Pull-based load distribution in large heterogeneous systems

A. Stolyar (Lehigh University)

June 2, 2015

Outline

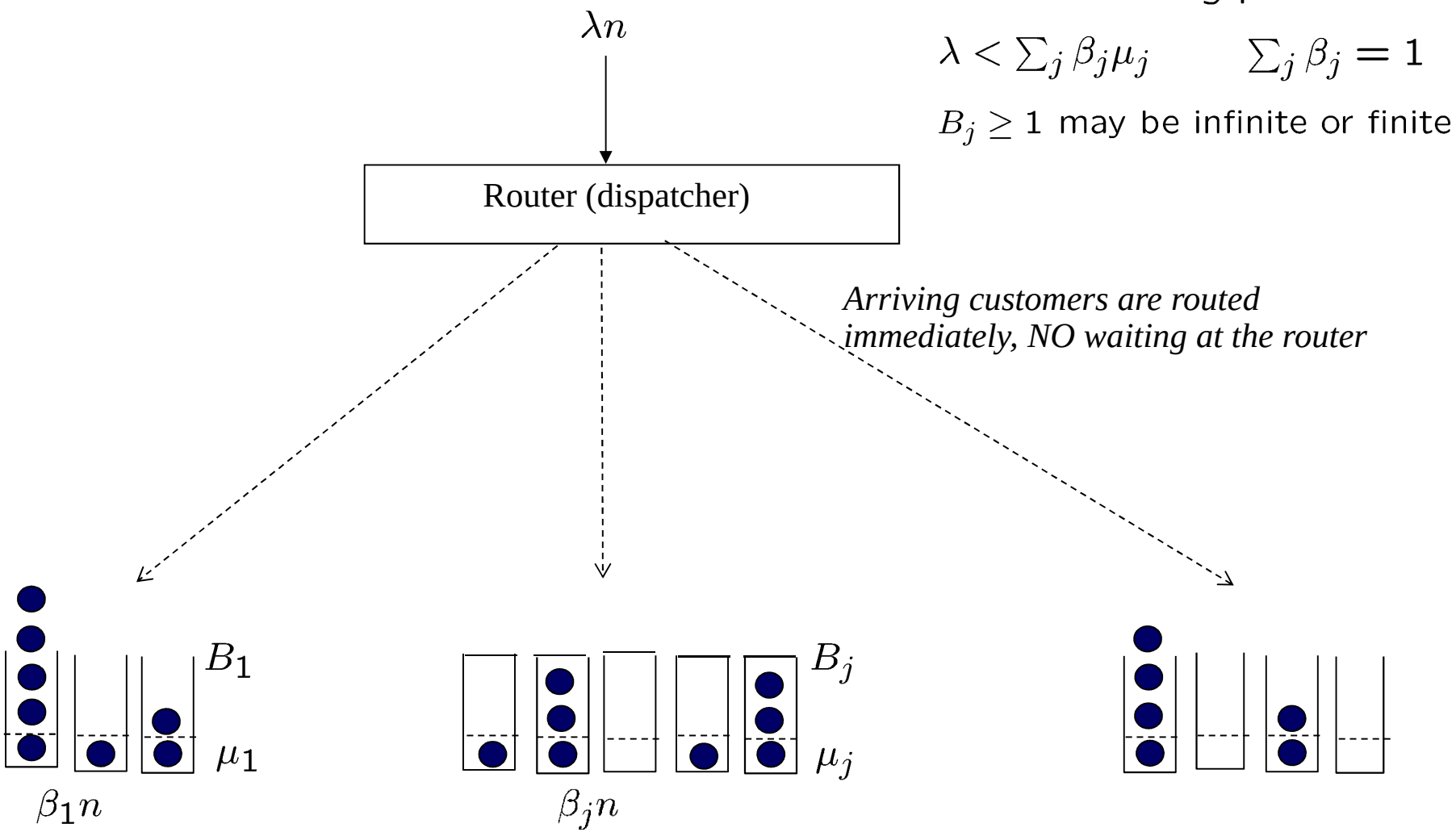
- Model: Large-scale service system with heterogeneous servers
  - Motivation: Job distribution (assignment) between cloud servers
- Basic problem: Minimize queues/delays/blocking
- Pull-based algorithm (PULL)
  - Main result (informally)
  - PULL Vs “Power-of-d-choices”
  - Main result
  - Proofs
  - Generalizations
- Open problems

Model

$n \to \infty$ scaling parameter

$\lambda < \sum_j \beta_j \mu_j \qquad \sum_j \beta_j = 1$

$B_j \geq 1$ may be infinite or finite



- Motivation: Job distribution among cloud servers
- Objectives: (a) Small queues/delays, (b) Small communication overhead between router and servers

PULL algorithm

PULL:

- When a server becomes idle, it sends a “pull-message” to the router
- When a customer arrives:
  - If router has available pull-messages, it chooses one randomly uniformly, routes the customer accordingly, and destroys the pull-message
  - If router has no pull-messages, it sends the customer to a randomly uniformly chosen server

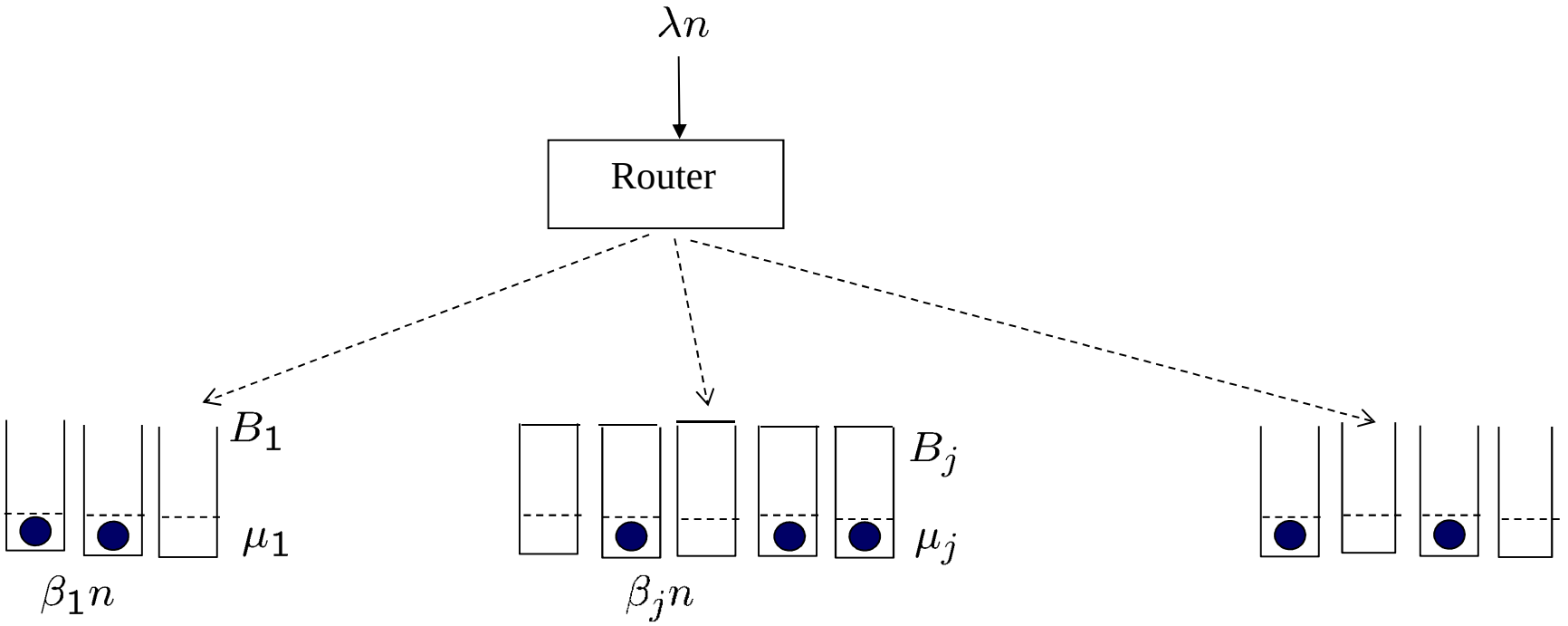
Equivalently (assuming pull-messages are never lost), Join-idle-queue (JIQ):

- When a customer arrives:
  - If there are idle servers, the customer goes to a randomly uniformly chosen idle server
  - If there are no idle servers, the customer goes to a randomly uniformly chosen server

Main result: PULL (asymptotically) eliminates waiting and blocking

THEOREM (informally): Under PULL and FIFO-at-each-server, and assuming the service time distributions have decreasing hazard rate (DHR) [e.g., exponential or Pareto]:
as $n \to \infty$, the steady-state probability of an arriving customer waiting or blocking vanishes.
[In the limit, each server has at most one customer, and there is a non-zero fraction of idle servers.]

Communication overhead: One (pull-)message per customer, in the $n \to \infty$ limit.



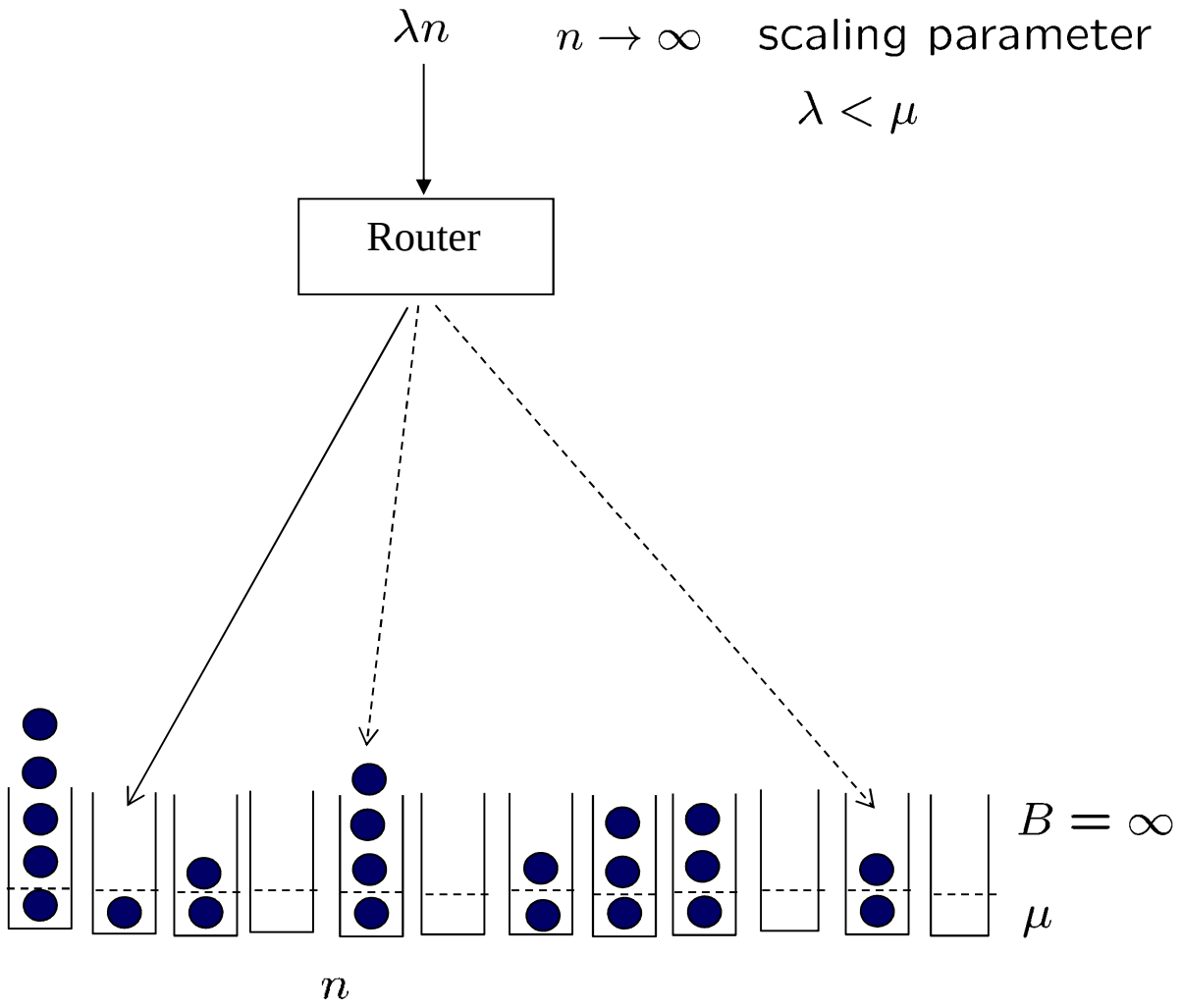
Previous work on pull-based algorithms

- "Pull-based" is as opposed to "push-based" algorithms, which poll server states before customer routing
  - "Power-of-d-choices" is a classical push-based algorithm
- Badonnel-Burgess'08. Dynamic pull-based load balancing for autonomic servers. Network Operations and Management Symposium, NOMS 2008, 751--754.
  - Systems work. Servers request (pull) new work from the router
- Lu-Xie-Kliot-Geller-Larus-Greenberg'11. Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services. Performance Evaluation 68, 1057--1071.
  - Homogeneous (single pool) model, but with multiple routers; the algorithm is essentially same as PULL (up to a generalization discussed later)
  - Different asymptotic regime
  - Analysis is insufficient to establish asymptotic optimality in our sense

PULL features

- No waiting or blocking
- Very small communication overhead (1 message/customer)
- PULL views the set of servers as a single pool
  - The notion of separate pools is purely logical, for analysis purposes
  - The actual physical server pools do not have to be large and can have heterogeneous servers
- Computation overhead is small. (Keeping the list of pull-messages/idle-servers and choosing one randomly.)
- Customer routing occurs immediately upon arrival (after a computation at the router)

Homogeneous system (one pool); exp service time; infinite buffers



- Classical algorithm: "Power-of-d-choices" = Join-shortest-queue(d) = JSQ(d)
- Vvedenskaya-Dobrushin-Karpelevich'96 [VDK'96]

JSQ(d) Vs PULL

$$p_k^n = P\{Q_1(\infty) \geq k\}, \quad k = 0, 1, 2, \ldots$$

$$\pi_k = \lim_{n\to\infty} p_k^n$$

THEOREM [VDK'96]: Under JSQ(d) algorithm:

$$\pi_k = \left(\frac{\lambda}{\mu}\right)^k, \qquad d = 1 \quad \text{(Random uniform routing)}$$

$$\pi_k = \left(\frac{\lambda}{\mu}\right)^{(d^k-1)/(d-1)}, \quad d \geq 2$$

Communication overhead: d = 1: ZERO
d > 1: 2d messages per customer

PULL: $\pi_2 = 0, \quad \pi_1 = \lambda/\mu$

Communication overhead: 1 message per customer

JSQ(d) Vs PULL. Decreasing hazard rate (DHR) distributions

Hazard rate: $h(z) = F'(z)/[1 - F(z)]$

DHR means *non-increasing h(z)*

Exponential distribution: constant hazard rate => DHR

Pareto distribution is DHR: $1 - F(z) = (1 + \sigma z)^{-\alpha}, \quad \sigma > 0, \ \alpha > 1$

Bramson-Lu-Prabhakar'12, '13: Generalization of VDK'96 to DHR service time distributions:

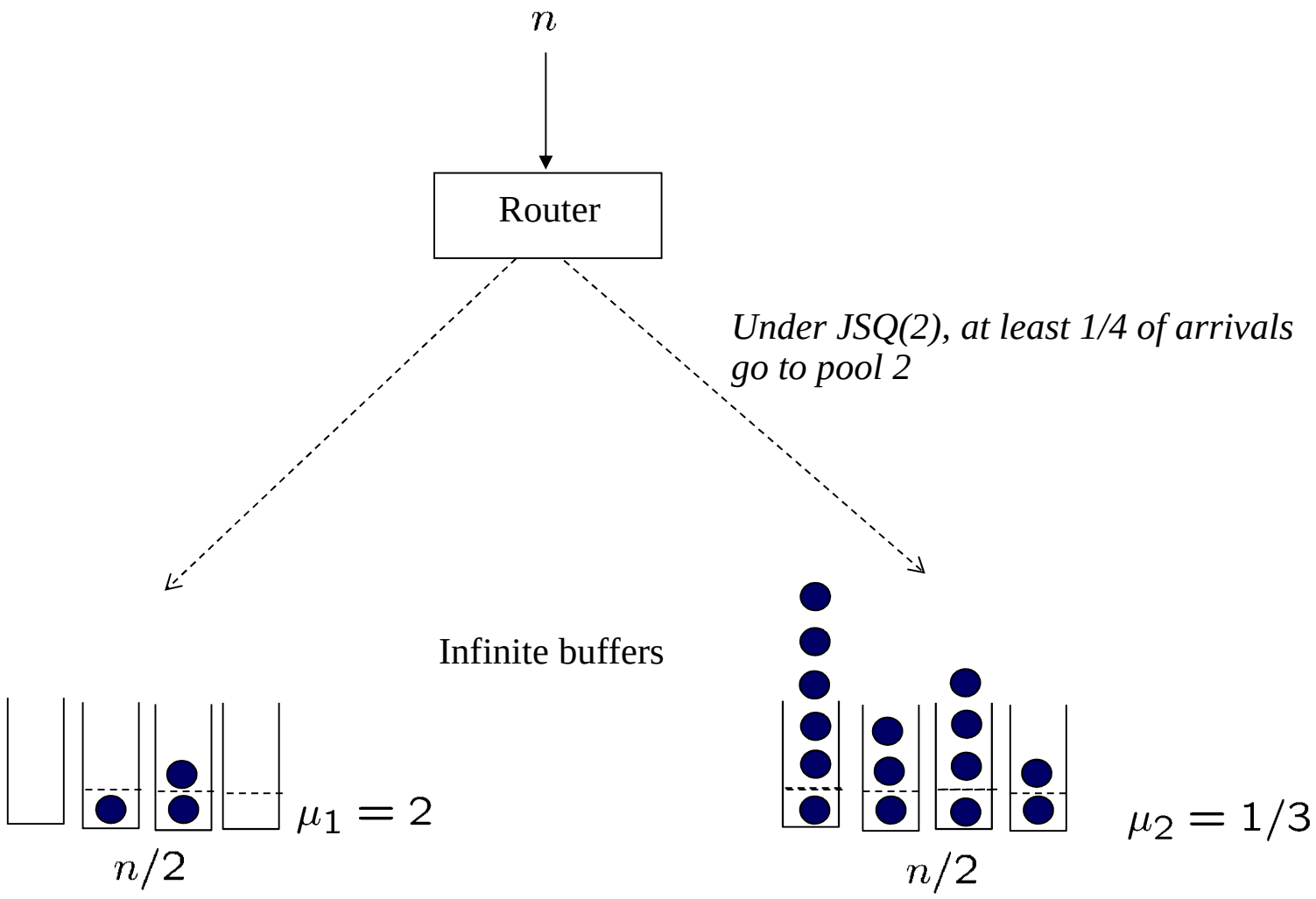
π_k more general, but still $\pi_k > 0, \forall k, \forall d$

Communication overhead: d = 1: ZERO
d > 1: 2d messages per customer

PULL: $\pi_2 = 0, \quad \pi_1 = \lambda/\mu$

Communication overhead: 1 message per customer

Heterogeneous system: JSQ(d) is unsuitable (can easily be unstable)



$n < (n/2)2 + (n/2)(1/3) = (7/6)n$ System is NOT overloaded

$n(1/4) > (n/2)(1/3)$ Pool 2 is overloaded

PULL algorithm analysis
(special case: exp service time distribution)

$x^n_{k,j}(t) \in [0, \beta_j]$ = the fraction of servers (out of the total number n), which are in pool *j* and have queue length $\geq k$ at time *t*

$x^n(t) = (x^n_{k,j}(t))$ = system state at time *t*

Common (for all *n*), *compact* state space:

$$X = \{x = (x_{kj},\ k \in Z_+,\ j \in J) \mid \beta_j = x_{0j} \geq x_{1j} \geq x_{2j} \geq \cdots \geq 0\}$$

Proper state (all queues finite): $\lim_{k\to\infty} x_{kj} = 0,\ \forall j$

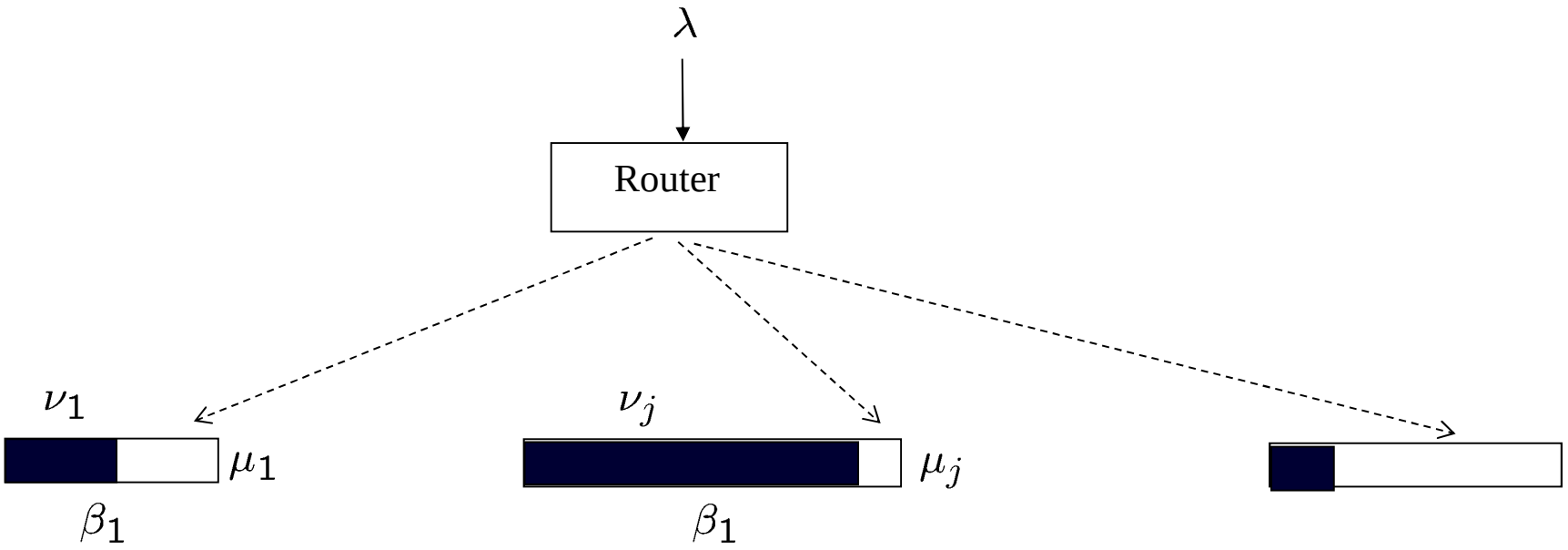
State with some queues infinite: $\lim_{k\to\infty} x_{kj} > 0$ for some j

Main result

Equilibrium point $x^* \in X \;:\; x^*_{1j} = \nu_j, \quad x^*_{kj} = 0, \; k \geq 2, \quad j \in J$

$$\lambda = \sum_j \nu_j \mu_j, \quad \nu_j \mu_j / (\beta_j - \nu_j) = \nu_\ell \mu_\ell / (\beta_\ell - \nu_\ell), \; \forall j, \ell \in J$$

THEOREM: For all sufficiently large *n* the system process is stable, and steady-states are such that $x^n(\infty) \Rightarrow x^*, \quad n \to \infty$



Monotonicity. Lower invariant measure

Monotonicity: If $x^n(0) \leq \bar{x}^n(0)$, then $x^n(\cdot)$ and $\bar{x}^n(\cdot)$ be coupled, so that

$$x^n(t) \leq \bar{x}^n(t), \forall t \geq 0.$$

Lower invariant measure: If the system is initially empty, then $x^n(t)$ is stochastically increasing in *t*, so that $x^n(t) \Rightarrow x^n(\infty)$, where the distribution of $x^n(\infty)$ is the lower invariant measure of the process.

Stability characterization: The process is stable (positive recurrent) if and only if the lower invariant measure is proper. If so, the lower invariant measure is the unique stationary distribution.

Necessary instability condition: If $x^n(\infty)$ is *not* proper, then for at least one j, we have $x^n_{1j}(\infty) = \beta_j$ almost surely.

Fluid (mean-field) limit trajectories

Fluid limit: $x(\cdot) = (x(t), t \geq 0)$ is a probability 1, u.o.c. limit

$$x(\cdot) = \lim_{n \to \infty} x^n(\cdot)$$

Fluid limit properties: (1) If $x(0)$ is the smallest, i.e. "empty", initial state,
then $x(t)$ is monotone increasing, and $x(t) \uparrow x^*$
(2) If $x(0) \geq x^*$ and $x_{2j}(0) \geq \epsilon > 0$, then $x(t) \geq x^*, \ \forall t$ and $x_{1,j}(\tau) \geq \nu_j + \delta$
for some $\delta > 0, \ \tau > 0$

Consider any subsequential limit: $x^n(\infty) \Rightarrow x(\infty)$

Proof of main result: (1) Almost surely, $x(\infty) \geq x^*$

(2) *Instability for infinitely many n is impossible*; otherwise could have $x_{1j}(\infty) = \beta_j$
and, in steady-state for large *n*, more work would be served than arrived
(3) *Impossible to have* $P\{x_{2j}(\infty) > 0\} > 0$; otherwise $P\{x_{1j}(\infty) > \nu_j\} > 0$
and again, in steady-state for large *n*, more work would be served than arrived

$$x(\infty) = x^*$$

Generalization to DHR distributions

Monotonicity under PULL is preserved for DHR + FIFO-at-each-server

More general state space, but can also be compactified, so can still work with the lower invariant measure, etc.

Same main result, same equilibrium point (in terms of queue sizes)

Generalization to queue-size dependent service rate (but service requirement distribution is exponential)

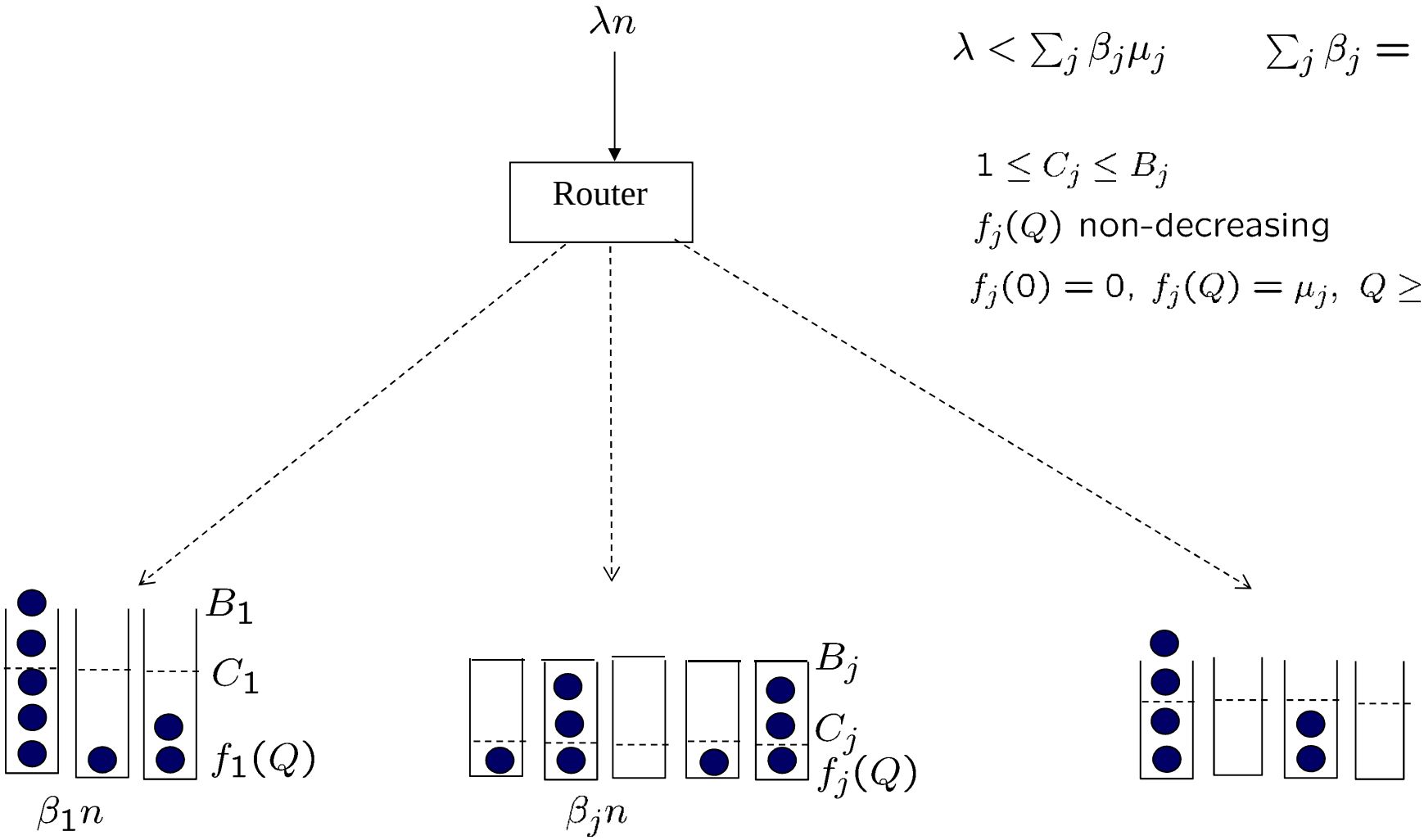
$n \to \infty$ scaling parameter

$\lambda < \sum_j \beta_j \mu_j \qquad \sum_j \beta_j = 1$

$1 \le C_j \le B_j$

$f_j(Q)$ non-decreasing

$f_j(0) = 0,\ f_j(Q) = \mu_j,\ Q \ge C_j$



- Motivation: Servers can work on multiple jobs simultaneously

Generalization to queue-size dependent service rate

PULL:

- When a server completes service AND as a result its queue length Q becomes $< C_j$, it sends a "pull-message" to the router. The number of available pull-messages from the server is $\max\{C_j - Q, 0\}$
- When a customer arrives:
  - If router has available pull-messages, it chooses one randomly uniformly, routes the customer accordingly, and destroys the pull-message
  - If router has no pull-messages, it sends the customer to a randomly uniformly chosen server

Monotonicity is preserved; same state space

Same main result, more general equilibrium point

Open problems / future work

- General service time distributions
- Different asymptotic regimes (e.g., with higher load)
- Multiple routers

Paper

A. Stolyar, “Pull-based load distribution in large-scale heterogeneous service systems”, *Queueing Systems*, 2015, DOI 10.1007/s11134-015-9448-8. arxiv: 1407.6343