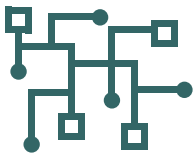

ISE

Industrial and
Systems Engineering

Pull-based load distribution among heterogeneous parallel servers: the case of multiple routers

Alexander L. Stolyar

Lehigh University

ISE Technical Report 16T-001

LEHIGH
UNIVERSITY

Pull-based load distribution among heterogeneous parallel servers: the case of multiple routers

Alexander L. Stolyar
Lehigh University
200 West Packer Avenue, Room 484
Bethlehem, PA 18015
stolyar@lehigh.edu

December 24, 2015

Abstract

The model is a service system, consisting of several large server pools. A server processing speed and buffer size (which may be finite or infinite) depend on the pool. The input flow of customers is split equally among a fixed number of routers, which must assign customers to the servers immediately upon arrival. We consider an asymptotic regime in which the customer total arrival rate and pool sizes scale to infinity simultaneously, in proportion to a scaling parameter n, while the number of routers remains fixed.

We define and study a multi-router generalization of the pull-based customer assignment (routing) algorithm PULL, introduced in [10] for the single-router model. Under PULL algorithm, when a server becomes idle it send a "pull-message" to a randomly uniformly selected router; each router operates independently – it assigns an arriving customer to a server according to a randomly uniformly chosen available (at this router) pull-message, if there is any, or to a randomly uniformly selected server in the entire system, otherwise.

Under Markov assumptions (Poisson arrival process and independent exponentially distributed service requirements), and under sub-critical system load, we prove asymptotic optimality of PULL: as $n \to \infty$, the steady-state probability of an arriving customer experiencing blocking or waiting, vanishes. Furthermore, PULL has an extremely low router-server message exchange rate of one message per customer. These results generalize some of the single-router results in [10].

Key words and phrases: Large-scale heterogeneous service systems; multiple routers (dispatchers); pull-based load distribution; PULL algorithm; load balancing; fluid limits; stationary distribution; asymptotic optimality

AMS 2000 Subject Classification: 90B15, 60K25

1 Introduction

Design of efficient load distribution algorithm for modern data processing systems is a challenging problem. An algorithm needs to have a good performance, which typically means low waiting times and blocking probabilities of the jobs. A big part of the challenge of achieving this goal efficiently, stems from the fact that such systems are heterogeneous (i.e., different servers may have different capabilities) and have very large scale. This makes the following features of a load distribution algorithm very desirable, and perhaps even required:

- An algorithm should be oblivious of the server types as much as possible.
- An algorithm should allow a distributed implementation, where assignment (routing) of jobs to servers is done by multiple routers, each handling a fraction of demand. This is because having a single router to handle a massive demand may be infeasible or impractical (see [6]).
- The router-server signaling overhead should be small.

Motivated by the challenges described above, in this paper we consider the following model. It is a service system, consisting of several large server pools. The system is heterogeneous in that a server processing speed and buffer size depend on the pool; the buffer size is the maximum number of jobs – or, customers – that can be queued at the server, and it can be finite or infinite. There is a finite number of *routers*. The flow of customer arrivals into the system is split equally among the routers, which must assign (or, route) "their" customers to the servers immediately upon arrival. This model is a generalization of the single-router model in [10].

We define and study (two versions of) a *pull-based* customer assignment (routing) algorithm PULL. (One of the algorithm versions is a generalization of the single-router PULL algorithm in [10].) Under this algorithm, when a server becomes idle it sends a "pull-message" to a randomly uniformly selected router; each router operates independently – it assigns an arriving customer to a server according to a randomly uniformly chosen available (at this router) pull-message, if there is any, or to a randomly uniformly selected server in the entire system, otherwise.

We make Markov assumptions: Poisson arrival process and independent exponentially distributed customer service requirements. We consider an asymptotic regime in which the number of routers remains fixed, but the customer total arrival rate and pool sizes scale to infinity simultaneously, in proportion to some scaling parameter n, while the system load is sub-critical (i.e., the load is smaller than the system capacity by cn for some $c > 0$).

The **main results** of this paper can be informally stated as follows (formal statements are given in Lemma 2, Theorems 3 and 4):

- The system process under PULL is stable (positive recurrent) for all sufficiently large n.
- PULL is asymptotically optimal: As $n \to \infty$, the steady-state probability of each pool having idle servers and each router having pull-messages goes to 1; consequently, the steady-state probability of an arriving customer experiencing blocking or waiting, vanishes. (Furthermore, in the limit, the idle servers in each pool have their pull-messages distributed equally among the routers.)
- Router-server signaling message exchange rate under PULL is at most two messages per customer for a pre-limit system, and equal to one message per customer in the $n \to \infty$ limit.

These results generalize the single-router results in [10], under Markov assumptions. (The main results in [10] hold under more general assumptions on the service requirement distributions – it suffices that they have a *decreasing hazard rate*.)

The following are some comments on the model and main results, which demonstrate that PULL algorithm does satisfy the listed above desired features of a load distribution algorithm:

- The assumption that the input flow of customers is equally split among the routers is non-restrictive. A real system can have a single "pre-router", which is a single point of entry of customers into the system. The pre-router's only function would be to distribute arriving customers in, say, round-robin fashion among the routers. Given the utmost simplicity of this function, there is no problem in handling all arriving customers by a single pre-router; this is in contrast to a router, which has to do actual customer-to-server assignment, requiring more processing per customer.

- The asymptotic optimality property of PULL shows that its performance in a large-scale system is perfect – waiting and/or blocking of arriving customers vanishes. In particular, its performance is much superior to that of the celebrated *power-of-d-choices*, or JSQ(d), algorithm [2, 3, 7, 11]. (JSQ(d) is defined and discussed below. See also [10] for more detailed PULL Vs. JSQ(d) comparisons.) The advantage of PULL over JSQ(d) is especially striking in heterogeneous systems, where JSQ(d) is not even applicable in general (see [10]).

- The router-server message exchange rate under PULL is extremely low: at most 2/customer and, in the limit, 1/customer. (It is $2d$ under JSQ(d), where d is a parameter whose values of interest are $d \geq 2$.)

- We want to emphasize that PULL algorithm does *not* try to explicitly balance the distribution of available pull-messages among routers – a perfect (in the limit) balancing occurs "automatically". This useful property is somewhat counterintuitive.

- In terms of technique, the generalization of the results of [10] to the multi-router case is *not straightforward*. The main reason is as follows. Process monotonicity plays a key role in this paper as it did in [10]. However, in the single-router case, there is the minimum (smallest) state of the process; this gives a simple characterization of the process steady-state as a limit of the stochastically monotone increasing process, starting from the smallest state; the analysis in [10] essentially relies on this fact. There is *no* smallest state in the multi-router case; this requires new approaches to establish the main results.

There exists a large amount of previous work on the load distribution in service systems. Majority of the previous work is focused on *load balancing* (in the sense of equalizing server loads) in homogeneous systems (with all servers identical); see e.g. [3, 6] for reviews. Load balancing is only one possible objective of load distribution. While it is a natural objective for homogeneous systems, it is not necessary to achieve asymptotic optimality. Our PULL algorithm is an example – it is asymptotically optimal, without attempting and without in general achieving load balancing among all servers in the system. (It does, however, equalize load among servers within each pool.)

Much attention, especially in recent years, was devoted to *power-of-d-choices*, or *join-the-shortest-queue(d)*, or JSQ(d), algorithm [2, 3, 7, 11]. Under this algorithm each arriving customer joins the shortest out of d randomly selected queues, where $d \geq 1$ is an algorithm parameter. JSQ(1) is just a random uniform routing, so that the interesting cases are when $d \geq 2$ and is small. The cited above work, as well as practically all work on JSQ(d), is for homogeneous systems. (For heterogeneous systems JSQ(d) with $d \geq 2$ is not applicable in general – see [10].) The advantage of JSQ(d) with small $d \geq 2$ over JSQ(1) is that it dramatically reduces the steady-state delays, at the cost of only small router-server signaling message exchange rate of $2d$/customer. Our results show that PULL algorithm asymptotically eliminates steady-state delays and has even smaller signaling message exchange rate, just 1/customer; and it is applicable to heterogeneous systems. Assuming the queue-length queries/responses are instantaneous, the JSQ(d) algorithm is unaffected by whether the system has a single or multiple routers.

A pull-based approach to load distribution is relatively recent [1, 6, 10]. Paper [1] proposed this approach in practical settings, and studied it by simulations, which demonstrated good performance. Paper [6] considers a homogeneous system model with multiple routers. It proposes a pull-based algorithm, called *join-idle-queue* (JIQ), which is essentially what we call PULL algorithm in this paper. (PULL has additional features; most importantly, the random uniform choice of pull-messages by each router.) The asymptotic regime in [6] is different from ours: both the numbers of servers and routers grow to infinity, with their ratio kept constant; heuristic arguments are used to conjecture the limit of the system steady-states; simulations both supported the conjecture validity and demonstrated good performance of the algorithm. Paper [10] considered a special – single-router – version of the model in this paper, and rigorously proved the asymptotic optimality of PULL, under assumptions more general than Markov (namely, for service time distributions with decreasing hazard rate). The main results of this paper generalize those in [10] to the multi-router model, under Markov assumptions.

Recent papers [4, 8] consider a single-router homogeneous system, under Markov assumptions, in the so called Halfin-Whitt asymptotic regime, where the number of servers scales in proportion to $n \to \infty$ and the system load is smaller than capacity by $c\sqrt{n}$ for some $c > 0$. These papers prove diffusion limits of the system transient behavior under *join-the-shortest-queue* routing (in [4]) and under its generalization which also includes JIQ as a special case (in [8]).

Finally, we note that pull-based algorithms can be applied to systems with more general server structures, beyond simple single server. For example, it is shown in [10] that the asymptotic optimality of PULL extends to (single-router) models, where server processing speed depends on its queue length. Another example is recent paper [9], which proposes and studies an algorithm, which can be viewed as a version of PULL, for (single-router) heterogeneous service systems with *packing constraints* at the servers.

1.1 Basic notation

Symbols $\mathbb{R}, \mathbb{R}_+, \mathbb{Z}, \mathbb{Z}_+$ denote the sets of real, real non-negative, integer, and integer non-negative numbers, respectively. For finite- or infinite-dimensional vectors, the vector inequalities are understood component-wise, unless explicitly stated otherwise. We use notation $x(\cdot) = (x(t),\ t \geq 0)$ for both a random process and its realizations, the meaning is determined by the context; the state space (of a random process) and the metric and/or topology on it are defined where appropriate, and we always consider Borel σ-algebra on the state space. Abbreviation *u.o.c.* means *uniform on compact sets* convergence, and *w.p.1* means *with probability 1*. Notations $\Rightarrow$ and $\stackrel{d}{=}$ signify convergence and equality *in distribution*, respectively, for random elements. For a process $x(\cdot)$, we denote by $x(\infty)$ a random element whose distribution is a *stationary distribution* of the process, assuming the latter exists. For $a, b \in \mathbb{R}$, $\lfloor a \rfloor$ denotes the largest integer less than or equal to a, $a \wedge b = \min\{a, b\}$. We use notation $Dist[X]$ for the probability distribution of a random element X; for a measure χ its total variation is denoted $\|\chi\|$. For two functions $z(u)$ and $\bar{z}(u)$ of $u \geq 0$ we write $z(u) \sim \bar{z}(u)$ to mean $z(u)/\bar{z}(u) \to 1$ as $u \downarrow 0$; left limit of a function (or a process) at a given point is denoted $z(t-)$; the right derivative at u is denoted $(d^+/du)z(u)$. Abbreviation *WLOG* means *without loss of generality*; *RHS* and *LHS* mean *right-hand side* and *left-hand side*, respectively.

1.2 Paper organization

The model and main results on stability (Lemma 2 and Theorem 3) and asymptotic optimality (Theorem 4) of PULL are given in Section 2, which also contains a discussion of PULL implementation mechanisms. Section 3 introduces an order relation on the process state space, and establishes monotonicity properties, which are a key tool for our analysis; it also discusses differences in implications of monotonicity in the single-router and multi-router cases. In Section 4 we study the process fluid limits, which are another important tool used in the paper. We study a special system where all buffer sizes are 1 in Section 5; these are auxiliary results, needed for analysis of the general case. Proofs of Theorems 3 and 4 are in Sections 6 and 7, respectively. We conclude in Section 8.

2 Model and main results

2.1 Model structure

Arrival process. Customers arrive according to a Poisson process of rate $\Lambda > 0$.

Servers. There are $J \geq 1$ server pools. Pool $j \in \mathcal{J} \equiv \{1, \ldots, J\}$ consists of N_j identical servers. Servers in pool 1 are indexed by $i \in \mathcal{N}_1 = \{1, \ldots, N_1\}$, in pool 2 by $i \in \mathcal{N}_2 = \{N_1 + 1, \ldots, N_1 + N_2\}$, and so on; $\mathcal{N} = \cup \mathcal{N}_j$ is the set of all servers. The service time of a customer at a server in pool j is an independent,

exponentially distributed random variable with mean $1/\mu_j \in (0,\infty)$, $j \in \mathcal{J}$. We assume that the customers at any server are served in the first-come-first-serve (FCFS) order. (That is, at any time only the head-of-the-line customer at each server is served.) The buffer size (maximum queue length) at any server in pool j is $B_j \geq 1$; we allow the buffer size to be either finite, $B_j < \infty$, or infinite, $B_j = \infty$. A new customer, routed to a server $i \in \mathcal{N}_j$, joins the queue at that server, unless B_j is finite and the queue length $Q_i = B_j$ – in this case the customer is blocked (or lost; i.e., it leaves the system immediately, without receiving any service).

Routing. We assume that the customer arrival process is distributed equally among R routers, labeled by $r \in \mathcal{R} = \{1,\dots,R\}$. Specifically, each arriving customer is assigned randomly uniformly to one of the routers; therefore, the arrival processes to the routers are independent Poisson processes of rate Λ/R. When a router receives an arriving customer, it has to either immediately route it to one of the servers or immediately block it from service (in which case the customer is lost).

2.2 Asymptotic regime

We consider the following (many-servers) asymptotic regime. The total number of servers $n = \sum_j N_j$ is the scaling parameter, which increases to infinity; the arrival rate and the server pool sizes increase in proportion to n, $\Lambda = \lambda n$, $N_j = \beta_j n$, $j \in \mathcal{J}$, where $\lambda, \beta_j, j \in \mathcal{J}$, are positive constants, $\sum_j \beta_j = 1$. (To be precise, the values of N_j need to be integer, e.g. $N_j = \lfloor \beta_j n \rfloor$. Such definition would not cause any problems, besides clogging notation, so we will simply assume that all $\beta_j n$ "happen to be" integer.) We assume that the subcritical load condition holds:

$$\lambda < \sum_j \beta_j \mu_j. \tag{1}$$

2.3 PULL routing algorithm

We study the following pull-based algorithm, labeled PULL.

Definition 1 (PULL)**.** *We consider two versions of the PULL algorithm, labeled PULL-1 and PULL-2. They have common components (a)-(d) below; the components (e.1) and (e.2) pertain only to PULL-1 and PULL-2, respectively.*

(a) At any given time, each idle server has exactly one pull-message *(containing the server identity), located at one of the routers. (Equivalently, each idle server is* associated *with one of the routers, and each router "knows" the subset of idle servers currently associated with it.) A pull-message located at router r is called r-pull-message.*

(b) When a server becomes idle after a service completion and departure of a customer, a pull-message from this server is generated and placed at one of the routers, chosen randomly uniformly.

(c) When a customer arrives at an idle server, the pull-message for this server is destroyed. (The pull-message is destroyed whether it was located at the router which sent the customer or at a different router.)

(d) When a customer arrives at router r, if there are available r-pull-messages, the customer is sent to one of the corresponding idle servers, chosen randomly uniformly.

(e.1) Under PULL-1, when a customer arrives at router r, if there are no available r-pull-messages, the customer is blocked (and lost).

(e.2) Under PULL-2, when a customer arrives at a router, if there are no available r-pull-messages, the customer is sent to one of the servers in the entire system, chosen randomly uniformly. (Recall that, according to the model structure, the customer then may be blocked at the server; this occurs if and only if the server's queue is "full", i.e. the queue length is equal to the finite buffer size.)

Note that, if we are interested in the system stationary distributions, then for PULL-1, WLOG, we can assume $B_j = 1$ for all j, because for any actual buffer sizes and any initial state, w.p.1 after a finite time there will be at most one customer at each server. We adopt this assumption for PULL-1 for the rest of the paper.

We also remark that the PULL algorithm in [10] is a special case (for $R = 1$) of PULL-2, but not of PULL-1.

2.4 Main results

Our main results hold for both versions, PULL-1 and PULL-2, of the algorithm. (Of course, this does *not* mean that the system *behavior* is same under both versions.) In general, throughout the paper, when we say that a certain fact holds for PULL algorithm, we mean that it holds for any fixed version of it, i.e. PULL-1 or PULL-2.

In the system with parameter n, the system state is given by $S^n = (Q_i^n,\ D_i^n,\ i \in \mathcal{N})$, where the components are defined as follows: $Q_i^n \in \mathbb{Z}_+$ is the queue length at server i; $D_i^n \in \mathcal{R}$ is the label of the router that "holds" the pull-message from server i, if it is idle ($Q_i^n = 0$), and $D_i^n = 0$, otherwise. Obviously, the components of the state satisfy the condition

$$\sum_{i \in \mathcal{N}_j} I\{Q_i^n = 0\} = \sum_{i \in \mathcal{N}_j} I\{D_i^n \neq 0\}, \quad \forall j \in \mathcal{J},$$

because this is the total number of idle servers in pool j.

Due to symmetry of servers within each pool, the alternative – *mean field*, or *fluid-scale* – representation of the process is as follows. Let $x_{k,j}^n$ denote the fraction of the (total number of) servers, which are in pool j and have queue length greater than or equal to k; let $\xi_{r,j}^n$ denote the fraction of the (total number of) servers, which are idle in pool j and have a pull-message at router r. Denote

$$x^n = (x_{k,j}^n,\ k \in \mathbb{Z}_+,\ j \in \mathcal{J}), \quad \xi^n = (\xi_{r,j}^n,\ r \in \mathcal{R},\ j \in \mathcal{J}).$$

Then

$$s^n = (x^n, \xi^n),$$

is (the mean field representation of) the system state. We will view states s^n, for any n, as elements of the common space with elements denoted by

$$s = (x, \xi), \quad x = (x_{k,j},\ k \in \mathbb{Z}_+,\ j \in \mathcal{J}), \quad \xi = (\xi_{r,j},\ r \in \mathcal{R},\ j \in \mathcal{J}).$$

Specifically, the common state space is

$$\mathcal{S} = \{s \mid \beta_j = x_{0j} \geq x_{1j} \geq x_{2j} \geq \cdots \geq 0 \text{ and } x_{0j} - x_{1j} = \sum_r \xi_{rj}, \forall j \in \mathcal{J}\},$$

equipped with metric

$$\rho(s, s') = \sum_j \sum_k 2^{-k} \frac{|x_{kj} - x'_{kj}|}{1 + |x_{kj} - x'_{kj}|} + \sum_j \sum_r |\xi_{rj} - \xi'_{rj}|, \tag{2}$$

and the corresponding Borel σ-algebra. (The corresponding topology is that of componentwise convergence.) Space $\mathcal{S}$ is compact.

For any n, the process $S^n(t),\ t \geq 0$, – and its projection $s^n(t),\ t \geq 0$, – is a continuous-time, countable state space, irreducible Markov process. (For any n, the state space of $s^n(\cdot)$ is a countable subset of $\mathcal{S}$.) If the buffer sizes B_j are finite in *all* pools j, the state space is obviously finite, and therefore the process $S^n(\cdot)$ (and then $s^n(\cdot)$) is positive recurrent (and then ergodic), with unique stationary distribution. Therefore, the following fact is trivially true – we present it for ease of reference.

Lemma 2. *Consider the system under PULL algorithm. If $B_j < \infty$ for all $j \in \mathcal{J}$, the Markov process $s^n(\cdot)$ is ergodic. (Recall that, under PULL-1, all buffer sizes are finite, $B_j = 1,\ j \in \mathcal{J}$, by definition.)*

Our first main result is that $s^n(\cdot)$ is ergodic under arbitrary (possibly infinite) buffer sizes, when n is large enough.

Theorem 3. *Consider the system under PULL algorithm. Suppose that $B_j = \infty$ for some (or all) $j \in \mathcal{J}$. (This implies that the algorithm is necessarily PULL-2.) Then, for all sufficiently large n, the Markov process $s^n(\cdot)$ is ergodic.*

Define numbers $\nu_j \in (0, \beta_j)$, $j \in \mathcal{J}$, uniquely determined by the conditions

$$\lambda = \sum_j \nu_j \mu_j, \quad \nu_j \mu_j / (\beta_j - \nu_j) = \nu_\ell \mu_\ell / (\beta_\ell - \nu_\ell),\ \forall j, \ell \in \mathcal{J}. \tag{3}$$

Let us define the *equilibrium point* $s^* \in \mathcal{S}$ by

$$x^*_{1,j} = \nu_j, \quad x^*_{k,j} = 0,\ k \geq 2, \quad j \in \mathcal{J}, \tag{4}$$

$$\xi^*_{r,j} = (\beta_j - \nu_j)/R, \quad r \in \mathcal{R},\ j \in \mathcal{J}. \tag{5}$$

The equilibrium point s^* definition in (3)-(5) has the following interpretation. Condition (4) means that point s^* is such that the fraction $\nu_j < \beta_j$ of servers (out of the total number of servers) in pool j is occupied by exactly one customer, while the remaining servers in pool j are idle. Moreover, according to (5), *the idle servers is each pool are in equal quantities associated with (have a pull-message at) different routers;* this in particular means that each router has pull-messages available, and therefore all new arrivals are routed according to pull-messages. Finally, the numbers ν_j are uniquely determined by the condition (3), which simply says that the rate at which new arrivals are routed to pool j (it is proportional to $\beta_j - \nu_j$) is equal to the service completion rate at pool j (it is proportional to $\nu_j \mu_j$).

By Lemma 2 and Theorem 3 process $s^n(\cdot)$ is ergodic for all large n (and if all buffers are finite – for all n). Denote by $s^n(\infty)$ a random element with the distribution equal to the stationary distribution of $s^n(\cdot)$.

Our second main result is the following

Theorem 4. *Under PULL algorithm, $s^n(\infty) \Rightarrow s^*$ as $n \to \infty$.*

Given the definition of s^*, Theorem 4 implies that, in the $n \to \infty$ limit, there are always pull-messages available at each router. (Furthermore, the pull-messages from idle servers in each pools are distributed equally among the routers.) Therefore, PULL algorithm is *asymptotically optimal* in the following sense: as $n \to \infty$, the steady-state probability of an arriving customer experiencing blocking or waiting, vanishes. (Together, Theorems 3 and 4 generalize Theorem 2 in [10] to the model with multiple routers.) We discuss the implications of Theorem 4 in more detail in Section 2.5.

2.5 PULL algorithm implementation mechanisms

We will now discuss some natural *mechanisms* for the PULL algorithm practical implementation. This discussion will demonstrate that very simple and extremely efficient mechanisms do exist, which is a major motivation for the PULL algorithm in the first place. It is also important to emphasize that *as far as the algorithm analysis is concerned*, the actual implementation mechanism is, of course, irrelevant – Definition 1 of the algorithm is sufficient for the analysis.

PULL-1. When a server is initialized in the actual system, it is idle, and it sends a pull-message to one of the routers chosen uniformly at random. After that, the server sends a new pull-message to a randomly

uniformly chosen router, after any service completion that leaves the server idle. (By definition of PULL-1, any service completion leaves the server idle.) Each pull-message contains the server identity. When a customer arrives at a router, if this router has available pull-messages, it picks one of those pull-messages uniformly at random, sends the customer to the corresponding server, and immediately destroys the "used" pull-message. When a customer arrives at a router, if this router has no available pull-messages, the customer is blocked (and discarded).

PULL-2. When a server is initialized in the actual system, it is idle, and it sends a pull-message to one of the routers chosen uniformly at random. After that, the server sends a new pull-message to a randomly uniformly chosen router, after any service completion that leaves the server idle. (Under PULL-2, in general, not every service completion leaves the server idle.) Each pull-message contains the server identity. Each time a server sends a pull-message, it "remembers" the identity of the router, to which it was sent. When a customer arrives at a router, if this router has available pull-messages, it picks one of those pull-messages uniformly at random, sends the customer to the corresponding server, and immediately destroys the "used" pull-message. When a customer arrives at a router, if this router has no available pull-messages, the customer is sent to one of the servers in the entire system, chosen uniformly at random. Every customer sent from a router to a server is supplied with one additional bit of information, indicating whether or not a pull-message was used for its routing or not. When a customer arrives to an idle server, the server checks whether or not a pull-message was used in routing; if not, the server sends a special "pull-remove-message" (containing identity of the server) to the router which "held" the pull-message from the server. When a router receives a pull-remove-message, it destroys the pull-message of the corresponding server.

Next we discuss key properties of these implementation mechanisms.

2.5.1 Asymptotic optimality is achieved by a very simple mechanism.

PULL algorithm implementation is clearly very simple. Routers do not keep track of the exact states of the servers. Each router only needs to have the list of servers (which is necessary under any algorithm, as discussed in Section 2.5.2 of [10]), where for each server one additional bit is used to indicate the presence of a pull-message. Each server only needs to have the list of routers (which it needs under any feedback-based algorithm), and one additional variable to "remember" which router currently holds its pull-message.

Also note that there is *no* additional mechanism to balance the numbers of pull-messages at different routers; such balancing (in the limit) occurs automatically. (A pull-based algorithm proposed in [6] for the case of multiple routers, does suggest a pull-message balancing mechanism. Our results prove that when the ratio of servers to routers is large – which is our asymptotic regime – such additional mechanism is unnecessary.)

2.5.2 Performance: message exchange rate.

It is clear that, under PULL-1, the steady-state rate of message exchange between the routers and the servers is *at most one* message per customer. Specifically, one pull-message is sent per each customer upon its service completion. (Some fraction of customers is blocked, so for a finite n not every customer generates a pull-message.) As $n \to \infty$, the steady-state blocking probability vanishes; therefore, in the $n \to \infty$ limit, the message exchange rate is exactly one message per customer.

Under PULL-2, the steady-state router/server message exchange rate is *at most two* messages per customer. Specifically, for each customer, at most one pull-remove-message is sent upon its arrival at a server and at most one pull-message is sent upon its service completion (when it leaves the server idle). In the $n \to \infty$ limit, the message exchange rate is only one message per customer (same as for PULL-1), because the probability that a customer is routed *without* using a pull-message vanishes.

The asymptotic message exchange rate of one per customer is much lower than that under JSQ(d) algorithm, for which it is $2d$. (See [10] for a more detailed discussion.)

2.5.3 Performance: absence of routing delay.

Pull-messages do *not* contribute to the routing delay: an arriving customer in not waiting at the router for any pull-message or pull-remove-message, the routing decision is made immediately. This is unlike JSQ(d) algorithm, where each arriving customer waits for the queue-length request/response message exchange to complete, before being routed. (See also [6] for a discussion of this issue.)

2.5.4 The notion of servers pools is purely logical.

Neither routers nor servers use the notion of a server pool. Indeed, each router need *not* know anything about each server parameters (service speed, buffer size) or state (queue length), besides currently having or not a pull-message from it. This means that from the "point of view" of each router all servers form a single pool. The notion of a server pool that we have in the model is purely logical, used for analysis only. A real system may consist of a single or multiple pools of *non-identical* servers. In this case, we consider all servers of a particular type as forming a *logical* pool. Our results still apply, as long as the number of servers of each type in the entire system is large.

3 The process in the compactified state space. Monotonicity

All results in this section concern a system with a fixed n. Also, they hold for any fixed version of PULL – either PULL-1 or PULL-2.

3.1 Compactified state space. Order relation.

It will be convenient to consider a more general system and the Markov process. Namely, we assume that the queue length at any server $i \in \mathcal{N}_j$ within a pool j with infinite buffer size ($B_j = \infty$), can be infinite. In other words, $Q_i^n(t)$ can take values in the set $\bar{\mathbb{Z}}_+ \doteq \mathbb{Z}_+ \cup \{\infty\}$, which is the one-point compactification of $\mathbb{Z}_+$, containing the "point at infinity." We consider the natural topology and order relation on $\bar{\mathbb{Z}}_+$. Obviously, $\bar{\mathbb{Z}}_+$ is compact. (Note that if A is a finite subset of $\mathbb{Z}_+$, then sets A and $\bar{\mathbb{Z}}_+ \setminus A$ are both closed and open.)

Therefore, the state space of the generalized version of Markov process $S^n(\cdot)$ is the compact set $\bar{\mathbb{Z}}_+^n \times [\mathcal{R} \cup \{0\}]^n$. The process transitions are defined in exactly same way as before, with the following additional convention. If $Q_i^n(t) = \infty$, then neither new arrivals into this queue nor service completions in it, change the system state; this means, in particular, that $Q_i^n(\tau) \equiv \infty$ for all $\tau \geq t$.

The corresponding generalized version of the process $s^n(\cdot)$ is defined as before; if at time t some of the queues in pool j are infinite, then $x^n(t)$ is such that $\lim_{k\to\infty} x_{k,j}^n(t) > 0$. Note that the state space of the generalized $s^n(\cdot)$ is still the compact set $\mathcal{S}$, as defined above.

It is easy to see that, for each n, the (generalized versions of) processes $S^n(\cdot)$ and $s^n(\cdot)$ are Feller continuous.

We will consider the following natural order relation on the process $S^n(\cdot)$ state space. Vector inequality $Q' \leq Q''$ for $Q', Q'' \in \bar{\mathbb{Z}}_+^n$ is understood component-wise. If $D', D'' \in [\mathcal{R} \cup \{0\}]^n$, then $D' \leq D''$ is understood as the following property:

$$D_i'' \neq 0 \;\text{ implies }\; D_i' = D_i'', \quad \forall i \in \mathcal{N}. \tag{6}$$

Then, $S' = (Q', D') \leq S'' = (Q'', D'')$ is defined as $Q' \leq Q''$ and $D' \leq D''$. The meaning of this order relation is clear: the state S' is dominated by S'' if its queue length is smaller or equal at each server, and the set of r-pull-messages in S' is a superset of that in S'' for each r at each pool j.

The stochastic order relation $S' \leq_{st} S''$ between two *random elements* means that they can be constructed on the same probability space so that $S' \leq S''$ holds w.p.1.

The corresponding order relation on the state space $\mathcal{S}$ for the mean-field process $s^n(\cdot)$ is as follows. For $s' = (x', \xi'), s'' = (x'', \xi'') \in \mathcal{S}$, $s' \leq s''$ is defined as $x' \leq x''$ component-wise and the property:

$$\xi'_{rj} \geq \xi''_{rj} \quad \forall r \in \mathcal{R},\ \forall j \in \mathcal{J}. \tag{7}$$

The stochastic order relation $s' \leq_{st} s''$ between two *random elements* in $\mathcal{S}$ means that they can be constructed on the same probability space so that $s' \leq s''$ holds w.p.1.

In the rest of the paper, for a state $s^n(t)$ (with either finite $t \geq 0$ or $t = \infty$), we denote by

$$x^n_{\infty,j}(t) \doteq \lim_{k\to\infty} x^n_{k,j}(t) \tag{8}$$

the fraction of queues that are in pool j and are infinite. (Note that $x^n_{\infty,j}(t)$ is a function, but *not a component*, of $x^n(t)$.) Also, denote by $y^n_{k,j}(t)$ the fraction of queues that are in pool j and have queue size *exactly* $k \in \bar{\mathbb{Z}}_+$:

$$y^n_{k,j}(t) \doteq x^n_{k,j}(t) - x^n_{k+1,j}(t), \quad k \in \mathbb{Z}_+, \tag{9}$$

$$y^n_{\infty,j}(t) \doteq x^n_{\infty,j}(t) = \lim_{k\to\infty} x^n_{k,j}(t). \tag{10}$$

3.2 Monotonicity.

The following monotonicity property holds. (For a general definition and discussion of stochastic processes' monotonicity properties, cf. [5].) It is a rather straightforward generalization of Lemma 3 in [10], where it was given for the special case of single router. (We give a proof for completeness.)

Lemma 5. *Consider two version of the process, $S^n(\cdot)$ and $\bar{S}^n(\cdot)$ [resp. $s^n(\cdot)$ and $\bar{s}^n(\cdot)$], with fixed initial states $S^n(0) \leq \bar{S}^n(0)$ [resp. $s^n(0) \leq \bar{s}^n(0)$]. Then, the processes can be constructed on a common probability space, so that, w.p.1, $S^n(t) \leq \bar{S}^n(t)$ [resp. $s^n(t) \leq \bar{s}^n(t)$] for all $t \geq 0$. Consequently, $S^n(t) \leq_{st} \bar{S}^n(t)$ [resp. $x^n(t) \leq_{st} \bar{x}^n(t)$] for all $t \geq 0$. This property still holds if the second process $\bar{S}^n(\cdot)$ [resp. $\bar{s}^n(\cdot)$] is for a system with possibly larger buffer sizes: $B_j \leq \bar{B}_j \leq \infty,\ \forall j \in \mathcal{J}$.*

Proof. It suffices to prove the result for $S^n(\cdot)$ and $\bar{S}^n(\cdot)$. We will refer to the systems, corresponding to $S^n(\cdot)$ and $\bar{S}^n(\cdot)$, as "smaller" and "larger", respectively. Let us consider PULL-2 algorithm first. It is clear how to couple the service completions in the two systems, so that any service completion preserves the $S^n(t) \leq \bar{S}^n(t)$ condition. We make the arrival process to each router r common for both systems. It suffices to show that condition $S^n(t) \leq \bar{S}^n(t)$ is preserved after an arrival at a given router r at time t. Suppose the (joint) system state just before this arrival is such that $S^n \leq \bar{S}^n$. If router r does not have pull-messages in either system, i.e. $\sum_j I\{D^n_i = r\} = \sum_j I\{\bar{D}^n_i = r\} = 0$, we make a *common* random uniform assignment of the arrival to one of the servers in the entire system. If router r has pull-messages in the smaller system, but not in the larger system, i.e. $\sum_j I\{D^n_i = r\} > \sum_j I\{\bar{D}^n_i = r\} = 0$, we make *independent* assignments in the two systems, according to the algorithm. In the case when router r has pull-messages in both systems, i.e. $\sum_j I\{D^n_i = r\} \geq \sum_j I\{\bar{D}^n_i = r\} > 0$, the corresponding set of idle servers in the larger system forms a subset of that in the smaller one. In this case we do the following. We make uniform random choice of a pull-message in the smaller system, and assign the arrival to the corresponding idle server; if the chosen pull-message in the smaller system is also present in the larger system, we assign the arrival to the same idle server in the larger system; if the chosen pull-message in the smaller system is not present in the larger system, we do an additional step of choosing randomly uniformly a pull-message in the larger system, and assigning the arrival accordingly. Clearly, condition $S^n(t) \leq \bar{S}^n(t)$ is preserved in each case, and the procedure conforms to the PULL-2 algorithm in both systems.

The coupling construction for the PULL-1 algorithm is same, except it is simpler: if an arrival at router r does not find a pull-message there, it is immediately blocked. The condition $S^n(t) \leq \bar{S}^n(t)$ is still preserved after each arrival. □

Note that the state space has a *maximum* ("largest") state, namely the "full" state, with $Q_i = B_j$ for all j and $i \in \mathcal{N}_j$. Then, from Lemma 5 we obtain the following

Corollary 6. *Consider the process $S^n(\cdot)$ starting from the maximum initial state. Then the process $S^n(\cdot)$ is* monotone non-increasing*; namely, for any $0 \le t_1 \le t_2 \le \infty$, $S^n(t_2) \le_{st} S^n(t_1)$.*

We will also need an order relation on a subset of servers. We say that *state S' is dominated by state S'' on a subset $\widehat{\mathcal{N}} \subseteq \mathcal{N}$ of servers*, if $D' \le D''$ (understood as (6)) and $Q'_i \le Q''_i,\ i \in \widehat{\mathcal{N}}$. Therefore, this relation is such that if $i \notin \widehat{\mathcal{N}}$ and $Q''_i > 0$ then condition $Q'_i \le Q''_i$ is *not* required.

3.3 Implications of monotonicity: singe-router Vs. multi-router case

Monotonicity is one of the main tools used in this paper, as it was in [10] for the special case of a single router. However, some key parts of the analysis in [10] *cannot* be extended to the multi-router case. The principal reason is that in the single-router case the state space has the minimum state – this gives a simple characterization of the stationary distribution as the lower invariant measure. In the multi-router case there is *no minimum state*, which substantially complicates the analysis. We now discuss these issues in more detail.

Single router. In the special case of a single router, $R = 1$, studied in [10], the state space has the *minimum* ("smallest") element, namely the state with all servers idle. In that case, for a given system and parameter n, we considered the process, starting from the minimum state $S^n(0)$. Such process, as easily follows from Lemma 5, is stochastically non-decreasing in time

$$S^n(t_1) \le_{st} S^n(t_2) \quad [\text{resp. } s^n(t_1) \le_{st} s^n(t_2)], \quad 0 \le t_1 \le t_2 < \infty,$$

and since the state space is compact, we have

$$S^n(t) \Rightarrow S^n(\infty) \quad [\text{resp. } s^n(t) \Rightarrow s^n(\infty)], \quad t \to \infty,$$

where the distribution of $S^n(\infty)$ [resp. $s^n(\infty)$] is the *lower invariant measure* of process $S^n(\cdot)$ [resp. $s^n(\cdot)$]. (The lower invariant measure is a stationary distribution of the process, stochastically dominated by any other stationary distribution. Cf. [5], in particular Proposition I.1.8(d).) Then, as shown in [10], the following is true for the process $S^n(\cdot)$ [resp. $s^n(\cdot)$], *as originally defined (without infinite queues)*: the process is ergodic if and only if $S^n(\infty)$ [resp. $s^n(\infty)$] is proper in the sense that

$$\mathbb{P}\{Q_i^n(\infty) < \infty,\ \forall i\} = 1 \quad [\text{resp. } \mathbb{P}\{x^n_{\infty,j}(\infty) = 0,\ \forall j\} = 1].$$

And if the original process is ergodic, the lower invariant measure is its unique stationary distribution.

Multiple routers. The more general model that we consider in this paper, with multiple routers, $R \ge 1$, is substantially different. There is a finite number (greater than one) of minimal states; *a single minimum state does not exist.* There exists the *maximum* ("largest") state, namely the "full" state, with $Q_i = B_j$ for all j and $i \in \mathcal{N}_j$. In the special case when *all buffer sizes are finite*, $B_j < \infty,\ \forall j$, the situation is in a sense analogous to that for the single router in [10]. First, since the state space is finite the process is automatically ergodic. Second, we can consider the process starting with the maximum state; by Corollary 6 it is stochastically monotone non-increasing, and the distribution of $S^n(t)$ converges to the *upper invariant measure*, which is the unique stationary distribution. However, if buffer sizes are infinite in some or all pools, the situation is much more complicated. The maximum state and the upper invariant measure still exist, but this upper invariant measure is not necessarily (and typically is not) the stationary distribution of the original process. For example, if all buffer sizes are infinite, $B_j = \infty$ for all j, then the maximum state (with all queues infinite) is invariant, so the upper invariant measure is concentrated on it; but it is not the stationary distribution the original process, if the latter happens to be ergodic. Thus, the upper invariant measure is not useful in the case of infinite buffers, which means that a key element of the approach in [10] does not extend to multiple routers.

4 Fluid limits

In this section we consider limits of the sequence of process $s^n(\cdot)$ as $n \to \infty$. First, define fluid sample paths (FSP), which arise as limits of the trajectories $s^n(\cdot)$ as $s \to \infty$. The FSP construction will apply to both PULL-1 and PULL-2; it will be clear which parts of the construction become trivial (or redundant) for either PULL-1 or PULL-2. Similarly, when we establish properties of the FSPs, they hold for both PULL-1 and PULL-2, unless explicitly stated otherwise.

Without loss of generality, assume that Markov process $s^n(\cdot)$ for each n is driven by a common set of primitive processes, as defined next.

Let $A_r^n(t),\ t \geq 0$, denote the number of exogenous arrivals into the system, via router r, in the interval $[0, t]$. Assume that

$$A_r^n(t) = \Pi_r^{(a)}(\frac{\lambda}{R} nt), \tag{11}$$

where $\Pi_r^{(a)}(\cdot)$ is an independent unit rate Poisson process. The functional strong law of large numbers (FSLLN) holds:

$$\frac{1}{n}\Pi_r^{(a)}(nt) \to t,\ u.o.c.,\quad w.p.1,\quad \forall r \in \mathcal{R}. \tag{12}$$

Denote by $D_{k,j,r}^n(t),\ t \geq 0,\ 1 \leq k < \infty$, the total number of departures in $[0, t]$ from servers in pool j with queue length k, which are "associated" with router r; this association with router r is only used when $k = 1$, in which case an r-pull-message is generated. (Also note that in case of PULL-1, $D_{k,j,r}^n(t) \equiv 0$ for any $k \geq 2$.) Assume that

$$D_{k,j,r}^n(t) = \Pi_{k,j,r}^{(d)}\left(\frac{1}{R}\int_0^t n y_{k,j}^n(w)\mu_j dw\right), \tag{13}$$

where $\Pi_{k,j,r}^{(d)}(\cdot)$ are independent unit rate Poisson processes. (Recall that departures from – and arrivals to – infinite queues can be ignored, in the sense that they do not change the system state.) Similarly to (12),

$$\frac{1}{n}\Pi_{k,j,r}^{(d)}(nt) \to t,\ u.o.c.,\quad w.p.1,\quad 1 \leq k < \infty,\ \forall j,\ \forall r \in \mathcal{R}. \tag{14}$$

The routing of new arrivals, going through router r, is constructed as follows. There are two sequences of i.i.d. random variables,

$$\chi_r(1), \chi_r(2), \ldots, \quad \text{and} \quad \zeta_r(1), \zeta_r(2), \ldots,$$

uniformly distributed in $[0, 1)$. The routing of the m-th arrival (via router r) into the system is determined by the values of r.v. $\chi_r(m)$ and $\zeta_r(m)$, as follows. (We will drop index m, because we consider one arrival.) Let s^n denote the system state just before the arrival. If $\sum_j \xi_{rj}^n = 0$, i.e. there are no pull-messages at r, then under PULL-1 the arrival is blocked, and under PULL-2 the routing is determined by ζ_r as follows. The customer is sent to a server with k, $k \geq 0$, customers in pool 1, if $\zeta_r \in [x_{k+1,1}^n, x_{k,1}^n)$, and to a server with $k = \infty$ customers in pool 1, if $\zeta_r \in [0, x_{\infty,1}^n)$; the customer is sent to a server with k, $k \geq 0$, customers in pool 2, if $\zeta_r \in [\beta_1 + x_{k+1,2}^n, \beta_1 + x_{k,2}^n)$, and to a server with $k = \infty$ customers in pool 2, if $\zeta_r \in [\beta_1, \beta_1 + x_{\infty,2}^n)$; and so on. If $\sum_j \xi_{rj}^n > 0$, i.e. there are pull-messages at r, the routing is determined by χ_r as follows. Let $a = \sum_j \xi_{rj}^n$, $p_j = \xi_{rj}^n / a$. If $\chi_r \in [0, p_1)$, the customer is routed to an idle server associated with r in pool 1; if $\chi_r \in [p_1, p_1 + p_2)$ – it is routed to an idle server associated with r in pool 2; and so on.

Denote

$$f_r^n(w, u) \doteq \frac{1}{n}\sum_{m=1}^{\lfloor nw \rfloor} I\{\xi_r(m) \leq u\},\quad g_r^n(w, u) \doteq \frac{1}{n}\sum_{m=1}^{\lfloor nw \rfloor} I\{\zeta_r(m) \leq u\},$$

where $w \geq 0$, $0 \leq u < 1$. Obviously, from the strong law of large numbers and the monotonicity of $f^n(w, u)$ and $g^n(w, u)$ on both arguments, we have the following FSLLN:

$$f_r^n(w, u) \to wu,\quad g_r^n(w, u) \to wu,\quad \text{u.o.c.},\qquad w.p.1,\quad \forall r \in \mathcal{R}. \tag{15}$$

It is easy (and standard) to see that, for any n, w.p.1, the realization of the process $s^n(\cdot)$ is uniquely determined by the initial state $s^n(0)$ and the realizations of the driving processes $\Pi_r^{(a)}(\cdot)$, $\Pi_{k,j,r}^{(d)}(\cdot)$, $\chi_r(\cdot)$ and $\zeta_r(\cdot)$.

Denote by $A_{k,j,r}^n(t),\ k \in \mathbb{Z}_+,\ t \geq 0$, the total number of arrivals in $[0,t]$ via router r into servers in pool j with queue length k. (Recall that arrivals to infinite queues can be ignored. Also note that, under PULL-1, $A_{k,j,r}^n(t) \equiv 0$ if $k \geq 1$.) Obviously, for any $0 \leq t_1 \leq t_2 < \infty$

$$\sum_j \sum_{0 \leq k < \infty} [A_{k,j,r}^n(t_2) - A_{k,j,r}^n(t_1)] \leq A_r^n(t_2) - A_r^n(t_1), \quad r \in \mathcal{R}.$$

(Note that, under PULL-1,

$$[A_r^n(t_2) - A_r^n(t_1)] - \sum_j [A_{0,j,r}^n(t_2) - A_{0,j,r}^n(t_1)],$$

is the number of arrivals via r in $(t_1, t_2]$ which are blocked.)

We define the fluid-scaled the arrival and departure processes:

$$a_{k,j,r}^n(t) = \frac{1}{n} A_{k,j,r}^n(t), \quad 0 \leq k < \infty,$$

$$d_{k,j,r}^n(t) = \frac{1}{n} D_{k,j,r}^n(t), \quad 1 \leq k < \infty.$$

Definition 7 (Fluid sample path)**.** *A set of uniformly Lipschitz continuous functions $s(\cdot) = [x_{k,j}(\cdot),\ k \in \mathbb{Z}_+,\ j \in \mathcal{J};\ \xi_{r,j}(\cdot),\ r \in \mathcal{R},\ j \in \mathcal{J}]$ on the time interval $[0,\infty)$ we call a* fluid sample path *(FSP), if there exist realizations of the primitive driving processes, satisfying conditions (12), (14) and (15) and a fixed subsequence of n, along which*

$$s^n(\cdot) \to s(\cdot), \quad u.o.c., \tag{16}$$

and also

$$a_{k,j,r}^n(\cdot) \to a_{k,j,r}(\cdot), \quad u.o.c., \quad 0 \leq k < \infty,\ j \in \mathcal{J},\ r \in \mathcal{R}, \tag{17}$$

$$d_{k,j,r}^n(\cdot) \to d_{k,j,r}(\cdot), \quad u.o.c., \quad 1 \leq k < \infty,\ j \in \mathcal{J},\ r \in \mathcal{R}, \tag{18}$$

for some functions $a_{k,j,r}(\cdot)$ and $d_{k,j,r}(\cdot)$.

Given the metric (2) on $\mathcal{S}$, condition (16) is equivalent to component-wise convergence:

$$x_{k,j}^n(\cdot) \to x_{k,j}(\cdot), \quad u.o.c., \quad k \in \mathbb{Z}_+,\ j \in \mathcal{J},$$

$$\xi_{r,j}^n(\cdot) \to \xi_{r,j}(\cdot), \quad u.o.c., \quad r \in \mathcal{R},\ j \in \mathcal{J}.$$

Also, since all functions $a_{k,j,r}^n(\cdot)$ and $d_{k,j,r}^n(\cdot)$ are non-decreasing, the FSP definition easily implies that all limit functions $a_{k,j,r}(\cdot)$ and $d_{k,j,r}(\cdot)$ are necessarily non-decreasing and uniformly Lipschitz continuous.

In what follows, WLOG, we use the convention that each FSP $s(\cdot)$ has an associated set of functions $a_{k,j,r}(\cdot)$ and $d_{k,j,r}(\cdot)$, which are a part of its definition. (A cleaner, but more cumbersome FSP definition should include functions $a_{k,j,r}(\cdot)$ and $d_{k,j,r}(\cdot)$ as FSP components. We do not do this to simplify notation.)

For any FSP $s(\cdot)$, almost all points $t \geq 0$ (w.r.t. Lebesgue measure) are *regular*; namely, all component functions (including $a_{k,j,r}(\cdot)$ and $d_{k,j,r}(\cdot)$) have proper (equal right and left) derivatives. Note that $t = 0$ is *not* a regular point.

Analogously to notation in (8) - (10), we will denote:

$$x_{\infty,j}(t) \doteq \lim_{k \to \infty} x_{k,j}(t)$$

$$y_{k,j}(t) \doteq x_{k,j}(t) - x_{k+1,j}(t), \quad k \in \mathbb{Z}_+,$$
$$y_{\infty,j}(t) \doteq x_{\infty,j}(t) = \lim_{k\to\infty} x_{k,j}(t).$$

For two FSPs $s(\cdot)$ and $\bar{s}(\cdot)$, $s(\cdot) \le \bar{s}(\cdot)$ is defined as $s(t) \le \bar{s}(t),\ t \ge 0$.

Lemma 8. *Consider a sequence in n of processes $s^n(\cdot)$ with deterministic initial states $s^n(0) \to s(0) \in \mathcal{S}$. Then w.p.1 any subsequence of n has a further subsequence, along which*

$$s^n(\cdot) \to s(\cdot) \quad \textit{u.o.c.},$$

where $s(\cdot)$ is an FSP.

Proof. The proof is analogous to that of Lemma 5 in [10]. We give it here for completeness. All processes $a^n_{k,j,r}(\cdot)$ and $d^n_{k,j,r}(\cdot)$ are non-decreasing. W.p.1 the primitive processes satisfy the FSLLN (12), (14) and (15). From here it is easy to observe the following: w.p.1 any subsequence of n has a further subsequence along which the u.o.c. convergences

$$a^n_{k,j,r}(\cdot) \to a_{k,j,r}(\cdot), \quad d^n_{k,j,r}(\cdot) \to d_{k,j,r}(\cdot),$$

hold for all (k,j,r), with the limiting functions $a_{k,j,r}(\cdot)$ and $d_{k,j,r}(\cdot)$ being non-decreasing, uniformly Lipschitz continuous. Then, we use obvious relations between pre-limit process components:

$$x^n_{1,j}(t) - x^n_{1,j}(0) = \sum_r a^n_{0,j,r}(t) - \sum_r d^n_{1,j,r}(t) - \sum_r a^n_{1,j,r}(t), \quad j \in \mathcal{J},$$

$$x^n_{k,j}(t) - x^n_{k,j}(0) = \sum_r a^n_{k-1,j,r}(t) - \sum_r d^n_{k,j,r}(t), \quad j \in \mathcal{J},\ 2 \le k < \infty,$$

$$\xi^n_{r,j}(t) - \xi^n_{r,j}(0) = d^n_{1,j,r}(t) - a^n_{0,j,r}(t), \quad j \in \mathcal{J},\ r \in \mathcal{R}.$$

The result easily follows. □

Lemma 9. *(i) Suppose two FSPs, $s(\cdot)$ and $\bar{s}(\cdot)$, are such that $s(0) \le \bar{s}(0)$, and for some $\tau > 0$ (including possibly $\tau = \infty$) these FSPs are the unique in $[0,\tau)$ for the initial states $s(0)$ and $\bar{s}(0)$, respectively. Then $s(t) \le \bar{s}(t)$ for $0 \le t < \tau$.*
(ii) For an FSP $s(\cdot)$, at any $t \ge 0$ such that

$$\sum_j \xi_{rj}(t) > 0, \forall r \in \mathcal{R}, \tag{19}$$

$(d/dt)x_{k,j}(t)$ and $(d/dt)\xi_{r,j}(t)$ exist (for $t = 0$, right derivatives exist) and

$$\frac{d}{dt}x_{1,j}(t) = \frac{\lambda}{R}\sum_r \frac{\xi_{rj}(t)}{\sum_\ell \xi_{r\ell}(t)} - \mu_j y_{1,j}(t),\ j \in \mathcal{J}, \tag{20}$$

$$\frac{d}{dt}x_{k,j}(t) = -\mu_j y_{k,j}(t) \le 0, \quad 2 \le k < \infty,\ j \in \mathcal{J}, \tag{21}$$

$$\frac{d}{dt}\xi_{rj}(t) = \frac{1}{R}y_{1j}\mu_j(t) - \frac{\lambda}{R}\frac{\xi_{rj}(t)}{\sum_\ell \xi_{r\ell}(t)},\ r \in \mathcal{R},\ j \in \mathcal{J}. \tag{22}$$

(iii) If initial state $s(0)$ of an FSP is such that (19) holds (for $t = 0$) and $\sum_j x_{2,j}(0) = 0$, then the FSP is unique in the interval $[0,\tau)$, where τ is the smallest time t when (19) no longer holds; $\tau = \infty$ if such t does not exist.
(iv) The FSP $s(\cdot)$ with initial condition $s(0) = s^*$ is unique, and it is stationary, i.e. $s(t) \equiv s^*$.
(v) Suppose: $\xi_{rj}(t) = 0,\ \forall r,\ \forall j$; $y_{1j}(t)\mu_j > 0,\ \forall j$; $\sum_j y_{1j}(t)\mu_j > \lambda$. Then, the following right derivatives exist and equal to

$$\frac{d^+}{dt}\xi_{rj}(t) = \frac{1}{R}\left[y_{1j}(t)\mu_j - \frac{y_{1j}(t)\mu_j}{\sum_\ell y_{1\ell}(t)\mu_\ell}\lambda\right],\ r \in \mathcal{R},\ j \in \mathcal{J}. \tag{23}$$

(vi) Consider the system with $B_j = 1$ for all j, and an FSP $\bar{s}(\cdot)$ starting with the maximum initial state, namely $x_{1,j}(0) = \beta_j,\ x_{2,j}(0) = 0$, for all j. Such an FSP $\bar{s}(\cdot)$ is unique, monotone decreasing, $\bar{s}(t_1) \geq \bar{s}(t_2),\ t_1 \leq t_2$, and is such that $\bar{s}(t) \to s^*$.
(vii) For any $\epsilon > 0$, there exist $\tau > 0$ and $\delta > 0$, such that the following holds. If at time $t \geq 0$, $s(t) \geq s^*$, $x_{1,j}(t) = \nu_j$ for all $j \in \mathcal{J}$, and $x_{2,\ell}(t) \geq \epsilon$ for some fixed ℓ, then

$$x_{1,\ell}(\tau) \geq \nu_\ell + \delta.$$

Proof. (i) Consider the sequences of *processes* $s^n(\cdot)$ and $\bar{s}^n(\cdot)$, with fixed initial states such that $s^n(0) \leq \bar{s}^n(0)$, and $s^n(0) \to s(0),\ \bar{s}^n(0) \to \bar{s}(0)$ as $n \to \infty$. By Lemma 5, $s^n(t) \leq_{st} \bar{s}^n(t)$ for all $t \geq 0$. This, Lemma 8, and the FSPs uniqueness in $[0, \tau)$ imply the result.

(ii) Consider a fixed $t \geq 0$, for which (19) holds. If $s^n(\cdot)$ is a sequence of pre-limit trajectories defining FSP $s(\cdot)$, then in a fixed small neighborhood of t, condition $\sum_j \xi^n_{rj}(u) > 0, \forall \mathcal{R}$, holds for all sufficiently large n. This means that, for large n, all new arrivals in that time neighborhood are routed to idle servers; furthermore, the fraction of arrivals to router r that are sent to pool j is close to $\xi_{rj}(t)/[\sum_\ell \xi_{r\ell}(t)]$. Given the fact that FSP components are Lipschitz, and using the FSLLN properties of driving trajectories, we easily obtain (20)-(22).

(iii) From (ii) we in particular have the following. For an FSP $s(\cdot)$, at any $t \geq 0$ such that conditions (19) and

$$x_{2,j}(t) = 0 \ \ [\text{or, equivalently, } y_{1,j}(t) = x_{1,j}(t)], \quad \forall j, \tag{24}$$

hold, we have

$$(d/dt)x_{k,j}(t) = 0, \quad k \geq 2,\ \forall j.$$

Therefore, the condition (24) must hold in the interval $[0, \tau)$. In that interval, $x_{1,j}(t) = y_{1,j}(t) = \beta_j - \sum_r \xi_{rj}$, and therefore (22) becomes

$$\frac{d}{dt}\xi_{rj}(t) = \frac{1}{R}(\beta_j - \sum_{r'} \xi_{r'j}(t))\mu_j - \frac{\lambda}{R}\frac{\xi_{rj}}{\sum_\ell \xi_{r\ell}},\ r \in \mathcal{R},\ j \in \mathcal{J}. \tag{25}$$

We see that, in $[0, \tau)$, the FSP trajectory is uniquely determined by the solution to ODE (25).

(iv) By (ii) and the definition of x^*, $(d/dt)s(t) = 0$ if $s(t) = s^*$. Then we apply (iii).

(v) The proof will be in two steps. Since we consider the right derivatives only, WLOG, to simplify notation, let $t = 0$, and we consider an FSP with time being $u \geq 0$.

Step 1. We will use the following notation: $w_j(u) = y_{1j}(u)\mu_j$. For any r,

$$\frac{d^+}{du}\sum_j \xi_{rj}(0) = \frac{1}{R}\sum_j w_j(0) - \frac{1}{R}\lambda. \tag{26}$$

The meaning of this property is simple: the first term in the RHS is the (fluid-scaled) rate at which new r-pull-messages are generated, and the second term is the rate at which they are "used" (and removed). The formal argument is as follows. The FSP is such that

$$\frac{d^+}{du}\sum_r \sum_j d_{1,j,r}(0) = \sum_j w_j(0);$$

this is the total rate (at $u = 0$) at which new pull-messages of all types are created. The derivative

$$\frac{d^+}{du}\sum_r \sum_j a_{0,j,r}(0) \leq \lambda;$$

this is the total rate (at $u = 0$) at which pull-messages are used. Therefore, in a small interval $[0, \epsilon]$, the FSP is such that, for any $u' \in (0, \epsilon]$, there exists r' for which $\sum_j \xi_{r'j}(u) > 0$ in a small interval $(u' - 2\epsilon', u' + 2\epsilon')$. This means that the pre-limit trajectories, defining this FSP, are such that, for all sufficiently large n, in $(u' - \epsilon', u' + \epsilon')$, the customers arriving at router r will have pull-messages to use, and will be routed only according to pull-messages (as opposed to the random uniform routing to a server in the entire system). Consider the interval $(u' - \epsilon', u' + \epsilon')$. Using analogous argument, we can find that there is a smaller interval $(u' - \epsilon'', u' + \epsilon'') \subset (u' - \epsilon', u' + \epsilon')$ and $r'' \neq r'$, such that $\sum_j \xi_{r''j}(u) > 0$ in $(u' - \epsilon'', u' + \epsilon'')$. Continuing this argument, we obtain that $\sum_j \xi_{rj}(u) > 0, \forall r$, for $u = u'$, and therefore for all $u \in (0, \epsilon]$. In other words, (19) holds in $(0, \epsilon]$, so that (22) holds as well. Summing up (22) over j, we see that for $u \in (0, \epsilon]$ the derivative $\frac{d}{du} \sum_j \xi_{rj}(u)$ is equal to the RHS of (26) with time 0 replaced by u. Then we use the continuity of $y_{1j}(\cdot)$. This proves Step 1.

Step 2. Now we will establish (23), which, using the notations of this proof, can be written as

$$\frac{d^+}{du}\xi_{rj}(0) = b_j \equiv \frac{1}{R}\left[w_j(0) - \frac{w_j(0)}{\sum_\ell w_\ell(0)}\lambda\right] = \frac{w_j(0)}{R}\left[\frac{\sum_\ell w_\ell(0) - \lambda}{\sum_\ell w_\ell(0)}\right]. \tag{27}$$

By (26), condition (19) holds in a sufficiently small interval $(0, \epsilon]$, $\epsilon > 0$, so that (22) holds as well. Moreover, in $(0, \epsilon]$, $w_j(u) > 0$ for any j. Then, from (22) we see that $\xi_{rj}(u) > 0$ for $u \in (0, \epsilon]$, any j and any r. (If $\xi_{rj}(u) = 0$ for some $u \in (0, \epsilon]$, then by (22) $(d/du)\xi_{rj}(u) > 0$ at that u, which is impossible.)

Fix any j and r. Consider function

$$g_{rj}(u) = \ln[\xi_{rj}(u)/(b_j u)].$$

For $u \in (0, \epsilon]$,

$$\frac{d}{du}\xi_{rj}(u) = \frac{1}{R}w_j(u) - \frac{\lambda}{R}\frac{\xi_{rj}(u)}{\sum_\ell \xi_{r\ell}(u)},\ j \in \mathcal{J}.$$

We can write

$$g'_{rj}(u) = \frac{\xi'_{rj}(u)}{\xi_{rj}(u)} - \frac{1}{u} = \frac{\xi'_{rj}(u) - \xi_{rj}(u)/u}{\xi_{rj}(u)} =$$
$$\frac{\frac{1}{R}w_j(u) - \frac{\lambda}{R}\frac{\xi_{rj}(u)/u}{\sum_\ell \xi_{r\ell}(u)/u} - \xi_{rj}(u)/u}{\xi_{rj}(u)}.$$

Now, using the Lipschitz continuity of the FSP, the facts that $w_j(0) > 0$, $\sum_\ell w_\ell(0) - \lambda > 0$, $\xi_{rj}(u)/u$ is bounded (because $\xi_{rj}(\cdot)$ is Lipschitz), and (26), we obtain

$$g'_{rj}(u) \sim \frac{\frac{1}{R}w_j(0) - \lambda\frac{\xi_{rj}/u}{\sum_\ell w_\ell(0) - \lambda} - \xi_{rj}(u)/u}{\xi_{rj}(u)} = \frac{\frac{1}{R}w_j(0)[\sum_\ell w_\ell(0) - \lambda] - [\sum_\ell w_\ell(0)]\xi_{rj}(u)/u}{[\sum_\ell w_\ell(0) - \lambda]\xi_{rj}(u)},$$

or, using definition of b_j in (27) and a little further algebra in the RHS,

$$g'_{rj}(u) \sim \frac{[1 - \frac{\xi_{rj}(u)}{b_j u}]b_j \sum_\ell w_\ell(0)}{[\sum_\ell w_\ell(0) - \lambda]\xi_{rj}(u)}. \tag{28}$$

Since FSP is Lipschitz, $\xi_{rj}(u) \leq cu,\ u \geq 0$, for some $c > 0$. Consequently, $g_{rj}(u) \leq c'$ and $\xi_{rj}(u)/(b_j u) \leq c'$ for all $u > 0$. We obtain the following property. For any $\delta > 0$, we can choose $\epsilon > 0$ small enough, so that for any $u \in (0, \epsilon]$, condition $g_{rj}(u) \geq \delta$ implies $g'_{rj}(u) \leq -c_1/(c_2 u)$, where the constant c_1 depends on δ, and constant c_2 does not. Since $g_{rj}(u) \leq c',\ u > 0$, and $\int_0^u (1/u')du' = \infty$, we must have $g_{rj}(u) \leq \delta$ for all $u \in (0, \epsilon]$. Since this holds for an arbitrarily small $\delta > 0$, we obtain

$$\limsup_{u \downarrow 0} g_{rj}(u) \leq 0, \quad \text{or, equivalently,} \quad \limsup_{u \downarrow 0} \frac{\xi_{rj}(u)}{u} \leq b_j. \tag{29}$$

However, (26) means

$$\lim_{u \downarrow 0} \frac{\sum_j \xi_{rj}(u)}{u} = \sum_j b_j.$$

This implies that (27) must hold, which completes Step 2.

(vi) By (v), condition (19) holds for $\bar{s}(\cdot)$ at least in some small interval $(0, \epsilon]$, and it is unique in that interval. (This is because (v) gives unique initial condition for the ODE.) Moreover, $\bar{s}(\cdot)$ is unique up to the first time $\tau > 0$, when condition (19) no longer holds; at this point in the proof we can only claim that $\tau > 0$ – but it could be infinite or finite. Now, for each n, consider the *processes* $s^n(\cdot)$, starting from the maximum initial state. By Corollary 6, this process is monotone non-increasing. Let $n \to \infty$. Then, using Lemma 8, process $s^n(\cdot)$ monotonicity, and FSP $\bar{s}(\cdot)$ uniqueness in $[0, \tau)$, we obtain that FSP $\bar{s}(\cdot)$ must be monotone non-increasing $[0, \tau)$. But then τ cannot be finite. We proved that FSP $\bar{s}(\cdot)$ is unique, monotone non-increasing in the entire interval $[0, \infty)$.

Since $\bar{s}(\cdot)$ is non-increasing, as $t \to \infty$, $\bar{s}(t) \to s^{**}$ for some s^{**}. Let us show that $s^{**} = s^*$. First, since $\bar{s}(0) \geq s^*$, and the FSPs starting from those initial states are unique, by (i) we must have $\bar{s}(t) \geq s^*$ for all $t \geq 0$. Therefore, $s^{**} \geq s^*$. Furthermore, it is easy to see from the structure of the ODE that governs $\bar{s}(\cdot)$ that $s^{**} \neq s^*$ is impossible; otherwise, $\sum_j \bar{x}_{1j}(t) \downarrow \sum_j x^{**}_{1j} > \sum_j x^*_{1j}$, which would imply that $(d/dt) \sum_j \bar{x}_{1j}$ must stay negative bounded away from zero for all t.

(vii) From (ii) and definition of ν_j, using relation $y_{1,j}(t) = x_{1,j}(t) - x_{2,j}(t)$, we have

$$(d/dt)x_{1,j}(t) = \mu_j x_{2,j}(t), \quad j \in \mathcal{J}.$$

(For $t = 0$ it is the right derivative.) Also from (ii), we observe that in a sufficiently small fixed neighborhood of time t, the expression for the derivative $(d/ds)x_{1,\ell}(s)$ must be uniformly Lipschitz continuous. This implies that, for an arbitrarily small $\epsilon_1 > 0$, in a (further reduced) small neighborhood t, $(d/ds)x_{1,\ell}(s) \geq \mu_j \epsilon - \epsilon_1$; which in turn implies the desired property. □

5 System with unit buffer sizes

Here we consider a special system where all buffer sizes are equal to 1, i.e. $B_j = 1$ for all j. The process $s^n(\cdot)$ is automatically positive recurrent (Lemma 2).

Lemma 10. *Suppose $B_j = 1$ for all j. Then*

$$s^n(\infty) \Rightarrow s^*. \tag{30}$$

Proof. Since space $\mathcal{S}$ is compact, any subsequence of n has a further subsequence, along which

$$s^n(\infty) \Rightarrow s^\circ(\infty), \tag{31}$$

where $s^\circ(\infty)$ is a random element in $\mathcal{S}$. We need to prove that

$$s^\circ(\infty) = s^*, \quad w.p.1. \tag{32}$$

For each n, consider the process $s^n(\cdot)$, starting from the maximum initial state: $\xi^n_{rj} = 0$ for all j and r. By Corollary 6, process $s^n(\cdot)$ is monotone non-increasing. Consider any fixed pair (r, j). Fix arbitrary $\epsilon > 0$, and choose $T > 0$ large enough so that the FSP $\bar{s}(\cdot)$ starting from the maximum initial condition (as in Lemma 9(vi)) is such that $\bar{\xi}_{rj}(T) \geq \xi^*_{rj} - \epsilon/2$. Then, by Lemma 8, $\mathbb{P}\{\xi^n_{rj}(T) \geq \xi^*_{rj} - \epsilon\} \to 1$. From here and the fact that $s^n(\cdot)$ is monotone non-increasing, we obtain

$$\liminf_{n\to\infty} \mathbb{P}\{\xi^n_{rj}(\infty) \geq \xi^*_{rj} - \epsilon\} \geq \liminf_{n\to\infty} \mathbb{P}\{\xi^n_{rj}(T) \geq \xi^*_{rj} - \epsilon\} = 1.$$

Therefore, since $\{\xi_{rj} \geq \xi^*_{rj} - \epsilon\}$ is a closed set,

$$\mathbb{P}\{\xi^\circ_{rj}(\infty) \geq \xi^*_{rj} - \epsilon\} \geq \limsup_{n\to\infty} \mathbb{P}\{\xi^n_{rj}(\infty) \geq \xi^*_{rj} - \epsilon\} \geq 1.$$

This holds for any $\epsilon > 0$, so we have $\mathbb{P}\{\xi^\circ_{rj}(\infty) \geq \xi^*_{rj}\} = 1$ for any pair (r, j). This is equivalent to (recall that here we consider system with $B_j = 1$ for all j)

$$s^\circ(\infty) \leq_{st} s^*. \tag{33}$$

Note that (33) implies $x^\circ_j(\infty) \leq x^*_j = \nu_j, \ \forall j$, w.p.1. To prove lemma, it remains to show that, in fact, (32) holds.

For a system with fixed parameter n, the (fluid-scaled) steady-state rate at which customers enter service (equal to the rate at which they leave service) is lower bounded by

$$\sum_r (\lambda/R)\mathbb{P}\{\sum_j \xi^n_{rj}(\infty) > 0\},$$

which by (33) converges to λ as $n \to \infty$. But, this rate is obviously upper bound by λ. Therefore, for the steady-state rate at which customers leave service, we must have

$$\lim_{n\to\infty} \mathbb{E}\sum_j \mu_j x^n_j(\infty) = \lambda.$$

But,

$$\lim_{n\to\infty} \mathbb{E}\sum_j \mu_j x^n_j(\infty) = \mathbb{E}\sum_j \mu_j x^\circ_j(\infty).$$

We conclude that

$$\mathbb{E}\sum_j \mu_j x^\circ_j(\infty) = \lambda. \tag{34}$$

Suppose now that (33) holds, but (32) does not; namely, for at least one j, we have $\mathbb{P}\{x^\circ_j(\infty) < \nu_j\} > 0$. This is impossible, because then $\mathbb{E}\sum_j \mu_j x^\circ_j(\infty) < \sum_j \mu_j \nu_j = \lambda$, which contradicts (34). This proves (32). □

6 Proof of Theorem 3

Recall that we consider a system with fixed buffer sizes B_j, some of which are infinite. Denote by $\mathcal{J}_\infty$ the subset of pools j with $B_j = \infty$.

6.1 Instability characterization.

Here we consider a system with the scaling parameter n being fixed. System *instability* means the originally defined process (i.e., with original – not compactified – state space) is not ergodic (i.e., not positive recurrent).

Lemma 11. *Suppose the system is unstable. Then there exists $j \in \mathcal{J}_\infty$ and $\delta > 0$ such that the following holds. For any $C > 0$, there exists $T = T(C) > 0$ such that, uniformly on all initial states, for any $i \in \mathcal{N}_j$,*

$$\mathbb{P}\{Q^n_i(T) \geq C\} \geq \delta. \tag{35}$$

Proof. It suffices to prove that the lemma statement holds uniformly on all *idle* initial states (i.e., such that $Q^n_i(0), \ \forall i$). This follows from the fact that any initial state dominates *one of the idle states*, and from Lemma 5. Furthermore, for any $T_0 > 0$, uniformly on the idle initial states, with probability at least some $\delta_0 > 0$, the system in time T_1 makes transition to the state where all queues are equal to 1: $Q^n_i(T_0) = 1, \ \forall i$. Therefore, it suffices to prove the lemma statement for the single initial condition: $Q^n_i(0) = 1, \ \forall i$; this is the initial condition we assume from now on.

The system is unstable, which means that $\sum_i Q_i^n \stackrel{P}{\to} \infty$. Then, for any $C > 0$, we can pick a sufficiently large $T > 0$, such that

$$\mathbb{P}\{\sum_i Q_i^n(T) \geq Cn\} \geq 1/2.$$

But,

$$\mathbb{P}\{\sum_i Q_i^n(T) \geq Cn\} \leq \sum_i \mathbb{P}\{Q_i^n(T) \geq C\}.$$

Then, there exists j and $i \in \mathcal{J}$ for which $\mathbb{P}\{Q_i^n(T) \geq C\} \geq \delta = 1/(2n)$; then, by symmetry, this is true for all $i \in \mathcal{J}$. This is true for a sequence of $C \uparrow \infty$ and the corresponding sequence of $T = T(C)$; we choose a subsequence, for which the corresponding j is same. Clearly, this $j \in \mathcal{J}_\infty$. Therefore, the desired property holds for this j and $\delta = 1/(2n)$. □

Lemma 12. *Suppose the system is unstable. Consider $j \in \mathcal{J}_\infty$ picked as in Lemma 11. Then for some $\delta > 0$ and for any $C > 0$ there exists $T = T(C) > 0$ such that, uniformly on all initial states,*

$$\mathbb{P}\{\min_{i \in \mathcal{N}_j} Q_i^n(T) \geq C\} \geq \delta. \tag{36}$$

Proof. The proof is by induction on subsets of $\mathcal{N}_j$. Suppose the lemma statement holds for subsets $\hat{\mathcal{N}}_j \subset \mathcal{N}_j$ of cardinality at most $\kappa < |\mathcal{N}_j|$. Namely, for some $\delta > 0$ and for any $C > 0$ there exists $T = T(C) > 0$ such that, uniformly on all initial states, for any subset $\hat{\mathcal{N}}_j \subset \mathcal{N}_j$ with $|\hat{\mathcal{N}}_j| \leq \kappa$,

$$\mathbb{P}\{\min_{i \in \hat{\mathcal{N}}_j} Q_i^n(T) \geq C\} \geq \delta. \tag{37}$$

Note that by Lemma 11 this is true for $\kappa = 1$; this is the induction base. Pick any $\hat{\mathcal{N}}_j \subset \mathcal{N}_j$ with $|\hat{\mathcal{N}}_j| = \kappa$, and an element $m \in \mathcal{N}_j \setminus \hat{\mathcal{N}}_j$. Since there only a finite number of subsets of $\mathcal{N}_j$, the induction step will be completed if we show that for some $\delta' > 0$ and for any $C > 0$ there exists $T' = T'(C) > 0$ such that, uniformly on all initial states,

$$\mathbb{P}\{\min_{i \in \hat{\mathcal{N}}_j \cup \{m\}} Q_i^n(T') \geq C\} \geq \delta'. \tag{38}$$

Indeed, pick δ, arbitrarily large C and the corresponding $T = T(C)$, so that (38) holds for the subset $\hat{\mathcal{N}}_j$. Pick any $\epsilon \in (0, \delta)$. For the same δ, we can pick $C_1 > 0$ so large that, if $Q_m^n(0) \geq C_1$ then

$$\mathbb{P}\{Q_m^n(T) \geq C\} \geq 1 - \epsilon.$$

We can and do pick $T_1 = T_1(C_1)$, so that (35) holds with $i = m$, $T = T_1$, and $C = C_1$.

With arbitrary fixed inital state, we see that $\mathbb{P}\{Q_m^n(T_1) \geq C_1\} \geq \delta$. We also have

$$\mathbb{P}\{\min_{i \in \hat{\mathcal{N}}_j} Q_i^n(T_1 + T) \geq C \mid Q_m^n(T_1) \geq C_1\} \geq \delta$$

and

$$\mathbb{P}\{Q_m^n(T_1 + T) \geq C \mid Q_m^n(T_1) \geq C_1\} \geq 1 - \epsilon.$$

Therefore,

$$\mathbb{P}\{\min_{i \in \hat{\mathcal{N}}_j \cup \{m\}} Q_i^n(T_1 + T) \geq C\} \geq \delta(\delta - \epsilon).$$

This proves the induction step, with $\delta' = \delta(\delta - \epsilon)$, and $T' = T_1 + T$ chosen, as a function of C, as described above. □

Lemma 13. *Suppose the system is unstable. Consider $j \in \mathcal{J}_\infty$ picked as in Lemma 11 (and Lemma 12). Then, uniformly on all initial states,*

$$\min_{i \in \mathcal{N}_j} Q_i^n(t) \stackrel{P}{\to} \infty, \text{ as } t \to \infty. \tag{39}$$

Proof. First, we show that if the statement of Lemma 12 holds for some fixed $\delta > 0$, then it must hold for any $\delta \in (0,1)$. To show this, it suffices to verify the following fact: if the statement of Lemma 12 holds for some fixed $\delta > 0$, then it also holds for $\delta' = \delta(1-\epsilon) + (1-\delta)\delta = \delta(2-\delta-\epsilon)$ with arbitrarily small fixed $\epsilon > 0$. We will do just that.

Fix arbitrary initial state. Fix $\epsilon > 0$. Pick δ as in Lemma 12, then pick arbitrarily large C and the corresponding $T = T(C)$, so that (36) holds. Choose $C_1 > 0$ so large that, if $\min_{i\in\mathcal{N}_j} Q_i^n(0) \geq C_1$ then

$$\mathbb{P}\{\min_{i\in\mathcal{N}_j} Q_i^n(T) \geq C\} \geq 1-\epsilon.$$

Then, again using Lemma 12, choose $T_1 = T_1(C_1)$ so that (36) holds with C and T replaced by C_1 and T_1, respectively. We have

$$\mathbb{P}\{\min_{i\in\mathcal{N}_j} Q_i^n(T_1) \geq C_1\} \geq \delta.$$

Considering the system state at time $T_1 + T$, conditioning on the event $\{\min_{i\in\mathcal{N}_j} Q_i^n(T_1) \geq C_1\}$ occurring or not, we obtain

$$\mathbb{P}\{\min_{i\in\mathcal{N}_j} Q_i^n(T_1+T) \geq C\} \geq \delta(1-\epsilon) + (1-\delta)\delta = \delta'.$$

Thus, the statement of Lemma 12 indeed holds, with δ rechosen to be δ', and function $T(C)$ rechosen to be $T'(C)$. Consequently, the statement of Lemma 12 holds for any $\delta > 0$.

Then, the statement of this lemma easily follows, because for any $\delta \in (0,1)$ and arbitrarily large $C > 0$ we can choose $T = T(C)$ such that, for any initial state

$$\mathbb{P}\{\min_{i\in\mathcal{N}_j} Q_i^n(t) \geq C\} \geq \delta$$

for $t = T$. But then this is also true for all $t \geq T$. □

Consider $j \in \mathcal{J}_\infty$ picked as in Lemma 11 (and Lemmas 12 and 13). Consider a variant of the system, such that all queues in pool j are infinite, $Q_i^n(t) \equiv \infty$, $i \in \mathcal{N}_j$, and buffer sizes in all other pools are equal to 1, $B_\ell = 1$, $\ell \neq j$. Recall that if a queue size is infinite at time 0, it remains infinite forever. Therefore, the process for this specific system, let us denote it $\hat{S}^n(\cdot)$, is a Markov chain with a *finite* state space. It has unique stationary distribution; then $\hat{S}^n(\infty)$ is its random state in stationary regime. Since the state space is finite, the convergence to the stationary distribution is uniform on the initial states:

$$\lim_{t\to\infty} \max_{\hat{S}^n(0)} \|Dist[\hat{S}^n(t)] - Dist[\hat{S}^n(\infty)]\| = 0.$$

Lemma 14. *Suppose the system is unstable. Consider $j \in \mathcal{J}_\infty$ picked as in Lemma 11 (and Lemmas 12 and 13). Let an arbitrary initial state $S^n(0)$ be fixed. Consider a random element $S^n(\infty)$, which is any limit in distribution of $S^n(t)$ along a subsequence of $t \to \infty$. (Recall that our process is viewed as having the compact state space, with possibly infinite queues. Then, any subsequence of $t \to \infty$ has a further subsequence, along which $S^n(t)$ converges in distribution.) Then, $\hat{S}^n(\infty) \leq_{st} S^n(\infty)$.*

Proof. It will suffice to prove the following property; let us label it (P1):
For any $\epsilon_1 > 0$, $\epsilon_2 > 0$ and any $C > 0$, there exists $T' = T'(C)$ such that the following holds uniformly on all initial states $S^n(0)$. Process $S^n(\cdot)$ can be constructed on a common probability space with another process $\tilde{S}^n(\cdot)$, so that there exists an event G with $\mathbb{P}\{G\} \geq 1-\epsilon_1$, such that

$$\mathbb{P}\{\min_{i\in\mathcal{N}_j} Q_i^n(T') \geq C \mid G\} = 1,$$

$$\tilde{S}^n(T') \text{ is dominated by } S^n(T') \text{ on the subset of servers } \mathcal{N}\setminus\mathcal{N}_j, \quad \text{on the event } G,$$

$$\|Dist[\tilde{Q}^n(T')] - Dist[\hat{Q}^n(\infty)]\| \leq \epsilon_2.$$

We proceed with proving property (P1). Let us first choose $T_2 > 0$ sufficiently large so that, uniformly on the initial states $\hat{S}^n(0)$ of the process $\hat{S}^n(\cdot)$,

$$\|Dist[\hat{Q}^n(T_2) - Dist[\hat{Q}^n(\infty)]\| \leq \epsilon_2. \tag{40}$$

Then, we choose $C_1 > C$, sufficiently large so that if $\min_{i\in\mathcal{N}_j} Q_i^n(0) \geq C_1$ then with probability at least $1-\epsilon_1/2$, $\min_{i\in\mathcal{N}_j} \min_{0\leq t\leq T_2} Q_i^n(t) \geq C$. Then we choose $T_1 = T_1(C_1) > 0$ sufficiently large so that, uniformly on the initial states $S^n(0)$, the event

$$\{\min_{i\mathcal{N}_j} Q_i^n(T_1) \geq C_1\}$$

occurs with probability at least $1 - \epsilon_1/2$. We choose $T' = T_1 + T_2$ and event G to be

$$G = \{\min_{i\in\mathcal{N}_j} Q_i^n(T_1) \geq C_1, \quad \min_{i\in\mathcal{N}_j} \min_{T_1\leq t\leq T'} Q_i^n(t) \geq C\}.$$

We construct the process $\tilde{S}^n(\cdot)$ as follows. In the time interval $[0, T_1)$, $\tilde{S}^n(\cdot)$ coincides with $S^n(\cdot)$. At time T_1 we change the model for the process $\tilde{S}^n(\cdot)$ by keeping buffer sizes in pool j to be infinite, $\tilde{B}_j = \infty$, and making buffer sizes in all other pools to be equal to 1, $\tilde{B}_\ell = 1$ for $\ell \neq j$; we also change all queue sizes in pool j to be infinite, $\tilde{Q}_i^n(T_1) = \infty,\ i \in \mathcal{N}_j$, and leaving at most one customer in each queue in the other pools,

$$\tilde{Q}_i^n(T_1) = 1 \wedge \tilde{Q}_i^n(T_1-),\ i \in \mathcal{N}_\ell,\ \ell \neq j.$$

Starting time T_1, the process $\tilde{S}^n(\cdot)$ is coupled to the process $S^n(\cdot)$, as described in Lemma 5, until the random time $T_1 \leq \tau \leq \infty$ when $\min_{i\in\mathcal{N}_j} Q_i^n(t) = 0$ for the first time after T_1. Obviously, $\tau > T'$ on the event G. It remains to observe that in the interval $[T_1, \tau)$, the process $S^n(\cdot)$ dominates $\tilde{S}^n(\cdot)$ on the subset $\mathcal{N} \setminus \mathcal{N}_j$ of servers. □

Observe that $\breve{S}^n(\infty) \leq_{st} \hat{S}^n(\infty)$, where $\breve{S}^n(\cdot)$ is the process for the system with $B_j = 1$ for all j, considered in Lemma 10. Let $\breve{s}^n(\infty)$ and $\hat{s}^n(\infty)$ be the random states (in stationary regime) for the corresponding mean-field (or, fluid-scaled) processes. Then, we obtain the following mean-field version of Lemma 14.

Lemma 15. *Suppose the system is unstable. Consider $j \in \mathcal{J}_\infty$ picked as in Lemma 11 (and Lemmas 12 – 14). Let an arbitrary initial state $s^n(0)$ be fixed. Consider a random element $s^n(\infty)$, which is any limit in distribution of $s^n(t)$ along a subsequence of $t \to \infty$. Then, $\breve{s}^n(\infty) \leq_{st} \hat{s}^n(\infty) \leq_{st} s^n(\infty)$.*

Finally, note that Lemma 10 proves that, $\breve{s}^n(\infty) \Rightarrow s^*$ as $n \to \infty$.

6.2 Theorem 3 proof.

For each n, consider the process, starting from any fixed initial state $s^n(0)$. We have

$$\limsup_{T\to\infty} (1/T) \int_0^T [\mathbb{E} \sum_\ell \mu_j x_{1\ell}^n(t)]dt \leq \lambda, \tag{41}$$

because the LHS is the lim sup of the time-averaged expected (fluid-scaled) number of customer service completions over the interval $[0, T]$, which cannot exceed the limit of the time-averaged expected (fluid-scaled) number of customer arrivals.

Suppose the system is unstable for infinitely many n. This leads to a contradiction as follows. Consider an infinite subsequence of those n, for which the system is unstable. Moreover, we choose, if necessary, a further subsequence, along which we can apply Lemma 15 with the same value of j. Namely, for each such n – and same fixed j – choose a sequence of values of time T increasing to infinity (this sequence may depend on n), along which $s^n(T) \Rightarrow s^n(\infty)$, where $\breve{s}^n(\infty) \leq_{st} \hat{s}^n(\infty) \leq_{st} s^n(\infty)$. Along this sequence of T,

$$\lim_{T\to\infty} (1/T) \int_0^T [\mathbb{E} \sum_\ell \mu_j x_{1\ell}^n(t)]dt = \mathbb{E} \sum_\ell \mu_\ell x_{1\ell}^n(\infty). \tag{42}$$

Recall that $\breve{s}^n(\infty) \Rightarrow s^*$ as $n \to \infty$. This implies that if we consider any subsequential (in n) distributional limit $\hat{s}^n(\infty) \Rightarrow \hat{s}(\infty)$, it is such that $\hat{x}_{1\ell}(\infty) \geq \nu_\ell$ for all j; but, from the definition of process $\hat{s}^n(\cdot)$, $\hat{x}_{1j}(\infty) = \beta_j$. We see that, as $n \to \infty$, the RHS in (42) converges to $\mu_j\beta_j + \sum_{\ell \neq j} \mu_\ell \nu_\ell > \lambda$. This contradicts the fact that (41) must hold for each n. $\square$

7 Proof of Theorem 4

Space $\mathcal{S}$ is compact. Therefore, any subsequence of n has a further subsequence, along which

$$s^n(\infty) \Rightarrow s^\circ(\infty), \tag{43}$$

where $s^\circ(\infty)$ is a random element in $\mathcal{S}$. Therefore, to prove Theorem 4 it suffices to show that any limit in (43) is equal to s^* w.p.1.

From Lemma 10, we obtain the following Corollary 16, because $s^n(\infty)$ for a system with arbitrary buffer sizes $B_j \geq 1$ stochastically dominates that for the system with $B_j = 1$ for all j, and therefore this relation holds also for the limit $s^\circ(\infty)$.

Corollary 16. *Any subsequential limit $s^\circ(\infty)$ in (43) is such that*

$$s^* \leq s^\circ(\infty), \quad w.p.1.$$

Proof of Theorem 4. Consider any subsequential limit $s^\circ(\infty)$ in (43), along a subsequence of n; for the rest of the proof, we consider this subsequence. By Corollary 16,

$$\mathbb{E} \sum_j \mu_j x^\circ_{1,j}(\infty) \geq \lambda.$$

On the other hand, for any n, the (fluid-scaled) steady-state rate of service completions in the system is

$$\mathbb{E} \sum_j \mu_j x^n_{1,j}(\infty) \leq \lambda;$$

this implies

$$\mathbb{E} \sum_j \mu_j x^\circ_{1,j}(\infty) = \lim_{n\to\infty} \mathbb{E} \sum_j \mu_j x^n_{1,j}(\infty) \leq \lambda.$$

Therefore,

$$\mathbb{E} \sum_j \mu_j x^\circ_{1,j}(\infty) = \lambda,$$

which (by Corollary 16) implies

$$x^\circ_{1,j}(\infty) = \nu_j, \ j \in \mathcal{J}, \quad w.p.1. \tag{44}$$

Now, (44) means that, w.p.1 for each j, $\sum_r \xi^\circ_{rj}(\infty) = \sum_r \xi^*_{rj}$, which alond with Corollary 16 implies

$$\xi^\circ_{rj}(\infty) = \xi^*_{rj}, \quad j \in \mathcal{J}, \ r \in \mathcal{R}, \quad w.p.1.$$

It remains to show that

$$x^\circ_{2,j}(\infty) = 0, \ j \in \mathcal{J}, \quad w.p.1. \tag{45}$$

Suppose not, namely, for at least one ℓ, $\mathbb{P}\{x^\circ_{2,\ell}(\infty) > \epsilon\} = 2\epsilon_1$, for some $\epsilon > 0$, $\epsilon_1 > 0$. Then, for all sufficiently large n, $\mathbb{P}\{x^n_{2,\ell}(\infty) > \epsilon\} > \epsilon_1$. For each sufficiently large n, consider a stationary version of $s^n(\cdot)$, that is $s^n(t) \stackrel{d}{=} s^n(\infty)$ for all $t \geq 0$. Then $\mathbb{P}\{x^n_{2,\ell}(0) > \epsilon\} > \epsilon_1$. Now, employing Lemma 8 and Lemma 9(vii), we can easily show that, for some $\tau > 0$, $\delta > 0$, and all large n,

$$\mathbb{P}\{x^n_{1,\ell}(\tau) \geq \nu_\ell + \delta/2\} > \epsilon_1/2.$$

But then

$$\mathbb{P}\{x^\circ_{1,\ell}(\infty) \geq \nu_\ell + \delta/2\} \geq \limsup_{n\to\infty} \mathbb{P}\{x^n_{1,\ell}(\tau) \geq \nu_\ell + \delta/2\} \geq \epsilon_1/2,$$

a contradiction with (44), which proves (45). $\square$

8 Conclusions

The main conclusion of this paper is that a pull-based approach for load distribution can be applied in large-scale heterogeneous systems with multiple independent routers. We proposed (two versions of) a specific algorithm PULL – along with a specific mechanism for its implementation – and proved its asymptotic optimality. PULL algorithm in this paper is a generalization of the single-router algorithm in [10]. The generalization of the algorithm itself is very natural, but a priori it is far from obvious that it is provably optimal for the multi-router model. In fact, to make our proofs work, the version PULL-2 of the algorithm is such that, in addition to pull-messages, it requires occasional "pull-remove-messages" to keep the pull-messages at the routers "up to date." This additional mechanism potentially increases the router-server message exchange rate to 2/message; however, as our results show, in the large-scale asymptotic limit, pull-remove-messages are never used, and therefore the limiting exchange rate is only 1/message, same as in the single-router case. We conjecture that the additional pull-remove-message mechanism is *not* necessary for PULL-2 asymptotic optimality to hold. However, its absence substantially complicates the analysis. Verifying this conjecture may be one of the subjects of future work.

References

[1] Badonnel, R. and Burgess, M. (2008). Dynamic pull-based load balancing for autonomic servers. *Network Operations and Management Symposium, NOMS 2008*, 751–754.

[2] Bramson, M., Lu, Y., and Prabhakar, B. (2012). Asymptotic independence of queues under randomized load balancing. *Queueing Systems 71*, 247–292.

[3] Bramson, M., Lu, Y., and Prabhakar, B. (2013). Decay of tails at equlibrium for fifo join the shortest queue networks. *The Annals of Applied Probability 23*, 1841–1878.

[4] Eschenfeldt, P. and Gamarnik, D. (2015). Join the shortest queue with many servers. the heavy traffic asymptotics. arXiv:1502.00999.

[5] Liggett, T. M. (1985). *Interacting Particle Systems*. Springer.

[6] Lu, Y., Xie, Q., Kliot, G., Geller, A., Larus, J., and Greenberg, A. (2011). Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services. *Performance Evaluation 68*, 1057–1071.

[7] Mitzenmacher, M. (2001). The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems* **12**, 10, 1094–1104.

[8] Mukherjee, D., Borst, S., van Leeuwaarden, J., and Whiting, P. (2015). Universality of load balancing schemes on diffusion scale. arXiv:1510.02657.

[9] Stolyar, A. L. (2015a). Large-scale heterogeneous service systems with general packing constraints. arXiv:1508.07512.

[10] Stolyar, A. L. (2015b). Pull-based load distribution in large-scale heterogeneous service systems. *Queueing Systems* **80**, 4, 341–361.

[11] Vvedenskaya, N., Dobrushin, R., and Karpelevich, F. (1996). Queueing system with selection of the shortest of two queues: an asymptotic approach. *Problems of Information Transmission* **32**, 1, 20–34.