FCFS parallel service systems and matching models

FCFS Parallel Service Systems and Matching Models

Ivo Adan[a], Igor Kleiner[b,1], Rhonda Righter[c], Gideon Weiss[b,1,*]

[a] *Eindhoven University of Technology*
[b] *Department of Statistics, The University of Haifa, Mount Carmel 31905, Israel*
[c] *University of California at Berkeley*

Abstract

We consider three parallel service models in which customers of several types are served by several types of servers subject to a bipartite compatibility graph, and the service policy is first come first served. Two of the models have a fixed set of servers. The first is a queueing model in which arriving customers are assigned to the longest idling compatible server if available, or else queue up in a single queue, and servers that become available pick the longest waiting compatible customer, as studied by Adan and Weiss, 2014. The second is a redundancy service model where arriving customers split into copies that queue up at all the compatible servers, and are served in each queue on FCFS basis, and leave the system when the first copy completes service, as studied by Gardner et al., 2016. The third model is a matching queueing model with a random stream of arriving servers. Arriving customers queue in a single queue and arriving servers match with the first compatible customer and leave immediately with the customer, or they leave without a customer. The last model is relevant to organ transplants, to housing assignments, to adoptions and many other situations.

We study the relations between these models, and show that they are closely related to the FCFS infinite bipartite matching model, in which two infinite sequences of customers and servers of several types are matched FCFS according to a bipartite compatibility graph, as studied by Adan et al., 2017. We also introduce a directed bipartite matching model in which we embed the queueing systems. This leads to a generalization of Burke's theorem to parallel service systems.

Keywords: parallel service queueing systems; FCFS; redundancy service; infinite matching.

---

1. Introduction

We consider three parallel service models in which customers of several types, indexed by $c_i \in \mathcal{C} = \{c_1, \dots, c_I\}$ are served by several types of servers, indexed by $s_j \in \mathcal{S} = \{s_1, \dots, s_J\}$, subject to a bipartite compatibility graph, $\mathcal{G} = (\mathcal{S}, \mathcal{C}, \mathfrak{E})$, $\mathfrak{E} \subseteq \mathcal{S} \times \mathcal{C}$, such that $(s_j, c_i) \in \mathfrak{E}$ if customer type c_i can be served by server type s_j. We focus on the first come first served (FCFS) policy in all the models, i.e. customers are prioritized by their order of arrival, and servers are prioritized by the order in which they become available.

---

[*] Corresponding author

Email addresses: `iadan@win.tue.nl` (Ivo Adan), `igkleiner@gmail.com` (Igor Kleiner), `rrighter@ieor.berkeley.edu` (Rhonda Righter), `gweiss@stat.haifa.ac.il` (Gideon Weiss)

[1] Research supported in part by Israel Science Foundation Grant 286/13.

Two of the models have a fixed set of servers, while the third model has a random stream of arriving servers. Briefly stated the models are as follows:

- *FCFS-ALIS Parallel Queueing Model:* There are J servers of types $\mathcal{S}$ and a stream of customers of types $\mathcal{C}$. An arriving customer is assigned to the longest idle server which is compatible with it (ALIS - assign longest idle server) if such is available, or else it joins the queue of waiting customers. A server that completes a service picks up the longest waiting customer which is compatible with it (FCFS), if such is available, or else it joins the queue of idle servers. This model was studied by Adan and Weiss [1].
- *A Redundancy Service Model:* There are J servers of types $\mathcal{S}$, each with his FCFS queue, and a stream of arriving customers of types $\mathcal{C}$. An arriving customer splits upon arrival into several copies that join the queues of the servers which are compatible with it. Service of a customer can then proceed simultaneously at several compatible servers. The customer and all its copies leave the system when the first of its copies completes service. This model was studied by Gardner et al. [2].
- *A Parallel FCFS Matching Queue:* There is an arrival stream of customers of types $\mathcal{C}$, and an independent arrival stream of servers of types $\mathcal{S}$. When a customer arrives it joins a queue of customers waiting for service. When a server arrives it scans the queue of customers and matches with the longest waiting customer that is compatible with its type, and the matched customer then leaves the system with the server. If the server does not find a match it leaves immediately without a match.

  The matching queue model is relevant to many types of service systems: It can describe organ transplants, where patients are waiting to receive organs, and donated organs arrive in a random stream, and organs are assigned to compatible recipients in FCFS order, or are lost if no compatible recipient is waiting [3]. It can also describe an adoption process, where families are waiting for available babies to be adopted (this may only be approximate since unmatched babies do not disappear). It was used to model assignment of project houses to families in Boston public housing, by Kaplan [4, 5]. Another application is to call centers with inbound and outbound calls, where differently skilled agents (servers) start outbound calls if there are no waiting inbound calls that match their skill sets. Here the state would be the set of customers waiting in the queue, and would not include those in service. Our matching queue model, although it seems very relevant to the study of organ transplants and to various other systems, has not, to the best of our knowledge, been analyzed in any level of detail.

We assume Poisson arrivals and exponential server-dependent service times for all three models so that their evolution is Markovian and can be described by various discrete-space continuous-time Markov chains.

These models are closely related to a fourth model:

- *The FCFS infinite bipartite matching model:* This was introduced in [6, 7] and studied in more detail recently by Adan, Busic, Mairesse and Weiss [8]. In this model there are two infinite sequences, drawn independently, one from $\mathcal{C}$, the other from $\mathcal{S}$, and the two sequences are then matched FCFS according to the compatibility graph $\mathcal{G}$. This model is much simpler than either of the above models since it does not involve arrival times and service times, and servers and customers play a completely symmetric role.

In this paper we explore the relations among the three service models, and their connections to the FCFS infinite matching model. Our results here are:

- The continuous-time Markov chains that describe all three service models share the same stationary distribution. This leads the way to comparing their performance measures.
- We note that the redundancy service model and the matching queue are equivalent, in that they share the same continuous-time Markov chain.
- We compare the performance of the Redundancy Service model and the FCFS-ALIS model, and consider when either should be preferred. In particular we study their performance for the 'N'-system (see Figure 5, Section 4), and obtain sharp stochastic bounds on the difference in the number in queue for each policy.
- We introduce a new discrete FCFS infinite matching model, which we call the FCFS infinite directed matching model, that is similar to the model of [8].
- We derive properties of this new FCFS infinite directed matching model.
- We embed the three service models in the infinite directed bipartite matching model.
- We obtain a version of Burke's Theorem for the redundancy service and for the matching queue systems.

For related work, see Ayesta et al., [9].

The rest of the paper is structured as follows: In Section 2 we describe the three models, and in Section 3 we compare their performance. In Section 4 we study the performance of the 'N'-system under FCFS-ALIS and under Redundancy Service, and present computational and simulation results. In Section 5 we describe the relevant properties of the FCFS infinite bipartite matching model. In Section 6 we introduce the new FCFS infinite directed matching model, and derive properties of the process. In Section 7 we show how to embed the three service models in this new matching model, and discover some surprising consequences of this embedding. We complete the proofs of our results in appendices.

Notation

We let $\mathcal{S}(c_i)$ denote the subset of server types compatible with c_i, and $\mathcal{C}(s_j)$ denote the subset of customer types compatible with s_j. For $C \subset \mathcal{C}$, $S \subset \mathcal{S}$ we let $\mathcal{S}(C) = \bigcup_{c_i \in C} \mathcal{S}(c_i)$, $\mathcal{C}(S) = \bigcup_{s_j \in S} \mathcal{C}(s_j)$, and denote by $\mathcal{U}(S) = (\mathcal{C}(S^c))^c$ those customer types that are compatible only with server types in S.

We associate with c_i a rate λ_{c_i}, and with s_j a rate μ_{s_j}; these are rates for exponential distributions. We also let $\bar{\lambda} = \sum_{i=1}^{I} \lambda_{c_i}$ and $\bar{\mu} = \sum_{j=1}^{J} \mu_{s_j}$. For subsets $C \subset \mathcal{C}$, $S \subset \mathcal{S}$ we let $\lambda_C = \sum_{c_i \in C} \lambda_{c_i}$, $\mu_S = \sum_{s_j \in S} \mu_{s_j}$.

In what follows we will denote quantities related to the FCFS-ALIS model by a superscript ^{q}, those related to the Redundancy Service model by a superscript ^{r}, and those related to the Matching model by a superscript ^{m}. In addition, we denote quantities related to the FCFS infinite bipartite matching model by a superscript $^\infty$, and those related to the FCFS infinite directed matching model by a superscript $^{\downarrow\infty}$.

2. The Service Models

2.1. A stability condition

Theorem 2.1. *All three service models are stable, in the sense that Markov chains describing them are ergodic, if and only if the following condition holds:*

$$\lambda_C < \mu_{S(C)} \text{ for every } C \subseteq \mathcal{C}. \tag{1}$$

Proof. The proof for the FCFS-ALIS system is given in [1] Theorem 2.1. The proof for the redundancy system is given in [2], Theorem 1. The proof for the matching system follows from Theorem 3.1 later in this paper.

The conditions for stability can also be verified from the solutions to the balance equations, given in the following Theorems 2.2–2.5. It is easy to see that the solutions converge if and only if (1) holds. □

Figure 1 illustrates the compatibility graph for an example we will use throughout the paper. In this example there are 3 types of customers and 3 types of servers, customers of type c_2 (type c_3) can only be served by server of type s_2 (type s_3), while customers of type c_1 can be served by all types of servers. A variant of this model (without s_1) is referred to in the literature as the 'W'-model.

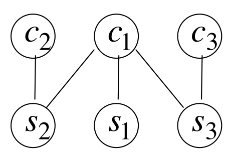

Figure 1: Compatibility graph for customer and server types

The stability condition for this example is:

$$\lambda_2 < \mu_2, \quad \lambda_3 < \mu_3, \quad \bar{\lambda} < \bar{\mu}.$$

2.2. The FCFS-ALIS parallel queueing model

Customers arrive in independent Poisson streams, with rate λ_{c_i} for type c_i. There are J servers of types $\{s_1, \ldots, s_J\}$, and service by server s_j is exponential with rate μ_{s_j}. The service policy as described in the introduction is FCFS-ALIS. Figure 2 illustrates a possible state for our example. In this figure all customers

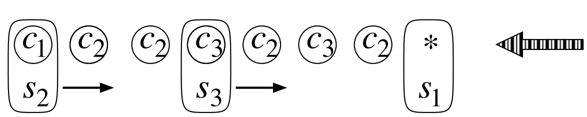

Figure 2: A current state under FCFS-ALIS

in the system are displayed in order of arrival, with earlier arrivals more to the left. Customers in service are shown together with their server. The oldest customer in the system is of type c_1 and it is served by server s_2, server s_3 is serving a customer of type c_3 after skipping two incompatible customers of type c_2. Server s_1 is idle. In the future, new customers will arrive from the right and join the end of the queue, with or without a compatible server, and on completion of service a customer departs, and the server moves to the right and scans waiting customers until he find a compatible customer or else he joins the end of the queue of idle servers .

In [1] the system is described by the process $Y^q(t) = (S_1, n_1, \ldots, S_i, n_i, S_{i+1}, \ldots, S_J)$ where $S_1, \ldots, S_J$ is a permutation of the servers, servers $S_1, \ldots, S_i$ are busy with S_1 serving the oldest customer in the system, S_2 has skipped n_1 customers and is serving the second oldest customer currently in service, and so on. n_j is the number of skipped customers between S_j and S_{j+1}. The remaining servers, $S_{i+1}, \ldots, S_J$ are idle, ordered by length of time they were idle, with S_J the longest idle. This describes the system at time t. Adan and Weiss proved:

Theorem 2.2 (Adan and Weiss [1])**.** *The process $Y^q(t)$ is a continuous-time discrete state Markov chain. It is ergodic if and only if (1) holds. Its stationary distribution is given, up to a normalizing constant, by*

$$P^q(S_1, n_1, \ldots, S_i, n_i, S_{i+1}, \ldots, S_J) \propto \prod_{j=1}^{i} \frac{(\lambda_{\mathcal{U}(\{S_1,\ldots,S_j\})})^{n_j}}{(\mu_{\{S_1,\ldots,S_j\}})^{n_j+1}} \times \prod_{j=i+1}^{J} \frac{1}{\lambda_{\mathcal{C}(\{S_j,\ldots,S_J\})}} \tag{2}$$

Adan and Weiss [1] also calculated the normalizing constant.

We introduce an alternative process to describe the system, $X^q(t) = (c^1, c^2, \ldots, c^L, s^1, \ldots, s^K)$ where c^ℓ is the random type of the ℓth oldest customer in the system that is waiting and has not started service yet, and s^k is the type of the kth longest idling server in the system, all this at time t. Note that L, the number of waiting customers, corresponds to $n_1 + \cdots + n_i$ of $Y^q(t)$, and can take any value ≥ 0, while $s^1, \ldots, s^K$ correspond to $S_J, \ldots, S_{i+1}$ of $Y^q(t)$, which is the ordered subset of idle servers, with no replications, so that $K \leq J$. We then have:

Theorem 2.3. *The process $X^q(t)$ is a continuous-time discrete state Markov chain. It is ergodic if and only if the stability condition (1) holds. Its stationary distribution is given, up to a normalizing constant, by:*

$$P^q(c^1, c^2, \ldots, c^L, s^1, \ldots, s^K) \propto \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \times \prod_{k=1}^{K} \frac{\mu_{s^k}}{\lambda_{\mathcal{C}(\{s^1,\ldots,s^k\})}} \tag{3}$$

The proof of this theorem, by partial balance, appears in Appendix A. In particular, the following corollary is immediate:

Corollary 1. *The process $X^q(t)$, conditional on the event that all servers are busy, has the stationary distribution, up to a normalizing constant, given by:*

$$P^q(c^1, c^2, \ldots, c^L \mid \textit{all busy}) \propto \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \tag{4}$$

2.3. The redundancy service model

There are servers $s_1, \ldots, s_J$, and each of them has its own FCFS queue of compatible customers, and service by server s_j is exponential with rate μ_{s_j}. Customers arrive in independent Poisson streams, with rate

λ_{c_i} for customers of type c_i. Each arriving customer, upon arrival, splits into copies of the same type, and one copy joins the queue of each of the servers with which it is compatible. Service of a customer can then be performed at several compatible servers simultaneously. The customer departs from the system, with all its copies, at the instant at which service of one of its copies is complete.

Figure 3 illustrates a possible state for our example. In it we display the list of customer types, in order of

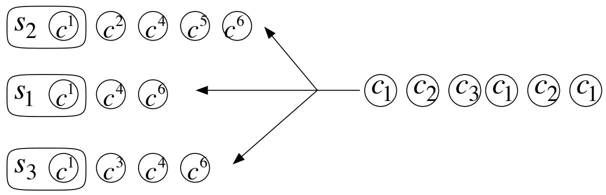

Figure 3: A current state with redundant queueing

arrival, on the right side, and on the left side are the servers and their queues. The first customer, c^1 (where the superscript ^{1} indicates his place in the sequence of customers in the system) is of type c_1, and is currently being served simultaneously be all three servers. The second and third customers are of types c_2 and c_3 and queue up for servers s_2, s_3 respectively. The fourth and sixth customer, c^4, c^6 are again of type c_1 and queue up at all three servers.

Gardner et al. [2] have studied this system and defined the following process to describe it: $X^r(t) = (c^1, \ldots, c^L)$, where $c^1, \ldots, c^L$ are the types of all the customers in the system at time t, ordered by their arrival times, with c^1 the oldest. They have shown:

Theorem 2.4 (Gardner, Zbarsky, Doroudi, Harchol-Balter, Hyytia and Scheller-Wolf [2])**.** *The process $X^r(t)$ is a continuous-time discrete state Markov chain. It is ergodic if and only if the stability condition (1) holds. Its stationary distribution is given, up to a normalizing constant, by:*

$$P^r(c^1, c^2, \ldots, c^L) \propto \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \tag{5}$$

2.4. The FCFS parallel matching queue

Customers of various types arrive in independent Poisson streams of rates λ_{c_i} and queue up in order of arrival. Servers of various types arrive in independent Poisson streams of rates μ_{s_j}. An arriving server scans the queue of customers and matches with the longest waiting customer that is compatible with it, and the two leave the system immediately. If the server does not find a compatible customer in the queue it leaves immediately without a customer.

Figure 4 illustrates a possible history of this system, for our example. The figure shows a sequence of

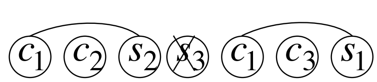

Figure 4: A partial history of the matching queue

customers and servers specified by their types, ordered in the order of arrival from left to right. A customer of type c_1 arrived first, followed by a customer of type c_2. Next a server of type s_2 arrived and was immediately matched to the first customer and they departed together. Next a server of type s_3 arrived and left immediately without a match. This was followed by a customer of type c_1, then a customer of type c_3 and finally by a server of type s_1 that matched immediately with the third customer, of type c_1, and both departed. At this point in time there was a queue of two customers, the earlier of type c_2, the later of type c_3.

We describe this system by the process $X^m(t) = (c^1, \ldots, c^L)$, where there are L customers in total, their types (random) are $c^1, \ldots, c^L$, ordered in order of arrival, with c^1 the longest waiting, and the time is t.

Theorem 2.5. *The process $X^m(t)$ is a Markov chain. It is ergodic if and only if the stability condition (1) holds, and its stationary distribution is given, up to a normalizing constant, by:*

$$P^m(c^1, \ldots, c^L) \propto \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \tag{6}$$

The proof of this theorem is identical to the proof of Theorem 2.4. It also follows directly from Theorem 3.1.

3. Comparison of the three service models

As we see from Theorems 2.4, 2.5 and Corollary 1, all three parallel service systems are associated with a Markov chain with the same stationary distribution. Furthermore this stationary distribution is similar to that of the FCFS infinite matching model of [6, 7, 8] that we will describe in Section 5. We now explore the relations among these models.

3.1. Equivalence of the redundancy service system and the matching queue

Note that although the redundancy system can have idle servers, and the matching queue cannot, the state of the redundancy system is completely determined by the sequence of customers in the system; servers are idle at a given time if there are no compatible customers in the system at that time. We will show that the matching and redundancy queues are sample-path equivalent in the sense that if we start them with the same customer state, and we couple the customer arrival processes in the two queues, and we couple potential service completions in the redundancy queue with service arrivals in the matching queue, then the sample paths for the state processes of the two systems will be identical, with probability one.

Theorem 3.1. *The redundancy service system and the matching queue are equivalent in the sense that the processes $X^r(t)$ and $X^m(t)$ are sample path equivalent. In particular this means that for each customer, the sojourn time in the system is the same for both models, so the mean sojourn times are the same for each customer type.*

Proof. Consider the situation at time t where the customers in the both systems, in order of arrival, are of types $c^1, \ldots, c^L$, i.e. $X^r(t) = X^m(t) = (c^1, \ldots, c^L)$. To show that the two systems have the same Markov chain, we will show that all the transition rates are the same. Transitions are of two kinds, arrivals or departures. An arrival of a customer of type c_i will occur at rate λ_{c_i}, and this customer then joins the queue as $c^{L+1} = c_i$ in both systems.

Consider now departures. We will compare the departure rates of each of the customers currently in the system. If $\mathcal{S}(c^\ell)\backslash\mathcal{S}(c^1 \ldots, c^{\ell-1}) = \emptyset$ then the departure rate for c^ℓ in both systems will be 0: in the redundancy system, all the servers that can serve c^ℓ are currently serving customers that are earlier in the queue (some of the earlier customers may be served by several servers simultaneously), and in the matching system, any arrival of a server that can match with c^ℓ will in fact match to an earlier customer. On the other hand, if $s_j \in \mathcal{S}(c^\ell)\backslash\mathcal{S}(c^1 \ldots, c^{\ell-1}) \neq \emptyset$, then in the redundancy system, customer c^ℓ will be at the head of the line for server s_j and will be in service, and in the matching system an arrival of a server of type s_j will result in matching with c^ℓ and its departure. Hence we have shown that departure of customer c^ℓ occurs in both systems at rate $\mu_{\mathcal{S}(c^\ell)\backslash\mathcal{S}(c^1 \ldots, c^{\ell-1})}$.

This completes the proof. Note also that the total rate of departures in both systems is $\mu_{\mathcal{S}(c^1,\ldots,c^\ell)}$. In the redundancy system servers $\mathcal{S}(c^1, \ldots, c^\ell)$ are all busy, while the remaining servers are idle, and in the matching system any arrival of a server of any type in $\mathcal{S}(c^1, \ldots, c^\ell)$ will result in a matching and a departure, while an arrival of a server of type $s_j \notin \mathcal{S}(c^1, \ldots, c^\ell)$ will not result in a departure.

□

3.2. Comparing the FCFS-ALIS and the redundancy service systems

In contrast, the situation is different when we compare the FCFS-ALIS system with the redundancy system. We list some points for comparison:

- The process $X^q(t)\big|$all busy and $X^r(t)$ have the same stationary distribution, but $X^r(t)$ includes all customers in the system, those waiting and those being served, while $X^q(t)$ only includes waiting customers, so there is an additional set of customers which are currently in service in the FCFS-ALIS system.

  It is in fact shown in [10] that the stationary distribution of the types of customers that are in service in the FCFS-ALIS system cannot be expressed in product form, even for the simple 'N' compatibility graph.
- One can regard the FCFS-ALIS system also as a system in which customers split on arrival into several copies that queue up at all the compatible servers, as in the redundancy queue. However, at the instant that service of one copy starts, all the other copies disappear. This happens either when the customer has been waiting at several queues, and reaches the server in one of these queues, or when on arrival finds several compatible servers, in which case it will be processed by the longest idle server, so there is no simultaneous processing.
- It is worth mentioning that the FCFS-ALIS system is equivalent to a system in which customers have full information about all the processing times in the system, and each arriving customer joins the compatible server with the smallest workload. This join the smallest workload policy (JSW) leads to a Nash equilibrium determined by selfish customers.
- With the same set of customers $c^1, \ldots, c^L$, and the same set of idle servers $s^1, \ldots, s^K$ in the system, under FCFS-ALIS each busy server serves a different customer, while in the redundancy system different servers

may serve the same customer simultaneously. Therefore, although the stationary distributions of $X^q(t)|\text{busy}$ and $X^r(t)$ are the same, they are not sample path equivalent.

- Because all processing times are exponentially distributed, there is no loss of processing time when a customer is served simultaneously by more than one server. In fact, if a set of servers are processing jobs, the next service completion will be at the same time whether they work on different customers or are processing the same customer simultaneously.
- If in the two systems there is the same set of customers (both waiting and in service), then the number of busy servers in the redundancy system is greater than or equal to the number of busy servers in the FCFS-ALIS systems, because simultaneous service is allowed under the redundancy system.
- Under the Redundancy service policy flexible customers have an advantage over less flexible customers. As a result, the composition of customers in the system under Redundancy service may include more inflexible and fewer flexible customers than under FCFS-ALIS policy. This may result in forced idleness when too many inflexible customers accumulate, and there are no flexible customers left in the system.

The last two considerations indicate that comparison of performance of the two service policies may depend on the parameters of the system, such as workloads and service rates. In the next section we take a closer look at this question through a detailed study of the special case of the 'N'-system.

4. A comparison of FCFS-ALIS and Redundancy Service for the 'N'-System

In this section, we compare the performance of the *FCFS-ALIS* policy and the *Redundancy Service* policy for the 'N'-system. In the comparison of the expected sojourn times and number of customers in steady state under the two policies, we find that neither policy dominates the other. We then consider a coupled realization of both systems, and analyze how the sample paths under the two policies differ, and prove a theorem on the difference. We also present some simulation results that illustrate typical behavior in light traffic and in heavy traffic.

The 'N'-System is illustrated in figure 5. There are two servers and two customer types. Type 1 customers arrive at rate λ_1 and are flexible, and can be served by either server, type 2 customers arrive at rate λ_2 and can only be served by server 2. Server 2 is flexible and can serve both types of customers, at rate μ_2, while server 1 can only serve type 1 customers, at rate μ_1. Recall that $\bar{\lambda} = \sum_{i=1}^{I} \lambda_{c_i}$ and $\bar{\mu} = \sum_{j=1}^{J} \mu_{s_j}$. We assume that $\lambda_2 < \mu_2$ and $\bar{\lambda} < \bar{\mu}$.

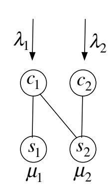

Figure 5: The 'N'-system

The sojourn time for the 'N'-system under the Redundancy Service policy is derived in Theorems 2 and 3 of [2]. From this we obtain the expected sojourn times for type 1 and type 2 customers:

$$E(W_1^r) = \frac{1}{\bar{\mu} - \bar{\lambda}}, \qquad E(W_2^r) = \frac{1}{\mu_2 - \lambda_2} - \frac{1}{\bar{\mu} - \lambda_2} + \frac{1}{\bar{\mu} - \bar{\lambda}}.$$

The expected waiting times and the service times for type 1 and type 2 customers under the FCFS-ALIS policy can be calculated using results of section 4 in [10] (see also [1]). Using these results we obtain first the expected waiting times V_1^q, V_2^q, and then the expected service times S_1^q, S_2^q. The waiting times are:

$$E(V_1^q) = B \tfrac{1}{(\bar{\mu}-\bar{\lambda})^2} \left(\tfrac{1}{\mu_1} + \tfrac{1}{\mu_2-\lambda_2} \right),$$

$$E(V_2^q) = B \left(\tfrac{1}{\lambda_1(\mu_2-\lambda_2)^2} + \tfrac{1}{\mu_1(\bar{\mu}-\bar{\lambda})^2} + \tfrac{1}{(\mu_2-\lambda_2)^2(\bar{\mu}-\bar{\lambda})} + \tfrac{1}{(\mu_2-\lambda_2)(\bar{\mu}-\bar{\lambda})^2} \right)$$

where

$$B = \left(\frac{1}{\lambda_1 \bar{\lambda}} + \frac{1}{\bar{\lambda}^2} + \frac{1}{\mu_1 \bar{\lambda}} + \frac{1}{\lambda_1(\mu_2 - \lambda_2)} + \frac{1}{\mu_1(\bar{\mu} - \bar{\lambda})} + \frac{1}{(\mu_2 - \lambda_2)(\bar{\mu} - \bar{\lambda})} \right)^{-1}.$$

To calculate the service times of customers of type 1, we note that the total stationary probability that servers 1 and 2 are busy, denoted here as b_1^q, b_2^q, are given [1] by:

$$b_1^q = P(\text{server 1 busy}) = B \left(\tfrac{1}{\mu_1} \tfrac{1}{\bar{\lambda}} + \tfrac{1}{\mu_1} \tfrac{1}{\bar{\mu}-\bar{\lambda}} + \tfrac{1}{\mu_2-\lambda_2} \tfrac{1}{\bar{\mu}-\bar{\lambda}} \right),$$

$$b_2^q = P(\text{server 2 busy}) = B \left(\tfrac{1}{\mu_2-\lambda_2} \tfrac{1}{\lambda_1} + \tfrac{1}{\mu_1} \tfrac{1}{\bar{\mu}-\bar{\lambda}} + \tfrac{1}{\mu_2-\lambda_2} \tfrac{1}{\bar{\mu}-\bar{\lambda}} \right).$$

and therefore the stationary probability that server 2 is working on customer of type 1 is

$$P(\text{server 2 working on a customer of type 1}) = b_2^q - \frac{\lambda_2}{\mu_2}.$$

From these we get expressions for the expected service times of the customers in steady state:

$$E(S_1^q) = \frac{b_1^q}{\mu_1 b_1^q + \mu_2(b_2^q - \frac{\lambda_2}{\mu_2})} + \frac{b_2^q - \frac{\lambda_2}{\mu_2}}{\mu_1 b_1^q + \mu_2(b_2^q - \frac{\lambda_2}{\mu_2})}, \qquad E(S_2^q) = \frac{1}{\mu_2}.$$

The expected number of customers in the system can now also be obtained, by Little's Law. This enables us to compare expected sojourn times and number in system under the two policies. In the following Figure 6 we plot the difference in expected number in system under the two policies. It is seen from the plots that one policy does not always dominate the other.

To learn more about the behavior of the 'N'-system under the two policies, we now study coupled versions of the two systems, which we define as follows: We have 4 independent Poisson processes of rates $\lambda_1, \lambda_2, \mu_1, \mu_2$ which are shared by the two systems. The first two give the arrival times of the two types of customers. The second two give the sequences of potential service completions of the two servers. When a potential service is completed at either of the systems, if a customer is in service at that server, the customer leaves. In the system with redundant service, if both servers are serving the same customer, then that customer will leave if either server has a potential service completion. If the server is idle at a service completion there is no change

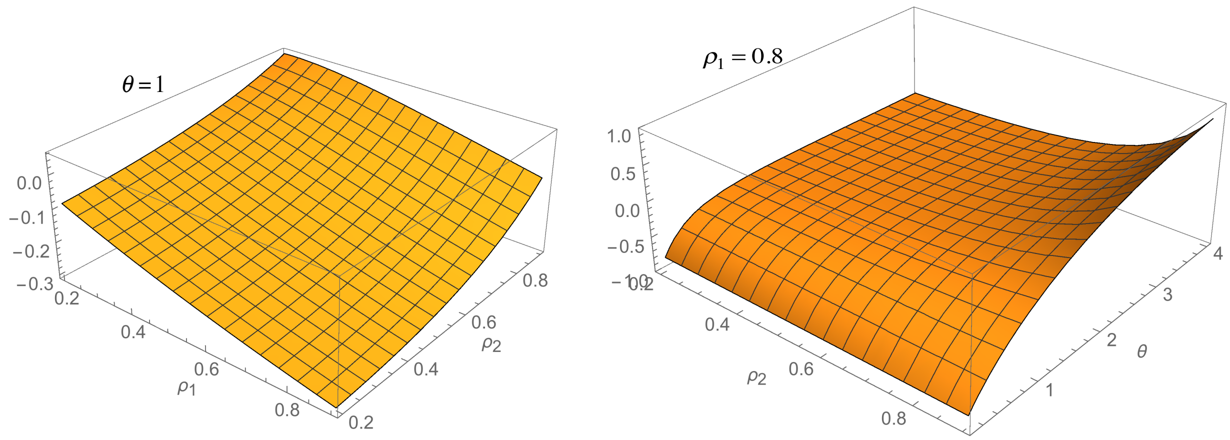

Figure 6: Plots of $E(N_1^r + N_2^r - N_1^q - N_2^q)$ as a function of $\rho_1 = \frac{\lambda_1}{\mu_1}$, $\rho_2 = \frac{\lambda_2}{\mu_2}$, $\theta = \frac{\mu_1}{\mu_2}$

in the state. This coupling means that arrival times and potential service completions occur at the same times in both systems, but we may have, for a potential service completion, that the type of customer that leaves is different, or even that under one policy there is a departure, and under the other policy there is no departure.

Let N^q, N^r denote the random variables counting the total number of customers in each of the coupled systems, when the systems are in steady state, and let N_1^q, N_1^r, N_2^q, N_2^r, count the number of customers of each type.

Theorem 4.1. *For the coupled systems in steady state, the number of customers in system under FCFS-ALIS can exceed the number of customers in system under redundancy service by no more than 1. On the other hand, the number of customers in system under redundancy service minus the the number of customers in system under FCFS-ALIS can be unboundedly large. This holds for the total number as well as for the number of customers of each type. Stated in terms of the distributions,*

$$N^r \geq_{ST} N^q - 1, \qquad N_1^r \geq_{ST} N_1^q - 1, \qquad N_2^r \geq_{ST} N_2^q - 1. \tag{7}$$

The proof of this theorem is based on coupling, and is given in Appendix D.

It is instructive to explain under what conditions either policy may be advantageous. In light traffic, when the servers are not overloaded, the redundant system may be preferred, since there will be many occasions when there is a single type 1 customer in the system, in which case under FCFS-ALIS one of the servers is idle, while under Redundancy Service both servers are working.

This is illustrated in the following coupled sample realization in Figure 7. The parameters for this example are:

$$\text{Example 1:} \quad \lambda_1 = 3,\ \lambda_2 = 2,\ \mu_1 = 6,\ \mu_2 = 6, \qquad E(N_1^r + N_2^r) = 1.014, \quad E(N_1^q + N_2^q) = 1.194.$$

We plot the number of customers $N_1^r + N_2^r - N_1^q - N_2^q$. We see that for an appreciable fraction of the time the difference equals -1.

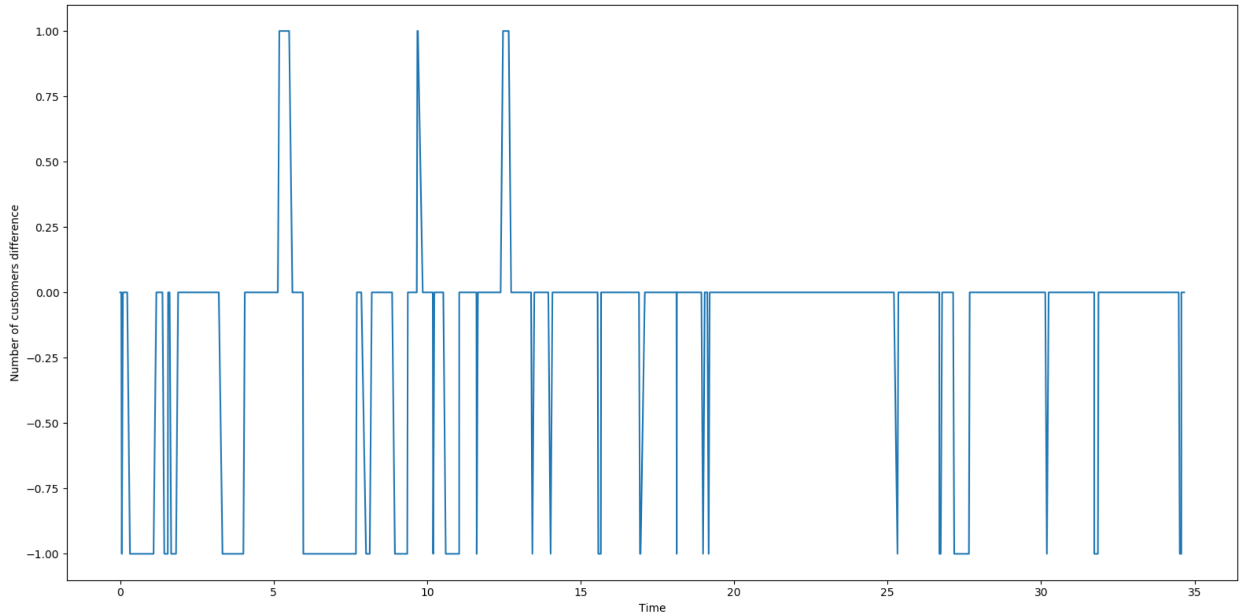

Figure 7: Example 1: Difference in the number of customers for: $\lambda_1 = 3, \lambda_2 = 2, \mu_1 = 6, \mu_2 = 6$

On the other hand, when the flexible server 2 is heavily loaded by inflexible customers of type 2, then whenever server 2 is "helping" server 1 by serving a type 1 customer under Redundancy Service, customers of type 2 accumulate, and so redundancy can have more congestion than FCFS-ALIS. This is clearly illustrated in the following coupled sample realization in Figure 8.

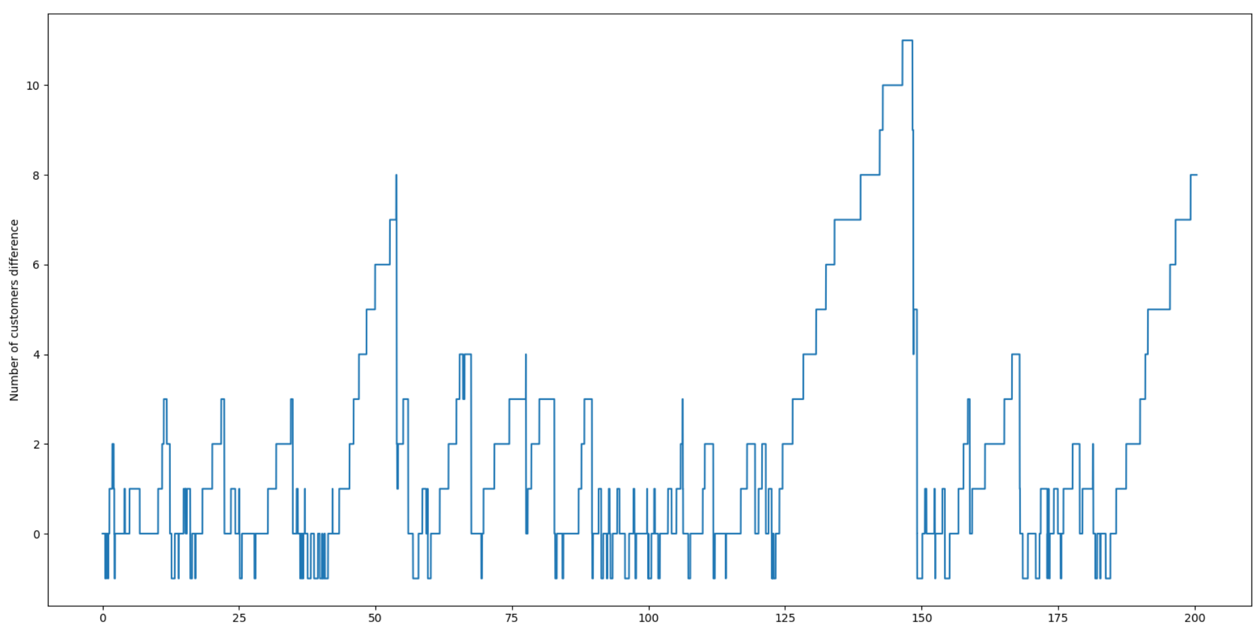

Figure 8: Example 2: Difference in the number of customers for: $\lambda_1 = 2, \lambda_2 = 145, \mu_1 = 3, \mu_2 = 150$

Example 2: $\lambda_1 = 2$, $\lambda_2 = 145$, $\mu_1 = 3$, $\mu_2 = 150$, $\quad E(N_1^r + N_2^r) = 35.375$, $\quad E(N_1^q + N_2^q) = 32.4993$.

We see that the difference is positive most of the time and can be as high as 10. Note that in this case, if we use dedicated service of customers of type 1 by server 1 only, and customers of type 2 by server 2 only, then $E(N_1) = 2$, $E(N_2) = 29$, i.e., flexibility hurts performance. Of course, under an optimal scheduling policy (as opposed to FCFS), flexibility can only help. Gardner et al. [11] show that the optimal scheduling policy prioritizes less flexible (type 2) customers.

5. The FCFS Infinite Bipartite Matching Model

We note that all three service models have similar stationary distributions. Furthermore, these distributions are also similar to stationary distributions that were obtained for the infinite bipartite matching model introduced by Kaplan, Caldentey and Weiss [6] and further studied in [7, 8]. We briefly describe the results of [7, 8].

There are two independent doubly infinite series of customers $\ldots, c^{-2}, c^{-1}, c^0, c^1, c^2, \ldots$ drawn i.i.d. from $\mathcal{C}$ according to the probabilities α, and of servers $\ldots, s^{-2}, s^{-1}, s^0, s^1, s^2, \ldots$, drawn i.i.d. from $\mathcal{S}$ according to the probabilities β, and they are matched FCFS according to the compatibility graph $\mathcal{G}$.

What we mean by FCFS is that if s^n is matched with c^m, then there is no earlier (unmatched) $s^k \in \mathcal{S}(c^m)$ which is unmatched, and no earlier $c^\ell \in \mathcal{C}(s^n)$ which is unmatched. Figure 9 illustrates FCFS infinite bipartite matching with the compatibility graph of Figure 1, for a window of the sequences. In this figure one customer

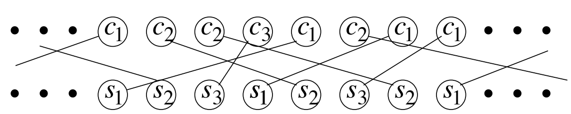

Figure 9: FCFS infinite bipartite matching

and one server were matched to an earlier (left of the window) customer and server, and one customer and one server remain unmatched at the end of the window, and are matched to a later (right of the window) customer and server.

Definition 1. *We say that this system has complete resource pooling if the following equivalent conditions hold for any $S \subset \mathcal{S}$, $S \neq \emptyset, \mathcal{S}$ and $C \subset \mathcal{C}$, $C \neq \emptyset, \mathcal{C}$:*

$$\alpha_C < \beta_{\mathcal{S}(C)}, \qquad \beta_S < \alpha_{\mathcal{C}(S)}, \qquad \alpha_{\mathcal{U}(S)} < \beta_S. \tag{8}$$

The following theorem was proved by Adan et al. [8]:

Theorem 5.1 (Adan, Busic, Mairesse and Weiss [8])**.** *If complete resource pooling (8) holds then almost surely there exists a FCFS matching of the two sequences (such that no customers or servers are unmatched) and this matching is unique.*

We define the following transformation on the matched sequences:

Definition 2. *For given matched sequences, the exchange transformation exchanges the position, or index, of each matched pair, so that if s^n was matched to c^m in the original system, then in the exchanged system we have $\tilde{c}^n$ matched to $\tilde{s}^m$. This defines a permutation of the original sequence $\ldots, c^{-2}, c^{-1}, c^0, c^1, c^2, \ldots$ to a new sequence $\ldots, \tilde{c}^{-2}, \tilde{c}^{-1}, \tilde{c}^0, \tilde{c}^1, \tilde{c}^2, \ldots$, and the original sequence $\ldots, s^{-2}, s^{-1}, s^0, s^1, s^2, \ldots$ to a new sequence $\ldots, \tilde{s}^{-2}, \tilde{s}^{-1}, \tilde{s}^0, \tilde{s}^1, \tilde{s}^2, \ldots$.*

Figure 10 illustrates the exchanged sequences obtained by the exchange transformation from the illustration in Figure 9.

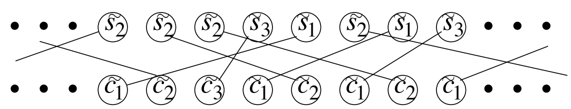

Figure 10: The exchange transformation applied to Figure 9

The following reversibility result is proved in [8]

Theorem 5.2 (Adan, Busic, Mairesse and Weiss [8])**.** *The exchanged sequences $\ldots, \tilde{c}^{-2}, \tilde{c}^{-1}, \tilde{c}^0, \tilde{c}^1, \tilde{c}^2, \ldots$, $\ldots, \tilde{s}^{-2}, \tilde{s}^{-1}, \tilde{s}^0, \tilde{s}^1, \tilde{s}^2, \ldots$, are independent and each is i.i.d. from $\mathcal{C}$ and from $\mathcal{S}$ according to α, β. Furthermore, the original matching is now the almost surely unique FCFS matching of the exchanged sequences in reversed time.*

Using the reversibility, it is easy to obtain stationary distributions for several discrete time Markov chains associated with this system. We consider making all the FCFS matches of s^ℓ, c^k for $k, \ell \le n$, and define the process $X^\infty(n) = (c^{i_1}, \ldots, c^{i_L}, s^{j_1}, \ldots, s^{j_L})$, where customers in positions $i_1, \ldots, i_L$ and servers in positions $j_1, \ldots, j_L$ were left unmatched, and $c^{i_1}, \ldots, c^{i_L}, s^{j_1}, \ldots, s^{j_L}$ are the types of these unmatched customers and servers.

Theorem 5.3 (Adan, Busic, Mairesse and Weiss [8])**.** *The process $X^\infty(n)$ is a discrete time discrete state Markov chain. It is ergodic if and only if complete resource pooling condition (8) holds. Its stationary distribution is given, up to a normalizing constant, by:*

$$P^\infty(c^{i_1}, c^{i_2}, \ldots, c^{i_L}, s^{j_1}, \ldots, s^{j_L}) \propto \prod_{\ell=1}^{L} \frac{\alpha_{c^{i_\ell}}}{\beta_{\mathcal{S}(\{c^{i_1}, \ldots, c^{i_\ell}\})}} \times \prod_{\ell=1}^{L} \frac{\beta_{s^{j_\ell}}}{\alpha_{\mathcal{C}(\{s^{j_1}, \ldots, s^{j_\ell}\})}} \tag{9}$$

We note the close resemblance of this formula to the stationary distributions given in (3), (5), (6).

6. A Novel FCFS Infinite Directed Matching Model

The similarity of the stationary distributions of the continuous time processes X^q, X^r, X^m and the FCFS infinite bipartite matching discrete time process X^∞ suggests that they may be more closely related. In this section we introduce a new FCFS infinite matching model. It is similar to the model of Section 5 and of

[6, 7, 8]. It is also related to the model studied in [12], and is even more closely related to X^q, X^r, X^m. We use this new process to derive some more properties of X^q, X^r, X^m, in Section 7.

We consider a single infinite sequence of customers and servers, which is generated as follows: each successive item in the list is a customer of type c_i with probability $\alpha_{c_i} = \frac{\lambda_{c_i}}{\bar\lambda+\bar\mu}$, and it is a server of type s_j with probability $\beta_{s_j} = \frac{\mu_{s_j}}{\bar\lambda+\bar\mu}$, and successive items in the sequence are independent. The result is a sequence $\dots, z^1, z^2, \dots$, where each item z^n indicates either a type of customer or a type of server. We then perform FCFS matching of the customers and servers according to the compatibility graph $\mathcal{G}$, *utilizing only matches of servers to earlier customers*. This means in particular that a server $z^n = s_j$ for which there is no earlier unmatched compatible customer will remain unmatched. We call this the *FCFS single stream infinite directed bipartite matching model*, or directed matching model for short.

We define the discrete-time process $X^{\downarrow\infty}(n)$ to describe the matching process for the directed matching model. Assume we have performed all the possible matches in the sequence $\dots, z^1, z^2, \dots$ up to and including z^n. Then $X^{\downarrow\infty}(n) = (c^1, \dots, c^L)$ is the ordered list of the customers that are still unmatched.

Theorem 6.1. *$X^{\downarrow\infty}(n)$ is a discrete-time discrete-state Markov chain. It is ergodic if and only if the stability condition (1) holds, and its stationary distribution, up to a normalizing constant, is given by:*

$$P^{\downarrow\infty}(c^1, \dots, c^L) \propto \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\dots,c^\ell\})}}. \tag{10}$$

The fraction of servers that remain unmatched is $1 - \frac{\bar\lambda}{\bar\mu}$.

Proof. It is seen immediately that the Markov chain $X^{\downarrow\infty}(n)$ is the jump chain of the process $X^m(t)$. Furthermore, the process $X^m(t)$ has jumps in which its state changes, at the uniform rate of $\bar\lambda + \bar\mu$, irrespective of the state. The theorem follows. □

Remark: The fraction of unmatched servers of each type s_j can be calculated from (10). It is the sum of $P^{\downarrow\infty}(c^1, \dots, c^L)$ over all sequences $c^1, \dots, c^L$ that contain only customers that are incompatible with s_j.

We define a more detailed process to describe the dynamics of the FCFS infinite directed matching model. The process $U(n) = (u^1, \dots, u^K)$ records the ordered sequence of the unmatched customers as well as the servers that are left unmatched between them, after all matches of customers and servers in the sequence $\dots, z^1, z^2, \dots$ up to and including z^n have been made. We refer to $U(n)$ as the augmented Markov chain of the infinite directed matching process. Here $U(n)$ starts with the earliest customer that remained unmatched up to z^n, $u^1 \in \mathcal{C}$. If all customers up to z^n have been matched we say that the matching is perfect, and we define $U(n) = \emptyset$ (we also denote it by 0). Clearly by Theorem 6.1, if the stability condition (1) holds, then $U(n)$ is an ergodic Markov chain.

We now formulate two theorems for the FCFS infinite directed matching model. Their proofs are similar to the proofs of Theorems 5.1, 5.2 of Section 5, (they are Theorems 3 and 4 in Adan et al. [8]). Because there are still some essential differences in the proofs, compared to [8], we include the proofs in Appendix B and Appendix C.

Theorem 6.2. *Let* $\ldots, z^{-1}, z^0, z^1, \ldots$ *be a sequence of customer and server types defined as above. If (1) holds then almost surely there exists a directed FCFS matching of servers to cover all the customers, and this matching is unique.*

We define an exchange transformation for the FCFS infinite directed matching model:

Definition 3. *For the FCFS infinite directed matching model, if all matches were made on* $\ldots, z^{-1}, z^0, z^1, \ldots$, *we define the exchanged sequence* $\ldots, \tilde{z}^{-1}, \tilde{z}^0, \tilde{z}^1, \ldots$ *as follows: If* $z^m = c_i$ *was matched to* $z^n = s_j$, *where* $m < n$, *then in the exchanged sequence we will have* $\tilde{z}^m = s_j$, $\tilde{z}^n = c_i$. *If* $z^n = s_j$ *was unmatched, then* $\tilde{z}^n = z^n$.

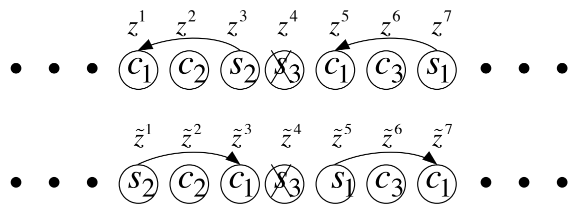

Figure 11: directed matching and its reversal

Figure 11 describes directed matching for a window of time in the doubly infinite sequence of customers and servers on the top panel. In it, $z^3 = s_2$ is matched with earlier $z^1 = c_1$, and z^7 is matched with z^5, while $z^2 = c_2$, $z^6 = c_3$ are not yet matched, and $z^4 = s_3$ will remain unmatched for ever. On the bottom panel of Figure 11 we see the exchange transformation of the top panel, with the matchings directed in reversed time.

Theorem 6.3. *The sequence* $\ldots, \tilde{z}^{-1}, \tilde{z}^0, \tilde{z}^1, \ldots$ *obtained from the sequence* $\ldots, z^{-1}, z^0, z^1, \ldots$ *by the exchange transformation is an i.i.d. sequence. The unique directed matches for the new sequence performed in reversed time, result in exactly the reversed matches of the original sequence, almost surely.*

7. Embeddings and a Version of Burke's Theorem

We have already noticed, and used it in the proof of Theorem 6.1, that the process $X^{\downarrow\infty}(n)$ is the jump process of the continuous-time matching queue process $X^m(t)$. Likewise, by Theorem 3.1 it is the jump process of $X^r(t)$.

We now use this embedding to prove a version of Burke's Theorem for the FCFS parallel service system under the redundancy policy, and for the FCFS parallel servers matching queue. Our proof here is direct, and follows from the reversibility property. An indirect proof of this result is givin by Bonald and Comte [13].

Theorem 7.1. *Let* $D_{c_i}(t)$, $i = 1, \ldots, I$ *be the departure process of customers of type* c_i *from the stationary parallel FCFS Redundancy Service queue, or from the stationary parallel FCFS Matching queue (counting all departures in* $(0, t]$*).*

(i) $D_{c_i}(t)$ *are independent Poisson processes of rates* λ_{c_i}.

(ii) The ordered sequence of customers in the system at time t for either system, given by $X^m(t)$ or $X^r(t)$, is independent of past departures, $D_{c_i}(s)$, $i = 1, \dots, I$ for all $s < t$.

Proof. We will use the reversibility result of $X^{\downarrow\infty}(n)$ in Theorem 6.3.

We consider a path of $X^m(t)$, $-\infty < t < \infty$ (the same goes for $X^r(t)$). The sample path is determined by the doubly infinite sequence of arriving customers and servers, $\dots, z^{-1}, z^0, z^1, \dots$, and the path of the independent Poisson process of rate $\bar\lambda + \bar\mu$, which determines the arrival time of z^n at t_n. The sequence then determines a sample path of the FCFS infinite directed bipartite matching process $X^{\downarrow\infty}(n)$, with the relation that $X^m(t) = X^{\downarrow\infty}(n)$ in the interval $[t_n, t_{n+1})$. Consider now the FCFS infinite directed matching for the sequence $\dots, z^{-1}, z^0, z^1, \dots$. In the matching process, if $z^m = c_i$ is matched to $z^n = s_j$ where $m < n$, then a customer of type c_i arrived at time t_m, and a server s_j arrived at t_n and was matched to that customer, so the customer of type c_i that arrived at time t_m departed at time t_n.

Now we perform the exchange transformation, so we now have the exchanged sequence $\dots, \tilde z^{-1}, \tilde z^0, \tilde z^1, \dots$. We then proceed with FCFS directed matching for the exchanged sequence, in reverse order. By Theorem 6.3 in this FCFS directed matching in reverse order, the same pairs will be matched (almost surely), so now $\tilde z^n = c_i$ is matched with $\tilde z^m = s_j$.

Consider now the process $X^m(t)$, and its jump process $X^{\downarrow\infty}(n)$. Take the exchange transformation of the sequence of FCFS directed matchings, reverse the discrete time, and perform FCFS directed matching, to obtain the discrete time exchanged and reversed process $\overleftarrow{X}^{\downarrow\infty}(n)$, and using the reversed sequence of time intervals between jumps in $X^m(t)$, construct from $\overleftarrow{X}^{\downarrow\infty}(n)$ the continuous-time process $\overleftarrow{X}^m(t)$.

By Theorem 6.3, the stationary $X^{\downarrow\infty}(n)$ and the stationary $\overleftarrow{X}^{\downarrow\infty}(n)$ are stochastically identical. The Poisson process of arrival with rates $\bar\lambda + \bar\mu$ is time reversible and so $X^m(t)$ and $\overleftarrow{X}^m(t)$ are stochastically identical. in particular, the sequence of arrivals of $\overleftarrow{X}^m(t)$ consists of independent Poisson process of arrivals of customers of type c_i at rates λ_{c_i}, and the state of the process, $X^m(t)$ is independent of the arrivals at all time $s > t$. But these arrivals are exactly the departures of $X^m(t)$ in reversed time. This completes the proof. □

Corollary 2. *Networks of parallel service systems under the redundancy service policy, as well as networks of parallel matching queues have product form stationary distributions.*

Proof. This version of Burke's Theorem, as given by Theorem 7.1, implies that the process $X^{\downarrow\infty}(n)$ is quasi-reversible. It is proven in [14, 15] that networks of quasi reversible Markovian systems have a product form stationary distribution. □

Another consequence of the embedding is a relaxation of the Poisson-exponential assumptions.

Theorem 7.2. *The stationary distribution of the FCFS matching queue $X^m(t)$, at times t immediately following transitions, remains the same as given in (6) if the arrivals of customers and servers are a general stationary point process, as long as types of arrivals are i.i.d. so that each arrival is a customer of type c_i with probability $\frac{\lambda_{c_i}}{\bar\lambda + \bar\mu}$, and it is a server of type s_j with probability $\frac{\mu_{s_j}}{\bar\lambda + \bar\mu}$.*

Proof. Consider the FCFS matching queue model, when arrivals are a stationary point process, and the arrival types are i.i.d. as above. Let t_n, $n = \dots, -1, 0, 1, \dots$ denote the arrival times. Then at the times t_n the ordered sequence of customers in the system, given by $X^m(t_2)$ is exactly the state of $X^{\downarrow\infty}(n)$, where in this FCFS infinite directed matching model arrivals are i.i.d., and so the stationary distribution of $X^{\downarrow\infty}(n)$ is given by (10), which is the same as (6). So the stationary distribution of $X^m(t)$ at the times t_n is given by (6). □

7.1. Another Interpretation of the Matching Queue

In our description and interpretation of the parallel matching queue we thought of customers waiting for servers, and arriving servers match to the oldest waiting compatible customer, or are lost. This system is stable when $\bar{\mu} > \bar{\lambda}$. It may describe a situation in which patients are waiting for a transplant of an organ, and patients have enough patience to wait for the right organ to arrive, and the supply of organs is sufficient, but organs cannot be conserved.

In reality the situation may be different, the organs may be conserved for a while, but there are more patients than organs. So now $\bar{\lambda} > \bar{\mu}$, and patients may be lost. We now consider the following process and policy: servers arrive and queue up waiting for customers, customers arrive, and each arriving customer then matches to the longest waiting compatible server and leaves immediately, or if no compatible server is found, the customer leaves immediately without a match. All we did in this model is to switch the roles of customers and servers, and all the results of Sections 2.4 and 7 hold, with c_i and s_j switching roles. Denote by $Y^m(t) = (s^1, \dots, s^L)$ the process that records the ordered sequence of available servers at time t, with s^1 the longest waiting. Then the stationary distribution of $Y^m(t)$ is given by:

$$P(Y^m(t) = s^1, \dots, s^L) \propto \prod_{\ell=1}^{L} \frac{\mu_{s^\ell}}{\lambda_{\mathcal{C}(\{s^1,\dots,s^\ell\})}}. \tag{11}$$

7.2. Embedding the FCFS-ALIS queues in an Infinite Matching Model

The process $X^q(t)$ can also be embedded in an infinite matching model, by considering the same sequences $\dots, z^{-1}, z^0, z^1, \dots$, but using a different matching mechanism: We now match each successive server $z^n = s_j$ to the earliest unmatched compatible customer $z^m = c_i$ where $m < k$ and k is the earliest position in the sequence with $k > n$, $z^k = s_j$. If no such match exists, the server z^n remains unmatched.

We define the process $X^{q\infty}(n)$ to describe the system after all possible matches that involve server z^k and customer z^ℓ for all $k, \ell \le n$ have been made. Then $X^{q\infty}(n) = (c^1, c^2, \dots, c^L, s^1, \dots, s^K)$. Here $c^1, c^2, \dots, c^L$ are the types of the customers in positions $\le n$ that are still unmatched, ordered as they appeared in the sequence, and $s^1, \dots, s^K$ are the types of servers in positions $\le n$ that have not been matched but may still be matched to a customer later in the sequence, ordered as they appeared in the sequence. Note that any of $c^1, c^2, \dots, c^L$ are incompatible with any of $s^1, \dots, s^K$, and that the server types $s^1, \dots, s^K$ are all different, so that $K \le J$.

One can see that this process is the discrete time jump process of $X^q(t)$, and analogues of Theorems 6.1 and 6.2 hold.

Appendix: Completion of Proofs

Appendix A. FCFS-ALIS stationary distribution

Proof of Theorem 2.3. The proof is by verifying that (3) satisfies partial balance. It is similar to the proof of Theorem 2.2 given in [10, 1], and to the proof of Theorem 2.4 given in [2].

We consider a state $x = (c^1, \ldots, c^L, s^1, \ldots, s^K)$. We list transitions in and out of the state x and their rates:

(i) Transition out of x due to arrival of type c_i that joins the queue, at rate λ_{c_i}, where $c_i \notin \mathcal{C}(\{s^1, \ldots, s^K\})$.

(ii) Transition out of x due to arrival of type c_i that matches to one of the idle servers, at rate $\lambda_{\mathcal{C}(\{s^1,\ldots,s^K\})}$.

(iii) Transition out of x due to completion of service, where server type s_j becomes idle, at rate: μ_{s_j}, for $s_j \notin S(\{c^1, \ldots, c^L\})$.

(iv) Transition out of x due to completion of service and start of service of a waiting customer, at rate: $\mu_{\mathcal{S}(\{c^1,\ldots,c^L\})}$

(v) Transition into state x due to arrival of c^L, at rate λ_{c^L}.

(vi) Transition into state x due to an arrival that matched with idle server s^* that was in position $k+1$, at rate: $\lambda_{\mathcal{C}(s^*)\backslash\mathcal{C}(\{s^1,\ldots,s^k\})}$, where $s^* \notin \mathcal{S}(\{c^1, \ldots, c^L\})$

(vii) Transition into state x due to a service completion, and server becoming idle, at rate μ_{s^K}.

(viii) Transition into state x due to a service completion, where a server is starting service of a customer c^* that was in position $\ell + 1$, at rate: $\mu_{\mathcal{S}(c^*)\backslash\mathcal{S}(\{c^1,\ldots,c^\ell\})}$.

We now show by substitution of the conjectured values from (3), that partial balance equations hold.

- Balance of (iv) with (v):

$$P^q(c^1, \ldots, c^L, s^1, \ldots, s^K) \times \mu_{\mathcal{S}(\{c^1,\ldots,c^L\})} = \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \prod_{k=1}^{K} \frac{\mu_{s^k}}{\lambda_{\mathcal{C}(\{s^1,\ldots,s^k\})}} \times \mu_{\mathcal{S}(\{c^1,\ldots,c^L\})};$$

$$P^q(c^1, \ldots, c^{L-1}, s^1, \ldots, s^K) \times \lambda_{c^L} = \prod_{\ell=1}^{L-1} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \prod_{k=1}^{K} \frac{\mu_{s^k}}{\lambda_{\mathcal{C}(\{s^1,\ldots,s^k\})}} \times \lambda_{c^L}.$$

- Balance of (ii) with (vii):

$$P^q(c^1, \ldots, c^L, s^1, \ldots, s^K) \times \lambda_{\mathcal{C}(\{s^1,\ldots,s^K\})} = \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \prod_{k=1}^{K} \frac{\mu_{s^k}}{\lambda_{\mathcal{C}(\{s^1,\ldots,s^k\})}} \times \lambda_{\mathcal{C}(\{s^1,\ldots,s^K\})};$$

$$P^q(c^1, \ldots, c^L, s_1, \ldots, s^{K-1}) \times \mu_{s^K} = \prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \prod_{k=1}^{K-1} \frac{\mu_{s^k}}{\lambda_{\mathcal{C}(\{s^1,\ldots,s^k\})}} \times \mu_{s^K}.$$

• Balance of (i) with (viii):

For $c_i \notin \mathcal{C}(\{s^1, \ldots, s^K\})$

$$P^q(c^1, \ldots, c^L, s^1, \ldots, s^K) \times \lambda_{c_i} =$$
$$\prod_{\ell=1}^{L} \frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} \prod_{k=1}^{K} \frac{\mu_{s^k}}{\lambda_{\mathcal{C}(\{s^1,\ldots,s^k\})}} \times \lambda_{c_i};$$

$$\sum_{\ell=0}^{L} P^q(c^1, \ldots c^\ell, c_i, c^{\ell+1}, \ldots, c^L, s^1, \ldots, s^K)$$
$$\times \mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^\ell\})} =$$
$$= \sum_{\ell=0}^{L} \prod_{j=1}^{\ell} \frac{\lambda_{c^j}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}} \times \frac{\lambda_{c_i}}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^\ell\})}}$$
$$\times \prod_{j=\ell+1}^{L} \frac{\lambda_{c^j}}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^j\})}} \prod_{k=1}^{K} \frac{\mu_{s^k}}{\lambda_{\mathcal{C}(\{s^1,\ldots,s^k\})}}$$
$$\times \mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^\ell\})}.$$

To show that the two expressions do indeed balance, we need to show that:

$$\prod_{\ell=1}^{L} \frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}} = \sum_{\ell=0}^{L} \prod_{j=1}^{\ell} \frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}} \times \frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^\ell\})}}$$
$$\times \prod_{j=\ell+1}^{L} \frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^j\})}} \times \mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^\ell\})} \tag{A.1}$$

which follows by induction on L. For $L = 1$:

$$\frac{1}{\mu_{\mathcal{S}(c_i)}} \frac{1}{\mu_{\mathcal{S}(c_i,c^1)}} \mu_{\mathcal{S}(c_i)} + \frac{1}{\mu_{\mathcal{S}(c^1)}} \frac{1}{\mu_{\mathcal{S}(\{c_i,c^1\})}} \mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(c^1)}$$

$$= \frac{1}{\mu_{\mathcal{S}(\{c_i,c^1\})}} \frac{\mu_{\mathcal{S}(c^1)} + \mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(c^1)}}{\mu_{\mathcal{S}(c_1)}} = \frac{1}{\mu_{\mathcal{S}(c^1)}},$$

and assuming that (A.1) holds for $L-1$, we show that for L:

$$\begin{aligned}
&\sum_{\ell=0}^{L}\prod_{j=1}^{\ell}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}\times\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^\ell\})}}\\
&\times\prod_{j=\ell+1}^{L}\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^j\})}}\times\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^\ell\})}\\
&=\sum_{\ell=0}^{L-1}\prod_{j=1}^{\ell}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}\times\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^\ell\})}}\\
&\times\prod_{j=\ell+1}^{L-1}\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^j\})}}\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^L\})}}\times\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^\ell\})}\\
&+\prod_{j=1}^{L}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}\times\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^L\})}}\times\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^L\})}\\
&=\prod_{j=1}^{L-1}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}\times\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^L\})}}\\
&+\prod_{j=1}^{L}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}\times\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^L\})}}\times\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^L\})}
\end{aligned}$$

$$\begin{aligned}
&=\prod_{j=1}^{L-1}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}\times\frac{1}{\mu_{\mathcal{S}(\{c_i,c^1,\ldots,c^L\})}}\left(1+\frac{\mu_{\mathcal{S}(c_i)\setminus\mathcal{S}(\{c^1,\ldots,c^\ell\})}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^L\})}}\right)\\
&=\prod_{j=1}^{L}\frac{1}{\mu_{\mathcal{S}(\{c^1,\ldots,c^j\})}}
\end{aligned}$$

- Balance of (iii) with (vi):

For $s_j \notin \mathcal{S}(\{c^1,\ldots,c^L\})$

$$\begin{aligned}
&P^q(c^1,\ldots,c^L,s^1,\ldots,s^K)\times\mu_{s_j}=\\
&\prod_{\ell=1}^{L}\frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}}\prod_{k=1}^{K}\frac{\mu_{s^k}}{\lambda_{\mathcal{C}(\{s^1,\ldots,s^k\})}}\times\ \mu_{s_j};
\end{aligned}$$

$$\begin{aligned}
&\sum_{k=0}^{K}P^q(c^1,\ldots c^L,s^1,\ldots,s^k,s_j,s^{k+1},\ldots,s^K)\\
&\quad\times\lambda_{\mathcal{C}(s_j)\setminus\mathcal{C}(\{s^1,\ldots,s^k\})}\\
&=\sum_{k=0}^{K}\prod_{\ell=1}^{L}\frac{\lambda_{c^\ell}}{\mu_{\mathcal{S}(\{c^1,\ldots,c^\ell\})}}\prod_{i=1}^{k}\frac{\mu_{s^i}}{\lambda_{\mathcal{C}(\{s^1,\ldots,s^i\})}}\frac{\mu_{s_j}}{\lambda_{\mathcal{C}(\{s_j,s^1,\ldots,s^k\})}}\\
&\prod_{i=k+1}^{K}\frac{\mu_{s^i}}{\lambda_{\mathcal{C}(\{s_j,s^1,\ldots,s^i\})}}\times\ \lambda_{\mathcal{C}(s_j)\setminus\mathcal{C}(\{s^1,\ldots,s^k\})}.
\end{aligned}$$

To show that the two expressions do indeed balance, we need to show that:

$$\begin{aligned}
&\prod_{k=1}^{K}\frac{1}{\lambda_{\mathcal{C}(\{s^1,\ldots,s^k\})}}=\sum_{k=0}^{K}\prod_{i=1}^{k}\frac{1}{\lambda_{\mathcal{C}(\{s^1,\ldots,s^i\})}}\times\frac{1}{\lambda_{\mathcal{C}(\{s_j,s^1,\ldots,s^k\})}}\\
&\times\prod_{i=k+1}^{K}\frac{1}{\lambda_{\mathcal{C}(\{s_j,s^1,\ldots,s^i\})}}\times\lambda_{\mathcal{C}(s_j)\setminus\mathcal{C}(\{s^1,\ldots,s^k\})}
\end{aligned}\tag{A.2}$$

The proof of (A.2) is similar to the proof of (A.1) $\square$

Appendix B. Unique Path of the FCFS Infinite Directed Matching Model

In Appendix B and Appendix C we prove properties of the FCFS directed matching of the i.i.d sequence of customer and server types $\ldots, z^1, z^2, \ldots$, where servers are only matched to previous customers, and of the Markov chain $U(n)$ of the leftover unmatched customers and servers. We use the notation $\beta_j = \mu_{s_j}/(\bar{\lambda} + \bar{\mu})$.

Proof of Theorem 6.2. We prove the Theorem in several steps, requiring two lemmas and two propositions. The two lemmas are pathwise results which do not depend on any probabilistic assumptions, and they prove subadditivity and monotonicity. Following that, Proposition 1 shows forward coupling, and Proposition 2 shows backward coupling. The proof is then completed in a short paragraph. This proof is very similar to the proof of Theorem 3 in [8] □

Lemma 1 (Monotonicity)**.** *Consider a subsequence $z^1, \ldots, z^M$ of servers and customers, with all the possible FCFS matches of servers to previous customers. Assume there are K customers and L servers left unmatched. Consider now an additional element z^0 preceding z_1, and the complete FCFS matching of servers to previous customers of $z^0, z^1, \ldots, z^M$. Then:*

(i) If $z^0 = c^0$ is an additional customer, the sequence $z^0, z^1, \ldots, z^M$ will have no more than $K+1$ customers and L servers unmatched.

(ii) If $z^0 = s^0$ is an additional server, the sequence $z^0, z^1, \ldots, z^M$ will have exactly K customers and $L+1$ servers unmatched.

Proof. Statement (ii) is trivial; s^0 will be unmatched and all the other links in $s^0, z^1, \ldots, z^M$ will be unchanged from $z^1, \ldots, z^M$.

To prove (i), let $A = (z^1, \ldots, z^M)$. In the matching of (c^0, A), if c^0 has no match, then all the other links in the matching are the same as in the matching of A, so the total number of unmatched customers is $K+1$ and unmatched servers is L. If c^0 is matched to a server $z^n = s^n$ and s^n is unmatched in the matching of A then (c^0, s^n) is a new link, and all the other links in the matching of (c^0, A) are the same as in the matching of A, so the total number of unmatched customers is K and unmatched servers is $L-1$.

If c^0 is matched to z^{n_1} and $z^{n_1} = s^{n-1}$ was matched to $z^{m_1} = c^{m_1}$ in the A matching, then (c^0, s^{n_1}) is a new link, and the link (s^{n_1}, c^{m_1}) in the A matching is disrupted. We now look for a match for $z^{m_1} = c^{m_1}$ in the matching of (c^0, A). Clearly, c^{m_1} is not matched to any of $z^j = s^j$, $m_1 < j < n_1$, since in A any such server was either matched to an earlier customer, and this link is still there in the matching of c^0, A, or such a server is incompatible with c^{m_1}; otherwise c_{m_1} could not have been matched to s^{n_1} in A. So c^{m_1} will either remain unmatched, or it will be matched to some $z^{n_2} = s^{n_2}$, where $n_2 > n_1$. In the former case, all the links of the A matching except (s^{n_1}, c^{m_1}) remain unchanged in the matching of (c^0, A), and so the numbers of unmatched items in (c^0, A) is $K+1$ and L. In the latter case, there are again two possibilities: If s^{n_2} is unmatched in the A matching, it will now be matched to c^{m_1} and the (c^0, A) matching will have disrupted one link and added 2 links retaining all other links of the A matching, so the numbers of unmatched items are K and $L-1$. If s^{n_2} is matched to $z^{m_2} = c^{m_2}$ in the A matching, then the link s^{n_2}, c^{m_2} is disrupted, and we now look for a match for c^{m_2} in the (c^0, A) matching. Similar to c^{m_1}, either c^{m_2} remains unmatched, resulting in $K+1$ and L unmatched items in the (c^0, A) matching, or, by the same argument as before, c^{m_2} will be matched to s^{n_3}, where $n_3 > n_2$. Repeating these arguments for any additional disrupted links, we conclude that we either end up with one more link, so the number of unmatched items are K and $L-1$, or we have the same number of links and the number of unmatched items are $K+1$ and L. □

Lemma 2 (Subadditivity). *Let $A' = (z^1, \dots, z^m)$, $A'' = (z^{m+1}, \dots, z^M)$ and let $A = (z^1, \dots, z^M)$. Consider the complete FCFS matching of servers to earlier customers in A', in A'', and in A and let K', K'', K be the number of unmatched customers and L', L'', L be the number of unmatched servers in these three matchings. Then $K \le K' + K''$ and $L \le L' + L''$.*

Proof. Let $\hat{A}' = (\hat{z}^1, \dots, \hat{z}^{K'+L'})$ be the ordered unmatched customers and servers from the complete FCFS matching of A'. Then the FCFS matching of $(\hat{A}', A'')$ will have exactly the same ordered unmatched customers and servers as the FCFS matching of A. We now construct the matching of $(\hat{A}', A'')$ in steps, starting with the matching of $(\hat{z}^{K'+L'}, A'')$, next the matching of $(\hat{z}^{K'+L'-1}, \hat{z}^{K'+L'}, A'')$ and so on. At each step, by Lemma 1, if the added z^j is a server the number of unmatched servers increases by 1, and the number of unmatched customers remains unchanged. If the added z^j is a customer the number of unmatched servers remains unchanged or decreases by 1, and the number of unmatched customers increases by 1 or remains unchanged. It follows that the total number unmatched customers is $\le K' + K''$ and of unmatched servers is $\le L' + L''$. □

We assume that the stability condition (1) holds. By Theorem 6.1, the augmented Markov chain of the infinite directed matching $U(n)$ is ergodic. Using the Kolmogorov extension theorem [16], we may define (in a non-constructive way) a stationary version $U^* = (U^*(n))_{n=-\infty}^{\infty}$ of the Markov chain. Define also $U^{[k]} = (U^{[k]}(n))_{n=-k}^{\infty}$ the realization of the Markov chain that starts at $U^{[k]}(-k) = \emptyset$.

Our first task is to show forward coupling, namely that U^* and $U^{[0]}$ coincide after a finite time τ with $E(\tau) < \infty$. Following that we use standard arguments to show backward coupling and convergence to a unique matching.

Proposition 1 (Forward coupling). *The two processes $(U^*(n))_{n=-\infty}^{\infty}$ and $(U^{[k]}(n))_{n=-k}^{\infty}$ will couple after a finite time τ, with $E(\tau) < \infty$.*

Proof. Denote by $|u|$ the number of unmatched customers for any state u of process U; we refer to $|u|$ as the length of the state. Consider the sequence of times $0 \le M_0 < M_1 < \cdots < M_\ell, \cdots$ at which $U^*(M_\ell) = \emptyset$. This sequence is infinite with probability 1, and $E(M_\ell) = E(M_0) + \ell E(M_1 - M_0) < \infty, \ell \ge 0$ by the ergodicity. Consider the state $u_0 = U^{[0]}(M_0)$. Then $|u_0| \le M_0$. By the monotonicity results of Lemmas 1 and 2, the states of $U^{[0]}$ satisfy $|u_0| \ge |U^{[0]}(M_1)| \ge \ldots \ge |U^{[0]}(M_\ell)|$, i.e. the length of the state of $U^{[0]}$ at the times $M_0, M_1, \ldots$ is non-increasing. This is because each block of customers and servers in times between $M_{\ell-1}$ and M_ℓ on its own has 0 unmatched. Furthermore, if the first unmatched customer in $U^{[0]}(M_\ell)$ is c_i, and the following item in the infinite sequence of customers and servers, $z^{M_\ell+1}$ is $s_j \in S(c_i)$, then $M_{\ell+1} = M_\ell + 1$, and $|U^{[0]}(M_\ell + 1)| = |U^{[0]}(M_\ell)| - 1$. This will happen with probability $\geq \delta = \min(\beta_1, \ldots, \beta_J)$. Hence, there will be coupling after at most $\sum_{j=1}^{|u_0|} L_j$ perfect matching blocks of U^*, where L_j are i.i.d. geometric random variables with probability δ of success. So coupling occurs almost surely, and the coupling time τ satisfies $E(\tau) \leq E(M_0) \left(\frac{1}{\delta} E(M_1 - M_0)\right)$.

The proof for $U^{[k]}$ is the same. $\square$

Note that once $U^{[k]}$ and U^* couple, they stay together forever. We now need to show backward coupling.

Proposition 2 (Backward coupling)**.** *Let U^* be the stationary version of the Markov chain $U(\cdot)$, and let $U^{[-k]}$ be the process starting empty at time $-k$. Then $\lim_{k\to\infty} U^{[-k]}(n) = U^*(n)$ for all $-\infty < n < \infty$ almost surely.*

Proof. The statement of almost surely refers to the measure of the infinite sequences $\ldots, z^{-1}, z^0, z^1, \ldots$.

Define $T_k = \inf\{n \geq -k : U^{[-k]}(n) = U^*(n)\}$. By the forward coupling Proposition 1, we get that T_k is almost surely finite. Let $\hat{T}_K = \max_{0\leq k\leq K} T_k$. Then $\hat{T}_K \geq 0$ and is also almost surely finite for any K. $\hat{T}_K$ is the time at which all the processes starting empty at time $-k$, where $0 \leq k \leq K$, couple with U^*, and remain merged forever. Define the event $E_K = \{\omega : \forall \ell \geq 0,\, U^{[-\ell]}(\hat{T}_K) = U^*(\hat{T}_K)\}$, i.e., those ω for which the process starting empty at any time before 0 will merge with U^* by time $\hat{T}_K$. We claim that $P(E_K) > 0$. We evaluate $P(\overline{E}_K)$. For any fixed $\ell \geq 0$, let $E_{\ell,K}$ be the event that $U^{[-\ell]}$ couples with U^* by time $\hat{T}_K$. We have $E_K = \bigcap_{\ell\geq 0} E_{\ell,K} = \bigcap_{\ell>K} E_{\ell,K}$ (by definition of $\hat{T}_K$, $E_{\ell,K}$ is always true for $\ell \leq K$, so we only need to consider $\ell > K$), so $\overline{E_K} = \bigcup_{\ell>K} \overline{E_{\ell,K}}$.

The event $\overline{E_{\ell,K}}$ will happen if starting at the last time prior to $-K$ at which the process $U^{[-\ell]}$ was empty, the next time that it is empty is after time 0. The reason for that is that otherwise the process $U^{[-\ell]}$ reaches state $\emptyset$ at some time $k \in [-K, 0]$ and from that time onwards it is coupled with $U^{[-k]}$, and will couple with U^* by time $\hat{T}_K$.

Define for $\ell > K$, $D_\ell = \{\omega : O^{[-\ell]}(m) \neq \emptyset \text{ for all} - \ell < m \leq 0\}$. Clearly, from the above, $\bigcup_{\ell>K} \overline{E_{\ell,K}} \subseteq \bigcup_{\ell>K} D_\ell$. Let τ denote the recurrence time of the empty state. Then:

$$P(\overline{E_K}) = P(\bigcup_{\ell>K} \overline{E_{\ell,K}}) \leq P(\bigcup_{\ell>K} D_\ell) \leq \sum_{\ell>K} P(\tau > \ell).$$

By the ergodicity $\sum_{l=0}^{\infty} P(\tau > l) = E(\tau) < \infty$. Hence we have that $P(\overline{E_K}) \to 0$ as $K \to \infty$, and therefore $P(E_K) > 0$ for large enough K, and $P(E_K) \to 1$ as $K \to \infty$. Note also that $E_K \subseteq E_{K+1}$.

Define now $\hat{T} = \sup_{k\geq 0} T_k$. We claim that $\hat{T}$ is finite a.s. Consider any ω. Then by $P(E_K) \to 1$ as $K \to \infty$ and by the monotonicity of E_K, almost surely for this ω there exists a value ℓ such that $\omega \in E_\ell$. But if $\omega \in E_l$, then $\hat{T}(\omega) \leq \hat{T}_\ell < \infty$.

So, all processes starting empty before time 0 will couple with U^* by time $\hat{T}$. By the stationarity of the sequences $(z^n)_{n=-\infty}^{\infty}$ and of U^*, we then also have that all processes $U^{[-k]}(n)$ starting empty before $-k$ will couple with U^* by time 0, if $k \geq \hat{T}$. Hence using the Loynes' scheme of starting empty at $-k$ and letting

$k \to \infty$ the constructed process will merge with U^* at time 0. But the same argument holds not just for 0, but for any negative time $-n$. Hence $U^{[-k]}$ and U^* couple at $-n$ (and stay coupled) for any $k > n + \hat{T}$. This completes the proof. $\square$

End of proof of Theorem 6.2. We saw that $\lim_{k\to\infty} U^{[-k]}(n) = U^*(n)$ for all n almost surely. Each process $U^{[-k]}(n)$ determines matches uniquely for all $n > -k$, so if we fix n, matches from n onwards are uniquely determined by $\lim_{k\to\infty} U^{[-k]}(n)$. Hence $(U^*(n))_{n=-\infty}^{\infty}$ determines, for every customer $z^n = c^n$, its match uniquely, almost surely. This proves the theorem. $\square$

Appendix C. Time Reversal of the FCFS Infinite Directed Matching Model

To prove Theorem 6.3, we consider blocks of the form $z^1, \dots, z^n$ such that all the customers in the block are matched to servers further in the sequence, we refer to those as perfect blocks. We first show in Lemma 3 that the exchange transformation implies time reversal in each block. Next in Lemma 4 we show that perfect blocks and their reversal have the same probability. The proof of the theorem then follows by considering the Palm measure and the time stationary measure of the exchanged sequence.

Lemma 3. *Let $z^1, \dots, z^n$ be a perfect block of customers and servers, and let $\tilde{z}^1, \dots, \tilde{z}^n$ be the block obtained from $z^1, \dots, z^n$ by the exchange transformation. Then $\tilde{z}^n, \dots, \tilde{z}^1$ is also a perfect block. In other words, if we have a block where all the customers are matched FCFS to servers ahead of them in the sequence, and we exchange the positions of matched pairs of customers and servers and retain the links, then the resulting matching is FCFS of servers to customers ahead of them in the sequence in reversed time.*

Proof. Consider the sequence $\tilde{z}^n, \dots, \tilde{z}^1$, and assume that $\tilde{z}^k = c_i$ is coupled in the exchanged sequence to $\tilde{z}^l = s_j$, with l ahead of k in the reversed sequence, i.e. $l < k$. Then we look at $\tilde{z}^{l'} = s_{j'}$ with $l < l' < k$. There are two possibilities: If $\tilde{z}^{l'}$ is unmatched, then $z^{l'} = \tilde{z}^{l'}$ because it was not exchanged. Hence in the original sequence $z^l = c_i$ precedes $z^{l'} = s_{j'}$ precedes $z^k = s_j$. But then $s_{j'}$ must be incompatible with c_i, or else $z^l = c_i$ would have matched with $z^{l'} = s_{j'}$ in the original sequence. The other possibility is that $\tilde{z}^{l'}$ has

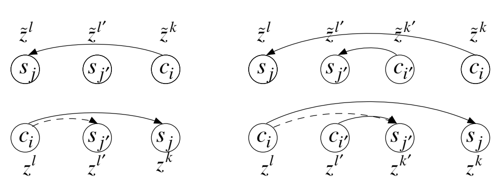

Figure C.12: Illustration of the proof of time reversal

been matched and exchanged with $\tilde{z}^{k'} = c_{i'}$. Assume now that $s_{j'}$ is compatible with c_i. Then we must show that $k' > k$. Assume to the contrary $l < l' < k' < k$. Then in the original sequence $\tilde{z}^l = c_i$ precedes $\tilde{z}^{l'} = c_{i'}$ precedes $\tilde{z}^{k'} = s_{j'}$ precedes $\tilde{z}^k = c_i$. But then $z^l = c_i$ would have matched with $z^{k'} = s_{j'}$ in the original sequence. This completes the proof. The proof is illustrated in Figure C.12. $\square$

Lemma 4. *Consider the FCFS directed matching of* $\dots, z^{-1}, z^0, z^1, \dots$*, and let* $z^{m+1}, \dots, z^{m+M}$ *be the block of customers and servers in positions* $[m+1, m+M]$*. Then the probability of observing these values, conditional on the event that the FCFS directed matching of these values is a perfect match is:*

$$\begin{aligned}
&P\Big((z^{m+1}, \dots, z^{m+M})\Big|(z^{m+1}, \dots, z^{m+M}) \text{ has perfect match}\Big) \\
&= \kappa_M \prod_{i=1}^{I} \alpha_{c_i}{}^{\#c_i} \prod_{j=1}^{J} \beta_{s_j}{}^{\#s_j}
\end{aligned}$$

where κ_M *is a constant that may depend on* M*, and* $\#c_i$ *and* $\#s_j$ *count the number of type* c_i *customers and type* s_j *servers in the block.*

Proof. The conditional probability is calculated using Bayes formula:

$$\begin{aligned}
&P(\text{seeing } z^{m+1}, \dots, z^{m+M} \mid \text{having a perfect match}) \\
&= \frac{P(\text{having a perfect match } |\text{seeing } z^{m+1}, \dots, z^{m+M})}{P(\text{having a perfect match of length } M)} \\
&\quad \times P(\text{seeing } z^{m+1}, \dots, z^{m+M}) \\
&= \kappa_M \times \mathbf{1}_{\{(z^{m+1}, \dots, z^{m+M}) \text{ is a perfect match}\}} \times \prod_{i=1}^{I} \alpha_{c_i}{}^{\#c_i} \prod_{j=1}^{J} \beta_{s_j}{}^{\#s_j}
\end{aligned}$$

where $\kappa_M = 1/P(\text{having a perfect match of length } M)$. ☐

Corollary 3. *Consider the FCFS directed matching of* $\dots, z^{-1}, z^0, z^1, \dots$*. Let* $z^{m+1}, \dots, z^{m+M}$ *be the block of customers and servers in positions* $[m+1, m+M]$*, which has perfect matching, and let* $\tilde{z}^{m+1}, \dots, \tilde{z}^{m+M}$ *be its exchange transformation. Replace* $z^{m+1}, \dots, z^{m+M}$ *by* $\tilde{z}^{m+M}, \dots, \tilde{z}^{m+1}$*. Then* $\tilde{z}^{m+1}, \dots, \tilde{z}^{m+M}$ *will be a perfectly matched block in the new directed matching of the complete sequence, and:*

$$P(\tilde{z}^{m+1}, \dots, \tilde{z}^{m+M}) = P(z^{m+1}, \dots, z^{m+M}).$$

Proof. That $\tilde{z}^{m+1}, \dots, \tilde{z}^{m+M}$ is a FCFS directed perfectly matched block follows from Lemma 3 and that $P(\tilde{z}^{m+M}, \dots, \tilde{z}^{m+1}) = P(z^{m+1}, \dots, z^{m+M})$ follows from Lemma 4. ☐

We now assume that the system is ergodic, i.e., the Markov chain $U(n)$ is ergodic (which holds if and only if the stability condition (1) holds). We have shown in Theorem 6.2 that for i.i.d sequences of servers and customers $\dots, z^{-1}, z^0, z^1, \dots$, under ergodicity, there exists a.s. a unique FCFS directed matching, which corresponds to the stationary version of the Markov chain U (generated by the Loynes' construction). We now define the following augmented process, with paths $\mathfrak{p}$ where the state consists of $\mathfrak{p}(n) = (U(n), z^n, v_n)$, and where v_n records the location of the element that is matched with z^n. That is, if $z^n = c_i$ is matched to $z^m = s_j$ where $m > n$ then $v_n = m$, if $z^n = s_j$ is matched to $z^m = c_i$ where $m < n$ then $v_n = m$, and if $z^n = s_j$ is unmatched then $v_n = n$. The path $\mathfrak{p}$ is uniquely determined by the i.i.d. sequence $\dots, z^{-1}, z^0, z^1, \dots$. We denote by $\mathfrak{P}$ the probability distribution of the paths $\mathfrak{p}$. We now define paths $\psi\mathfrak{p}$ by the exchange transformation followed by time reversal. From $\mathfrak{p}(n) = (U(n), z^n, v_n)$ we define $\psi\mathfrak{p}(-n) = (\tilde{U}(-n), \tilde{z}^{-n}, v_{-n})$

where $\tilde{z}^{-n} = z^{v_n}$, $\tilde{v}_{-n} = -v_n$, and define $\tilde{U}(m)$, for every m, as the customers that are unmatched and the servers that are unmatched starting from the last unmatched customer, in the sequence of $\tilde{z}^r, \tilde{v}_r$, $r \leq m$ (i.e., it behaves just like $U(n)$, but is obtained from the given matching that is already determined by $\tilde{z}, \tilde{v}$). For every path $\mathfrak{p}$ there is a transformed path $\psi\mathfrak{p}$.

We denote by $\psi\mathfrak{P}$ the distribution of the transformed paths $\psi\mathfrak{p}$. Our goal is to show that $\psi\mathfrak{P} = \mathfrak{P}$.

Let $\mathfrak{P}^0$ be the Palm version of the measure $\mathfrak{P}$, with respect to the state $\emptyset$ of U, i.e., $\mathfrak{P}^0$ is the law of $(U(n), z^n, v_n)$ conditioned on the event $\{U(0) = \emptyset\}$. A realization of a process of law $\mathfrak{P}^0$ can be obtained by considering a bi-infinite sequence of perfectly matched blocks of i.i.d customers and servers. Denote by $O(m)$, $m = 0, \pm 1, \pm 2, \ldots$ a sequence of i.i.d. minimal perfectly matched blocks. Then the resulting paths of these will have the distribution $\mathfrak{P}^0$. Now perform the exchange transformation on this sequence, followed by time reversal, and let $\psi\mathfrak{P}^0$ be the probability distribution of the result. For $\psi\mathfrak{P}^0$, let $\psi\mathfrak{P}$ be the corresponding stationary version of $\psi\mathfrak{P}^0$. According to Lemma 3, we have $\psi\mathfrak{P}^0 = \mathfrak{P}^0$. Because $\psi\mathfrak{P}$ is the stationary version of $\psi\mathfrak{P}^0$ and $\mathfrak{P}$ is the stationary version of $\mathfrak{P}^0$, we deduce that $\psi\mathfrak{P} = \mathfrak{P}$.

The key to the argument is the link between time-stationarity and event-stationarity. For general background on Palm calculus, see for instance Chapter 1 in [17]. We obtain the following result.

Proposition 3. *Consider a FCFS directed matching model under the stability condition (1). Let* $\ldots, z^{-1}, z^0, z^1, \ldots$ *be the independent i.i.d. sequence of customers and servers, with the unique FCFS matching between them. Then the exchanged sequence* $\ldots, \tilde{z}^{-1}, \tilde{z}^0, \tilde{z}^1, \ldots$ *is also i.i.d. with the same law. Furthermore, the FCFS directed matching for the exchanged sequence in reversed time consists of the same links as the matching of the original sequence.*

Proof of Theorem 6.3. That $\ldots, \tilde{z}^{-1}, \tilde{z}^0, \tilde{z}^1, \ldots$ is an i.i.d. sequence follows from the identity of $\psi\mathfrak{P}$ and $\mathfrak{P}$. That the Loynes' construction in reversed time will use the same links follows because the links of $\ldots, \tilde{z}^{-1}, \tilde{z}^0, \tilde{z}^1, \ldots$ give a set of links which are the FCFS directed matchings in reversed time between $\ldots, \tilde{z}^{-1}, \tilde{z}^0, \tilde{z}^1, \ldots$, and by Theorem 6.2 this matching is unique. □

Appendix D. Proof of Theorem 4.1, on the Comparison of 'N'-system Policies

We will prove the stronger result. For $h = q, r$, let $N^h(t) = N_1^h(t) + N_2^h(t)$, and assume $N_i^h(0) = 0$ for $i = 1, 2$, $h = q, r$.

Proposition 4. *Starting from an empty system, for every $t > 0$ the following holds:* $\{N^r(t), t \geq 0\} \geq_{st} \{N^q(t) - 1, t \geq 0\}$, $\{N_1^r(t), t \geq 0\} \geq_{st} \{N_1^q(t) - 1, t \geq 0\}$, $\{N_2^r(t), t \geq 0\} \geq_{st} \{N_2^q(t) - 1, t \geq 0\}$,.

This will by ergodicity prove that Theorem 4.1 holds.

Proof of Proposition 4. We will show the result path-wise (with probability 1) for coupled systems. We refer to the system under the Redundancy policy as system-r and to the system under FCFS-ALIS as system-q. We assume that system-q and system-r share a sequence of event times generated according to a Poisson process at rate $\bar{\lambda}+\bar{\mu}$, and the types of events are coupled so that each event, independently, is an arrival of type i with probability $\lambda_i/(\bar{\lambda}+\bar{\mu})$, and is a potential service completion at server i with probability $\mu_i/(\bar{\lambda}+\bar{\mu})$. We will show our result by induction on T, where T is the number of events so far, and, abusing notation, we define $N_i^h(T)$ to be the state immediately after event T occured. Our result is trivially true at time 0, and, if it is true at event T, it is true for all times between event T and before event $T+1$. Let us suppose the result is true for all time up until just before event $T+1$, and, in particular, $N^r(T) \geq N^q(T)-1$, $N_2^r(T) \geq N_2^q(T)-1$, and $N_1^r(T) \geq N_1^q(T)-1$. We first consider the result for $N^r(T+1)$ and $N^q(T+1)$.

It is easy to see that if $N^r(T) \geq N^q(T)$ then $N^r(T+1) \geq N^q(T+1)-1$, because $N^h(T+1) \geq N^h(T)-1$, for $h=q,r$, so suppose $N^r(T)=N^q(T)-1$. If event $T+1$ is an arrival, then $N^r(T+1)=N^q(T+1)-1$. If event $T+1$ is a potential service completion at server i and server i is busy in system-q, then either $N^r(T+1)=N^q(T+1)-1$ or $N^r(T+1)=N^q(T+1)$.

Hence, it remains only to show that $N^r(T+1) \geq N^q(T+1)-1$ given that $N^r(T)=N^q(T)-1$, event $T+1$ is a potential service completion at server i, and server i is idle in system-q for $i=1,2$. We will show that in this case, server i will also be idle in system-r, so $N^r(T+1)=N^r(T)=N^q(T)-1=N^q(T+1)-1$.

If $i=2$, because server 2 is idle in system-q, we must have $N^q(T) \leq 1$ and $N^q(T+1)=N^q(T)$, so $N^q(T+1)-1 \leq 0 \leq N^r(T+1)$.

If $i=1$, because server 1 is idle in system-q, $N_1^q(T)=0$ or 1.

Assume first that $N_1^q(T)=0$. Then $N^q(T)=N_2^q(T)$. It follows from $N^r(T)=N^q(T)-1$ and $N_2^r(T) \geq N_2^q(T)-1$ that $N_2^r(T)=N_2^q(T)-1=N^r(T)$ and $N_1^r(T)=0$. So we have shown that server 1 is idle in system-r.

Assume now that $N_1^q(T)=1$ and server 1 is idle in system-q. If $N_1^r(T)=0$, then server 1 is idle in system-r and we are done. It only remains to consider the possibility that $N_1^r(T)=N_1^q(T)=1$. We will show that this leads to a contradiction with the assumption that server 1 is idle in system-q. Assuming $N_1^r(T)=N_1^q(T)=1$, because $N^r(T)=N^q(T)-1$ we have $N_2^r(T)=N_2^q(T)-1$. Consider then the additional customer of type 2 that is present in system-q and not in system-r. By FCFS it must have been the earliest arrival among the $N_2^q(T)$ type 2 customers. Also, because it has already left system-r, it must have arrived before the single type 1 customer in both systems. Hence this customer of type 2 must be in service at server 2 in system-q. Therefore the single customer of type 1 must be in service at server 1 in system-q. This contradicts our assumption that server 1 is idle in system-q.

We therefore have the result for the total number of customers, so let us now consider the result for type 2 customers. Arguing as above, the only difficult case is the one where $N_2^r(T)=N_2^q(T)-1$, event $T+1$ is a potential service completion at server 2, and server 2 is serving a type-2 customer in system-r (so $N_2^r(T) \geq 1$ and $N_2^q(T) \geq 2$). We now show that in this case, server 2 must also be serving a type 2 customer in system-q, so $N_2^r(T+1)=N_2^q(T+1)-1$.

The argument for this is the same as above: The single type 2 customer that is p resent in system-q and has already left system-r must be the earliest customer in system-q, and so it must be in service at server 2 in system-q at time T.

Next consider the result for type 1 customers. We assume by induction that $N_1^r(T) \geq N_1^q(T) - 1$. As before, we only need to consider the case that $N_1^r(T) = N_1^q(T) - 1$, and we only need to consider potential service completions.

First suppose the event $T+1$ is a potential service completion at server 1. If server 1 is idle in system-r, clearly $N_1^r(T+1) \geq N_1^q(T+1) - 1$. If server 1 is working on a type 1 customer in system-r, then $N_1^r \geq 1$ and $N_1^q \geq 2$, so at least one of the type 1 customers must be in service at server 1 in system-q and we will have $N_1^q(T+1) = N_1^r(T+1) - 1$.

Now suppose the event $T+1$ is a potential service completion at server 2. If in system-q server 2 is serving a type 1 customer, then $N_1^r(T+1) \geq N_1^q(T+1) - 1$. If in system-q server 2 is serving a type 2 customer, and in system-r server 2 is either idle or serving a type 2 customer, then $N_1^q(T+1) = N_1^r(T+1) - 1$.

Thus, the only difficult possibility is if the event $T+1$ is a potential service completion at server 2, server 2 is serving a type 2 customer in system-q, while server 2 is serving a type 1 customer in system-r. We show by the induction hypothesis that this case is impossible.

We have assumed that $N_1^r(T) = N_1^q(T) - 1$, and we have shown that $N^r(T) \geq N^q(T) - 1$, therefore $N_2^r(T) = N_2^q(T)$. Therefore in both coupled systems, after event T there is the same set of type 2 customers. If in system-r server 2 is serving a type 1 customer, this means that this customer has arrived earlier than all the type 2 customers. But, using the same argument as before, this means that the additional type 1 customer in system-q has arrived even earlier. This contradicts the possibility that in system-q server 2 is serving a type 2 customer and in system-r server 2 is serving a type 1 customer.

This completes the proof.

□

The same argument can be used to show the following for the "W" system, where we redefine customer types so that type i customers can only be served at server i, $i = 1, 2$, and type 0 customers can be served at either server.

Proposition 5. $\{N_0^r(t) + N_i^r(t), t \geq 0\} \geq_{st} \{N_0^q(t) + N_i^q(t) - 1, t \geq 0\}$, $\{N_i^r(t), t \geq 0\} \geq_{st} \{N_i^q(t) - 1, t \geq 0\}$, $i = 1, 2$.

References

[1] I. Adan, G. Weiss, A skill based parallel service system under fcfs-alis – steady state, overloads, and abandonments, Stochastic Systems 4 (1) (2014) 250–299.

[2] K. Gardner, S. Zbarsky, S. Doroudi, M. Harchol-Balter, E. Hyytiä, A. Scheller-Wolf, Queueing with redundant requests: exact analysis, Queueing Systems 83 (3-4) (2016) 227–259.

[3] X. Su, S. A. Zenios, Patient choice in kidney allocation: A sequential stochastic assignment model, Operations Research 53 (3) (2005) 443–455.

[4] E. H. Kaplan, Managing the demand for public housing, Ph.D. thesis, Massachusetts Institute of Technology (1984).

[5] E. H. Kaplan, A public housing queue with reneging, Decision Sciences 19 (2) (1988) 383–391.

[6] R. Caldentey, E. H. Kaplan, G. Weiss, Fcfs infinite bipartite matching of servers and customers, Advances in Applied Probability 41 (03) (2009) 695–730.

[7] I. Adan, G. Weiss, Exact fcfs matching rates for two infinite multitype sequences, Operations Research 60 (2) (2012) 475–489.

[8] I. Adan, A. Busic, J. Mairesse, G. Weiss, Reversibility and further properties of fcfs infinite bipartite matching, arXiv preprint arXiv:1507.05939v2; Mathematics of Operations Research, to appear.

[9] U. Ayesta, T. Bodas, I. Verloop, A unifying framework for redundancy models: product form and impact of independence assumption.

[10] J. Visschers, I. Adan, G. Weiss, A product form solution to a system with multi-type jobs and multi-type servers, Queueing Systems 70 (3) (2012) 269–298.

[11] K. Gardner, M. Harchol-Balter, E. Hyytiä, R. Righter, Scheduling for efficiency and fairness in systems with redundancy, Performance Evaluation 116 (2017) 1–25.

[12] J. Mairesse, P. Moyal, Stability of the stochastic matching model, Journal of Applied Probability 53 (4) (2016) 1064–1077.

[13] T. Bonald, C. Comte, Balanced fair resource sharing in computer clusters, Performance Evaluation 116 (2017) 70–83.

[14] F. P. Kelly, Reversibility and stochastic networks, Cambridge University Press, 2011.

[15] J. Walrand, An introduction to queueing networks, Prentice Hall, 1988.

[16] B. Øksendal, Stochastic differential equations, Springer, 2003.

[17] F. Baccelli, P. Brémaud, Elements of Queueing Theory: Palm Martingale Calculus and Stochastic Recurrences, Springer, 2003.