Decentralized Learning in Online Queuing Systems

Flore Sentenac

ENSAE Paris

Etienne Boursier

ENS Paris-Saclay

Vianney Perchet

ENSAE Paris
Criteo AI Lab

NeurIPS 2021

Motivations

Second-by-second packet routing

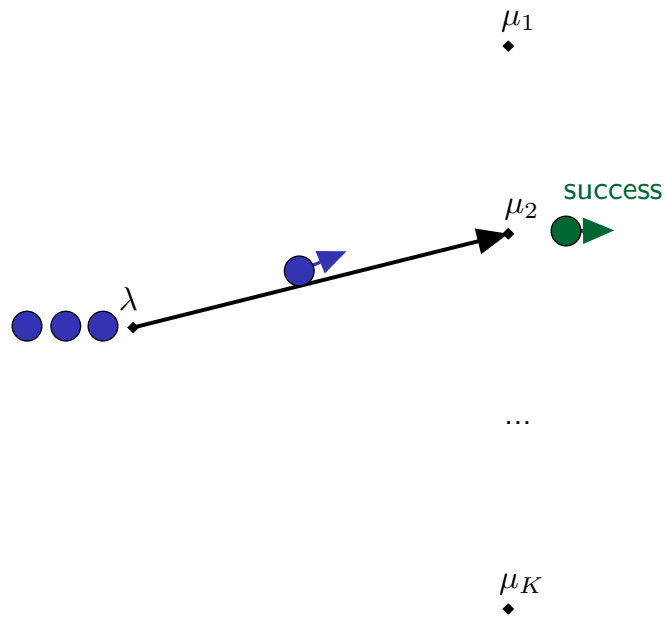
Dropped packets have to be resent in next rounds

$\rightarrow$ Learning in repeated games **with carryover**?

Queuing Systems: Single Queue

At each round $t = 1, \ldots, \infty$:

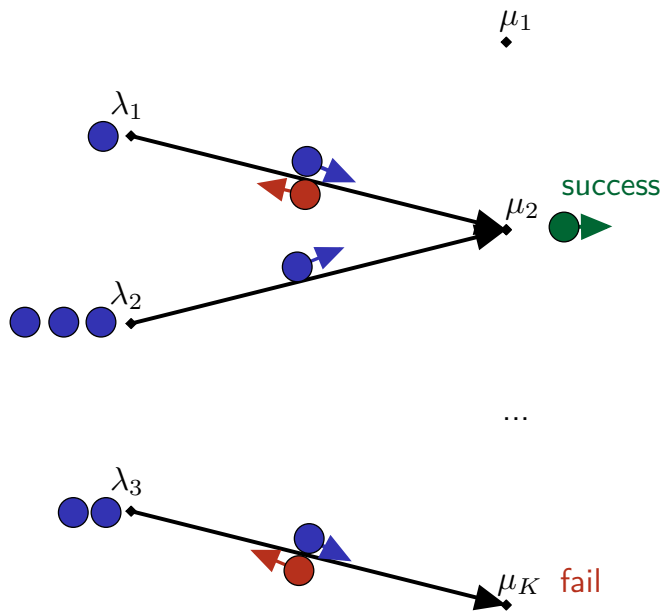
- packet arrives with proba λ
- sends a packet to server $k \in [K]$
- server k clears with proba μ_k
- if fails $\rightarrow$ packet back in queue



Queuing Systems: Multiple Queues

At each round $t = 1, \ldots, \infty$:

- N queues ($N \leq K$)
- Heterogeneous arrival rates λ_i
- each queue chooses $a_t^i \in [K]$
- Server treats one packet at a time, the oldest one



Stability

$Q_t^i :=$ number of packets in queue i at time t.

A queue i is **stable** if for any r, there is a constant $C_r > 0$ such that

$$\mathbb{E}[(Q_t^i)^r] \leq C_r \qquad \forall t \in \mathbb{N}.$$

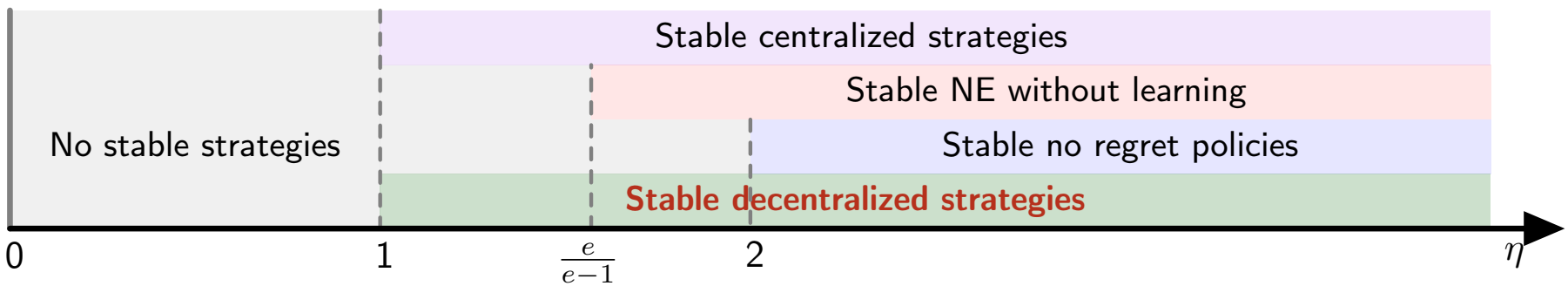
System is stable if all queues are stable.

Overview: Stability as a function of Slack

Note $\lambda_{(1)} \geq \ldots \geq \lambda_{(N)}$ and $\mu_{(1)} \geq \ldots \geq \mu_{(K)}$.

Slack of the System

Largest η s.t. $\eta \sum_{i=1}^{n} \lambda_{(i)} \leq \sum_{i=1}^{n} \mu_{(i)}, \quad \forall n \leq N.$



Previous results: Stable no regret policies

Define regret

$$R_i(T) = \max_{k\in[K]} \sum_{t=1}^{T} \nu_t^i(k) - \sum_{t=1}^{T} \overbrace{\nu_t^i(a_t^i)}^{\text{1 if clears}} .$$

(?)

If $\eta > 2$ and all queues follow a no regret strategy: $R_i(T) = o(T)$ w.h.p., then the system is stable.

Previous results: Stable NE without learning

Define game $\mathcal{G} = ([N], (c_i)_{i=1}^n, \boldsymbol{\mu}, \boldsymbol{\lambda})$ with

- Action Space: $p_i \in \mathcal{P}([K])$, queue i chooses server $a_t^i \sim p_i$ at time t,
- Cost Function:

$$c_i(p_i, \boldsymbol{p_{-i}}) = \lim_{t \to +\infty} \frac{T_t^i}{t}$$

where T_t^i is the age of the oldest packet in queue i at time t.

(?)

If $\eta > \frac{e}{e-1}$, all Nash equilibria of $\mathcal{G}$ are stable.

No Policy-regret

Define the policy regret of queue i:

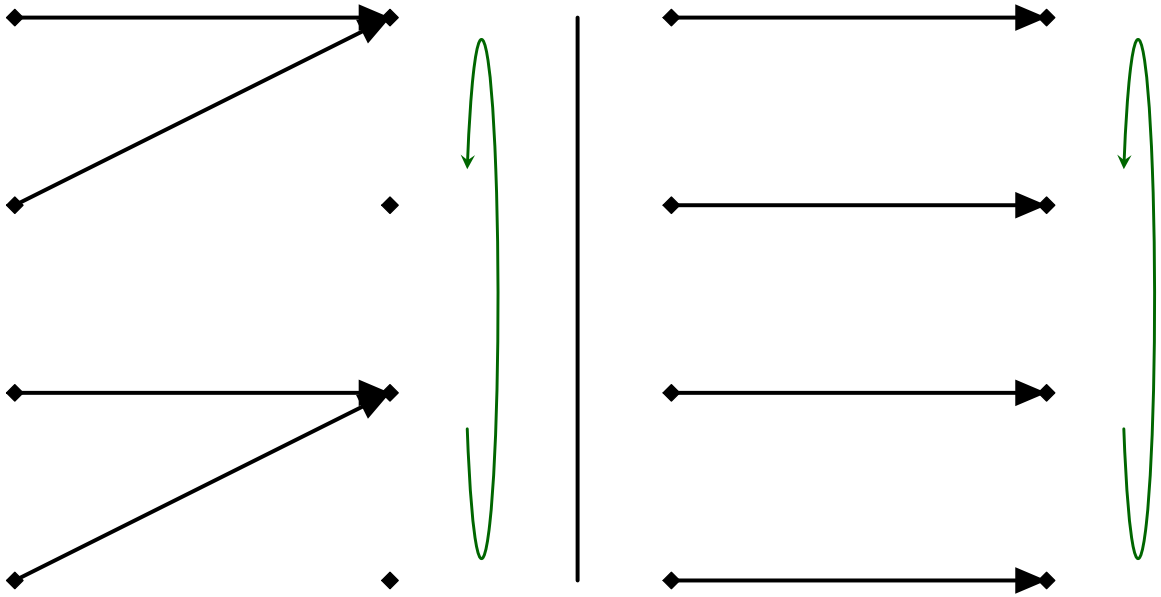
$$\max_{p\in\mathcal{P}([K])} \sum_{t=1}^{T} \overbrace{\mathbb{E}_{\tilde{a}^i_{\mathbf{1}:t}\sim\otimes_{i=\mathbf{1}}^{t} p}[\nu_t(\tilde{a}^i_{1:t})]}^{\text{reward } i \text{ would get if playing } p \text{ for all rounds}} - \sum_{t=1}^{T} \mathbb{E}[\nu_t(a^i_{1:t})]$$

and recall the definition of regret :

$$R_i(T) = \max_{k\in[K]} \sum_{t=1}^{T} \nu_t^i(k) - \sum_{t=1}^{T} \overbrace{\nu_t^i(a_t^i)}^{\text{1 if clears}} .$$

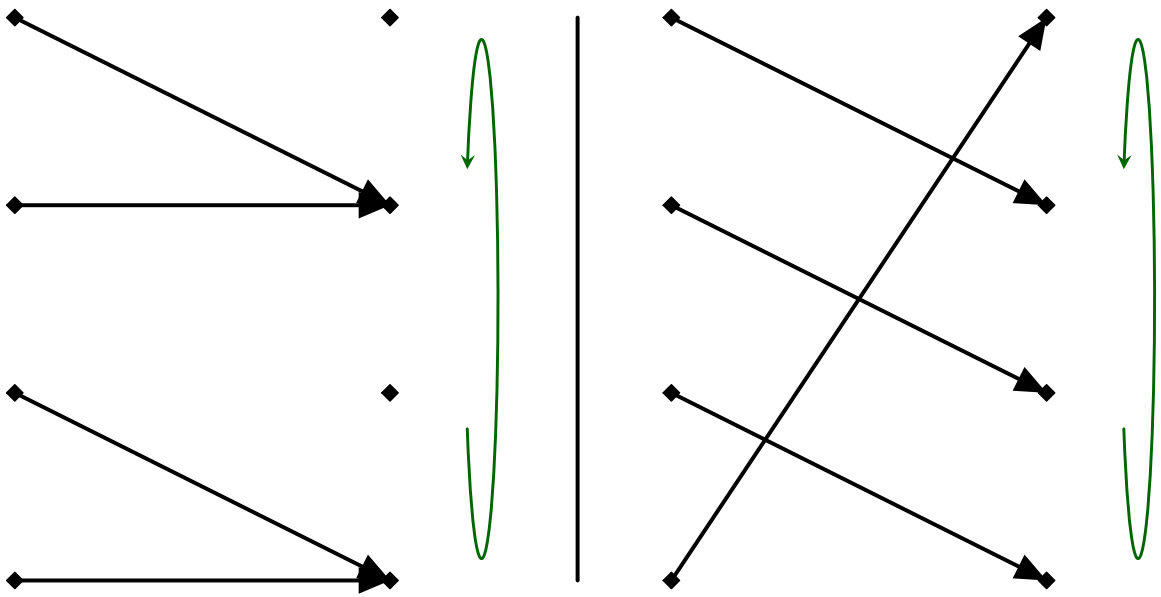
There is an instance with $\eta = 2 - O(\frac{1}{N})$, s.t. each queue's policy regret is $o(T)$, but the system is not strongly stable.

An unstable no-policy regret strategy



$$\forall i \in [N],\ \lambda_i = \frac{1}{N},\ \mu_i = \frac{2}{N} - \frac{1}{N^2}$$

An unstable no-policy regret strategy



$$\forall i \in [N],\ \lambda_i = \frac{1}{N},\ \mu_i = \frac{2}{N} - \frac{1}{N^2}$$

A stable learning strategy

Assumptions:

- all queues know N and have attributed ranks $i \in [N]$
- shared randomness between queues

Theorem

If $\eta > 1$ and all queues follow ADeQuA, *then the system is stable.*

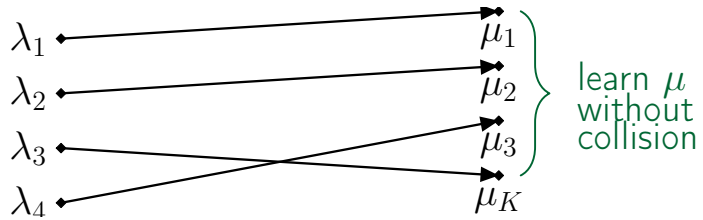
ADeQuA: A DEcentralized QUeuing Algorithm

for $t = 1, \ldots, \infty$ **do**
  Draw $\omega_1 \sim \mathrm{Bernoulli}(\varepsilon_t)$ `// shared randomness`
  if $\omega_1 = 1$ **then** **Explore**
  else **Exploit**
end

Exploration

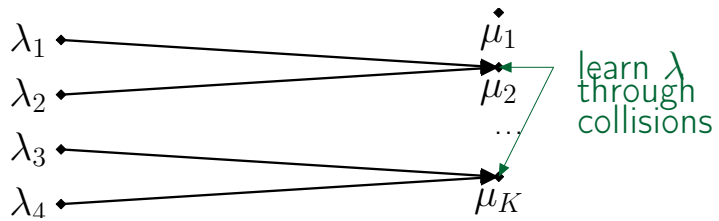
All queues explore simultaneously and explore either μ or λ

Explore μ: queues choose servers without colliding
$\rightarrow$ accurate estimations of μ_k



Explore λ: when queue i explores queue j, both choose same server k **with packet generated at t (if it exists)**

Assumption: servers break ties in packets' age uniformly at random
i clears with probability $(1 - \frac{\lambda_j}{2})\mu_k$
$\rightarrow$ estimate λ_j



Exploitation: centralized

When centralized:

- $\phi : (\hat{\lambda}, \hat{\mu}) \mapsto$ marginals ensuring stability (dominant mapping)
- $\psi : P \mapsto$ coupling without collision (Birkhoff von Neumann decomposition)

Centralized exploitation	
Draw $\omega_2 \sim \mathcal{U}(0,1)$	`// shared randomness`
Play $\psi(\phi(\hat{\lambda}, \hat{\mu}))(\omega_2)$	

When decentralized:

- compute mapping $\hat{A}^i = \psi(\phi(\hat{\lambda}^i, \hat{\mu}^i)) : [0,1] \to \mathbb{R}^N$
- play $\hat{A}^i(\omega_2)(i)$

Exploitation: decentralized

Compute mapping $\hat{A}^i = \psi(\phi(\hat{\lambda}^i, \hat{\mu}^i))$

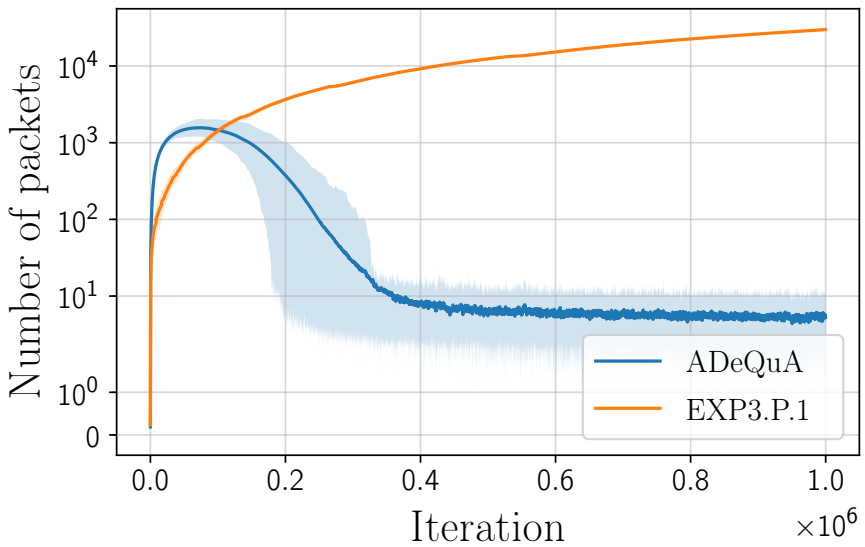
Problem: estimates $(\hat{\lambda}^i, \hat{\mu}^i)$ differ (but are close)
But general dominant mappings and BvN decompositions are non-continuous

$\|\hat{A}^i - \hat{A}^j\|$ arbitrarily large $\implies$ too many collisions

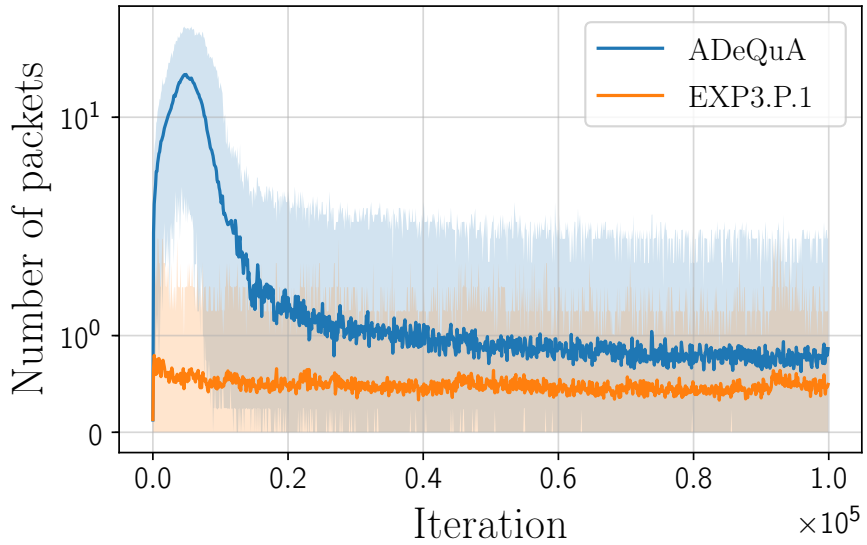
If ϕ and ψ **regular** $\rightarrow \|\hat{A}^i - \hat{A}^j\|$ small
$\implies$ small amount of collisions

Challenge: design **regular** dominant mapping and BvN decomposition

Simulations



Hard instance, $\eta < 2$.



Easy instance, $\eta > 2$.

- No regret strategies: unstable
- ADeQuA: stable & number of packets decreases after learning

- both strategies stable
- No regret better suited to easy instances?

Conclusion

Recap

- cooperation required in queuing systems
- patience not enough for stable learning
- first decentralized stable learning strategy (for $\eta > 1$)

Perspectives

- decrease the maximal number of accumulated packets at each time
- asynchronous case: queues don't share a common time clock
- enhancement with additional observation (eg collision information)

Thank you!

Join us at poster session ☺