\ldots, p_n$
- With these costs, ∃ Nash equilibria (Kakutani's theorem)
- Proof long-run rates as given relies heavily on concentration and careful accounting of priorities

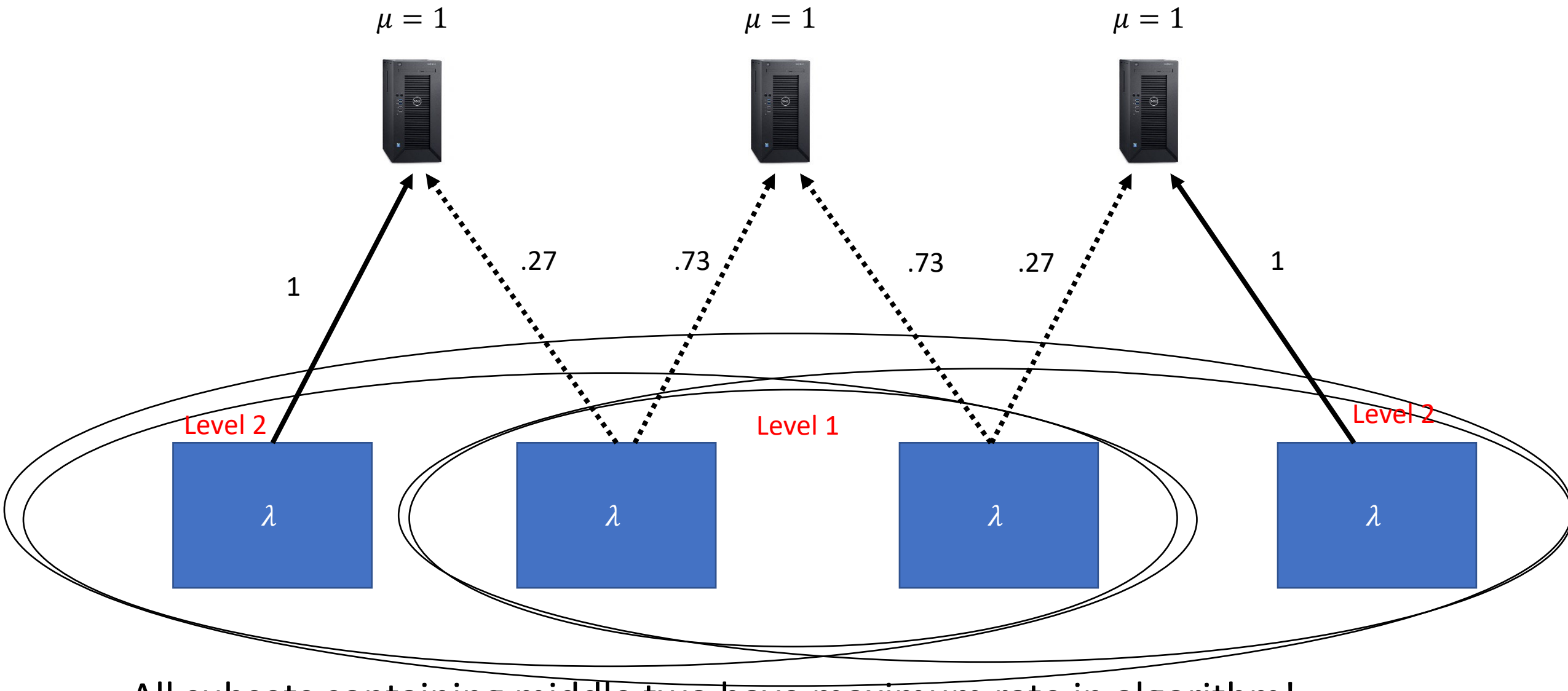
Price of Anarchy: Proof Sketch

- Idea: continuously deform a Nash profile to one with "proportional" loads on each server
  - Given total load on each server, worst case is symmetrized profile
- Want to show deformation process only increases aging rate via Nash property
- Key idea: use structural properties (closure under unions and intersection) of tight subsets



Price of Anarchy: Proof Sketch

- Tight subsets inside S_1 (fastest-aging group) form levels
- Queue incentives essentially determined by level



All subsets containing middle two have maximum rate in algorithm!

Price of Anarchy: Proof Sketch

- Tight subsets inside S_1 (fastest-aging group) form levels
- Queue incentives essentially determined by level
- Shift all highest-level queues first towards symmetrized profile

→ Succeeds directly from Nash definition

- Inductively proceed to next level

→ Can show this schedule preserves relevant incentives from Nash

- Bound aging rate at final symmetrized profile

Conclusions

Learning players achieve high social welfare many games (even some with carryover effect),

- but no-regret learning can be too simplistic: can one obtain better bounds using better learning algorithms?

What may be a good learning method for the players?

My first guess: learning with low policy regret

Sentenac, Boursier, Perchet, NeurIPS'21: not true

- Offers cooperative solution

Maybe more natural learning processes do better (even just no-regret)?