UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

A Token-Based Central Queue with Order-Independent Service Rates

Ayesta, U.; Bodas, T.; Dorsman, J.L.; Verloop, I.M.

DOI
10.1287/opre.2020.2088
Publication date
2021
Document Version
Submitted manuscript
Published in
Operations Research

Link to publication

Citation for published version (APA):
Ayesta, U., Bodas, T., Dorsman, J. L., & Verloop, I. M. (2021). A Token-Based Central Queue with Order-Independent Service Rates. *Operations Research*, *70*(1), 545-561. https://doi.org/10.1287/opre.2020.2088

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:25 Jun 2022

A token-based central queue with order-independent service rates

Urtzi Ayesta, Tejas Bodas, Jan-Pieter Dorsman, Ina Maria Verloop

▶ **To cite this version:**

Urtzi Ayesta, Tejas Bodas, Jan-Pieter Dorsman, Ina Maria Verloop. A token-based central queue with order-independent service rates. Operations Research, INFORMS, In press. hal-02934633

HAL Id: hal-02934633

https://hal.archives-ouvertes.fr/hal-02934633

Submitted on 9 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A token-based central queue with order-independent service rates

U. Ayesta[1,3,5,6], T. Bodas[2], J.L. Dorsman[*,4], and I.M. Verloop[1,5]

[1]CNRS, IRIT, 2 rue Charles Camichel, 31071 Toulouse, France
[2]Indian Institute of Technology Dharwad, Karnataka, 580011, India
[3]IKERBASQUE - Basque Foundation for Science, 48011 Bilbao, Spain
[4]Korteweg-de Vries Institute for Mathematics, University of Amsterdam, P.O. Box 94248, 1090 GE Amsterdam, The Netherlands
[5]Université de Toulouse, INP, 31071 Toulouse, France
[6]UPV/EHU, University of the Basque Country, 20018 Donostia, Spain

September 7, 2020

Abstract

We study a token-based central queue with multiple customer types. Customers of each type arrive according to a Poisson process and have an associated set of compatible tokens. Customers may only receive service when they have claimed a compatible token. If upon arrival, more than one compatible token is available, an *assignment rule* determines which token will be claimed. The *service rate* obtained by a customer is state-dependent, i.e., it depends on the set of claimed tokens and on the number of customers in the system. Our first main result shows that, provided the *assignment rule* and the *service rates* satisfy certain conditions, the steady-state distribution has a product form. We show that our model subsumes known families of models that have product-form steady-state distributions including the order-independent queue of [20] and the model of [22]. Our second main contribution involves the derivation of expressions for relevant performance measures such as the sojourn time and the number of customers present in the system. We apply our framework to relevant models, including an M/M/K queue with heterogeneous service rates, the MSCCC queue and multi-server models with redundancy. For some of these models, we present expressions for performance measures that have not been derived before.

Keywords: product form, token-based, order-independent queue, redundancy system, matching model

1 Introduction

The discovery of queueing systems with a steady-state product-form distribution is probably one of the most fundamental contributions in queueing theory. In a pioneering work, [17] showed that in a queueing network formed by M/M/1 nodes, the joint steady-state distribution is given by the product of the marginal distributions of the individual nodes. Roughly speaking, this implies that the stationary distribution of the network can be obtained by multiplying the stationary distributions of the individual nodes assuming that each node is in isolation. Due to this property, the analysis of a queueing network reduces to that of single-node queues, simplifying the analysis tremendously. Product-form distributions provide insight into the impact of parameters on the performance and allow efficient calculation of performance measures. As a consequence, since Jackson's discovery, considerable effort has been put in understanding the conditions such that a stochastic model has a product-form steady-state distribution. Important steps forward were made by [7] and [19], who introduced BCMP networks and Kelly networks, respectively, which have product-form steady-state distributions. These networks demonstrate that models with multiple types of customers and general service time distributions could also have a product-form distribution. Since then, further studies have shown that networks with negative arrivals, instantaneous signals and blocking might have a product-form distribution, see [9] for an overview.

*Corresponding author: j.l.dorsman@uva.nl

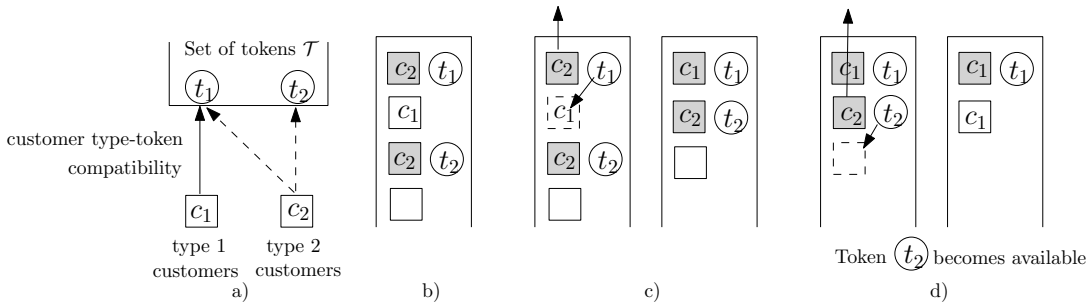

Figure 1: An example of a token-based model with token set $\mathcal{T} = \{t_1, t_2\}$. Grey-shaded customers are paired with tokens and can thus receive service. Figure b) depicts the initial state. Figure c) then sketches what happens when the first customer would leave: token t_1 scans the queue for the next compatible customer, and is then claimed by the oldest compatible customer, which is of type c_1. Figure d) describes what happens if the customer holding token t_2 would then leave the system. Token t_2 again scans the rest of the queue for compatible customers, but it turns out that the last customer is of type c_1, and token t_2 becomes available.

Recent years have witnessed a surge of interest in parallel server models with different types of customers. The main application is in the study of data centers, which consists of a pool of resources that are interconnected by a communication network. Indeed, data centers provide the main infrastructure to support many internet applications, enterprise operations and scientific computations. In two relevant studies, [22] and [20], sufficient conditions have been obtained for a multi-server system to have a product form. We note that these product-form distributions are not expressed as the product of per-type or per-server terms. In fact, they are expressed as a product of terms that correspond to a unique customer in the system. In that respect, they do not allow an interpretation in terms of a product of marginal distributions, as is the case with classical product-form distributions for Jackson, BCMP and Kelly networks. A notable difference between the two papers is in the state descriptor considered therein. In the multi-type customer and server model of [22], the authors consider an aggregated descriptor that keeps track of the servers being active but not of the type of customers being served or waiting. On the other hand, in the order-independent queue of [20], the state descriptor keeps track of the type of customers in the system, but not of the servers being active. These two modelling approaches have led to two separate streams of papers, where each of the approaches covers applications that are not covered by the other. Some of the applications studied are systems with blocking, redundancy and computer clusters. A natural question that arises is whether the original models of [22] and [20] can be generalised while preserving the product-form distribution in steady state.

We answer this question in the affirmative in this paper. We analyse a token-based central queue with multiple types of customers and multiple tokens. As will be proved in the paper, this model is a generalisation of both the model of [22] and the order-independent queue of [20]. Customers of each type arrive according to a Poisson process and have an associated set of compatible tokens. To receive service, a customer must claim a compatible token. Therefore, an arriving customer will immediately claim a compatible token if there is one available, otherwise it will wait until it can claim one.

For illustrational purposes, we regard an example of the token-based central queue depicted in Figure 1a). In this example, there are two customer types, namely c_1 and c_2, and two tokens t_1 and t_2. We assume type-c_1 customers can only claim token t_1, whereas type-c_2 customers are compatible to both token t_1 and token t_2. Figure 1b) represents a particular state of this system. Namely, there are four customers (represented by squares) in the queue, of which the first is a type-c_2 customer. Upon arrival, this customer immediately claimed token t_1 according to an *assignment rule*, which we will elaborate on later. The second customer that arrived, does not hold a token. This is partly because t_1 was already claimed by the first customer upon arrival of the second customer. Since it did not claim token t_2 upon arrival either, token t_2 must have been incompatible with the type of the second customer, so that the second customer must be of type c_1. The third customer that arrived, however, has claimed token t_2, and hence must be of type c_2. For the last customer in the queue, we do not have any information regarding its type, since upon arrival, all tokens were already claimed by other customers. As the first and third customer hold tokens, they are provided service with a state dependent service rate or *departure rate*.

Our first main result shows that, provided the *assignment rule* and the *departure rate function* satisfy the required conditions, the steady-state distribution of the token-based central queue has a product form. As in the

case of [22] and [20], this product-form distribution cannot be expressed as the product of per-type or per-token terms. We further show that the order-independent queue and the multi-type customer and server model of [22] are particular instances of our model and that our model includes examples that were not covered by either. In other words, our model and main results provide a unifying framework for parallel-server models with a product-form distribution. Our second main contribution is that we use the steady-state distribution of the general model to characterise transforms of relevant performance measures, including the sojourn time and the number of customers in the system. We illustrate the applicability of the framework by computing the steady-state distribution and analysing the performance of many relevant models, including an M/M/K queue with heterogeneous service rates, the MSCCC queue and multi-server models with redundancy. For some of these models, we present expressions for performance measures that have not been derived before. It is important to note that, even though our model is based on a central-queue architecture, some of the applications, in particular the redundancy models, correspond to topologies without a central queue, where instead every server has its own queue.

The rest of the paper is organised as follows. In the next section, we discuss studies related to this paper. Section 3 then describes the token-based central queue that we study in more detail and introduces the required notation. Section 4 shows that the token-based central queue has a product-form stationary distribution, which allows for the calculation of other performance measures in Section 5. Finally, we show in Section 6 that the models of [20] and [22] are captured by our model, and we discuss several applications of our model.

2 Related work

As mentioned in the introduction, there has been a surge of interest in multi-server queueing models in recent years. The main two references related to our work are [22] and [20], which identify classes of models that have a product-form stationary measure.

Subsequently, several studies have used the results of these two models to analyse a variety of other models. An important application area that has received a lot of attention is formed by redundancy models. While there are several variants of a redundancy-based system, the general notion of redundancy is to create multiple copies of the same customer that will be sent to a subset of servers. Depending on when replicas are deleted, there are two classes of redundancy systems: cancel-on-start (COS) and cancel-on-completion (COC). In redundancy systems with COC, once a copy has completed service, the other copies are deleted and the customer is said to have received service. On the other hand, in redundancy systems with COS, copies are removed as soon as one copy starts being served. In [8] the authors observe that the COC model is a special case of the order-independent queue in [20], which enables the authors to derive the steady-state distribution directly. We also refer to [15] for a thorough analysis of the COC system. On the other hand, [6] shows that while the COS based redundancy system is not an order-independent queue, it fits within the multi-type customer and server model of [22]. They also show that, while the COC model does not fit the framework of [22], it does fit an extension of it, where the state descriptor used in [22] is endowed with a more general departure rate function. We will use the resulting state descriptor also in this paper (see Section 3 for more details).

An important application area, which fits the framework of [22], is that of matching models, which have been studied in several recent papers, see for instance [1]. We also refer to [4] and [3], where the authors explore the relation between redundancy and matching models.

Another important related work is [5]. The model considered therein is similar to the one of [22] with the exception that the assignment policy *'assign longest idle server'* (ALIS) is used. Under the ALIS-policy, a new arrival that could be served by more than one inactive server, is assigned to the longest-idle server. To implement this policy, the state descriptor is enriched with information on the idleness of every inactive server. The authors prove that the steady-state distribution of this model has a product form. In our paper, we do not consider the ALIS variant. However, from the analysis of [5], we expect that all our results would carry over to this case. We discuss this in more detail in Section 4.

We conclude this section by mentioning the work of [10, 11], which discusses a token-based model that is somewhat related to ours. More particularly, in that study, a token-based mechanism is devised for the purpose of dynamic load balancing. This mechanism is described in terms of a tripartite compatibility graph and differs from ours mainly in that arriving customers which do not claim a token are immediately lost.

3 Model description

We now proceed with a detailed description of the model.

Customers and tokens. The model that we study represents a central-queue system with multi-type customers. The set of all customer types is denoted by $\mathcal{C}$ and customers of type $c \in \mathcal{C}$ arrive according to a Poisson process with rate λ_c. The total arrival rate of customers to the system is $\lambda := \sum_{c\in\mathcal{C}} \lambda_c$. In order for customers to receive service, they must hold a *token*. To this end, a set of K tokens denoted by $\mathcal{T} = \{t_1, \ldots, t_K\}$ is associated with the model, where $\mathcal{T}$ could be an infinitely large set (i.e., K could be equal to infinity, see Section 4.3 for more details). In particular, a type-c customer type is characterised by a token set $\mathcal{T}_c \subseteq \mathcal{T}$ which consists of the compatible tokens that can be held by customers of this type. Similarly, associated with a token $t \in \mathcal{T}$ is a set of customer types $\mathcal{C}_t \subseteq \mathcal{C}$ that can choose this token. As an example, we have in Figure 1 that $\mathcal{T}_{c_1} = \{t_1\}$, $\mathcal{T}_{c_2} = \{t_1, t_2\}$, $\mathcal{C}_{t_1} = \{c_1, c_2\}$ and $\mathcal{C}_{t_2} = \{c_2\}$.

Assignment of customers to tokens. At any point in time, the set of available tokens is denoted by $\mathcal{T}^{(a)}$, $\mathcal{T}^{(a)} \subseteq \mathcal{T}$, while the set of unavailable tokens is given by $\mathcal{T}\backslash\mathcal{T}^{(a)}$. To receive service, customers are required to hold a compatible token. Hence, when a customer of type $c \in \mathcal{C}$ arrives, it will claim a single token from the set $\mathcal{T}_c \cap \mathcal{T}^{(a)}$ (if it is non-empty), and then join the central queue. In case no compatible token is available upon arrival ($|\mathcal{T}_c \cap \mathcal{T}^{(a)}| = 0$), the customer joins the queue and waits until a token in the set $\mathcal{T}_c$ becomes available. If multiple compatible tokens are available, i.e., $|\mathcal{T}_c \cap \mathcal{T}^{(a)}| > 1$, an *assignment rule* decides which of the tokens will be claimed by the arriving customer. This assignment rule constitutes a randomised policy which, given $\mathcal{T}^{(a)}$ and the type of the arriving customer, dictates the probability with which the customer should claim a particular token. We assume this assignment rule to satisfy a so-called *assignment condition*, which is specified in Condition 1 below. Once a token t is selected by a customer, it is no longer available for selection (i.e. $\mathcal{T}^{(a)} := \mathcal{T}^{(a)}\backslash\{t\}$) until the customer completes service. Upon release, the token will immediately be reclaimed by the longest waiting tokenless customer of a type from the set $\mathcal{C}_t$. If there are no such customers, the token is added back to the set $\mathcal{T}^{(a)}$ ($\mathcal{T}^{(a)} := \mathcal{T}^{(a)} \cup \{t\}$). This event for example occurs in Figure 1d). We shall refer to customers with tokens as *active customers* and identify such customers by their associated tokens. Customers in the central queue without tokens are referred to as *inactive customers*. In Figure 1b), for instance, the first and third customer in line are active, the second and fourth are inactive and both tokens are claimed: $\mathcal{T}^{(a)} = \emptyset$.

Departure rates of customers. We assume service requirements of customers to be exponentially distributed. Since only customers possessing tokens receive service, the departure rate of active customers from the system is non-negative, while that of inactive customers is zero. Throughout the paper, we assume that the departure rates associated with active customers satisfy a condition that is specified in Condition 2 below. Since this condition is reminiscent of the order-independent queue as studied in [20], we call this the *order-independent condition.*

Markovian state descriptor. Due to the memoryless properties of the arrival and departure processes, the token-based central queue can be represented as a Markov process. We now introduce a suitable state descriptor, which in Section 4.1 is indeed shown to lead to a Markovian system. The state descriptor is of the form $(T_1, n_1, \ldots, T_i, n_i)$. This descriptor retains the order of arriving customers in the central queue from left to right. When the model is in state $(T_1, n_1, \ldots, T_{i-1}, n_{i-1}, T_i, n_i)$, it has i active customers which have claimed tokens $T_1, \ldots, T_i$. Furthermore, there are n_j inactive customers in the central queue that arrived between the two customers that have claimed tokens T_j and T_{j+1}, respectively, for $1 \leq j \leq i-1$. Inactive customers at the end of the queue are denoted by n_i. Since tokens are always claimed by the longest waiting eligible customer, we have for example that n_1 represents inactive customers which have token T_1 as their only compatible token. The set of such customer types is denoted by $\mathcal{U}(\{T_1\}) := \{c \in \mathcal{C} : \mathcal{T}_c = \{T_1\}\}$. In general, for $1 \leq j \leq i$, we denote the set of customer types that can claim tokens only from the set $\{T_1, \ldots, T_j\}$ by $\mathcal{U}(\{T_1, T_2, \ldots, T_j\}) := \{c \in \mathcal{C} : \mathcal{T}_c \subseteq \{T_1, \ldots T_j\}\}$. Thus, the customer types of the n_j customers between those with tokens T_j and T_{j+1} must belong to the set $\mathcal{U}(\{T_1, T_2, \ldots, T_j\})$. As the state descriptor retains the order of arrival, the oldest customer in a state is represented by token T_1. The youngest customer is one of the n_i customers, or in case $n_i = 0$, it is the active customer with token T_i. Furthermore, when $1 \leq j < k \leq i$, all n_j customers between T_j and T_{j+1} arrived before the n_k customers between T_k and T_{k+1}. We denote the state space of the resulting Markov process by $\mathcal{X}$, where any generic state $x \in \mathcal{X}$ is of the form $x = (T_1, n_1, \ldots, T_i, n_i)$. The only exception is the empty state with no customers present, which we denote by (0).

As an illustration of the state descriptor, the state depicted in Figure 1b) is described by $(t_1, 1, t_2, 1)$: the

first customer holds token t_1, the second customer waits for token t_1, the third customer holds token t_2, and the fourth customer waits for any compatible token. Likewise, the states in Figure 1c) and Figure 1d) are described by $(t_1, t_2, 1)$ and $(t_2, 1)$, respectively. Note that these state descriptors do not include the actual types of the customers themselves. Furthermore, for Figure 1 we have $\mathcal{U}(\{t_2\}) = \emptyset, \mathcal{U}(\{t_1\}) = \{c_1\}$ and $\mathcal{U}(\{t_1, t_2\}) = \{c_1, c_2\}$.

Assignment rule and assignment condition. In state $x = (T_1, n_1, \ldots, T_i, n_i)$, the arrival rate of *customers* that will initially be inactive is given by $\lambda_{\mathcal{U}(\{T_1,\ldots,T_i\})} := \sum_{c\in\mathcal{U}(\{T_1,\ldots,T_i\})} \lambda_c$, while the arrival rate of *customers* that become active immediately is given by $\lambda - \lambda_{\mathcal{U}(\{T_1,\ldots,T_i\})}$. When multiple compatible tokens are available upon a customer's arrival, an *assignment rule* determines the probability with which any of these tokens is assigned to the customer. Given the nature of the assignment rule, we denote by $\lambda_t(\{T_1, \ldots, T_j\})$ the rate at which arriving customers claim token t, provided that $\{T_1, \ldots, T_j\}$ is the set of all unavailable tokens. While $\lambda_t(\{T_1, \ldots, T_j\})$ depends on the assignment rule, it holds for any assignment rule that

$$\lambda - \lambda_{\mathcal{U}(\{T_1,\ldots,T_i\})} = \sum_{t\in\mathcal{T}\setminus\{T_1,\ldots,T_i\}} \lambda_t(\{T_1, \ldots, T_i\}). \tag{1}$$

As in [22], for the system to have a product-form stationary distribution, we require that an assignment rule satisfies the following assignment condition.

Condition 1. *For any possible combination of i tokens $T_1, \ldots, T_i$, $i = 1, \ldots, K$,*

$$\prod_{j=1}^{i} \lambda_{T_j}(\{T_1, \ldots T_{j-1}\}) = \prod_{j=1}^{i} \lambda_{\bar{T}_j}(\{\bar{T}_1, \ldots \bar{T}_{j-1}\}) \tag{2}$$

for every permutation $\bar{T}_1, \ldots \bar{T}_i$ of $T_1, \ldots T_i$.

[2] shows that there always exists an assignment rule for which Condition 1 is satisfied.

Order-independent condition. For any state $x = (T_1, n_1, \ldots, T_i, n_i)$, let $\mu_{T_j}(x)$ denote the departure rate of the active customer holding token T_j. Furthermore, let $\mu(x) = \sum_{j=1}^{i} \mu_{T_j}(x)$ be the total departure rate in state x. Additionally, we denote by $\phi(x) = i + \sum_{j=1}^{i} n_j$ the total number of customers in state x. The order-independent condition, which the departure rates in this model must satisfy, now reads as follows.

Condition 2. *In a given state $x = (T_1, n_1, \ldots, T_i, n_i)$, each of the departure rates $\mu_{T_j}(x)$, $j = 1, \ldots, i$, can be written as*

$$\mu_{T_j}(x) = \eta(\phi(x)) s_j(T_1, \ldots, T_i), \tag{3}$$

where

1. *$s_j(\cdot)$ is a non-negative real-valued function for which $s_j(T_1, \ldots, T_i) = s_j(T_1, \ldots, T_j)$, $1 \le j \le i$,*
2. *$k(T_1, \ldots, T_i) := \sum_{j=1}^{i} s_j(T_1, \ldots, T_j)$ is independent of any permutation of $(T_1, \ldots, T_i)$ and*
3. *$\eta(\cdot)$ is a non-negative real-valued function for which $\eta(j) > 0$ for $j = 1, 2, \ldots$.*

These restrictions on the functions $s_j(\cdot)$, $k(\cdot)$ and $\eta(\cdot)$ have the following implications. First, by the restriction on $s_j(\cdot)$, the departure rate of an active customer may depend on the types of the active customers ahead of it, but not on those behind. Note that $s_j(\cdot)$ may equal zero, so that it is possible for active customers to still receive no service. Second, $k(\cdot)$ is defined such that the total departure rate of customers from the system is the same for any permutation of the active customers. Finally, the function $\eta(\cdot)$ allows the departure rate of customers to depend on the total number of customers present in the system, but at the same time the departure rate is indifferent to the types of the inactive customers. Next, based on the definition of $\mu(x)$, we conclude that

$$\mu(x) = \eta(\phi(x)) k(T_1, \ldots, T_i). \tag{4}$$

As mentioned earlier, Condition 2 is reminiscent of the order-independent queue introduced in [20]. The difference, however, stems from the fact that we consider a different state descriptor, which captures a broader set of systems (cf. Section 6.2). It is also important to note that this condition allows our model to be more general than

that of [22], as will become clear in Section 6.3.

Further notation. We conclude this section with notation needed to describe several important performance measures. At an arbitrary point in time, let N denote the number of inactive customers in the system. More particularly, N_j denotes the number of inactive customers in the central queue between the two customers that have claimed tokens T_j and T_{j+1}. Thus, when the system is in state $x = (T_1, n_1, \ldots, T_i, n_i)$, it holds that $N_j = n_j$ and $N = \sum_{j=1}^{i} n_j$. Moreover, the number of type-c customers among these N_j customers is denoted by $N_j^{(c)}$. As a consequence, the total number of inactive type-c customers, denoted by $N^{(c)}$, satisfies $N^{(c)} = \sum_{j=1}^{i} N_j^{(c)}$. Using the same style of notation, M denotes the total number of customers present in the system. Furthermore, $M_j = N_j + 1$ represents the number of customers in the 'j-th' part of the system, where the added single customer is the one that holds token T_j. Of these M_j customers, $M_j^{(c)}$ are of type c, so that $M^{(c)}$, the number of type-c customers present in the system, satisfies $M^{(c)} = \sum_{j=1}^{i} M_j^{(c)}$. Note that the state descriptor does in general not include the types of both active customers and inactive customers. However, in special cases of the model, the values of $M_j^{(c)}$ and $N_j^{(c)}$ can still be retrieved.

We define the *time-till-token* of a customer to be the duration of the period between its arrival and the moment the customer claims a token. The time-till-token and the sojourn time of a type-c customer is denoted by W_c and S_c, respectively. Likewise, the quantities W and S refer to the time-till-token and the sojourn time of an arbitrary customer. Finally, the indicator function $\mathbb{1}_{\{A\}}$ on the event A returns one if event A is true, and zero otherwise.

4 Product-form stationary distribution

In this section, we derive the stationary distribution of the token-based central queue. To do this, in Sections 4.1 and 4.2 we use the methods and techniques of [22, Section 3], while accounting for a more elaborate token structure and using a slightly different notation. Afterwards, we provide notes on computational efficiency and stability in Sections 4.3 and 4.4, respectively.

4.1 Transition rates and balance equations

To derive the stationary distribution, we first note that, as is the case in [22], the model contains three types of transitions.

- *Arrival transitions.* An arriving customer either joins the central queue as an inactive customer (when it finds no compatible tokens in the set $\mathcal{T}^{(a)}$) or joins it as an *active customer*. In a given state $x = (T_1, n_1, \ldots, T_i, n_i)$, the arrival rate of inactive customers $\lambda_{\mathcal{U}(\{T_1,\ldots,T_i\})}$ forms the transition rate from state x to state $(T_1, n_1, \ldots, T_i, n_i + 1)$. Likewise, customers that immediately claim a token t upon arrival, $t \notin \{T_1, \ldots, T_i\}$, arrive at rate $\lambda_t(\{T_1, \ldots, T_i\})$. Therefore, the transition rate from state x to state $(T_1, n_1, \ldots, T_i, n_i, t)$ is given by $\lambda_t(\{T_1, \ldots, T_i\})$.

- *Departure transitions where tokens become available.* Transitions to a state $x = (T_1, n_1, \ldots, T_i, n_i)$ due to a departure of a customer where a token T is released are possible from states of the form

$$\text{release}_{k,n}(x, T) = (T_1, n_1, \ldots, T_k, n_k - n, T, n, T_{k+1}, n_{k+1}, \ldots, T_i, n_i),$$

  where $k \in \{0, \ldots, i\}$, $n \in \{0, \ldots, n_k\}$ and $T \in \mathcal{T} \backslash \{T_1, \ldots, T_i\}$. It is straightforward to verify that $\phi(\text{release}_{k,n}(x, T)) = \phi(x) + 1$, so that

$$\begin{aligned} \mu_T(\text{release}_{k,n}(x, T)) &= \eta(\phi(x) + 1) s_T(T_1, \ldots, T_k, T, T_{k+1}, \ldots, T_i) \\ &= \eta(\phi(x) + 1) \left(k(T_1, \ldots, T_k, T) - k(T_1, \ldots, T_k)\right). \end{aligned} \tag{5}$$

  To obtain the transition rate from state $\text{release}_{k,n}(x, T)$ to x, $\mu_T(\text{release}_{k,n}(x, T))$ must be multiplied with the probability that the token T is indeed released from activity after the departure. This probability is given by $r_{k,n}(x, T) = \beta_k(T)^n \beta_{k+1}(T)^{n_{k+1}} \cdots \beta_i(T)^{n_i}$, where

$$\beta_k(T) = \frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_k\})}}{\lambda_{\mathcal{U}(\{T_1,\ldots,T_k,T\})}} \tag{6}$$

is the probability that a customer waiting in the k-th portion of the central queue can not be served by token T. As a special case, we define $\beta_0(T) = 0$ for any token $T \in \mathcal{T}$. It now follows that the transition rate from $\text{release}_{k,n}(x,T)$ to x is given by $\mu_T(\text{release}_{k,n}(x,T))r_{k,n}(x,T)$. In reference to [22], note that $\text{release}_{k,n}(x,T)$, $r_{k,n}(x,T)$ and $\beta_k(T)$ are equivalent to $\mathit{insert}^M_{kn}(x)$, $p^T_{kn}(x)$ and $\delta_k(T)$, respectively, in the notation of that paper.

- *Departure transitions where tokens are reassigned.* Departures where a token T_j is immediately reclaimed by another customer leading to transitions to $x = (T_1, n_1, \ldots, T_i, n_i)$ are possible from states of the form

$$\text{shift}_{k,n}(x, T_j) = (T_1, n_1, \ldots, T_k, n_k - n, T_j, n, T_{k+1}, n_{k+1}, \ldots, T_{j-1}, n_{j-1} + 1 + n_j, T_{j+1}, n_{j+1}, \ldots, T_i, n_i),$$

where $1 \le k \le i$, $k+1 < j \le i$ and $n \in \{0, \ldots, n_k\}$. Again, $\phi(\text{shift}_{k,n}(x,T)) = \phi(x) + 1$, so that

$$\begin{aligned}\mu_T(\text{shift}_{k,n}(x,T_j)) &= \eta(\phi(x)+1)s_{T_j}(T_1,\ldots,T_k,T_j,T_{k+1},\ldots,T_i)\\ &= \eta(\phi(x)+1)\left(k(T_1,\ldots,T_k,T_j) - k(T_1,\ldots,T_k)\right). \end{aligned} \tag{7}$$

Similar to the previous case, the transition rate from state $\text{shift}_{k,n}(x,T_j)$ to state x can be argued to be equal to $\mu_T(\text{shift}_{k,n}(x,T_j))s_{k,n}(x,T_j)$, where

$$s_{k,n}(x,T_j) = \beta_k(T_j)^n \beta_{k+1}(T_j)^{n_{k+1}} \cdots \beta_{j-1}(T_j)^{n_{j-1}}(1 - \beta_{j-1}(T_j)),$$

with $\beta_k(T_j)$ as defined in (6). In [22], $\text{shift}_{k,n}(x,T_j)$ and $s_{k,n}(x,T_j)$ are denoted by $\mathit{swap}^{T_j}_{kn}(x)$ and $q^{T_j}_{kn}$, respectively.

Taking these transitions into account, denoting the stationary distribution by $\{\pi(x) : x \in \mathcal{X}\}$ and recalling that the total departure rate from a state x is simply $\mu(x)$, one can now conclude that the global balance equations are, for $x = (T_1, n_1, \ldots, T_i, n_i) \in \mathcal{X}\backslash\{(0)\}$, given by

$$\begin{aligned}(\lambda + \mu(x))\pi(x) = {} & \mathbb{1}_{\{n_i>0\}}\lambda_{\mathcal{U}(\{T_1,\ldots,T_i\})}\pi(T_1,n_1,\ldots,T_i,n_i-1)\\ & + \mathbb{1}_{\{n_i=0\}}\lambda_{T_i}(\{T_1,\ldots,T_{i-1}\})\pi(T_1,n_1,\ldots,T_{i-1},n_{i-1})\\ & + \sum_{T\in\mathcal{T}\backslash\{T_1,\ldots,T_i\}}\sum_{k=0}^{i}\sum_{n=0}^{n_k}\mu_T(\text{release}_{k,n}(x,T))r_{k,n}(x,T)\pi(\text{release}_{k,n}(x,T))\\ & + \sum_{j=1}^{i}\sum_{k=0}^{j-1}\sum_{n=0}^{n_k}\mu_{T_j}(\text{shift}_{k,n}(x,T_j))s_{k,n}(x,T_j)\pi(\text{shift}_{k,n}(x,T_j)),\end{aligned} \tag{8}$$

and, moreover,

$$\lambda\pi((0)) = \sum_{T\in\mathcal{T}}\mu_T((0,T))\pi((0,T)). \tag{9}$$

4.2 Derivation of the stationary distribution

In the following theorem, we present one of the main contributions of this paper: when both the assignment condition and the order-independent condition (cf. Conditions 1 and 2) are satisfied, the stationary distribution of the token-based central queue has a product form.

Theorem 1. *If the token-based central queue is stable and Conditions 1 and 2 are satisfied, then, for each $x = (T_1, n_1, \ldots, T_i, n_i) \in \mathcal{X}$, the stationary distribution is given by*

$$\pi(x) = \pi((0))\frac{\Pi_\lambda(\{T_1,\ldots,T_i\})}{\Pi_k(T_1,\ldots,T_i)}\prod_{j=1}^{i}{\alpha_j}^{n_j}\prod_{j=1}^{\phi(x)}\frac{1}{\eta(j)}, \tag{10}$$

where

$$\Pi_\lambda(\{T_1,\ldots,T_i\}) = \prod_{j=1}^{i}\lambda_{T_j}(\{T_1,\ldots,T_{j-1}\}), \Pi_k(T_1,\ldots,T_i) = \prod_{j=1}^{i}k(T_1,\ldots,T_j) \text{ and } \alpha_j = \frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_j\})}}{k(T_1,\ldots T_j)}.$$

The normalising constant $\pi((0))$ is given by

$$\pi((0)) = \left(1 + \sum_{i=1}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \frac{\Pi_\lambda(\{T_1,\ldots,T_i\})}{\Pi_k(T_1,\ldots,T_i)} \sum_{(n_1,\ldots,n_i)\in\mathbb{N}^i} \prod_{j=1}^{i} {\alpha_j}^{n_j} \prod_{j=1}^{i+\sum_{k=1}^{i} n_k} \frac{1}{\eta(j)}\right)^{-1}, \tag{11}$$

where $\mathcal{T}^i$ denotes the set of all possible combinations of i tokens from the set $\mathcal{T}$.

Proof. The proof is similar to that of [22, Theorem 2], but accounts for the more general order-independent service rates. In Appendix A, we show that (10) satisfies (8). It is furthermore straightforward to verify that (10) satisfies (9). This guarantees that (10) represents the unique stationary distribution. The constant $\pi((0))$ in (11) follows from normalisation. □

Remark 1. The expression in (10) is not in closed form, since the normalising constant $\pi((0))$ contains infinite sums. For some specific cases of the function $\eta(\cdot)$, though, given that the token set is finite, $\pi((0))$ allows for a closed-form expression. For example, when $\eta(\cdot) = 1$, (11) reduces to

$$\pi((0)) = \left(1 + \sum_{i=1}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \frac{\Pi_\lambda(\{T_1,\ldots,T_i\})}{\Pi_k(T_1,\ldots,T_i)} \prod_{j=1}^{i} \frac{1}{1-\alpha_j}\right)^{-1}, \tag{12}$$

which is in closed form. We will see in Section 6 that $\eta(\cdot)$ is a constant function in many applications.

Remark 2. In the literature, most notably [5], a different assignment mechanism has been studied called ALIS: 'Assign Longest Idle Server'. Stated in our context, the key feature of an ALIS queue is that an arriving customer who finds multiple compatible tokens upon arrival, will activate the token that has been available the longest. Since this mechanism cannot be captured by an assignment rule as described in Section 3, the state descriptor would have to be extended to keep track of which token has been available the longest. The new state descriptor would be of the form $(T_1, n_1, \ldots, T_i, n_i, T_{i+1}, \ldots, T_K)$, where $T_{i+1}, \ldots, T_K$ are the available tokens in ascending order of the time they have been idle. In other words, if an arriving customer is eligible to claim token T_K, it will do so. Otherwise, it will claim T_{K-1} if it is able to do so, and so on. By following the lines of proof of [5], we expect that the stationary distribution for the token-based central queue with an ALIS mechanism also has a product form.

4.3 Aggregation of states for indistinguishable tokens

Having derived the stationary distribution, we now point out how its computation, and especially the computation of the normalising constant in (11) can be made more efficient. We do so by realising that the tokens may be indistinguishable in certain model instances. This also sllows for the modelling of an infinite number of tokens.

To define the notion of indistinguishability, we write the token set $\mathcal{T}$ as a union of disjoint token sets $\tilde{\mathcal{T}}_1, \ldots, \tilde{\mathcal{T}}_l$, where l is assumed to be finite and it holds for any two tokens $s, t \in \tilde{\mathcal{T}}_i$, $i \in \{1, \ldots, l\}$ that $\mathcal{C}_s = \mathcal{C}_t$, $\lambda_s(T_1, \ldots, T_j) = \lambda_t(T_1, \ldots, T_j)$, $\lambda_{T_j}(\{T_1, \ldots, T_{j-1}, s\}) = \lambda_{T_j}(\{T_1, \ldots, T_{j-1}, t\})$ and $k(T_1, \ldots, T_j, s) = k(T_1, \ldots, T_j, t)$ for $T_1, \ldots, T_j \in \mathcal{T}\backslash\{s, t\}$. We then call tokens which belong to the same token set $\tilde{\mathcal{T}}_i$ indistinguishable.

We say that a token t has a token label l_k whenever $t \in \tilde{\mathcal{T}}_k$, $k \in \{1, \ldots, l\}$. Then, two tokens s and t from the set $\tilde{\mathcal{T}}_k$ can be addressed by their token label l_k. This leads to a state descriptor of the form $x^{(L)} = (L_1, n_1, \ldots, L_i, n_i)$, where L_i represents the label of the token held by the i-th active customer in the system. We denote the state space under this state descriptor by $\mathcal{X}^{(L)}$. Let $l(t)$ denote the label of token $t \in \mathcal{T}$, i.e. $l(t) = l_j$ if $t \in \tilde{\mathcal{T}}_j$. Then, by aggregation of states in (10),

$$\begin{aligned}\pi((L_1, n_1, \ldots, L_i, n_i)) &= \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i : l(T_j)=L_j \forall j\in\{1,\ldots,i\}} \pi((T_1, n_1, \ldots, T_i, n_i)) \\ &= \pi((0)) \prod_{j=1}^{i} \frac{\lambda_{L_j}(\{L_1, \ldots, L_{j-1}\})}{k(L_1, \ldots, L_j)} \prod_{j=1}^{i} \left(\frac{\lambda_{\mathcal{U}(\{L_1,\ldots,L_j\})}}{k(L_1, \ldots, L_j)}\right)^{n_j} \prod_{j=1}^{\phi(x)} \frac{1}{\eta(j)}.\end{aligned} \tag{13}$$

Here, $\lambda_{L_j}(\{L_1, \ldots, L_{j-1}\}) = \sum_{t\in\mathcal{T}:l(t)=L_j} \lambda_t(T_1, \ldots, T_{j-1})$ (with $l(T_1) = L_1, l(T_2) = L_2, \ldots$) represents the arrival rate of customers that immediately claim a token with label L_j, when there are $j-1$ active customers that have claimed tokens from labels $L_1, \ldots, L_{j-1}$. Likewise, when $L_j = l(T_j)$ for $j \in \{1, \ldots, i\}$,

$k(L_1,\ldots,L_j) = k(T_1,\ldots,T_j)$ and $\mathcal{U}(\{L_1,\ldots,L_j\}) = \mathcal{U}(\{T_1,\ldots,T_j\})$. The normalising constant $\pi((0))$ as given in (11) remains unchanged, but can now alternatively be written as

$$\pi((0)) = \left(1 + \sum_{i=1}^{K} \sum_{(L_1,\ldots,L_i)\in\mathcal{L}^i} \prod_{j=1}^{i} \frac{\lambda_{L_j}(\{L_1,\ldots,L_{j-1}\})}{k(L_1,\ldots,L_j)} \sum_{(n_1,\ldots,n_i)\in\mathbb{N}^i} \prod_{j=1}^{i} \left(\frac{\lambda_{\mathcal{U}(\{L_1,\ldots,L_j\})}}{k(L_1,\ldots,L_j)}\right)^{n_j} {}^{i+\sum_{k=1}^{i} n_k} \prod_{j=1}^{i} \frac{1}{\eta(j)}\right)^{-1},$$

where $\mathcal{L}^i$ represents the set of all possible combinations of i token labels. In the sequel, when working with the aggregated state descriptor, we will use $\mu_{L_j}(x^{(L)})$ and $s_j(L_1,\ldots,L_j)$ as notation for the equivalents of $\mu_{T_j}(x)$ and $s_j(T_1,\ldots,T_j)$.

4.4 Stability

From the stationary distribution (10), stability conditions can be derived. In particular, when the function $\eta(\cdot)$ has a limit $\eta := \lim_{j\to\infty}\eta(j)$, the system will be stable if $\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_i\})}}{k(T_1,\ldots,T_i)} < \eta$ for each $i \in \{1,\ldots,K\}$ and $\{T_1,\ldots,T_i\} \subset \mathcal{T}$, since (10) then constitutes a non-null and convergent solution of the equilibrium equations of the irreducible Markov process underlying the model. As such, it is implied by [13, Theorem 1] that the Markov process is ergodic, leading to stability. When $\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_i\})}}{k(T_1,\ldots,T_i)} > \eta$ for some $i \in \{1,\ldots,K\}$ and $\{T_1,\ldots,T_i\} \subset \mathcal{T}$, we have by (11) that $\pi((0)) = 0$, making the expected return time to state (0) infinite. Then, the Markov process is not ergodic and the token-based central queue is unstable. In case

$$\max_{\substack{i\in\{1,\ldots,K\}\\ \{T_1,\ldots,T_i\}\subset\mathcal{T}}} \frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_i\})}}{k(T_1,\ldots,T_i)} = \eta,$$

the existence of ergodicity depends on the way (and possibly the speed at which) the function $\eta(\cdot)$ converges to its limit η.

5 Performance analysis

In this section, we study several performance measures of the token-based central queue. By extending the techniques of [22, Section 4] to allow for the order-independent token structure, we study the (per-type) population size of inactive customers in Section 5.1, as well as their time-till-token using Little's law ([18]). Next, we consider performance measures in Section 5.2 that concern the total population of customers, namely the per-type population size and the sojourn time of customers.

5.1 Population of inactive customers

We first consider the population of inactive customers, i.e., the population of customers which are yet to claim a token. By following the analysis in [22, Section 4], we obtain the following expression for $N^{(c)}$, the number of present type-c customers that are inactive.

Theorem 2. *Let* $\theta_{c,j} := \frac{\lambda_c \mathbb{1}_{\{c\in\mathcal{U}(T_1,\ldots,T_j)\}}}{\lambda_{\mathcal{U}(T_1,\ldots,T_j)}}$ *for* $j \in \{1,\ldots,K\}$ *and* $c \in \mathcal{C}$. *Then, the joint PGF of* $\{N^{(c)} : c \in \mathcal{C}\}$ *is, for* $z_c \in \{\bar{c}\in\mathbb{C} : |\bar{c}| < 1\}$, *given by*

$$\mathbb{E}\left[\prod_{c\in\mathcal{C}} z_c^{N^{(c)}}\right] = \sum_{i=0}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \pi((0)) \frac{\prod_\lambda(\{T_1,\ldots,T_i\})}{\prod_k(T_1,\ldots,T_i)} \times$$
$$\times \prod_{j=1}^{i} \frac{1}{\eta(j)} \sum_{\{n_1,\ldots,n_i\}\in\mathbb{N}_0^i} \prod_{j=1}^{\sum_{k=1}^{i} n_k} \frac{1}{\eta(i+j)} \prod_{j=1}^{i} (\alpha_j \sum_{c\in\mathcal{C}} \theta_{c,j} z_c)^{n_j}, \qquad (14)$$

Proof. The proof extensively uses Theorem 1 and can be found in Appendix B. □

In [22], the waiting-time distribution of customers in their model has also been derived. In our model, however, although a customer may have claimed a token, it may still not receive any service. We will explicitly consider the time-till-token W_c of type-c customers (i.e. the time it takes for type-c customers to claim a token). In many applications, among which the model of [22], W_c coincides with the waiting time. To study the time-till-token, we note that the order in which type-c customers arrive is the same as the order in which type-c customers acquire a token, since tokens are always claimed by the longest waiting eligible customer. Therefore, $N^{(c)}$ and $W^{(c)}$ satisfy the assumptions required for the distributional form of Little's law to hold (cf. [18]). Little's law dictates that, for any $s \in \{\bar{c} \in \mathbb{C} : \Re(\bar{c}) > 0\}$,

$$\mathbb{E}\left[e^{-sW_c}\right] = \mathbb{E}\left[\left(\frac{\lambda_c - s}{\lambda_c}\right)^{N^{(c)}}\right]. \tag{15}$$

This leads to the following theorem.

Theorem 3. *The time-till-token of a type-c customer, W_c, satisfies, for any $s \in \{\bar{c} \in \mathbb{C} : \Re(\bar{c}) > 0\}$,*

$$\mathbb{E}\left[e^{-sW_c}\right] = \sum_{i=0}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \pi((0)) \frac{\prod_\lambda(\{T_1,\ldots,T_i\})}{\prod_k(T_1,\ldots,T_i)} \times \\ \times \prod_{j=1}^{i} \frac{1}{\eta(j)} \sum_{\{n_1,\ldots,n_i\}\in\mathbb{N}_0^i} \prod_{j=1}^{\sum_{k=1}^{i} n_k} \frac{1}{\eta(i+j)} \prod_{j=1}^{i} \left(\alpha_j \left(1 - \frac{s\mathbb{1}_{\{c\in\mathcal{U}(\{T_1,\ldots,T_j\})\}}}{\lambda_{\mathcal{U}(\{T_1,\ldots,T_j\})}}\right)\right)^{n_j}. \tag{16}$$

Proof. The theorem follows by substitution of $z_d = 1$ for all $d \neq c$ in (14) and combining the result with (15). □

5.2 Total population of customers

In this section, we present results on the per-type sizes of the total population (that is, both inactive and active) as well as the population's sojourn times. Let g_j be the type of the customer that holds token T_j and define $G_{c_1,\ldots,c_i}(T_1,n_1,\ldots,T_i,n_i) := \mathbb{P}\left(\bigcap_{j\in\{1,\ldots,i\}}\{g_j = c_j\} \mid x = (T_1,n_1,\ldots,T_i,n_i)\right)$. Then, based on the results of the previous section, we find that $M^{(c)}$, the number of type-c customers present in the system, satisfies the following theorem.

Theorem 4. *The joint PGF of $\{M^{(c)} : c \in \mathcal{C}\}$ is, for $z_c \in \{\bar{c} \in \mathcal{C} : |\bar{c}| < 1\}$ given by*

$$\mathbb{E}\left[\prod_{c\in\mathcal{C}} z_c^{M^{(c)}}\right] = \sum_{i=0}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \pi((0)) \frac{\prod_\lambda(\{T_1,\ldots,T_i\})}{\prod_k(T_1,\ldots,T_i)} \times \\ \times \prod_{j=1}^{i} \frac{1}{\eta(j)} \sum_{\{n_1,\ldots,n_i\}\in\mathbb{N}_0^i} \left(\sum_{\{c_1,\ldots,c_i\}\in\mathcal{C}^i} G_{c_1,\ldots c_i}(T_1,n_1\ldots,T_i,n_i) \prod_{j=1}^{i} z_{c_j}\right) \times \\ \times \prod_{j=1}^{\sum_{k=1}^{i} n_k} \frac{1}{\eta(i+j)} \prod_{j=1}^{i} (\alpha_j \sum_{c\in\mathcal{C}} \theta_{c,j} z_c)^{n_j}. \tag{17}$$

Proof. The proof is given in Appendix C. □

Remark 3. A general expression for $G_{c_1,\ldots,c_i}(T_1,n_1,\ldots,T_i,n_i)$, the probability that, provided the system is in state $x = (T_1,n_1,\ldots,T_i,n_i)$, tokens $T_1,\ldots,T_i$ are claimed by customers with types $c_1,\ldots,c_i$, respectively, seems hard to derive. For some applications, the derivation of an expression for $G_{c_1,\ldots,c_i}(T_1,n_1,\ldots,T_i,n_i)$ is, however, straightforward. For example, if the token sets $\mathcal{T}_c, c \in \mathcal{C}$, are disjoint, then $G_{c_1,\ldots,c_i}(T_1,n_1,\ldots,T_i,n_i) = \mathbb{1}_{\{\cap_{j=1}^{i}\{T_j\in\mathcal{T}_{c_j}\}\}}$.

We next focus on the sojourn time S_c of type-c customers. Deriving expressions for the sojourn time is generally hard, as type-c customers do not necessarily depart the system in the order of their arrival. Therefore, we only consider the sojourn time for instances of the model where type-c customers do depart the system in the

order of arrival. In such cases, again the distributional form of Little's law for the quantities $M^{(c)}$ and S_c holds true:

$$\mathbb{E}\left[e^{-sS_c}\right] = \mathbb{E}\left[\left(\frac{\lambda_c - s}{\lambda_c}\right)^{M^{(c)}}\right], \tag{18}$$

for any $s \in \{\bar{c} \in \mathbb{C} : \Re(\bar{c}) > 0\}$. This additional assumption holds in variety of applications and allows us to state the following theorem.

Theorem 5. *If type-c customers depart the system in the order of arrival, the LST of their sojourn time S_c is, for $s \in \{\bar{c} \in \mathbb{C} : \Re(\bar{c}) > 0\}$, given by*

$$\begin{aligned}\mathbb{E}\left[e^{-sS_c}\right] = \sum_{i=0}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \pi((0)) \frac{\prod_\lambda(\{T_1,\ldots,T_i\})}{\prod_k(T_1,\ldots,T_i)} \times \\ \times \prod_{j=1}^{i} \frac{1}{\eta(j)} \sum_{\{n_1,\ldots,n_i\}\in\mathbb{N}_0^i} \left(\sum_{\{c_1,\ldots,c_i\}\in\mathcal{C}^i} G_{c_1,\ldots c_i}(T_1, n_1 \ldots, T_i, n_i) \left(\frac{\lambda_c - s}{\lambda_c}\right)^{\sum_{j=1}^{i} \mathbb{1}_{\{c_i = c\}}} \right) \times \\ \times \prod_{j=1}^{\sum_{k=1}^{i} n_k} \frac{1}{\eta(i+j)} \prod_{j=1}^{i} \left(\alpha_j \left(1 - \frac{s\mathbb{1}_{\{c\in\mathcal{U}(\{T_1,\ldots,T_j\})\}}}{\lambda_{\mathcal{U}(\{T_1,\ldots,T_j\})}} \right) \right)^{n_j}. \end{aligned} \tag{19}$$

Proof. The proof is the same as that of Theorem 3, but instead of (14) and (15), (17) and (18) are used. □

In case $|\mathcal{T}_c| = 1$ for some type $c \in \mathcal{C}$, the assumption that type-c customers depart the system in the order of arrival is always valid. Then, if $\mathcal{T}_c = \{t\}$ and $\mathcal{C}_t = \{c\}$, (19) simplifies to

$$\begin{aligned}\mathbb{E}\left[e^{-sS_c}\right] = \sum_{i=0}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \pi((0)) \frac{\prod_\lambda(\{T_1,\ldots,T_i\})}{\prod_k(T_1,\ldots,T_i)} \frac{\lambda_c - s\mathbb{1}_{\{t\in\{T_1,\ldots,T_i\}\}}}{\lambda_c} \times \\ \times \prod_{j=1}^{i} \frac{1}{\eta(j)} \sum_{\{n_1,\ldots,n_i\}\in\mathbb{N}_0^i} \prod_{j=1}^{\sum_{k=1}^{i} n_k} \frac{1}{\eta(i+j)} \prod_{j=1}^{i} \left(\alpha_j \left(1 - \frac{s\mathbb{1}_{\{t\in\{T_1,\ldots,T_j\}\}}}{\lambda_{\mathcal{U}(\{T_1,\ldots,T_j\})}} \right) \right)^{n_j}. \end{aligned} \tag{20}$$

Remark 4. An expression for N, the total number of inactive customers in the system, can be obtained by noting that $\mathbb{E}\left[z^N\right] = \mathbb{E}\left[z^{\sum_{c\in\mathcal{C}} N^{(c)}}\right] = \mathbb{E}\left[\prod_{c\in\mathcal{C}} z^{N^{(c)}}\right]$ and $\sum_{c\in\mathcal{C}} \theta_{c,j} = \sum_{c\in\mathcal{C}} \frac{\lambda_c \mathbb{1}_{\{c\in\mathcal{U}(\{T_1,\ldots,T_j\})\}}}{\lambda_{\mathcal{U}(\{T_1,\ldots,T_j\})}} = 1$. Using this in (14) leads to

$$\mathbb{E}\left[z^N\right] = \sum_{i=0}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \pi((0)) \frac{\prod_\lambda(\{T_1,\ldots,T_i\})}{\prod_k(T_1,\ldots,T_i)} \prod_{j=1}^{i} \frac{1}{\eta(j)} \sum_{\{n_1,\ldots,n_i\}\in\mathbb{N}_0^i} \prod_{j=1}^{\sum_{k=1}^{i} n_k} \frac{1}{\eta(i+j)} \prod_{j=1}^{i} (\alpha_j z)^{n_j}. \tag{21}$$

Likewise, similar notions, along with the realisation that $\sum_{\{c_1,\ldots,c_i\}\in\mathcal{C}^i} G_{c_1,\ldots,c_i}(T_1, n_1, \ldots, T_i, n_i) = 1$, leads to

$$\mathbb{E}\left[z^M\right] = \sum_{i=0}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \pi((0)) \frac{\prod_\lambda(\{T_1,\ldots,T_i\})}{\prod_k(T_1,\ldots,T_i)} z^i \prod_{j=1}^{i} \frac{1}{\eta(j)} \sum_{\{n_1,\ldots,n_i\}\in\mathbb{N}_0^i} \prod_{j=1}^{\sum_{k=1}^{i} n_k} \frac{1}{\eta(i+j)} \prod_{j=1}^{i} (\alpha_j z)^{n_j}. \tag{22}$$

The overall time-till-token W and the overall sojourn time S can be reconstructed from $\mathbb{E}\left[e^{-sW}\right] = \sum_{c\in\mathcal{C}} \frac{\lambda_c}{\lambda} \mathbb{E}\left[e^{-sW_c}\right]$ and $\mathbb{E}\left[e^{-sS}\right] = \sum_{c\in\mathcal{C}} \frac{\lambda_c}{\lambda} \mathbb{E}\left[e^{-sS_c}\right]$, respectively.

Remark 5. In Appendix D, simplified expressions for several performance measures studied are given in case $\eta(\cdot) = 1$. These expressions are in closed form, and it follows from inversion of these expressions that $N^{(c)}$ and N can be interpreted as a weighted convolution of geometric random variables. Likewise, W_c can be interpreted as a weighted convolution of exponential random variables.

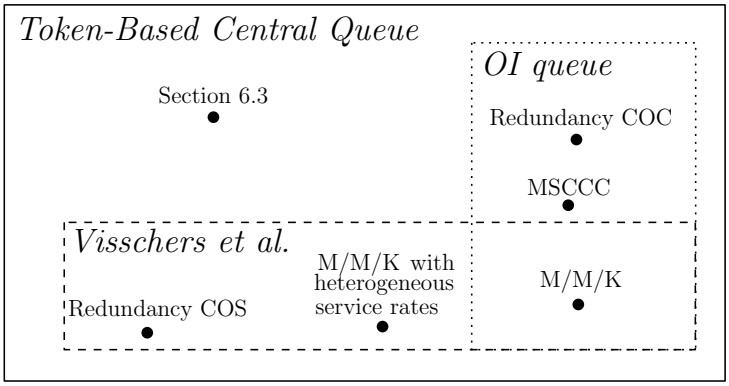

Figure 2: A classification of token-based central queues.

6 Generalisation of two existing classes of models

In this section, we show that both the multi-type customer and server model ([22]) as well as the order-independent queue ([20]) can be seen as special cases of the token-based central queue as analysed in this paper. The results are summarised in the Venn diagram presented in Figure 2.

6.1 Multi-type customer and server model

In the multi-type customer and server model of [22], type-c customers arrive at the system according to a Poisson process with rates λ_c and have an exponentially distributed service requirement with rate 1. There are K servers and server i works at rate μ_i. Each customer type has a set of compatible servers it can be served at. Whenever a server becomes idle, it takes the earliest arrived customer in the queue that it can process into service. An arriving customer that finds more than one compatible server idle is assigned to one of the servers according to a random assignment rule that satisfies a certain assignment condition.

6.1.1 Modelling as a token-based central queue

Introducing K tokens, letting each token represent a server and setting $\eta(\cdot) = 1$ and $s_j(T_1, \ldots, T_i) = \mu_{T_j}$ in our token-based central queue, we directly retrieve the multi-type customer and server model. The assignment condition of [22] coincides with Condition 1 and it is immediately seen that the order-independent condition (Condition 2) is also satisfied. We hence find that the model of [22] is a particular instance of our model.

6.1.2 Applications

In this section, we give two applications of the token-based central queue that fit the model of [22]. For illustrative purposes, we describe the M/M/K queue with heterogeneous service rates and explain how it can be interpreted as a token-based central queue. Then, we consider the redundancy-d COS model, and use results from Section 5 to derive performance measures that have not been derived before in the literature.

6.1.2.1 M/M/K queue with heterogeneous service rates

The M/M/K queue with heterogeneous service rates is a single-type queue that has K servers labeled $t_1, \ldots, t_K$, to which customers arrive according to a Poisson(λ) process. Upon arrival, the customer is assigned any idle server with equal probability. In case there are no idle servers, the customer waits in the queue. A customer served by server t_i requires an exponentially distributed service time with parameter $\mu(t_i)$. We denote the sum of the service rates by $\mu = \sum_{i=1}^{K} \mu(t_i)$.

To interpret this queue as a token-based central queue, we introduce a token for every server. Because of this, we label the tokens $t_1, \ldots, t_K$ as well. Hence, when a customer holds token t_i, it receives service from server t_i. Due to the uniform assignment of idle servers to arriving customers, the assignment rule is for $j = 1, \ldots, K$ and $t \in \mathcal{T} \backslash \{T_1, \ldots, T_{j-1}\}$ given by $\lambda_t(T_1, \ldots, T_{j-1}) = \frac{\lambda}{K-j+1}$. Condition 1 is now satisfied. Furthermore, since there is only one customer type, $\lambda_{\mathcal{U}(\mathcal{T})} = \lambda$ and $\lambda_{\mathcal{U}(\mathcal{S})} = 0$ for any strict subset $\mathcal{S} \subset \mathcal{T}$. To match the departure

rates of the M/M/K queue, we choose, for all $j \in \mathbb{N}$, $\eta(j) = 1$, $s_j(T_1, \ldots, T_i) = \mu(T_j)$ and $k(T_1, \ldots, T_i) = \sum_{j=1}^{i} \mu(T_j)$. These parameters satisfy Condition 2, and moreover make this model satisfy the framework of [22] (cf. Section 6.1.1). If the M/M/K queue would have homogeneous service rates (i.e. $\mu(T_i) = \mu(T_j)$ for all $i \neq j$), the queue is also order-independent, as all tokens would then be indistinguishable from one another. We explain why indistinguishable tokens lead to order-independent queues in Section 6.2.1. Finally, since the system has a single customer type ($\mathcal{C} = \{c\}$), we have that $G_{c,\ldots,c}(T_1, n_1, \ldots, T_i, n_i) = 1$.

Theorem 1 now leads to the stationary distribution, which was already derived in [16]. In particular, for any $x = (T_1, T_2, T_3, \ldots, T_k, n_k) \in \mathcal{X}$, we have that

$$\pi(x) = \pi((0)) \frac{\prod_{j=1}^{i} \frac{\lambda}{K-j+1}}{\prod_{j=1}^{i} \sum_{l=1}^{j} \mu(T_l)} \left(\frac{\lambda}{\mu}\right)^{\mathbb{1}_{\{i=K\}} n_K} = \pi((0)) \frac{\lambda^i (K-i)!}{K! \prod_{j=1}^{i} \sum_{l=1}^{j} \mu(T_l)} \left(\frac{\lambda}{\mu}\right)^{\mathbb{1}_{\{i=K\}} n_K}, \tag{23}$$

while $\pi(x) = 0$ for all other states. It is possible to drop the ordering of the tokens, while the system remains Markovian. In other words, states of the form $(n, \mathcal{R})$ can be introduced, where n is the number of waiting customers and $\mathcal{R}$ represents the (orderless) set of servers/tokens in service (i.e. $\mathcal{R} = \mathcal{T} \backslash \mathcal{T}^{(a)}$). By aggregation of states, we obtain

$$\pi(0, \mathcal{R}) = \frac{\pi((0)) \lambda^{|\mathcal{R}|} (K - |\mathcal{R}|)!}{K!} \sum_{(T_1, \ldots, T_{|\mathcal{R}|}) \in \underline{\mathcal{R}}} \frac{1}{\prod_{j=1}^{|\mathcal{R}|} \sum_{l=1}^{j} \mu(T_l)}$$

and

$$\pi(n, \mathcal{T}) = \frac{\pi((0)) \lambda^K}{K!} \left(\frac{\lambda}{\mu}\right)^n \sum_{(T_1, \ldots, T_K) \in \underline{\mathcal{T}}} \frac{1}{\prod_{j=1}^{K} \sum_{l=1}^{j} \mu(T_l)},$$

where $\underline{\mathcal{R}}$ ($\underline{\mathcal{T}}$) is the set of all possible permutations of the tokens in $\mathcal{R}$ ($\mathcal{T}$). In case $n > 0$ and $|\mathcal{R}| < K$, we obviously have that $\pi(n, \mathcal{R}) = 0$.

Since the M/M/K queue with heterogeneous service rates fits the framework of this paper, Theorems 2–4 now immediately offer characterizations of the size of the waiting customer population, the waiting times and the size of the overall customer population, respectively. In doing so, the size of the waiting customer population, conditional on the event all servers are busy, can be found to have a geometric distribution with failure parameter $\frac{\lambda}{\mu}$, while the waiting time of a customer, conditional on it being positive, is exponentially($\mu - \lambda$) distributed. To obtain expressions for the sojourn time distribution, however, Theorem 5 does not apply, since customers of equal types do not necessarily leave the system in the order of their arrival. Instead, through a PASTA-argument and by conditioning on the server that an arriving customer will be served by, we derive for any $s \in \{\bar{c} \in \mathbb{C} : \Re(\bar{c}) > 0\}$ that

$$\begin{aligned}
\mathbb{E}\left[e^{-sS}\right] &= \left(\sum_{\mathcal{R} \subset \mathcal{T} : \mathcal{T} \backslash \mathcal{R} \neq \emptyset} \pi(0, \mathcal{R}) \frac{\lambda}{K - |\mathcal{R}|} \sum_{T \in \mathcal{T} \backslash \mathcal{R}} \frac{\mu(T)}{\mu(T) + s} \right) + \\
&\quad + \sum_{n=0}^{\infty} \pi(n, \mathcal{T}) \mathbb{E}\left[e^{-sW} \mid W > 0\right] \sum_{T \in \mathcal{T}} \frac{\mu(T)}{\mu} \frac{\mu(T)}{\mu(T) + s} \\
&= \left(\sum_{\mathcal{R} \subset \mathcal{T} : \mathcal{T} \backslash \mathcal{R} \neq \emptyset} \pi(0, \mathcal{R}) \frac{\lambda}{K - |\mathcal{R}|} \sum_{T \in \mathcal{T} \backslash \mathcal{R}} \frac{\mu(T)}{\mu(T) + s} \right) + \left(\frac{\lambda}{\mu}\right)^K \frac{\mu - \lambda}{\mu - \lambda + s} \sum_{T \in \mathcal{T}} \frac{\mu(T)}{\mu} \frac{\mu(T)}{\mu(T) + s},
\end{aligned}$$

where terms between brackets represent the case where an arriving customer is immediately served.

6.1.2.2 The redundancy-d COS model

As pointed out in [6], an example of an application that fits the model of [22] is the redundancy-d cancel-on-start (COS) model. This model constitutes a system with K single-server FCFS queues and homogeneous servers providing service at unit speed. Customers arrive according to a Poisson(λ) process. Upon arrival, the customers choose at random d out of K queues, and to each of those queues, a copy of the customer is sent, each copy having its own independent, exponentially(μ) distributed service requirement. Once service on any of these copies has started, all the other copies of the same customer are removed from the system (cancelled), and only the sole remaining copy will be serviced. In case an arriving customer finds multiple of its d chosen servers idle upon arrival, one copy will go into service at any of these servers with uniform probability, and all other copies are cancelled immediately.

To interpret the COS-model as a token-based central queue, we follow much of the reasoning of [6]. We introduce a token set $\mathcal{T} = \{t_1, \ldots, t_K\}$, where token t_i has a one-to-one correspondence to the i-th of the K servers. We also introduce customer types corresponding to the set of servers/tokens an arriving customer replicates to: equal-type customers send copies to the same d out of K servers. As a consequence, there are $\binom{K}{d}$ lexicographically ordered customer types, labeled $c_1, \ldots, c_{\binom{K}{d}}$. When a token is claimed by a customer, the customer is taken into service by the server corresponding to the token, so that all other copies are cancelled. If type-c customers send copies to servers in the set $\mathcal{R} \subset \mathcal{T}$, we have that $\mathcal{T}_c = \mathcal{R}$. Since an arriving customer is of any of the $\binom{K}{d}$ types with uniform probability, we have $\lambda_{c_i} = \frac{\lambda}{\binom{K}{d}}$. Deriving $\lambda_t(T_1, \ldots, T_{j-1})$ is more intricate. An arriving customer that finds a tokens available will immediately claim any one of them with probability $\frac{1}{a}$. Thus, when tokens $(T_1, \ldots, T_{j-1})$ are active, this means that there are $\binom{K-j}{a-1}\binom{j-1}{d-a}$ customer types of which an arriving customer, upon arrival, would find a tagged token t among the a available tokens that it could immediately claim. That is, t is one of the eligible available tokens, there are $a-1$ others out of the $K-j$ available tokens ($\binom{K-j}{a-1}$ possibilities) and the remaining $d-a$ out of the d eligible tokens are among $T_1, \ldots, T_{j-1}$ ($\binom{j-1}{d-a}$ possibilities). Combining these observations and assuming that $\binom{m}{n} = 0$ for $0 \leq m < n$, we have for any $(T_1, \ldots, T_{j-1}) \in \mathcal{T}^{j-1}$ and any $t \in \mathcal{T} \backslash \{T_1, \ldots, T_{j-1}\}$ that

$$\lambda_t(T_1, \ldots, T_{j-1}) = \sum_{a=1}^{\min\{K-j+1,d\}} \frac{\lambda}{\binom{K}{d}} \frac{1}{a} \binom{K-j}{a-1} \binom{j-1}{d-a}.$$

Due to symmetry, it is immediate that Condition 1 is satisfied. We also reason that $\lambda_{\mathcal{U}(\{T_1,\ldots,T_i\})} = \frac{\lambda\binom{i}{d}}{\binom{K}{d}}$, since out of the $\binom{K}{d}$ customer types, there are $\binom{i}{d}$ that replicate to d servers/tokens in the set $\{T_1, \ldots, T_i\}$. As for the departure rate parameters; when a copy of a customer starts service (i.e., claims a token), its departure rate from the system equals μ. Therefore, $\eta(j) = 1$ for all $j \in \mathbb{N}$ and $s_j(T_1, \ldots, T_i) = \mu$ for all possible sets $(T_1, \ldots, T_i)$ of i tokens, so that $k(T_1, \ldots, T_i) = i\mu$. By probabilistic reasoning, we have that $G_{c_1,\ldots,c_i}(T_1, n_1, \ldots, T_i, n_i) = \frac{\lambda \mathbb{1}_{\{\bigcap_{j=1}^{i} \{T_j \in \mathcal{T}_{c_j}\}\}}}{\binom{K-1}{d-1}}$, since any server/token can be selected by $\binom{K-1}{d-1}$ customer types.

For this model, [6, Proposition 2] provides the PGF of $\mathbb{E}\left[z^N\right]$, the total number of waiting customers in the system. Since in generality, customers do not depart the system in the order of arrival, the distributional form of Little's law cannot be directly applied to this PGF to obtain an expression for the (PGF of the) waiting-time distribution W of the redundancy-d COS model. We therefore use the results of Section 5 for this purpose. For this model, the waiting time of a type-c customer coincides with its time-till-token. Hence, by using Theorem 3 and exploiting symmetry, we obtain for $s \in \{\bar{c} \in \mathbb{C} : \Re(\bar{c}) > 0\}$ that

$$\mathbb{E}\left[e^{-sW}\right] = \mathbb{E}\left[e^{-sW_{c_1}}\right] = \sum_{i=0}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \pi((0)) \prod_{j=1}^{i} \frac{\sum_{a=1}^{\min\{K-j+1,d\}} \frac{\lambda}{\binom{K}{d}} \frac{1}{a} \binom{K-j}{a-1} \binom{j-1}{d-a}}{j\mu - \lambda \frac{\binom{j}{d}}{\binom{K}{d}} + s\mathbb{1}_{\{j \geq d\}}},$$

where

$$\pi((0)) = \left(1 + \sum_{i=1}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \frac{\prod_{j=1}^{i} \sum_{a=1}^{\min\{K-j+1,d\}} \frac{\lambda}{\binom{K}{d}} \frac{1}{a} \binom{K-j}{a-1} \binom{j-1}{d-a}}{i!\mu^i} \prod_{j=d}^{i} \frac{\binom{K}{d} j\mu}{\binom{K}{d} j\mu - \lambda\binom{j}{d}}\right)^{-1}.$$

As for the sojourn time of customers, Theorem 5 again does not apply as same-type customers do not claim a token in the order of arrival. However, since each customer's exponential(μ) service time is independent of its waiting time, we have that $\mathbb{E}\left[e^{-sS}\right] = \frac{\mu\mathbb{E}\left[e^{-sW}\right]}{\mu+s}$.

6.2 The order-independent queue

The order-independent (OI) queue was first described in [20]. This model consists of a single central queue where customers of multiple types wait. The distinguishing feature of this model, as compared to a conventional FCFS queue, is that the service rate that the j-th customer receives is not necessarily zero for $j > 1$. Instead, the service rates of the customers satisfy an order-independent property. More particularly, in the OI queue, customers of type

i arrive according to a Poisson(λ_i) process and have an exponentially distributed service requirement with rate 1. The generic state descriptor as considered in [20] is $x^{(OI)} = (c_1, \ldots, c_n)$, where n is the number of customers in the system and c_j denotes the type of the j^{th} customer in the central queue. Let $\mathcal{X}^{(OI)}$ denote the corresponding state space. For a given state $x^{(OI)} \in \mathcal{X}^{(OI)}$, let $\mu_j^{(OI)}(x^{(OI)})$ denote the departure rate associated with the j-th customer. In an OI queue the following order-independent property holds.

Condition 3. *In a given state $x^{(OI)} = (c_1, \ldots, c_n)$, each of the rates $\mu_j^{(OI)}(x^{(OI)})$, $j = 1, \ldots, n$, can be written as*

$$\mu_j^{(OI)}(x^{(OI)}) = \eta^{(OI)}(n) s_j^{(OI)}(c_1, \ldots, c_n), \tag{24}$$

where

1. $s_j^{(OI)}(c_1, \ldots, c_n) = s_j^{(OI)}(c_1, \ldots, c_j)$ *for any* $1 \leq j \leq i$
2. $k^{(OI)}(c_1, \ldots, c_n) := \sum_{j=1}^{n} s_j^{(OI)}(c_1, \ldots, c_j)$ *is independent of any permutation of* $(c_1, \ldots, c_n)$ *and*
3. $\eta^{(OI)}(n) > 0$ *for* $n > 0$.

We see a close similarity with the order-independent condition as stated in Condition 2.

6.2.1 Modelling as a token-based central queue

Also the OI-queue can be interpreted as a token-based central queue, but seeing this is more intricate than the model of [22] before. The following theorem provides a connection between the OI queue and the token-based central queue.

Theorem 6. *For a given model, the following statements are equivalent:*

(1) the model fits in the OI queue framework;

(2) the model can be seen as a token-based central queue where the token sets associated with each of the customer types each consist of indistinguishable tokens.

Proof. To prove (1)→(2), one is given an OI queue, and to map this to a token-based central queue, one introduces an infinite number of tokens per customer type, and associates each customer with a token. To prove the opposite, i.e., (2)→(1), we notice that when in a token-based central queue the tokens are indistinguishable, it is clear to which type of customer an active token is associated, hence allowing for an OI state descriptor. For full details, we refer to Appendix E. □

The above theorem states that, given some model, one can interpret it as an OI queue if and only if the model can be interpreted as a token-based central queue where the token set of each customer type contains indistinguishable tokens. Recall from Section 4.3 that a model with indistinguishable tokens comes with the state descriptor $x^{(L)} = (L_1, n_1, \ldots, L_i, n_i)$. It is important to note the difference in the two state representations $x^{(OI)}$ and $x^{(L)}$: the types of all the customers are known in the OI queue, while only the customer types associated with the *active customers* can be known in our token-based representation. However, this sacrifice of detail leads to a richer class of models, as Theorem 6 shows that token-based central queues with distinguishable tokens handles a larger class of applications than OI queues.

For both state descriptors, a product-form solution for the steady-state distribution exists. For our state descriptor, the steady-state distribution follows from (13). Using this result and Theorem 6, the stationary distributions for the OI state descriptor can be recovered as is done in the corollary below. The proof can be found in Appendix F.

Corollary 7. *If the model fits in the OI queue framework, the steady-state distribution in terms of the OI state descriptor, denoted by $\pi^{(OI)}(x^{(OI)})$, is given by*

$$\pi^{(OI)}(x^{(OI)}) = \pi^{(OI)}((0)) \prod_{i=1}^{n} \frac{\lambda_{c_i}}{\eta^{(OI)}(i) k^{(OI)}(c_1, \ldots, c_i)}, \tag{25}$$

as was derived in [20].

6.2.2 Applications

We now proceed to give two applications of the token-based central queue that can be interpreted as OI queues. Based on Section 5, we derive performance measures for these models, which to the best of the authors' knowledge have not been obtained in the literature before.

6.2.2.1 The MSCCC queue

The first application that we consider is the Multi-server Station with Concurrent Classes of Customers (MSCCC) queue, as studied in [21] and [12]. As the name suggests, the MSCCC queue contains multiple servers and multiple types of customers, where at most one customer of any type can be in service. More particularly, the MSCCC queue consists of k identical servers serving customers at unit rate. Customers of type c_l, $l \in \mathbb{N}$, arrive according to a Poisson(λ_{c_l}) process and have exponential(μ) service requirements. Upon arrival, when a server is available and no other customer of his/her type is in service, the customer will go into service at an arbitrary free server. When no server is available or another customer of its type is already in service, the customer waits in line. When a server becomes available, it takes into service the longest waiting customer of a type not already in service.

To model the MSCCC queue using the token-based representation, we introduce for every customer type c_l a token t_l, which is dedicated to type-c_l customers. Thus, token t_l will always be held by the oldest type-c_l customer in the system if there is any, otherwise it is available. Given the one-to-one correspondence between customer types and tokens, we also refer to type-c_l customer as type-t_l customers (i.e., $\lambda_{c_l} = \lambda_{t_l}$, $z_{c_l} = z_{t_l}$ and so on). Then, it holds that $\lambda_{t_l}(T_1, \dots, T_i) = \lambda_{t_l}$ in case $t_l \notin \{T_1, \dots, T_i\}$. It follows trivially that $\lambda_{\mathcal{U}(T_1,\dots,T_i)} = \sum_{j=1}^{i} \lambda_{T_j}$. The departure rates are characterised by $\eta(j) = 1$, $s_j(T_1, \dots, T_i) = \mu \mathbb{1}_{\{j \le k\}}$ and $k(T_1, \dots, T_i) = \min\{i, k\}\mu$ for any $(T_1, \dots, T_i) \in \mathcal{T}^i$ and $j = 1, \dots, i$. These parameter settings satisfy Condition 1 as well as Condition 2 and lead to $G_{c_1,\dots,c_i}(T_1, n_1, \dots, T_i, n_i) = \mathbb{1}_{\{\bigcap_{j=1}^{i} \{\mathcal{C}_{T_j} = \{c_j\}\}\}}$.

The stationary distribution of this queue, which was already reported in [21] and [12] can now be reconstructed from (10). Additionally, after substitution of the model parameters derived above, Theorem 4 leads for $z_c \in \{\bar{c} \in \mathbb{C} : |\bar{c}| < 1\}$ to

$$\mathbb{E}\left[\prod_{c \in \mathcal{C}} z_c^{M^{(c)}}\right] = \sum_{i=0}^{K} \sum_{(T_1,\dots,T_i) \in \mathcal{T}^i} \pi((0)) \prod_{j=1}^{i} \frac{\lambda_{T_j} z_{T_j}}{k(T_1, \dots, T_j) - \sum_{l=1}^{j} \lambda_{T_l} z_{T_l}},$$

where $\pi((0)) = \left(\sum_{x \in \mathcal{X}} \frac{\prod_{j=1}^{i} \lambda_{T_j}}{\mu^i \min(i,k)! k^{\max(i-k,0)}}\right)^{-1}$. The expected queue lengths as reported in [12] and [20] can be derived from this expression. As for the sojourn time distribution, the MSCCC queue satisfies the condition that same-type customers depart the system in the order they arrive. Therefore, it follows after substitution of the model parameters from Theorem 5 (or more particularly, (20)) that, for $s \in \{\bar{c} \in \mathbb{C} : \Re(\bar{c}) > 0\}$,

$$\mathbb{E}\left[e^{-sS_{c_l}}\right] = \sum_{i=0}^{K} \sum_{(T_1,\dots,T_i) \in \mathcal{T}^i} \pi((0)) \prod_{j=1}^{i} \frac{\lambda_{T_j} - s\mathbb{1}_{\{T_j = t_l\}}}{\min(j,k)\mu - \sum_{l=1}^{j} \lambda_{T_l} + s\mathbb{1}_{\{t_l \in \{T_1,\dots,T_j\}\}}}.$$

The waiting-time distribution of a type-c_l customer is now given by $\frac{\mu+s}{\mu}\mathbb{E}\left[e^{-sS_{c_l}}\right]$.

6.2.2.2 The redundancy-d COC model

We now study another OI queue, namely the redundancy-d cancel-on-complete (COC) model. This model shares many characteristics with the redundancy-d COS model studied in Section 6.1.2. The only difference with that model is that redundant customer copies will now only be cancelled once any of the copies has *completed* service, rather than just having started service. Therefore, it is now possible that multiple copies of the same customer are in service at the same time. In [14], the sojourn time distribution of this model has been analysed in limiting regimes and the mean sojourn time in the general setting has been derived. The work of [8] in fact showed that this model is an OI queue, and therefore this model can also be interpreted as a token-based central queue. Using results from Section 5, we now supplement the analysis of [14] by giving a characterisation of the complete distribution of the sojourn time in the general setting. We also give an expression for the total number of customers in the system, which has not been derived before.

To interpret the redundancy-d COC model as a token-based central queue, the model parameters need to be chosen in a different way as compared to the COS model. We still introduce a customer type for every

choice of d out of K servers an arriving customer replicates to, so that there are $\binom{K}{d}$ customer types in total. However, we do not associate tokens with servers, but with customer types, as we did for the MSCCC queue. This is possible, since in a COC model only copies of the oldest of the customers of any type can receive service. We thus introduce a token set $\mathcal{T} = \{t_1, \dots, t_{\binom{K}{d}}\}$, where t_i corresponds to the i-th customer type. Since every customer type has its dedicated token, we have that $\lambda_{t_i}(T_1, \dots, T_{j-1}) = \lambda_{c_i} = \frac{\lambda}{\binom{K}{d}}$ when $t_i \notin \{T_1, \dots, T_{j-1}\}$. Similarly, we have that $\lambda_{\mathcal{U}(\{T_1,\dots,T_i\})} = \frac{i\lambda}{\binom{K}{d}}$ and Condition 1 is trivially satisfied. For the departure rates, we choose $\eta(j) = 1$ for all $j \in \mathbb{N}$. Recall that $s_j(T_1, \dots, T_i)$ is the departure rate of the customer that holds token T_j. This customer is the oldest of its type, and the j-th oldest overall among all the customers which are the oldest from their type. This customer's departure rate is given by μ (the service rate obtained from a single server) times the number of servers that are working on copies of this customer. These are the servers to which a copy of the customer with token T_j has been sent, but have not been sent a copy of any of the customers holding tokens $T_1, \dots, T_{j-1}$. In other words, if $F_j(T_1, \dots, T_i)$ refers to the number of servers that are able to serve copies of at least one of the customers holding $T_1, \dots, T_j$, $1 \leq j \leq i$, we have that $s_j(T_1, \dots, T_i) = \mu(F_j(T_1, \dots, T_i) - F_{j-1}(T_1, \dots, T_i))$. Note that, by nature of the function $F_j(\cdot)$, it is straightforward that $F_j(T_1, \dots, T_i) = F_j(T_1, \dots, T_j)$ and that $F_j(T_1, \dots, T_j) = F_j(\bar{T}_1, \dots, \bar{T}_j)$ for any permutation $(\bar{T}_1, \dots, \bar{T}_j)$ of $(T_1, \dots, T_j)$. As a consequence, $k(T_1, \dots, T_i) = \mu F_i(T_1, \dots, T_i)$ and Condition 2 holds. Finally, we have that $G_{c_1,\dots,c_i}(T_1, n_1, \dots, T_i, n_i) = \mathbb{1}_{\{\bigcap_{j=1}^{i}\{T_i = t_i\}\}}$.

Now that the model parameters are known, Theorem 1 provides the stationary distribution of the COC model as provided in [6, Proposition 7]. Moreover, (22) now implies for $z \in \{\bar{c} \in \mathbb{C} : |\bar{c}| < 1\}$ that

$$\mathbb{E}\left[z^M\right] = \sum_{i=0}^{\binom{K}{d}} \sum_{(T_1,\dots,T_i)\in\mathcal{T}^i} \frac{(\lambda z)^i \pi((0))}{\prod_{j=1}^{i} \mu\binom{K}{d} F_j(T_1, \dots, T_i) - j\lambda z},$$

where $\pi((0)) = \left(1 + \sum_{i-1}^{K} \sum_{(T_1,\dots,T_i)\in\mathcal{T}^i} \prod_{j=1}^{i} \frac{\lambda}{\mu\left(\binom{K}{d} F_j(T_1,\dots,T_i) - j\lambda\right)}\right)^{-1}$ is a normalisation constant. Furthermore, by applying Theorem 5 (or more particularly, (20)) and exploiting symmetry, we have for any $s \in \{\bar{c} \in \mathbb{C} : \Re(\bar{c}) > 0\}$ that

$$\mathbb{E}\left[e^{-sS}\right] = \mathbb{E}\left[e^{-sS_{c_1}}\right] = \sum_{i=0}^{\binom{K}{d}} \sum_{(T_1,\dots,T_i)\in\mathcal{T}^i} \frac{\lambda - s\binom{K}{d}\mathbb{1}_{\{t_1\in\{T_1,\dots,T_i\}\}}}{\lambda} \times$$
$$\times \frac{\lambda^i \pi((0))}{\prod_{j=1}^{i}\left(\mu\binom{K}{d} F_j(T_1, \dots, T_i) - j\lambda + s\binom{K}{d}\mathbb{1}_{\{t_1\in\{T_1,\dots,T_j\}\}}\right)}.$$

Remark 6. In the redundancy COC model in this section, as well as the redundancy COS model of Section 6.1.2.2, we have assumed that upon arrival, every customer selects exactly d servers to send copies to. Furthermore, we assumed that all servers each serve customers at an equal rate. However, conceptually, the results of this paper can be applied to a redundancy model where neither of these assumptions are satisfied, but at the cost of more intricate expressions.

Remark 7. The redundancy COC model is intimately related to other models, as discussed in [3]. More particularly, it is there shown that, among others, the redundancy COC model is equivalent to a parallel FCFS matching model. Such matching models arise naturally in many areas such as manufacturing, call centers and housing. Because of this equivalence, the framework of this paper also has applications in the field of matching.

6.3 Models that are neither a *multi-type customer and server* nor an *OI queue*

There also exist models that can be modelled as a token-based central queue, but do fit neither of the frameworks of [22] or [20]. In fact, a necessary and sufficient condition for a model to fall in this category is given as follows.

Condition 4. *A token-based central queue is not captured by the frameworks of [22] and [20] if and only if both of the following statements hold:*

a) The departure rate functions $\mu_{T_j}(x)$ are such that either $\eta(j)$ is not constant in j, or there exists no set of token-dependent values $\{\mu_t : t \in \mathcal{T}\}$ so that $s_j(T_1, \dots, T_i) = \mu_{T_j}$ for all states.

b) There is a token set $\mathcal{T}_c$ which contains at least one pair of tokens that are distinguishable from one another.

This condition can be argued as follows. If Condition 4a) is (not) satisfied, the model cannot (can) be a multi-type customer and server queue, because of the findings in Section 6.1.1. Similarly, Condition 4b) is the negation of the statement that the token-based central queue is also an OI queue by Theorem 6. Condition 4 thus allows us to design models that are token-based central queues but do not fit any of the two frameworks, and conversely to verify whether a given model falls into this category. We now introduce two such models, which shows that the token-based model extends the existing frameworks.

6.3.1 Dedicated and flexible customers

Consider a system with K servers and $K+1$ customer types. The set of compatible tokens for customer type c_i, $i \leq K$ is given by the infinite set $\mathcal{T}_{c_i} = \{t_{d_i}^{(1)}, t_{d_i}^{(2)}, \ldots\}$ of indistinguishable tokens, which we refer to as class-t_{d_i} tokens. We call these customer types *dedicated*, since they can only claim tokens of class t_{d_i}. Customer type c_{K+1} has as compatible token set

$$\mathcal{T}_{c_{K+1}} = \{\underbrace{t_1^{(1)}, \ldots, t_1^{(\gamma_1)}}_{\gamma_1}, \underbrace{t_2^{(1)}, \ldots, t_2^{(\gamma_2)}}_{\gamma_2}, \ldots, \underbrace{t_K^{(1)}, \ldots, t_K^{(\gamma_K)}}_{\gamma_K}\}.$$

The first γ_1 tokens in this set are mutually indistinguishable, as are the tokens in the other $K-1$ respective token subsets of sizes $\gamma_2, \gamma_3, \ldots, \gamma_K$. Customers of type c_{K+1} are henceforth called *flexible* in the sense that they can claim K different classes of tokens: $t_1, t_2, \ldots, t_K$.

In this system, the i-th server is associated to tokens t_i and t_{d_i}. This means that among the customers in the system holding a token of either class t_{d_i} or class t_i, only the one who arrived earliest will be served at rate μ_i, whereas the others do not receive service. Upon arrival, dedicated customers claim a token from their respective token set. Upon the arrival of a flexible customer, in case more than one token of the set $\mathcal{T}_{c_{K+1}}$ is available, an *assignment rule* will assign one of the tokens to the customer. When there is no token of $\mathcal{T}_{c_{K+1}}$ available, the customer waits.

The model can be interpreted as a load balancing system that implements redundancy, where customers are served in a FCFS fashion; see Figure 3. Upon arrival of a dedicated customer, the dispatcher always sends this customer immediately to the associated server, since a dedicated customer can always immediately claim a token from its infinite token set. If a flexible customer can also immediately claim a token from its set $\mathcal{T}_{c_{K+1}}$, it will likewise be dispatched to the corresponding server. When a flexible customer cannot claim a token directly upon arrival, this can be interpreted as having redundant copies of this customer dispatched to every server. The first copy that finds $\gamma_i - 1$ flexible customers ahead of it in its queue will be kept and eventually served by its corresponding server. All the other copies of the customer will be cancelled. This is the interpretation of a flexible customer claiming a token of class t_i which is released by another flexible customer that has its service completed by a server i. Figure 3 provides an example of this system.

The fact that the departure rate function is a state-dependent function and that this model involves mutually indistinguishable tokens ensures that Condition 4 is satisfied, so that this model is neither a multi-type customer and server queue nor an OI queue. It is tempting to think that for $\gamma_1 = \gamma_2 = 1$, as is the case in Figure 3, the model can be interpreted as an asymmetric redundancy CoS model in the spirit of Section 6.1.2.2, and as a result, that this model leads to a multi-type customer and server queue. However, this model does not incorporate a cancel-on-start mechanism. To illustrate, in Figure 3c), a copy of the customer has already been cancelled, while service for this customer has not started yet. Because of this reason, the model cannot be interpreted as a multi-type customer and server queue either.

6.3.2 M/M/K queue with a generalised service rate function

Let us consider the M/M/K queue with heterogeneous service rates of Section 6.1.2. That is, there are K tokens labeled $t_1, \ldots, t_K$, each associated with one of the K servers, and every arriving customer can claim any of them. Since the service rates are heterogeneous, the tokens are distinguishable. Recall that the service rate of the customer holding token T_j, $j \leq i$, is given by $\mu_{T_j}(x) = \eta(\phi(x))s_j(T_1, \ldots, T_i)$. We now look at the following variant of the model. Suppose that now either $s_j(T_1, \ldots, T_i)$ does not only depend on token T_j, or that the $\eta(\cdot)$-function is not constant (i.e. the service rates vary with the number of customers in the system). It then follows from Condition 4 that the model cannot be cast in the framework of [22] or [20]. Such a model could be motivated

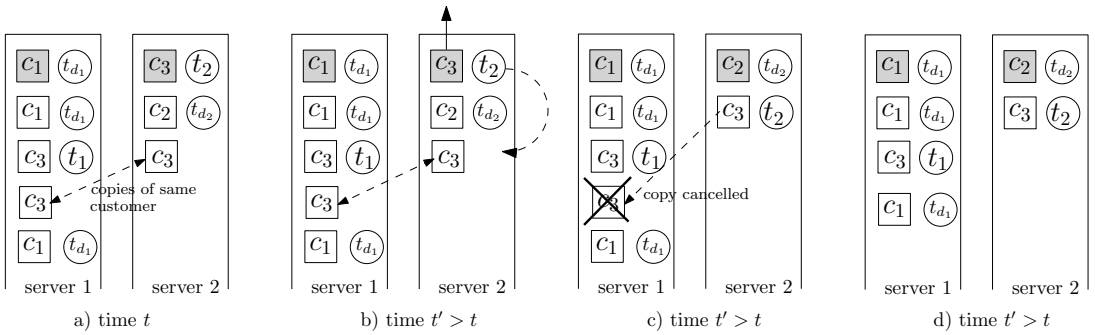

Figure 3: A two-server model with dedicated and flexible customers and $\gamma_1 = \gamma_2 = 1$. All dedicated customers have a token, and only the first flexible customer in each queue has a token. The servers serve the queues in a FCFS fashion, so only the first (shaded) customer in the queue gets service. Part $a)$ shows a snapshot of the system at some arbitrary time t. The second flexible customer of type c_3 has redundant copies in both servers, as it has not claimed token of classes t_1 or t_2 yet. Part $b)$ shows the system at a time $t' > t$, when the first customer from server 2 departs. Because of this, the next customer in line, which is a dedicated customer, starts being served, the flexible customer in server 1 obtains a token and its redundant copy in server 1 is cancelled (part c)). Part d) shows the final configuration.

by power saving systems, in which the speed of servers is modified depending on the number of customers in the system. The approach undertaken here, by generalising the service rate function, could be applied to other existing models. For example, by applying this to Section 6.1.2.2, one obtains a redundancy model that does not fit in the existing two frameworks.

Acknowledgements

The authors wish to thank the anonymous reviewers and associate editors for their comments, which improved the exposition of this paper. The research of U. Ayesta, T. Bodas and I.M. Verloop was partially supported by the French Agence Nationale de la Recherche (ANR) through the project ANR-15-CE25-0004 (ANR JCJC RACON). The research of U. Ayesta was also funded by the Department of Education of the Basque Government through the Consolidated Research Group MATHMODE (IT1294-19). The research of J.L. Dorsman was funded by the NWO Gravitation project NETWORKS, grant number 024.002.003.

References

[1] I. J. B. F. Adan, A. Busic, J. Mairesse, and G. Weiss. Reversibility and further properties of FCFS infinite bipartite matching. *Mathematics of Operations Research*, 43(2):598–621, 2018.

[2] I. J. B. F. Adan, C. Hurkens, and G. Weiss. A reversible Erlang loss system with multitype customers and multitype servers. *Probability in the Engineering and Informational Sciences*, 24(4):535–548, 2010.

[3] I. J. B. F. Adan, I. Kleiner, R. Righter, and G. Weiss. FCFS parallel service systems and matching models. *Performance Evaluation*, 127–128:253–272, 2018.

[4] I. J. B. F. Adan, R. Righter, and G. Weiss. FCFS parallel service systems and matching models. In *Proceedings of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools (Valuetools)*, pages 106–112. ACM, 2017.

[5] I. J. B. F. Adan and G. Weiss. A skill based parallel service system under FCFS-ALIS: steady state, overloads, and abandonments. *Stochastic Systems*, 4(1):250–299, 2014.

[6] U. Ayesta, T. Bodas, and I. M. Verloop. On a unifying product form framework for redundancy models. *Performance Evaluation*, 127-128:93 – 119, 2018.

[7] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22(2):248–260, 1975.

[8] T. Bonald and C. Comte. Balanced fair resource sharing in computer clusters. *Performance Evaluation*, 116:70–83, 2017.

[9] X. Chao. Networks with customers, signals, and product form solutions. In R. J. Boucherie and N. M. Van Dijk, editors, *Queueing Networks*, volume 154 of *International Series in Operations Research & Management Science*, pages 217–268. Springer US, 2011.

[10] C. Comte. Dynamic load balancing with tokens. In *Proceedings of the 17th International IFIP TC6 Networking Conference*, IFIP Networking 2018, pages 343–351, Piscataway, NJ, USA, 2018. Institute of Electrical and Electronics Engineers.

[11] C. Comte. Dynamic load balancing with tokens. *Computer Communications*, 144:76–88, 2019.

[12] S. Crosby and A. E. Krzesinski. Product from solutions for multiserver centres with concurrent classes of customers. *Performance Evaluation*, 11(4):265–281, 1990.

[13] F. G. Foster. On the stochastic matrices associated with certain queuing processes. *The Annals of Mathematical Statistics*, 24(3):355–360, 1953.

[14] K. Gardner, M. Harchol-Balter, A. Scheller-Wolf, M. Velednitsky, and S. Zbarsky. Redundancy-d: The power of d choices for redundancy. *Operations Research*, 65(4):1078–1094, 2017.

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

[16] H. Gumbel. Waiting lines with heterogeneous servers. *Operations Research*, 8(4):504–511, 1960.

[17] J.R. Jackson. Networks of waiting lines. *Operations Research*, 5:516–523, 1957.

[18] J. Keilson and L. D. Servi. The distributional form of Little's law and the Fuhrmann-Cooper decomposition. *Operations Research Letters*, 9:239–247, 1990.

[19] F. P. Kelly. *Stochastic Networks and Reversibility*. Wiley, Chichester, 1979.

[20] A. E. Krzesinski. Order independent queues. In R. J. Boucherie and N. M. Van Dijk, editors, *Queueing Networks*, volume 154 of *International Series in Operations Research & Management Science*, pages 85–120. Springer US, 2011.

[21] J. Le Boudec. A BCMP extension to multiserver stations with concurrent classes of customers. In *Proceedings of ACM SIGMETRICS*, pages 78–91, 1986.

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

A Completion of proof of Theorem 1

Proof. We complete the proof of Theorem 1 by showing that (10) satisfies (8). To this end, we follow the proof techniques of [22, Theorem 2]. In particular, we show below that (10) satisfies the equations

$$\begin{aligned}\mu(x)\pi(x) = {} & \mathbb{1}_{\{n_i>0\}}\lambda_{\mathcal{U}(\{T_1,\dots,T_i\})}\pi((T_1,n_1,\dots,T_i,n_i-1)) \\ & + \mathbb{1}_{\{n_i=0\}}\lambda_{T_i}(\{T_1,\dots,T_{i-1}\})\pi(T_1,n_1,\dots,T_{i-1},n_{i-1}),\end{aligned} \tag{26}$$

$$\lambda_T(\{T_1,\dots,T_i\})\pi(x) = \sum_{k=0}^{i}\sum_{n=0}^{n_k}\mu_T(\text{release}_{k,n}(x,T))r_{k,n}(x,T)\pi(\text{release}_{k,n}(x,T)) \tag{27}$$

and

$$\lambda_{\mathcal{U}(\{T_1,\dots,T_i\})}\pi(x) = \sum_{j=1}^{i}\sum_{k=0}^{j-1}\sum_{n=0}^{n_k}\mu_{T_j}(\text{shift}_{k,n}(x,T_j))s_{k,n}(x,T_j)\pi(\text{shift}_{k,n}(x,T_j)). \tag{28}$$

Next, summing (27) over all available tokens $T \in \mathcal{T}\backslash\{T_1, \ldots, T_i\}$ and adding (26) and (28), we conclude using (1) that (10) satisfies (8). The theorem then follows.

It is straightforward using (4) to see that (10) satisfies (26), in accordance with the reasoning in [22, Section 3.5]. We next show that (10) satisfies (27). By substitution of (10) in the right-hand side of (27) and subsequent rewriting, we obtain

$$
\begin{aligned}
&\sum_{k=0}^{i}\sum_{n=0}^{n_k}\mu_T(\text{release}_{k,n}(x,T))r_{k,n}(x,T)\pi(\text{release}_{k,n}(x,T)) \\
&= \sum_{k=0}^{i}\sum_{n=0}^{n_k}\eta(\phi(x)+1)(k(T_1,\ldots,T_k,T)-k(T_1,\ldots,T_k))\beta_k(T)^n\left(\prod_{j=k+1}^{i}\beta_j(T)^{n_j}\right)\times \\
&\qquad\times\pi((0))\frac{\Pi_\lambda(\{T_1,\ldots,T_k,T,T_{k+1},\ldots,T_i\})}{\Pi_k(T_1,\ldots,T_k,T,T_{k+1},\ldots,T_i)}\left(\prod_{j=1}^{k-1}\alpha_j^{n_j}\right)\alpha_k^{n_k-n}\left(\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_k,T\})}}{k(T_1,\ldots,T_k,T)}\right)^n\times \\
&\qquad\times\left(\prod_{j=k+1}^{i}\left(\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_j,T\})}}{k(T_1,\ldots,T_k,T,T_{k+1},\ldots,T_j)}\right)^{n_j}\right)\prod_{j=1}^{\phi(x)+1}\frac{1}{\eta(j)} \\
&= \pi(0)\left(\prod_{j=1}^{\phi(x)}\frac{1}{\eta(j)}\right)\sum_{k=0}^{i}(k(T_1,\ldots,T_k,T)-k(T_1,\ldots,T_k))\frac{\Pi_\lambda(\{T_1,\ldots,\ldots,T_i,T\})}{\Pi_k(T_1,\ldots,T_k,T,T_{k+1},\ldots,T_i)}\left(\prod_{j=1}^{k-1}\alpha_j^{n_j}\right)\times \\
&\qquad\times\left(\prod_{j=k+1}^{i}\left(\frac{\beta_j(T)\lambda_{\mathcal{U}(\{T_1,\ldots,T_j,T\})}}{k(T_1,\ldots,T_k,T,T_{k+1},\ldots,T_j)}\right)^{n_j}\right)\sum_{n=0}^{n_k}\alpha^{n_k-n}\left(\frac{\beta_k(T)\lambda_{\mathcal{U}(\{T_1,\ldots,T_k,T\})}}{k(T_1,\ldots,T_k,T)}\right)^n \\
&= \lambda_T(\{T_1,\ldots,T_i\})\pi(0)\frac{\Pi_\lambda(\{T_1,\ldots,T_i)\}}{\Pi_k(T_1,\ldots,T_k)}\left(\prod_{j=1}^{i}\alpha_j^{n_j}\right)\left(\prod_{j=1}^{\phi(x)}\frac{1}{\eta(j)}\right)\times \\
&\qquad\times\sum_{k=0}^{i}\frac{k(T_1,\ldots,T_k,T)-k(T_1,\ldots,T_k)}{k(T_1,\ldots,T_k,T)}\prod_{j=k+1}^{i}\left(\frac{\lambda_{\mathcal{U}}(\{T_1,\ldots,T_j\})}{\alpha_j k(T_1,\ldots,T_j,T)}\right)^{n_j}\times \\
&\qquad\times\prod_{j=k+1}^{i}\frac{k(T_1,\ldots,T_j)}{k(T_1,\ldots,T_j,T)}\sum_{n=0}^{n_k}\left(\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_k\})}}{\alpha_k k(T_1,\ldots,T_k,T)}\right)^n \\
&= \lambda_T(\{T_1,\ldots,T_i\})\pi(x)\sum_{k=0}^{i}\frac{k(T_1,\ldots,T_k,T)-k(T_1,\ldots,T_k)}{k(T_1,\ldots,T_k,T)}\times \\
&\qquad\times\prod_{j=k+1}^{i}\left(\frac{k(T_1,\ldots,T_j)}{k(T_1,\ldots,T_j,T)}\right)^{n_j+1}\sum_{n=0}^{n_k}\left(\frac{k(T_1,\ldots,T_k)}{k(T_1,\ldots,T_k,T)}\right)^n \\
&= \lambda_T(\{T_1,\ldots,T_i\})\pi(x)\sum_{k=0}^{i}\frac{k(T_1,\ldots,T_k,T)-k(T_1,\ldots,T_k)}{k(T_1,\ldots,T_k,T)}.
\end{aligned}
$$

where similar arguments are used as in [22, pp. 288–289], such as the use of Condition 1 in the second equality. The major difference stems from the fact that now Condition 2 and the straightforwardly verifiable fact that $\Pi_k(T_1,\ldots,T_j,T,T_{k+1},\ldots,T_i) = k(T_1,\ldots,T_i,T)\Pi_k(T_1,\ldots,T_i)\prod_{j=k+1}^{i}\frac{k(T_1,\ldots,T_j,T)}{k(T_1,\ldots,T_j)}$ is needed for the third equality to hold true. To verify the last equality, one can see that the outer sum of this line indeed equals one can be done by straightforward algebraic manipulation, but the probabilistic argument of [22, p. 289] can also be used.

Finally, we show that (10) satisfies (28) in a similar way, using arguments of [22, pp. 289–290]. By manipulation of the right-hand side of (28), we obtain

$$
\sum_{j=1}^{i}\sum_{k=0}^{j-1}\sum_{n=0}^{n_k}\mu_{T_j}(\text{shift}_{k,n}(x,T_j))s_{k,n,}(s,T_j)\pi(\text{shift}_{k,n}(x,T_j)))
$$

$$
\begin{aligned}
&= \sum_{j=1}^{i}\sum_{k=0}^{j-1}\sum_{n=0}^{n_k}\eta(\phi(x)+1)\left(k(T_1,\ldots,T_k,j)-k(T_1,\ldots,T_k)\right)\beta_k(T_j)^n\left(\prod_{l=k+1}^{j-1}\beta_l(T_j)^{n_l}\right)(1-\beta_{j-1}(T_j))\times\\
&\qquad\times\pi((0))\frac{\Pi_\lambda(\{T_1,\ldots,T_k,T_j,T_{k+1},\ldots,T_{j-1},T_{j+1},\ldots,T_i\})}{\Pi_k(T_1,\ldots,T_k,T_j,T_{k+1},\ldots,T_{j-1},T_{j+1},\ldots,T_i)}\left(\prod_{l=1}^{k-1}\alpha_l^{n_l}\right)\alpha_k^{n_k-n}\times\\
&\qquad\times\left(\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_k,T_j\})}}{k(T_1,\ldots,T_j,T_k)}\right)^n\left(\prod_{l=k+1}^{j-1}\left(\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_l,T_j\})}}{k(T_1,\ldots,T_k,T_j,T_{k+1},\ldots,T_l)}\right)^{n_l}\right)\times\\
&\qquad\times\left(\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_j\})}}{k(T_1,\ldots,T_k,T_j,T_{k+1},\ldots,T_{j-1})}\right)^{n_j+1}\times\\
&\qquad\times\left(\prod_{l=j+1}^{i}\left(\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_l\})}}{k(T_1,\ldots,T_k,T_j,T_{k+1},\ldots,T_{j-1},T_{j+1},\ldots,T_l)}\right)^{n_l}\right)\prod_{l=1}^{\phi(x)+1}\frac{1}{\eta(l)}.\\
&= \pi((0))\frac{\prod_\lambda(\{T_1,\ldots,T_i\})}{\prod_k(T_1,\ldots,T_i)}\left(\prod_{j=1}^{\phi(x)}\frac{1}{\eta(j)}\right)\sum_{j=1}^{i}\left(1-\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_{j-1}\})}}{\lambda_{\mathcal{U}(\{T_1,\ldots,T_j\})}}\right)\sum_{k=0}^{j-1}\left(k(T_1,\ldots,T_k,T_j)-k(T_1,\ldots,T_k)\right)\times\\
&\qquad\times\left(\frac{\prod_{l=k+1}^{j}k(T_1,\ldots,T_l)}{\prod_{l=k}^{j-1}k(T_1,\ldots,T_l,T_j)}\right)\left(\prod_{l=1}^{k-1}\alpha_l^{n_l}\right)\left(\prod_{l=k+1}^{j-1}\left(\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_l,T_j\})}}{k(T_1,\ldots,T_l,T_j)}\right)^{n_l}\right)\left(\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_j\})}}{k(T_1,\ldots,T_j)}\right)^{n_j+1}\times\\
&\qquad\times\left(\prod_{l=j+1}^{i}\left(\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_l\})}}{k(T_1,\ldots,T_l)}\right)^{n_l}\right)\sum_{n=0}^{n_k}\alpha_k^{n_k-n}\left(\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_k\})}}{k(T_1,\ldots,T_k,T_j)}\right)^n\\
&= \pi((0))\frac{\prod_\lambda(\{T_1,\ldots,T_i\})}{\prod_k(T_1,\ldots,T_i)}\prod_{j=1}^{i}\alpha_j^{n_j}\left(\prod_{j=1}^{\phi(x)}\frac{1}{\eta(j)}\right)\sum_{j=1}^{i}\left(\lambda_{\mathcal{U}(\{T_1,\ldots,T_j\})}-\lambda_{\mathcal{U}(\{T_1,\ldots,T_{j-1}\})}\right)\times\\
&\qquad\times\sum_{k=0}^{j-1}\left(1-\frac{k(T_1,\ldots,T_k)}{k(T_1,\ldots,T_k,T_j)}\right)\left(\prod_{l=k+1}^{j-1}\frac{k(T_1,\ldots,T_l)}{k(T_1,\ldots,T_l,T_j)}\right)\left(\prod_{l=k+1}^{j-1}\left(\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_l,T_j\})}}{\alpha_j k(T_1,\ldots,T_l,T_j)}\right)^{n_l}\right)\times\\
&\qquad\times\sum_{n=0}^{n_k}\left(\frac{\lambda_{\mathcal{U}(\{T_1,\ldots,T_k\})}}{\alpha_k k(T_1,\ldots,T_k,T_j)}\right)^n\\
&= \pi(x)\sum_{j=1}^{i}\left(\lambda_{\mathcal{U}(\{T_1,\ldots,T_j\})}-\lambda_{\mathcal{U}(\{T_1,\ldots,T_{j-1}\})}\right)\times\\
&\qquad\times\sum_{k=0}^{j-1}\left(1-\frac{k(T_1,\ldots,T_k)}{k(T_1,\ldots,T_k,T_j)}\right)\left(\prod_{l=k+1}^{j-1}\left(\frac{k(T_1,\ldots,T_l)}{k(T_1,\ldots,T_l,T_j)}\right)^{n_l+1}\right)\sum_{n=0}^{n_k}\left(\frac{k(T_1,\ldots,T_k)}{k(T_1,\ldots,T_k,T_j)}\right)^n\\
&= \lambda_{\mathcal{U}(\{T_1,\ldots,T_i\})}\pi(x),
\end{aligned}
$$

which is the left-hand side of (28). In the second equality, Conditions 1 and 2 are used. The final equality follows by using the fact that $\sum_{j=1}^{i}\left(\lambda_{\mathcal{U}(\{T_1,\ldots,T_j\})}-\lambda_{\mathcal{U}(\{T_1,\ldots,T_{j-1}\})}\right)=\lambda_{\mathcal{U}(\{T_1,\ldots,T_i\})}$. The sum involving the k-terms can again be shown to equal one by algebraic manipulation or the probabilistic argument of [22, p. 289]. As we have now rewritten the right-hand side of (28) into its left-hand side, we conclude that (10) satisfies (28), which completes the proof. □

B Proof of Theorem 2

Proof. The proof consists of applying the steps of [22, Section 4] to the token-based central queue. That is, we will consider $N_j^{(c)}$, the number of type-c customers among the N_j customers between those that have claimed T_j and T_{j+1}, out of which an expression for $\{N^{(c)}, c\in\mathcal{C}$ will follow.

From Theorem 1, we gather that the stationary distribution of the model at hand satisfies

$$\pi(T_1, n_1, \ldots, T_i, n_i) = \pi((0)) \frac{\prod_\lambda(\{T_1, \ldots, T_i\})}{\prod_k(T_1, \ldots, T_i)} \prod_{j=1}^{i} \alpha_j^{n_j} \prod_{j=1}^{i} \frac{1}{\eta(j)} \prod_{j=1}^{\sum_{k=1}^{i} n_k} \frac{1}{\eta(i+j)}. \tag{29}$$

By the dynamics of the arrival process, we next note that $N_j^{(c)}$ is binomially distributed with parameters N_j and $\theta_{c,j} := \frac{\lambda_c \mathbb{1}_{\{c \in \mathcal{U}(T_1, \ldots, T_j)\}}}{\lambda_{\mathcal{U}(T_1, \ldots, T_j)}}$. The indicator function in this expression reflects the fact that in order for $N_j^{(c)}$ to be positive, any token in the set $\mathcal{T} \backslash \{T_1, \ldots, T_j\}$ must reject type-c customers. More generally, the set $\{N_j^{(c)} : c \in \mathcal{C}\}$ is multinomially distributed with population size parameter N_j and probability parameters $\{\theta_{c,j} : c \in \mathcal{C}\}$. We also observe that, given the values of $N_1, N_2, \ldots$, the sets $\{N_1^{(c)} : c \in \mathcal{C}\}, \{N_2^{(c)} : c \in \mathcal{C}\}, \ldots$ are independent, so that

$$\mathbb{P}\left(\bigcap_{j \in \{1, \ldots, i\}, c \in \mathcal{C}} \{N_j^{(c)} = n_j^{(c)}\} \mid \bigcap_{j=1}^{i} \{N_j = n_j\} \right) = \prod_{j=1}^{i} \frac{n_j!}{\prod_{c \in \mathcal{C}} n_j^{(c)}!} \prod_{c \in \mathcal{C}} \theta_c^{n_{c,j}}. \tag{30}$$

Using (30) and applying Newton's binomium leads, for $z_{c,j} \in \{\bar{c} \in \mathbb{C} : |\bar{c}| \leq 1\}$, to

$$\begin{aligned} &\mathbb{E}\left[\prod_{c \in \mathcal{C}} \prod_{j=1}^{i} z_{c,j}^{N_j^{(c)}} \mid x = (T_1, n_1, \ldots, T_i, n_i) \right] \\ &= \sum_{\{n_j^{(c)} : c \in \mathcal{C}\} : \sum_{c \in \mathcal{C}} n_j^{(c)} = n_j} \frac{n_j!}{\prod_{c \in \mathcal{C}} n_j^{(c)}!} \prod_{c \in \mathcal{C}} (\theta_{c,j} z_{c,j})^{n_{c,j}} = \prod_{j=1}^{i} \left(\sum_{c \in \mathcal{C}} \theta_{c,j} z_{c,j} \right)^{n_j}. \end{aligned} \tag{31}$$

Unconditioning using (29) now leads to

$$\begin{aligned} &\mathbb{E}\left[\prod_{c \in \mathcal{C}} \prod_{j=1}^{K} z_{c,j}^{N_j^{(c)}} \right] \\ &= \sum_{i=0}^{K} \sum_{(T_1, \ldots, T_i) \in \mathcal{T}^i} \sum_{(n_1, \ldots, n_i) \in \mathbb{N}_0^i} \pi((T_1, n_1, \ldots, T_i, n_i)) \mathbb{E}\left[\prod_{c \in \mathcal{C}} \prod_{j=1}^{K} z_{c,j}^{N_j^{(c)}} \mid x = (T_1, n_1, \ldots, T_i, n_i) \right] \\ &= \sum_{i=0}^{K} \sum_{(T_1, \ldots, T_i) \in \mathcal{T}^i} \pi((0)) \frac{\prod_\lambda(\{T_1, \ldots, T_i\})}{\prod_k(T_1, \ldots, T_i)} \prod_{j=1}^{i} \frac{1}{\eta(j)} \sum_{\{n_1, \ldots, n_i\} \in \mathbb{N}_0^i} \prod_{j=1}^{\sum_{k=1}^{i} n_k} \frac{1}{\eta(i+j)} \prod_{j=1}^{i} (\alpha_j \sum_{c \in \mathcal{C}} \theta_{c,j} z_{c,j})^{n_j}, \end{aligned} \tag{32}$$

after which (14) follows since $\mathbb{E}\left[\prod_{c \in \mathcal{C}} z_c^{N^{(c)}}\right] = \mathbb{E}\left[\prod_{c \in \mathcal{C}} z_c^{\sum_{j=1}^{k} N_j^{(c)}}\right] = \mathbb{E}\left[\prod_{c \in \mathcal{C}} \prod_{j=1}^{k} z_c^{N_j^{(c)}}\right]$. □

C Proof of Theorem 4

Proof. We note that if there are at least j tokens activated, either $M_j^{(c)} = N_j^{(c)} + 1$ if token T_j is claimed by a type-c customer, or $M_j^{(c)} = N_j^{(c)}$ otherwise. This leads to

$$\begin{aligned} &\mathbb{E}\left[\prod_{c \in \mathcal{C}} \prod_{j=1}^{i} z_{c,j}^{M_j^{(c)}} \right] \\ &= \sum_{i=0}^{K} \sum_{(T_1, \ldots, T_i) \in \mathcal{T}^i} \sum_{(n_1, \ldots, n_i) \in \mathbb{N}_0^i} \pi((T_1, n_1, \ldots, T_i, n_i)) \mathbb{E}\left[\prod_{c \in \mathcal{C}} \prod_{j=1}^{i} z_{c,j}^{N_j^{(c)}} \mid x = (T_1, n_1, \ldots, T_i, n_i) \right] \times \\ &\qquad \times \sum_{(c_1, \ldots, c_i) \in \mathcal{C}^i} G_{c_1, \ldots, c_i}(T_1, n_1, \ldots, T_i, n_i) \prod_{j=1}^{i} z_{c,j} \end{aligned}$$

The theorem now follows by substitution of (29) and (31) into this expression and realising that $\mathbb{E}\left[\prod_{c\in\mathcal{C}} z_c^{M^{(c)}}\right] = \mathbb{E}\left[\prod_{c\in\mathcal{C}}\prod_{j=1}^{k} z_c^{M_j^{(c)}}\right]$. □

D Expressions for performance measures when $\eta(\cdot) = 1$

It follows by substitution and subsequent simplification of (14), (21), (22), (16) and (19) that, when $\eta(j) = 1$ for all $j \in \mathbb{N}$,

$$\mathbb{E}\left[\prod_{c\in\mathcal{C}} z_c^{N^{(c)}}\right] = \sum_{i=0}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \pi((0)) \frac{\prod_\lambda(\{T_1,\ldots,T_i\})}{\prod_k(T_1,\ldots,T_i)} \prod_{j=1}^{i} \frac{1}{1-\alpha_j \sum_{c\in\mathcal{C}} \theta_{c,j} z_c},$$

$$\mathbb{E}\left[z^N\right] = \sum_{i=0}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \pi((0)) \frac{\prod_\lambda(\{T_1,\ldots,T_i\})}{\prod_k(T_1,\ldots,T_i)} \prod_{j=1}^{i} \frac{1}{1-\alpha_j z},$$

$$\mathbb{E}\left[z^M\right] = \sum_{i=0}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \pi((0)) \frac{\prod_\lambda(\{T_1,\ldots,T_i\})}{\prod_k(T_1,\ldots,T_i)} z^i \prod_{j=1}^{i} \frac{1}{1-\alpha_j z}$$

and

$$\mathbb{E}\left[e^{-sW_c}\right] = \sum_{i=0}^{K} \sum_{(T_1,\ldots,T_i)\in\mathcal{T}^i} \pi((0)) \prod_{j=1}^{i} \frac{\lambda_{T_j}(T_1,\ldots,T_{j-1})}{k(T_1,\ldots,T_j) - \lambda_{\mathcal{U}(\{T_1,\ldots,T_j\})} + s\mathbb{1}_{\{c\in U(\{T_1,\ldots,T_j\})\}}},$$

with $\pi((0))$ as given in (12).

E Proof of Theorem 6

We will use the notion of indistinguishable tokens as introduced in Section 4.3, as well as the state descriptor of the form $x^{(L)} = (L_1, n_1, \ldots, L_i, n_i)$ and the corresponding steady-state distribution (cf. (13)). We start out with a preparatory lemma.

Lemma 8. *For any token-based central queue where each token set $\mathcal{T}_c$, $c \in \mathcal{C}$, consists of indistinguishable tokens, there exists a function $\tau : \mathcal{X}^{(OI)} \to \mathcal{X}^{(L)}$, where $\tau(x^{(OI)}) \in \mathcal{X}^{(L)}$ denotes the unique state $(L_1, n_1, \ldots, L_i, n_i)$ corresponding to the state $x^{(OI)} \in \mathcal{X}^{(OI)}$.*

Proof. Since each customer type has one token label it can select from, this guarantees that there is no ambiguity about how the tokens are distributed among the customers. By keeping track of the order of arrival and the token labels allotted to the customers, one can construct the unique state $x^{(L)} = (L_1, n_1, \ldots, L_i, n_i)$ corresponding to $x^{(OI)} = (c_1, \ldots, c_n)$, that is, the function $\tau(\cdot)$ as stated in the lemma exists. The quantity n_i represents customers without a claimed token. □

This lemma allows us to prove Theorem 6, which exposes the connection between OI queues and token-based central queues.

Proof. We first assume that (1) holds, that is, we are given a model that fits in the OI queue framework. In the remainder of this proof, we will use the notion of indistinguishable tokens and token labels as introduced in Section 4.3. We define the following token sets $\mathcal{T}_c$. Each token set of customer type c consists of an infinite number of indistinguishable tokens with label c. Thus, every customer type has its dedicated token label. Then, the state $x^{(L)} = (L_1, \ldots, L_n)$ gives exactly the same information as the state $x^{(OI)} = (c_1, \ldots, c_n)$, hence both state descriptors are equivalent. When setting $\mu_{L_j}(x^{(L)}) = \mu_j^{(OI)}(x^{(OI)})$, the token-based central queue describes exactly the same model as the OI queue.

What is left to show is that the token-based central queue satisfies Condition 1 and Condition 2. Condition 2 follows directly, since $\mu_{L_j}(x^{(L)}) = \mu_j^{(OI)}(x^{(OI)})$ and $\mu_j^{(OI)}$ satisfies Condition 3. Since each customer type c has its own dedicated set of indistinguishable tokens with label c, we have that $\lambda_c(\{L_1, \ldots, L_i\}) = \lambda_c$. Therefore,

$\prod_{j=1}^{i} \lambda_{L_j}(\{L_1, \dots L_{j-1}\}) = \prod_{j=1}^{i} \lambda_{L_j}$. This expression is independent of the permutation of the L_j's, and since tokens that bear the same label are indistinguishable, Condition 1 is satisfied. We have hence proved that (1) $\rightarrow$ (2).

We now assume that (2) of Theorem 6 holds, that is, we are given a token-based central queue where each token set consists of indistinguishable tokens. From Lemma 8, it follows that to a given state $x^{(OI)} = (c_1, \dots, c_n)$ (describing the type of each customer), there corresponds a unique state $x^{(L)} = (L_1, n_1, \dots, L_i, n_i)$, given by $\tau(x^{(OI)})$. We also define the function $\tilde{\tau}(x^{(OI)})$ that gives the unique activated tokens $(L_1, \dots, L_i)$ as a function of $x^{(OI)}$. To prove that (1) of Theorem 6 holds, that is, the model fits the OI queue framework, we will (i) define functions $\eta^{(OI)}(\cdot)$ and $s_j^{(OI)}(\cdot)$, (ii) show that these functions give rise to an OI queue and (iii) show that the departure rates under the token-based central queue and the OI queue are sample-path wise equal.

(i) Since $\phi(\tau(x^{(OI)})) = n$, we define

$$\eta^{(OI)}(n) := \eta(\phi(\tau(x^{(OI)}))).$$

Let $h(j, x^{(OI)}) := \sum_{c \in \mathcal{C}} \min(\sum_{l=1}^{j} \mathbf{1}_{(c_l = c)}, |\mathcal{T}_c|)$ denote the number of active customers among the first j customers (for ease of exposition, we assume that any two tokens from any two token sets $\mathcal{T}_{c_a}$ and $\mathcal{T}_{c_b}$, $c_a, c_b \in \mathcal{C}$, are not indistinguishable). When in state $x^{(OI)} = (c_1, \dots, c_n)$ and if $\sum_{i=1}^{j} \mathbf{1}_{(c_j = c_i)} \leq |\mathcal{T}_{c_j}|$, then the j-th customer is the $h(j, x^{(OI)})$-th customer that has a token. We therefore define

$$s_j^{(OI)}(x^{(OI)}) := \begin{cases} s_{h(j,x^{(OI)})}(\tilde{\tau}(x^{(OI)})) \text{ if } \sum_{i=1}^{j} \mathbf{1}_{(c_j = c_i)} \leq |\mathcal{T}_{c_j}|, \\ 0, \text{ otherwise.} \end{cases}$$

(ii) Since Condition 2 is satisfied, it is immediate that $\mu(x^{(OI)})$ satisfies Condition 3 and hence gives rise to an OI queue.

(iii) Consider the j-th customer. We now show that its departure rate in both systems is the same, which concludes the proof. If $\sum_{i=1}^{j} \mathbf{1}_{(c_j = c_i)} \leq |\mathcal{T}_{c(j)}|$, then the j-th customer has a token and is the $h(j, x^{(OI)})$-th active customer in the queue. Its departure rate in the OI queue is $\mu_j^{(OI)}(x^{(OI)}) = \eta^{(OI)}(n) s_j^{(OI)}(x^{(OI)}) = \eta(\phi(\tau(x^{(OI)}))) s_{h(j,x^{(OI)})}(\tilde{\tau}(x^{(OI)})) = \mu_{L_{h(j,x^{(OI)})}}(\tau(x^{(OI)}))$, which equals its departure rate in the token-based central queue. If the j-th customer is not active, then its departure rate in the OI queue is $\mu_j^{(OI)}(x^{(OI)}) = 0$, which equals its departure rate in the token-based central queue. □

F Proof of Corollary 7

Proof. In the proof of Theorem 6 (1) $\rightarrow$ (2) it was shown that the OI queue can be seen as a token-based central queue where a token set $\mathcal{T}_c$ of a customer type c consists of infinitely many indistinguishable tokens with label c. Noting that $\phi(x) = n$, $n_j = 0$ and $\lambda_c(\{L_1, \dots, L_j\}) = \lambda_c$, from (13) we recover the product-form stationary distribution (25) for the OI state descriptor. □