Faculty of Mathematical Sciences

University of Twente

University for Technical and Social Sciences

P.O. Box 217
7500 AE Enschede
The Netherlands
Phone: +31-53-4893400
Fax: +31-53-4893114
Email: memo@math.utwente.nl

Memorandum No. 1590

# Arrival first queueing networks with applications in kanban production systems

R.J. Boucherie, X. Chao[1] and M. Miyazawa[2]

October 2001

ISSN 0169-2690

[1]Department of Industrial Engineering, North Carolina State University, Raleigh, NC 27695-7906, U.S.A.
[2]Department of Information Sciences, Science University of Tokyo, Noda, Chiba 278, Japan

# Arrival first queueing networks with applications in kanban production systems

Richard J. Boucherie*
Faculty of Mathematical Sciences
University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands

Xiuli Chao†
Department of Industrial Engineering
North Carolina State University
Raleigh, NC 27695-7906, U.S.A.

Masakiyo Miyazawa
Department of Information Sciences
Science University of Tokyo
Noda, Chiba 278, Japan

**Abstract**

In this paper we introduce a new class of queueing networks called *arrival first networks*. We characterise its transition rates and derive the relationship between arrival rules, linear partial balance equations, and product form stationary distributions. This model is motivated by production systems operating under a kanban protocol. In contrast with the conventional *departure first networks*, where a transition is initiated by service completion of items at the originating nodes that are subsequently routed to the destination nodes (push system), in an arrival first network a transition is initiated by the destination nodes of the items and subsequently those items are processed at and removed from the originating nodes (pull system). These are similar to the push and pull systems in manufacturing systems.

Our characterisation provides necessary and sufficient conditions for the network to possess linear traffic equations, and sufficient conditions for the network to have a product form stationary distribution. We apply our results to networks operating under a kanban mechanism and characterise the rate at which items are pulled as well as the routing and blocking protocols that give rise to a product form stationary distribution.

**Keywords:** Arrival first networks, arrival rules, kanban, partial balance, product form solutions, pull system.

**AMS 1991 Subject classification:** Primary 60K25, Secondary 60J27.

*Research partially supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs, The Netherlands.
†Research partially supported by NSF under grant DMI-9908294.

# 1 Introduction

Over the last decades important new approaches have appeared in operations planning and control of production systems. Among these are materials requirements planning (MRP), kanban or just-in-time (JIT), and optimized production technology (OPT), see e.g. [7]. These innovative methods have changed the practice not only in manufacturing industries, but also in the service sectors. In contrast with these rapid developments, stochastic models used in operations research for analysing performance of these new methods have not reached a standard similar to that developed for classical production systems. This paper is motivated by and presents a stochastic model for kanban production systems.

Consider a production facility consisting of multiple machines or work stations. Raw parts arrive at the facility requiring operations at multiple stations, one after another. In the classical approach, work is driven by the availability of parts. A job starts upon arrival of the required parts, and when a job completes processing at one station, it is transferred to the buffer or queue of the next station. This procedure is referred to as a *push* system: jobs are pushed from one machine into the buffer of the next machine; the completion epochs of the job are dependent on their arrival epochs at the stations.

In contrast with push systems, under the kanban protocol the arrival epoch of a job at a station is determined by the desired completion epoch at that station. At each work station only those subcomponents that have been requested at the next stage of production are produced. This procedure is referred to as *pull* system: a work station pulls subcomponents from the previous stage on the route of a job; operations occur as they are needed or demanded. For example, when a worker on a production line begins drawing from a new bin, he removes the label (kanban in Japanese) from the bin and routes it back to the supplying work station, where it serves as an order for a new bin of parts. An important advantage of the kanban or just-in-time protocol, over the classical approach, is the reduction of inventory kept at the work stations. Additional advantages of the just-in-time protocol, such as increased flexibility and better quality of the production facility, are perhaps more related to a change in attitude when working under the JIT environment.

For push systems, the conventional class of queueing networks with tractable (e.g., product form) solution for their stationary distribution has been successfully used for performance analysis, e.g., Jackson networks [15] and BCMP networks [1]. In this respect partial balance equations have contributed to the development of a unified approach. The classical partial balance equations underlying networks of the Jackson type, see e.g. [20], have been generalized to capture other phenomena, such as *batch routing* (e.g. [6, 13, 18]), and networks with *signals* and *negative customers* (see e.g. [10]). The common feature of these networks is that a transition of the network is initiated by (a batch of) items departing from the nodes of the network. In the second step of the transition the items are routed to their destination nodes. In what follows we will refer to these networks as *departure first networks*. For obvious reasons departure first networks and arrival first networks can also be referred to as *push networks* and *pull networks*.

Due to the success of departure first queueing network models for performance analysis of classical job-shops and computer communication systems, models of the BCMP type have also been applied for the analysis of pull systems (e.g., [11, 19]). The processing of a job in [19] is initiated by service completions, i.e., by a push approach. The pull behaviour is modelled by assuming the network to be closed, which triggers an arrival to the network upon a service completion resulting in the departure of a job from the system. This is referred to as a constant work-in-progress strategy as the total number of jobs present in the system is constant. The stochastic model of the present paper avoids this assumption and enables implementation of pull behaviour at each station of the system. Almost all the research effort to evaluate pull systems using push networks resulted in analytically intractable solutions, and decomposition approximations are usually utilized for performance evaluation.

Networks with a transition structure *complementary* to those of Jackson type networks were first introduced in [2]. In these networks a transition is initiated by (a batch of) items arriving at the nodes, followed by departures from the nodes. This can be interpreted as items being pulled out of the nodes by the destination node. This general class of networks will be referred to as *arrival first networks,* and was used to model production systems operating under a just-in-time protocol in [3]. The aim of the present paper is to provide a *complete characterisation* of the structure of arrival first networks.

At first sight, departure first and arrival first networks with batch routing show similar behaviour. In fact, by allowing batches to become negative, the departure first network of [13] seems to transform into an arrival first network. This, however, is only part of the picture. By allowing batches to become negative the order in which the transition takes place is altered, which may considerably influence the boundary behaviour of the network, and results in a state dependent routing process.

In this paper we use a framework similar to that used in [18], also see [6, 8, 13, 17], to study arrival first networks. We obtain *necessary and sufficient conditions for a closed form stationary distribution* and a *characterisation of the* corresponding *arrival rule* for arrival first networks. Moreover, we provide a detailed study of product form preserving blocking protocols for these networks. Finally these results are applied to study kanban production systems.

This paper aims at developing an understanding of arrival first networks that might enable theoretical results similar to those for classical networks, and is organized as follows. After the general mathematical formulation in the next section, as a first step towards a complete characterisation, Section 3 further develops the notion of *backward local balance* first introduced in [2], and provides a general characterisation of arrival rules resulting in backward local balance equations. The special cases of product form solutions are discussed in Section 4. Product form preserving blocking protocols are investigated in Section 5, where the results of the paper are applied to obtain closed form expressions for the stationary distribution of manufacturing systems with finite queues operating under kanban production protocols.

# 2 Model Description

This section lays down the framework for arrival first networks.

Consider a continuous-time queueing network consisting of $N$ nodes with $I$ types of items. Let $X_t(j,u)$ be the number of type $u$ items in node $j$ at time $t$, and define

$$\boldsymbol{X}_t(j) = (X_t(j,1), X_t(j,2), \ldots, X_t(j,I))$$

for each node $j$, $j = 1, \ldots, N$. The state of the network at time $t$ is then given by

$$\boldsymbol{X}_t = (\boldsymbol{X}_t(1), \ldots, \boldsymbol{X}_t(N)).$$

Let $\mathcal{S}$ contain all admissible states of the queueing network, $\mathcal{S} \subset \{0, 1, 2, \ldots\}$. An element $\boldsymbol{n} \in \mathcal{S}$ is referred to as a network state.

Let $\{\boldsymbol{X}_t,\ t \geq 0\}$ be the stochastic process with state space $\mathcal{S}$ recording the state of the queueing network; $\boldsymbol{X}_t$ makes a transition each time the network changes its state. The type of an item may change as the item transfers from one node to another. Assume that $\boldsymbol{X}_t$ is regular and left-continuous in $t$. The network changes its state with rate $\mu(\boldsymbol{n})$ when $\boldsymbol{X}_t = \boldsymbol{n}$. Denote the instants of the state changes by $\tau(k)$, $k = 1, 2, \cdots$, and let $\tau(0) = 0$. At each $\tau(k)$, there are associated arrivals and departures. The arrival and departure processes are defined in terms of the random vectors:

$$\begin{aligned} \boldsymbol{A}_k &= (A_k(0), \boldsymbol{A}_k(1), \boldsymbol{A}_k(2), \ldots, \boldsymbol{A}_k(N)), \\ \boldsymbol{D}_k &= (D_k(0), \boldsymbol{D}_k(1), \boldsymbol{D}_k(2), \ldots, \boldsymbol{D}_k(N)), \end{aligned}$$

where $A_k(0)$ and $D_k(0)$ are the number of items arriving to and departing from the outside at time $\tau(k)$, respectively, and

$$\begin{aligned} \boldsymbol{A}_k(i) &= (A_k(i,1), A_k(i,2), \ldots, A_k(i,I)), \\ \boldsymbol{D}_k(i) &= (D_k(i,1), D_k(i,2), \ldots, D_k(i,I)), \quad i = 1, \ldots, N, \end{aligned}$$

where $A_k(i,u)$ and $D_k(i,u)$ are the number of type $u$ items arriving to and departing from node $i$ at time $\tau(k)$, respectively. The random vectors $\{\boldsymbol{A}_k\}$ and $\{\boldsymbol{D}_k\}$ are assumed to take values in space $\mathcal{A} \subset \mathbb{Z}_+^{NI}$. Note that $A_k(0)$ is the total number of the departures from the network, while $D_k(0)$ is the total number of the arrivals to the network. For $\boldsymbol{a} \in \mathcal{A}$, define the operator $^+$ as the operator deleting the first element of $\boldsymbol{a}$, i.e.,

$$\boldsymbol{a}^+ = (\boldsymbol{a}(1), \ldots, \boldsymbol{a}(N))$$

for

$$\boldsymbol{a} = (a(0), \boldsymbol{a}(1), \ldots, \boldsymbol{a}(N)).$$

Thus, for example, $\boldsymbol{A}_k^+$ describes the arrivals at nodes in the network at the $k$-th instant of state changes.

The evolution of the network is determined by the relation between $\{\boldsymbol{A}_k\}$, $\{\boldsymbol{D}_k\}$, and the network state $\boldsymbol{X}_{\tau(k)}$. To this end, two alternative dynamics can be formulated,

depending on the sequence according to which events take place. These dynamics will be referred to as *arrival first* and *departure first* dynamics.

**Definition 2.1 (Arrival first dynamics).** *Under arrival first dynamics, at transition epoch $\tau(k)$, first batch arrivals occur, then batch departures are triggered. For initial state $\boldsymbol{X}_0$, the network states are recursively determined by*

$$\boldsymbol{X}_{\tau(k+1)} = (\boldsymbol{X}_{\tau(k)} + \boldsymbol{A}_k^+) - \boldsymbol{D}_k^+, \qquad k \geq 0. \tag{1}$$

*The following stochastic assumptions determine the dynamics of the network.*

(2.1.1) *The arrival vector $\boldsymbol{A}_k$, given $\boldsymbol{X}_{\tau(k)}$ and the history of the process up to time $\tau(k)$, depends on $\boldsymbol{X}_{\tau(k)}$ only. This conditional arrival probability is denoted by*

$$b(\boldsymbol{n}, \boldsymbol{a}) = P\{\boldsymbol{A}_k = \boldsymbol{a} | \boldsymbol{X}_{\tau(k)} = \boldsymbol{n}\}.$$

(2.1.2) *At time $\tau(k)$, the departure vector $\boldsymbol{D}_k$ depends on $\boldsymbol{X}_{\tau(k)}$ and $\boldsymbol{A}_k$ only. The conditional departure probability is denoted by*

$$r_a((\boldsymbol{n}, \boldsymbol{a}), (\boldsymbol{n}', \boldsymbol{a}')) = P\{\boldsymbol{X}_{\tau(k+1)} = \boldsymbol{n}', \boldsymbol{D}_k = \boldsymbol{a}' | \boldsymbol{X}_{\tau(k)} = \boldsymbol{n}, \boldsymbol{A}_k = \boldsymbol{a}\}.$$

In $r_a((\boldsymbol{n}, \boldsymbol{a}), (\boldsymbol{n}', \boldsymbol{a}'))$ the state $\boldsymbol{n}' \equiv \boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+$ is included only for convenience of notation. We will further assume that $r_a((\boldsymbol{n}, \boldsymbol{a}), (\boldsymbol{n}', \boldsymbol{a}')) > 0$ only if $b(\boldsymbol{n}, \boldsymbol{a}) > 0$ and $|\boldsymbol{a}| = |\boldsymbol{a}'|$, where

$$|\boldsymbol{a}| = a(0) + \sum_{j=1}^{N} \sum_{u=1}^{I} a(j, u),$$

i.e., items cannot be lost in a transition.

The functions $b$ and $r_a$ are respectively referred to as *arrival* function and *routing* function or probability as they describe the probability for a batch arrival $\boldsymbol{a}$ and the probability that an arriving batch $\boldsymbol{a}$ is routed to generate a departure batch $\boldsymbol{a}'$, respectively. Under the arrival first dynamics $\{\boldsymbol{X}_t\}$ is a Markov process with transition rates $q_{\mathcal{S}}$ given by

$$q_{\mathcal{S}}(\boldsymbol{n}, \boldsymbol{n}') = \sum_{\boldsymbol{a}', \boldsymbol{a}} \mu(\boldsymbol{n}) b(\boldsymbol{n}, \boldsymbol{a}) r_a((\boldsymbol{n}, \boldsymbol{a}), (\boldsymbol{n}', \boldsymbol{a}')), \qquad \boldsymbol{n}, \boldsymbol{n}' \in \mathcal{S}\,. \tag{2}$$

**Remark 2.2.** The arrival and departure vectors $\boldsymbol{A}_k$ and $\boldsymbol{D}_k$ are indexed by $k$ only. Therefore, the Markov chain embedded at transition epochs has properties similar to $\{\boldsymbol{X}_t\}$. This discrete-time Markov chain can also be obtained by setting $\mu(\boldsymbol{n}) = 1$ and redefining $\boldsymbol{X}_k = \boldsymbol{X}_{\tau(k)}$. As a consequence, our results also go through for discrete-time models. □

In the standard formulation of a queueing network, at each transition epoch first items are served and leave the nodes, and then these items route among the nodes. This formulation is presented in the following definition.

**Definition 2.3 (Departure first dynamics).** *Under departure first dynamics, at a transition epoch $\tau(k)$ first batch departures occur, and then batch arrivals are triggered. For initial state $\boldsymbol{X}_0$, the network states are recursively determined by*

$$\boldsymbol{X}_{\tau(k+1)} = (\boldsymbol{X}_{\tau(k)} - \boldsymbol{D}_k^+) + \boldsymbol{A}_k^+, \quad k \geq 0. \tag{3}$$

*The following stochastic assumptions determine the dynamics of the network.*

(2.2.1) *The conditional probability of $\boldsymbol{D}_k$, given $\boldsymbol{X}_{\tau(k)}$ and the history of the process up to time $\tau(k)$, depends only on the network state $\boldsymbol{X}_{\tau(k)}$, i.e.,*

$$d(\boldsymbol{n}, \boldsymbol{a}) = P\{\boldsymbol{D}_k = \boldsymbol{a} | \boldsymbol{X}_{\tau(k)} = \boldsymbol{n}\}.$$

(2.2.2) *The arrival vector $\boldsymbol{A}_k$ depends on $\boldsymbol{X}_{\tau(k)}$ and $\boldsymbol{D}_k$ only:*

$$r_d((\boldsymbol{n}, \boldsymbol{a}), (\boldsymbol{n}', \boldsymbol{a}')) = P\{\boldsymbol{X}_{\tau(k+1)} = \boldsymbol{n}', \boldsymbol{A}_k = \boldsymbol{a}' | X_{\tau(k)} = \boldsymbol{n}, \boldsymbol{D}_k = \boldsymbol{a}\},$$

*where*

$$\begin{aligned} &\boldsymbol{n}' \equiv \boldsymbol{n} - \boldsymbol{a}^+ + (\boldsymbol{a}')^+, \\ &|\boldsymbol{a}| = |\boldsymbol{a}'|. \end{aligned}$$

*It is assumed that $r_d((\boldsymbol{n}, \boldsymbol{a}), (\boldsymbol{n}', \boldsymbol{a}'))$ vanishes unless $d(\boldsymbol{n}, \boldsymbol{a}) > 0$.*

**Remark 2.4.** To avoid possible confusion, the suffixes $a$ and $d$ are added to the routing functions for the arrival and departure first networks, respectively. We remark that the routing functions are different even when they describe the same model (see Example 2.5 below). □

We illustrate our notation using the network of [2, 3].

**Example 2.5.** Consider a network with $N$ nodes, a single type (i.e. $I = 1$), and transition rates

$$q_{\mathcal{S}}(\boldsymbol{n}, \boldsymbol{n} + \boldsymbol{e}_j^+ - \boldsymbol{e}_i^+) = \frac{\Psi(\boldsymbol{n} + \boldsymbol{e}_j^+)}{\Phi(\boldsymbol{n})} p_{i,j}(\boldsymbol{n} + \boldsymbol{e}_j^+), \quad i, j = 0, \ldots, N, \tag{4}$$

where $\boldsymbol{e}_j$ is the vector of zeros with a 1 in the $j$-th position, $\Psi$ and $p_{i,j}$ for each $i, j$ are arbitrary nonnegative functions on $\mathcal{S}$, and $\Phi$ is an arbitrary positive function on $\mathcal{S}$. In this network single items move at once, similar to Jackson networks. Define

$$\gamma(\boldsymbol{n}, \boldsymbol{e}_j) = \sum_{i=0}^{N} p_{i,j}(\boldsymbol{n}).$$

The rates $q_{\mathcal{S}}$ of (4) are obtained by defining $\mu$, $b$ and $r_a$ as

$$\begin{aligned} \mu(\boldsymbol{n}) &= \sum_{i=0}^{N} \frac{\Psi(\boldsymbol{n} + \boldsymbol{e}_i^+)}{\Phi(\boldsymbol{n})} \gamma(\boldsymbol{n} + \boldsymbol{e}_i^+, \boldsymbol{e}_i) \ , \\ b(\boldsymbol{n}, \boldsymbol{e}_j) &= \frac{\Psi(\boldsymbol{n} + \boldsymbol{e}_j^+) \gamma(\boldsymbol{n} + \boldsymbol{e}_j^+, \boldsymbol{e}_j)}{\mu(\boldsymbol{n}) \Phi(\boldsymbol{n})} \ , \\ r_a((\boldsymbol{n}, \boldsymbol{e}_j), (\boldsymbol{n} + \boldsymbol{e}_j^+ - \boldsymbol{e}_i^+, \boldsymbol{e}_i)) &= \frac{p_{i,j}(\boldsymbol{n} + \boldsymbol{e}_j^+)}{\gamma(\boldsymbol{n} + \boldsymbol{e}_j^+, \boldsymbol{e}_j)} \ . \end{aligned}$$

Observe that $\mu(\boldsymbol{n})$ is used to ensure that both $b$ and $r_a$ are well-defined probabilities. Thus, the model is formulated as an arrival first network.

Alternatively, the network can be formulated under departure first dynamics. To this end, define

$$\hat{\gamma}(\boldsymbol{n}, \boldsymbol{e}_i) = \sum_{j=0}^{N} \Psi(\boldsymbol{n} + \boldsymbol{e}_j^+) p_{i,j}(\boldsymbol{n} + \boldsymbol{e}_j^+).$$

The transition rates (5) under departure first dynamics are then obtained by setting

$$\begin{aligned}
\mu(\boldsymbol{n}) &= \sum_{i=0}^{N} \frac{\hat{\gamma}(\boldsymbol{n}, \boldsymbol{e}_i)}{\Phi(\boldsymbol{n})} \,, \\
d(\boldsymbol{n}, \boldsymbol{e}_i) &= \frac{\hat{\gamma}(\boldsymbol{n}, \boldsymbol{e}_i)}{\mu(\boldsymbol{n})\Phi(\boldsymbol{n})} \,, \\
r_d((\boldsymbol{n}, \boldsymbol{e}_i), (\boldsymbol{n} + \boldsymbol{e}_j^+ - \boldsymbol{e}_i^+, \boldsymbol{e}_j)) &= \frac{\Psi(\boldsymbol{n} + \boldsymbol{e}_j^+) p_{i,j}(\boldsymbol{n} + \boldsymbol{e}_j^+)}{\hat{\gamma}(\boldsymbol{n}, \boldsymbol{e}_i)} \,.
\end{aligned}$$

As we shall see in the next section, for this example the arrival first network formulation is convenient to find a tractable stationary distribution. □

The following remark further discusses the relationship between departure first and arrival first dynamics as well as their relationships with early and late arrival formulations for discrete-time queueing networks.

**Remark 2.6.** Consider a network in which batch departures $\{\boldsymbol{D}_k\}$ and batch arrivals $\{\boldsymbol{A}_k\}$ alternate. Let

$$\{\boldsymbol{Y}_k, \boldsymbol{A}_k, \tilde{\boldsymbol{Y}}_k, \boldsymbol{D}_k\}$$

denote the state of the network, where $\boldsymbol{Y}_k$ denotes the state of the network just before an arrival, and $\tilde{\boldsymbol{Y}}_k$ denotes the state of the network just before a departure. The evolution of the early arrival process $\{\boldsymbol{Y}_k\}$, and the late arrival process $\{\tilde{\boldsymbol{Y}}_k\}$ is described by the recursions

$$\boldsymbol{Y}_{k+1} = \boldsymbol{Y}_k + \boldsymbol{A}_k - \boldsymbol{D}_k, \tag{5}$$

$$\tilde{\boldsymbol{Y}}_{k+1} = \tilde{\boldsymbol{Y}}_k - \boldsymbol{D}_k + \boldsymbol{A}_{k+1}. \tag{6}$$

These recursions do not involve the probabilistic relation between the network states and the arrival and departure batches. Additional assumptions on the arrival and departure sequences are required for the network processes $\{\boldsymbol{Y}_k\}$ and $\{\tilde{\boldsymbol{Y}}_k\}$ to be Markovian. In particular, assuming that the network under dynamics (5) satisfies the assumptions of Definition 2.1, and is therefore a Markov chain, the network under dynamics (6) will, in general, not be a Markov chain as the transition probability from $\tilde{\boldsymbol{Y}}_k$ to $\tilde{\boldsymbol{Y}}_{k+1}$ generally also depends on $\boldsymbol{A}_k$. For more discussion on early and late arrival queueing network models the reader is referred to [10, Chapter 12]. □

In the following we concentrate on arrival first queueing networks. We derive conditions under which the model has linear traffic equations. The same models are generally non-linear under the departure first network formulation. The results are derived by analogy with, but are complementary to, the results of [18].

# 3 Linear Traffic Model

With the assumptions in Section 2, $\boldsymbol{X}_t$ is a Markov chain with state space $\mathcal{S}$ and transition rates $q_{\mathcal{S}}$ given in (2). However, as in [18] for the case of departure first networks, for obtaining a characterisation of the arrival rule, in this paper we deal with a more detailed Markov chain $\{(\boldsymbol{D}_{k(t)}, \boldsymbol{X}_t)\}$, where

$$k(t) = \sup\{\ell \geq 1 | \tau(\ell) < t\},$$

with transition rates:

$$q((\boldsymbol{a}, \boldsymbol{n}), (\boldsymbol{a}', \boldsymbol{n}')) = \mu(\boldsymbol{n}) b(\boldsymbol{n}, \boldsymbol{a}'') r_a((\boldsymbol{n}, \boldsymbol{a}''), (\boldsymbol{n}', \boldsymbol{a}')),$$

where, for a feasible transition, the arrival vector $\boldsymbol{a}''$ is uniquely determined by

$$\begin{aligned} (\boldsymbol{a}'')^+ &= \boldsymbol{n}' + (\boldsymbol{a}')^+ - \boldsymbol{n}, \\ |\boldsymbol{a}''| &= |\boldsymbol{a}'|. \end{aligned}$$

Assume that $\{(\boldsymbol{D}_{k(t)}, \boldsymbol{X}_t)\}$ has a stationary distribution $\pi_q$. Then $\pi_q$ satisfies the global balance equations

$$\pi_q(\boldsymbol{a}, \boldsymbol{n}) \mu(\boldsymbol{n}) = \sum_{\boldsymbol{a}', \boldsymbol{n}'} \pi_q(\boldsymbol{a}', \boldsymbol{n}') \mu(\boldsymbol{n}') b(\boldsymbol{n}', \boldsymbol{a}'') r_a((\boldsymbol{n}', \boldsymbol{a}''), (\boldsymbol{n}, \boldsymbol{a})). \tag{7}$$

Let $\pi$ be the marginal distribution of the state $\boldsymbol{n}$:

$$\pi(\boldsymbol{n}) = \sum_{\boldsymbol{a}} \pi_q(\boldsymbol{a}, \boldsymbol{n}). \tag{8}$$

We now develop local balance equations for $\pi$ and give necessary and sufficient conditions for $\pi$ to be the stationary distribution of $\{\boldsymbol{X}_t\}$. Note that by (7), (8)

$$\sum_{\boldsymbol{n}', \boldsymbol{a}'} \pi(\boldsymbol{n}') \mu(\boldsymbol{n}') b(\boldsymbol{n}', \boldsymbol{a}') r_a((\boldsymbol{n}', \boldsymbol{a}'), (\boldsymbol{n}, \boldsymbol{a})) = \pi_q(\boldsymbol{a}, \boldsymbol{n}) \mu(\boldsymbol{n}). \tag{9}$$

Hence, by (8) and (9), $r_a^*$ and $b^*$ defined as

$$r_a^*((\boldsymbol{n}, \boldsymbol{a}), (\boldsymbol{n}', \boldsymbol{a}')) = \frac{\pi(\boldsymbol{n}') \mu(\boldsymbol{n}') b(\boldsymbol{n}', \boldsymbol{a}') r_a((\boldsymbol{n}', \boldsymbol{a}'), (\boldsymbol{n}, \boldsymbol{a}))}{\pi_q(\boldsymbol{a}, \boldsymbol{n}) \mu(\boldsymbol{n})}, \tag{10}$$

$$b^*(\boldsymbol{n}, \boldsymbol{a}) = \frac{\pi_q(\boldsymbol{a}, \boldsymbol{n})}{\pi(\boldsymbol{n})}, \tag{11}$$

are proper probability distributions. For all $\boldsymbol{a}, \boldsymbol{a}' \in \mathcal{A}$, and $\boldsymbol{n}' = \boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+$, we have that

$$\pi(\boldsymbol{n}') \mu(\boldsymbol{n}') b(\boldsymbol{n}', \boldsymbol{a}') r_a((\boldsymbol{n}', \boldsymbol{a}'), (\boldsymbol{n}, \boldsymbol{a})) = \pi(\boldsymbol{n}) \mu(\boldsymbol{n}) b^*(\boldsymbol{n}, \boldsymbol{a}) r_a^*((\boldsymbol{n}, \boldsymbol{a}), (\boldsymbol{n}', \boldsymbol{a}')). \tag{12}$$

Let $\{(\boldsymbol{A}^*_{k(t)+1}, \boldsymbol{X}^*_t)\}$ represent the network process with arrival function $b^*$ and routing function $r_a^*$. Then $\{\boldsymbol{X}^*_t\}$ has arrival first dynamics. Kelly's Lemma [16, Section 1.7] implies

that $\pi$ is the stationary distribution of both $\boldsymbol{X}_t^*$ and $\boldsymbol{X}_t$. By analogy with [18, Lemma 2] it can also be shown that $\{(\boldsymbol{A}_{k(t)+1}^*, \boldsymbol{X}_t^*)\}$ is the time-reversed process of $\{(\boldsymbol{D}_{k(t)}, \boldsymbol{X}_t)\}$.

Adding up (12) for all $\boldsymbol{n}', \boldsymbol{a}'$ gives

$$\pi(\boldsymbol{n})\mu(\boldsymbol{n})b^*(\boldsymbol{n},\boldsymbol{a}) = \sum_{\boldsymbol{n}',\boldsymbol{a}'} \pi(\boldsymbol{n}')\mu(\boldsymbol{n}')b(\boldsymbol{n}',\boldsymbol{a}')r_a((\boldsymbol{n}',\boldsymbol{a}'),(\boldsymbol{n},\boldsymbol{a})). \tag{13}$$

The left hand side of (13) represents the rate out of state $\boldsymbol{n}$ due to an arrival of batch $\boldsymbol{a}$ in the time-reversed process, and the right hand side represents the rate into state $\boldsymbol{n}$ due to a batch departure $\boldsymbol{a}$ in the original process. In particular, if $b^* = b$, then we have

$$\pi(\boldsymbol{n})\mu(\boldsymbol{n})b(\boldsymbol{n},\boldsymbol{a}) = \sum_{\boldsymbol{a}'} \pi(\boldsymbol{n}')\mu(\boldsymbol{n}')b(\boldsymbol{n}',\boldsymbol{a}')r_a((\boldsymbol{n}',\boldsymbol{a}'),(\boldsymbol{n},\boldsymbol{a}))\mathbf{1}[\boldsymbol{n}' = \boldsymbol{n}+\boldsymbol{a}^+-(\boldsymbol{a}')^+], \tag{14}$$

where the indicator $\mathbf{1}[\boldsymbol{n}' = \boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+]$ is added to emphasize the relation between the states $\boldsymbol{n}$, $\boldsymbol{n}'$ and the batches $\boldsymbol{a}, \boldsymbol{a}'$. This is a kind of *partial balance* for $\pi$, stating that in steady state the *rate out* of state $\boldsymbol{n}$ *due to arrivals* of batch $\boldsymbol{a}$ equals the *rate into* the same state $\boldsymbol{n}$ *due to departures* of batch $\boldsymbol{a}$. For the network of Example 2.5, [2] studied this kind of local balance equations, and called them *backward local balance equations.*

In contrast, in partial balance as discussed in the queueing network literature, arrivals and departures are interchanged in the above interpretation. Those partial balance equations are obtained under the departure first dynamics and read

$$\pi_d(\boldsymbol{n})\mu(\boldsymbol{n})d(\boldsymbol{n},\boldsymbol{a}) = \sum_{\boldsymbol{a}'} \pi_d(\boldsymbol{n}')\mu(\boldsymbol{n}')d(\boldsymbol{n}',\boldsymbol{a}')r_d((\boldsymbol{n}',\boldsymbol{a}'),(\boldsymbol{n},\boldsymbol{a}))\mathbf{1}[\boldsymbol{n}' = \boldsymbol{n}-\boldsymbol{a}^+ + (\boldsymbol{a}')^+], \tag{15}$$

balancing for each state $\boldsymbol{n}$ the *rate out due to departure* of batch $\boldsymbol{a}$ with the *rate into* $\boldsymbol{n}$ *due to arrival* of batch $\boldsymbol{a}$. Local balance (15) is called group local balance in [6, 18].

A transition in (14) is initiated by the arrival of a batch, and not by the departure of a batch as in group local balance (15). Therefore, one could state that the transitions are oriented backwards. We will refer to (14) as *backward group local balance.*

The equation (14) is linear and purely dependent on the routing function, so it can also be regarded as a *linear traffic equation* concerning each batch $\boldsymbol{a}$ under state $\boldsymbol{n}$:

$$\hat{\pi}(\boldsymbol{n},\boldsymbol{a}) = \sum_{\boldsymbol{n}',\boldsymbol{a}'} \hat{\pi}(\boldsymbol{n}',\boldsymbol{a}')r_a((\boldsymbol{n}',\boldsymbol{a}'),(\boldsymbol{n},\boldsymbol{a})), \tag{16}$$

where

$$\hat{\pi}(\boldsymbol{n},\boldsymbol{a}) = \pi(\boldsymbol{n})\mu(\boldsymbol{n})b(\boldsymbol{n},\boldsymbol{a}).$$

Solving (16) is equivalent to finding the stationary measure for the routing probabilities. For this purpose, let us assume that the Markov chain with transition probabilities $r_a$ can be classified into recurrent subclasses, i.e., it has no transient state. Under this assumption, the state space $\mathcal{S} \times \mathcal{A}$ is decomposed into recurrent subspaces of the form

$$U_{\boldsymbol{j},m} = \{(\boldsymbol{n},\boldsymbol{a}) | \boldsymbol{n} + \boldsymbol{a}^+ = \boldsymbol{j}, |\boldsymbol{a}| = m\},$$

for each $\boldsymbol{j}$ and non-negative number $m$. Note that $U_{\boldsymbol{j},m}$ is a finite set, that may be divided further into irreducible subsets. Let $\langle \boldsymbol{n}, \boldsymbol{a} \rangle$ be a representative element in each irreducible subset containing state $(\boldsymbol{n}, \boldsymbol{a})$, and denote this irreducible subset by $V_{\langle \boldsymbol{n},\boldsymbol{a} \rangle}$. Clearly, there exists a positive stationary measure on $V_{\langle \boldsymbol{n},\boldsymbol{a} \rangle}$. By collecting these measures, we obtain a stationary measure for the routing function $r_a$. Let $\nu_0$ be such a solution, determined up to a multiplicative constant on each subset $V_{\langle \boldsymbol{n},\boldsymbol{a} \rangle}$, i.e.,

$$\nu_0(\boldsymbol{n}, \boldsymbol{a}) = \sum_{\boldsymbol{n}',\boldsymbol{a}'} \nu_0(\boldsymbol{n}', \boldsymbol{a}') r_a((\boldsymbol{n}', \boldsymbol{a}'), (\boldsymbol{n}, \boldsymbol{a})). \tag{17}$$

**Remark 3.1.** A similar construction was employed in [6, 18] for departure first networks. In these references the recurrent subspaces are defined as

$$U^d_{\boldsymbol{j},m} = \{(\boldsymbol{n}, \boldsymbol{a}) | \boldsymbol{n} - \boldsymbol{a}^+ = \boldsymbol{j}, |\boldsymbol{a}| = m\}.$$

Comparison of $U_{\boldsymbol{j},m}$ and $U^d_{\boldsymbol{j},m}$ shows that the *base state* for a transition from state $\boldsymbol{n}$ to state $\boldsymbol{n}' = \boldsymbol{n} - \boldsymbol{a}^+ + (\boldsymbol{a}')^+$ of the departure first network is $\boldsymbol{n} - \boldsymbol{a}^+$, whereas for the arrival first networks the base state is $\boldsymbol{n} + \boldsymbol{a}^+$. This is the main difference between group local balance (15) and backward group local balance (14). □

The proof of the following result is similar to that of [18, Theorem 3], hence we will only provide a sketch of the proof and refer the reader to that paper for further details.

**Theorem 3.2.** *For the arrival first queueing networks defined by (2.1.1) and (2.1.2), if there exists a stationary distribution $\pi$, then the following three conditions are equivalent.*

*(a)* $b = b^*$.

*(b) The backward group local balance equation (14) is satisfied.*

*(c) The routing function $r_a$ is recurrent, and the arrival rate function $\mu(\boldsymbol{n})b(\boldsymbol{n}, \boldsymbol{a})$ has the form*

$$\mu(\boldsymbol{n})b(\boldsymbol{n}, \boldsymbol{a}) = \frac{\Psi(\langle \boldsymbol{n}, \boldsymbol{a} \rangle)\nu_0(\boldsymbol{n}, \boldsymbol{a})}{\Phi(\boldsymbol{n})},$$

*where $\Psi$ and $\Phi$ are arbitrary nonnegative and positive functions, respectively, and $\nu_0(\boldsymbol{n}, \boldsymbol{a})$ is the solution of traffic equation (17), which exists by the recurrence of $r_a$.*

*Conversely, suppose (c) is satisfied without assuming the existence of $\pi$. Then, there always exists a stationary measure for the transition rate function $q$. In particular, if*

$$C_0 = \sum_{\boldsymbol{n},\boldsymbol{a}} \Psi(\langle \boldsymbol{n}, \boldsymbol{a} \rangle)\nu_0(\boldsymbol{n}, \boldsymbol{a})/\mu(\boldsymbol{n}) < \infty,$$

*then $\{\boldsymbol{X}_t\}$ and $\{(\boldsymbol{D}_{k(t)}, \boldsymbol{X}_t)\}$ have stationary distributions*

$$\pi(\boldsymbol{n}) = C_0^{-1}\Phi(\boldsymbol{n}), \qquad \boldsymbol{n} \in \mathcal{S},$$
$$\pi_q(\boldsymbol{a}, \boldsymbol{n}) = C_0^{-1}\Psi(\langle \boldsymbol{n}, \boldsymbol{a} \rangle)\nu_0(\boldsymbol{n}, \boldsymbol{a})/\mu(\boldsymbol{n}), \qquad (\boldsymbol{a}, \boldsymbol{n}) \in \mathcal{A} \times \mathcal{S},$$

*and $\pi$ satisfies backward group local balance (14).*

**Proof.** Equivalence of (a) and (b) follows from (13) and (14). Equivalence of (b) and (c) follows from (17) and (14) and observing that $\Psi(\langle \boldsymbol{n}, \boldsymbol{a} \rangle)$ is a constant at each $V_{\langle \boldsymbol{n},\boldsymbol{a} \rangle}$.

The reversed statement follows by insertion of $\pi$ and (c) in the backward group local balance equations. Then (c) and (a) are again equivalent, and $\pi_q(\boldsymbol{a}, \boldsymbol{n})$ is obtained as $\pi(\boldsymbol{n})b(\boldsymbol{n}, \boldsymbol{a})$, but can also be obtained from (7). □

**Remark 3.3.** From the definition of $V_{\langle \boldsymbol{n},\boldsymbol{a} \rangle}$ we obtain that

$$V_{\langle \boldsymbol{n},\boldsymbol{a} \rangle} \subset U_{\boldsymbol{n}+\boldsymbol{a}^+,|\boldsymbol{a}|} = \{(\boldsymbol{n}', \boldsymbol{a}') | \boldsymbol{n}' + (\boldsymbol{a}')^+ = \boldsymbol{n} + \boldsymbol{a}^+ \text{ and } |\boldsymbol{a}| = |\boldsymbol{a}'|\}.$$

A possible choice for $\Psi$ is

$$\Psi(\langle \boldsymbol{n}, \boldsymbol{a} \rangle) = g(\boldsymbol{n} + \boldsymbol{a}^+),$$

for some arbitrarily given non-negative function $g$, resulting in a typical arrival rate function

$$\mu(\boldsymbol{n})b(\boldsymbol{n}, \boldsymbol{a}) = \frac{g(\boldsymbol{n} + \boldsymbol{a}^+)\nu_0(\boldsymbol{n}, \boldsymbol{a})}{\Phi(\boldsymbol{n})}. \tag{18}$$

A slightly more general form of the arrival rate function is

$$\mu(\boldsymbol{n})b(\boldsymbol{n}, \boldsymbol{a}) = \frac{g(\boldsymbol{n} + \boldsymbol{a}^+, |\boldsymbol{a}|)\nu_0(\boldsymbol{n}, \boldsymbol{a})}{\Phi(\boldsymbol{n})}, \tag{19}$$

for arbitrary non-negative function $g$ on $\mathcal{S} \times \mathbb{Z}_+$. Observe the role of the base state $\boldsymbol{n}+\boldsymbol{a}^+$ in these arrival rate functions. □

**Example 3.4.** Let us apply Theorem 3.2 to the network of Example 2.5, where

$$\mu(\boldsymbol{n})b(\boldsymbol{n}, \boldsymbol{e}_j) = \frac{\Psi(\boldsymbol{n} + \boldsymbol{e}_j^+)\gamma(\boldsymbol{n} + \boldsymbol{e}_j^+, \boldsymbol{e}_j)}{\Phi(\boldsymbol{n})}.$$

By Remark 3.3, this arrival rate function leads to a linear traffic equation (17) if and only if there exist functions $H$ and $K$ such that

$$\nu_0(\boldsymbol{n}, \boldsymbol{e}_j) \equiv H(\boldsymbol{n})K(\boldsymbol{n} + \boldsymbol{e}_j^+)\gamma(\boldsymbol{n} + \boldsymbol{e}_j^+, \boldsymbol{e}_j)$$

is a stationary measure of the routing function $r_a$ given in Example 2.5, that is,

$$\begin{aligned} &H(\boldsymbol{n})K(\boldsymbol{n} + \boldsymbol{e}_j^+)\gamma(\boldsymbol{n} + \boldsymbol{e}_j^+, \boldsymbol{e}_j) \\ &= \sum_{i=0}^{N} H(\boldsymbol{n} + \boldsymbol{e}_j^+ - \boldsymbol{e}_i^+)K(\boldsymbol{n} + \boldsymbol{e}_j^+)\gamma(\boldsymbol{n} + \boldsymbol{e}_j^+, \boldsymbol{e}_i)r_a((\boldsymbol{n} + \boldsymbol{e}_j^+ - \boldsymbol{e}_i^+, \boldsymbol{e}_i), (\boldsymbol{n}, \boldsymbol{e}_j)) \,. \end{aligned}$$

Substituting $r_a$ shows that $H$ must satisfy

$$H(\boldsymbol{n}) \sum_{i=0}^{N} p_{i,j}(\boldsymbol{n} + \boldsymbol{e}_j^+) = \sum_{i=0}^{N} H(\boldsymbol{n} + \boldsymbol{e}_j^+ - \boldsymbol{e}_i^+)p_{j,i}(\boldsymbol{n} + \boldsymbol{e}_j^+) \,. \tag{20}$$

In this case, the stationary distribution is given by

$$\pi(\boldsymbol{n}) = C^{-1}\Phi(\boldsymbol{n})H(\boldsymbol{n}) \,. \tag{21}$$

Thus we obtain the result of [2, Theorem 3.1]. □

# 4 Product Form Solutions

Theorem 3.2 specifies the form of arrival rules for a linear traffic equation (17). The form of the arrival rule depends on $\nu_0$, the solution of this traffic equation. This dependency is not desired in applications. In many interesting cases, the effect of $\nu_0$ can be removed. In this section we consider classes of queueing networks with a product form stationary distribution. We will specify $\nu_0$ step by step and give its implications on the arrival rule.

Consider the case that there exists a solution $\nu_0$ of the form

$$\nu_0(\boldsymbol{n}, \boldsymbol{a}) = \tilde{\nu}(\boldsymbol{n})\omega_0(a(0))\omega_1(\boldsymbol{a}^+), \tag{22}$$

such that $\omega_1$ is an exponential function, i.e.,

$$\omega_1(\boldsymbol{n}_1 + \boldsymbol{n}_2) = \omega_1(\boldsymbol{n}_1)\omega_1(\boldsymbol{n}_2).$$

Then it is easily checked that $\nu_0$ solves the traffic equations (17) if and only if

$$\nu(\boldsymbol{n}) = \tilde{\nu}(\boldsymbol{n})/\omega_1(\boldsymbol{n})$$

satisfies

$$\nu(\boldsymbol{n})\omega_0(a(0)) = \sum_{\boldsymbol{a}'} \nu(\boldsymbol{n}')\omega_0(a'(0))r_a((\boldsymbol{n}', \boldsymbol{a}'), (\boldsymbol{n}, \boldsymbol{a}))\mathbf{1}[\boldsymbol{n}' = \boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+]. \tag{23}$$

We now specify the arrival rule under assumption (23). The effect of $\nu_0$ is eliminated from the arrival rule, and (as in Example 3.4) absorbed in the stationary distribution.

**Theorem 4.1.** *Consider a arrival first queueing network. If traffic equation (23) has solution $\nu$, then (a) and (b) of Theorem 3.2 are each equivalent to the existence of a non-negative function $\Psi$ and a positive function $\Phi$ such that the arrival rule is given by*

$$\mu(\boldsymbol{n})b(\boldsymbol{n}, \boldsymbol{a}) = \frac{\Psi(\langle \boldsymbol{n}, \boldsymbol{a} \rangle)\omega_0(a(0))}{\Phi(\boldsymbol{n})}. \tag{24}$$

*The stationary distribution satisfies backward group local balance (14) and is given by*

$$\pi(\boldsymbol{n}) = C^{-1}\Phi(\boldsymbol{n})\nu(\boldsymbol{n}), \qquad \boldsymbol{n} \in \mathcal{S}, \tag{25}$$

*where $C$ is the normalization constant. In addition, if $\omega_0$ is also exponential, then $\omega_0$ can be canceled from (24), i.e., (a) and (b) are equivalent to the existence of $\Psi$ and $\Phi$ such that*

$$\mu(\boldsymbol{n})b(\boldsymbol{n}, \boldsymbol{a}) = \frac{\Psi(\langle \boldsymbol{n}, \boldsymbol{a} \rangle)}{\Phi(\boldsymbol{n})}. \tag{26}$$

*For this arrival rule the stationary distribution $\pi$ is given by*

$$\pi(\boldsymbol{n}) = C^{-1}\Phi(\boldsymbol{n})\nu(\boldsymbol{n})\omega_0(|\boldsymbol{n}|), \qquad \boldsymbol{n} \in \mathcal{S}. \tag{27}$$

**Proof.** If $\nu$ solves (23) then (from the discussion preceding the theorem) there exists an exponential function $\omega_1$ such that

$$\nu(\boldsymbol{n}) = \tilde{\nu}(\boldsymbol{n})/\omega_1(\boldsymbol{n})$$

and for which $\nu_0(\boldsymbol{n}, \boldsymbol{a})$ specified in (22) satisfies (17). Thus, invoking the exponentiality of $\omega_1$,

$$\nu_0(\boldsymbol{n}, \boldsymbol{a}) = \tilde{\nu}(\boldsymbol{n})\omega_0(a(0))\omega_1(\boldsymbol{n} + \boldsymbol{a}^+)/\omega_1(\boldsymbol{n}).$$

Introducing a nonnegative $\tilde{\Psi}$ and a positive $\tilde{\Phi}$ the arrival rate function of Theorem 3.2 (c) can be expressed as

$$\mu(\boldsymbol{n})b(\boldsymbol{n}, \boldsymbol{a}) = \frac{\tilde{\Psi}(\langle \boldsymbol{n}, \boldsymbol{a} \rangle)\nu_0(\boldsymbol{n}, \boldsymbol{a})}{\tilde{\Phi}(\boldsymbol{n})} = \frac{\tilde{\Psi}(\langle \boldsymbol{n}, \boldsymbol{a} \rangle)\omega_1(\boldsymbol{n} + \boldsymbol{a}^+)}{\tilde{\Phi}(\boldsymbol{n})\omega_1(\boldsymbol{n})/\tilde{\nu}(\boldsymbol{n})}\omega_0(a(0)) = \frac{\Psi(\langle \boldsymbol{n}, \boldsymbol{a} \rangle)}{\Phi(\boldsymbol{n})}\omega_0(a(0)),$$

where

$$\begin{aligned} \Phi(\boldsymbol{n}) &= \tilde{\Phi}(\boldsymbol{n})\omega_1(\boldsymbol{n})/\tilde{\nu}(\boldsymbol{n}) \\ \Psi(\langle \boldsymbol{n}, \boldsymbol{a} \rangle) &= \tilde{\Psi}(\langle \boldsymbol{n}, \boldsymbol{a} \rangle)\omega_1(\boldsymbol{n} + \boldsymbol{a}^+), \end{aligned}$$

observing that $\boldsymbol{n} + \boldsymbol{a}^+ \in V_{\langle \boldsymbol{n}, \boldsymbol{a} \rangle}$. Thus, from Theorem 3.2 (c) we obtain that (24) is equivalent to both (a) and (b) of Theorem 3.2.

When $\omega_0$ is also exponential

$$\omega_0(a(0)) = \omega_0(|\boldsymbol{n}|)/\omega_0(|\boldsymbol{n}| - a(0)).$$

Since

$$|\boldsymbol{n} + \boldsymbol{a}^+| = |\boldsymbol{n}| + |\boldsymbol{a}| - a(0),$$

the term $\omega(|\boldsymbol{n}| - a(0))$ can be incorporated in $\Psi$. □

The stationary distributions obtained in Theorem 4.1 are a product of a routing part $\nu$, determined by the traffic equations, and an arrival part $\Phi$, obtained from the arrival rule. Let us now assume that the solution of the traffic equation (23) is exponential, i.e., there exist $\rho_1, \ldots, \rho_N$ such that, for some constant $c$,

$$\nu(\boldsymbol{a})\omega_0(a(0)) = c \prod_{i=0}^{N} \frac{1}{\rho_i^{a(i)}}.$$

The traffic equation (23) implies that the $\rho_i$'s satisfy the following traffic equation

$$\prod_{i=0}^{N} \rho_i^{a(i)} = \sum_{\boldsymbol{a}'} \prod_{i=0}^{N} \rho_i^{a'(i)} r_a((\boldsymbol{n}', \boldsymbol{a}'), (\boldsymbol{n}, \boldsymbol{a})), \qquad \boldsymbol{a} \in \mathcal{A},\ \boldsymbol{n}, \boldsymbol{n}' \in \mathcal{S}. \tag{28}$$

With this solution, the stationary distribution is given by

$$\pi(\boldsymbol{n}) = C^{-1}\Phi(\boldsymbol{n}) \prod_{i=1}^{N} \frac{1}{\rho_i^{n(i)}}, \qquad \boldsymbol{n} \in \mathcal{S}. \tag{29}$$

Thus we obtain a product form expression for the stationary distribution. In the special case that the routing function is state independent,

$$r_a((\boldsymbol{n}', \boldsymbol{a}'), (\boldsymbol{n}, \boldsymbol{a})) = r_a(\boldsymbol{a}', \boldsymbol{a}),$$

the traffic equation becomes

$$\prod_{i=0}^{N} \rho_i^{a(i)} = \sum_{\boldsymbol{a}'} \prod_{i=0}^{N} \rho_i^{a'(i)} r_a(\boldsymbol{a}', \boldsymbol{a}). \tag{30}$$

Observe that $\omega_0(|\boldsymbol{n}|)$ can be included in $\Phi(\boldsymbol{n})$. We summarize these results in the following corollary.

**Corollary 4.2.** *If the arrival rule takes the form (26), and if the traffic equation (23) has solution* $\prod_{i=0}^{N} \rho_i^{a(i)}$ *such that* $C = \sum_{\boldsymbol{n}} \Phi(\boldsymbol{n}) \prod_{i=1}^{N} \rho_i^{-n(i)} < \infty$*, then the stationary distribution of the network is of product form (30).*

The above result implies that a product form solution for the traffic equation translates to a product form solution for the stationary distribution of the network. Comparison with the analog for departure first networks shows that the stationary distribution involves the *reciprocal* of the solution to the traffic equation. In particular, a solution $\rho_i > 1$ of the traffic equation is desired for the stability of the arrival first networks.

# 5 Application

In this section, we model the kanban production system as a pull network and apply our results in earlier sections to analyze it. These networks are shown to have linear traffic equation under suitable blocking protocols adopted from the kanban protocol implemented in production systems. For simplicity of presentation, we restrict ourselves to the case of a single product type.

## 5.1 Kanban production

Consider a production facility consisting of multiple machines or work stations. Raw parts arrive at the facility requiring operations at multiple stages, one stage after another. In the classical approach, work is driven by the availability of parts. A job starts, if the machine is available, upon arrival of the required parts, and when a job is completed at one stage, it is placed in the buffer or queue of the next stage. This procedure results in a network with push dynamics. In contrast, under the kanban protocol the arrival epoch of a job is determined by the request at the next stage. For example, when a worker on a production line begins drawing from a new bin of parts, he removes the label (kanban in Japanese) from the bin and routes it back to the supplying or upstream work station, where it serves as an order for a new bin of parts. This bin of parts is then removed at the upstream work station and routed to the downstream (requesting) work station. As a consequence, routing of *requests* is initiated by the downstream station, and subsequently the requested *parts* are provided by the upstream station and delivered at the downstream station. With the state of the system recording the number of completed parts that is present in the buffers of the stations, this gives rise to a pull network.

To facilitate the description of the kanban protocol, we will first consider a tandem line in which batches have size 1 only. Consider a tandem production facility consisting of

$N$ single server stations, labelled $i = 1, \ldots, N$, in which items route among the stations in increasing order. Let $\boldsymbol{n}$ be the state of the network, where $n(i)$ is the number of items in station $i$. Under kanban production, work is driven by requests for items generated at the outside or at downstream stations. In particular, requests for completed items arrive from the outside to station $N$, and the server at station $j$ places requests for items at station $j-1$, $j = N, \ldots, 1$, where $j = 0$ denotes an outside supply of raw parts. When the server at station $j$ places a request at station $j-1$ this item is immediately taken from the buffer of station $j-1$, and placed at the server of station $j$, $j = 1, \ldots, N$. Similarly, requests for completed items arriving from the outside to station $N$ are immediately satisfied from the buffer of station $N$. Thus the flow of items among the stations in increasing order is driven by the stream of requests among the stations in decreasing order. The rate at which requests are generated at the downstream stations determines the flow of items, i.e., the system operates as a pull system. This mode of operation corresponds to kanban production. When a worker on a production line begins working on an item (e.g., starts drawing from a bin of parts), he places an order for a new bin of parts at the upstream station. When the time the worker needs to complete the bin of parts is exponentially distributed, requests are generated at exponentially distributed time intervals. Let $\alpha(\boldsymbol{e}(j))$ denote the rate at which the worker at station $j$ completes working on a bin of parts. Then the worker will place orders at station $j-1$ at rate $\alpha(\boldsymbol{e}(j))$, provided he is not idle due to lack of parts (the production line is blocked), and therefore the rate at which items route from station $j-1$ to station $j$ is $\alpha(\boldsymbol{e}(j))$. Similarly, with $\alpha(\boldsymbol{e}(N))$ denoting the rate at which finished items are demanded by customers, the rate at which the buffer of station $N$ is depleted is $\alpha(\boldsymbol{e}(N))$.

Clearly, for the kanban protocol to function properly, when an item is requested from the buffer of station $j$, this item must actually be present in the buffer. Otherwise the production manager must take action, which results in blocking protocols which we will illustrate in Section 5.3. Let us first consider the generic behaviour of a kanban system with batch routing.

In a general kanban system, the bin of parts required by a server or by the customers might contain items from different stations. Let $\alpha(\boldsymbol{n}, \boldsymbol{a})$ denote the request rate for a batch $\boldsymbol{a}$ of completed items when the network is in state $\boldsymbol{n}$. This request is immediately satisfied, and it places $a(i)$ items at the buffer of station $i$, $i = 0, 1, \ldots, N$, where $i = 0$ is meant the request from the outside. Satisfying such a request needs items from the buffer of other stations, say $a'(i)$ items are required from station $i$, including $a'(0)$ raw parts for $i = 0$. A fraction $p(\boldsymbol{a}, \boldsymbol{a}'; \boldsymbol{n} + \boldsymbol{a}^+)$ of batches $\boldsymbol{a}$ is produced using the batch $\boldsymbol{a}'$. The resulting rate at which the network produces a batch $\boldsymbol{a}$ satisfied by the requested batch $\boldsymbol{a}'$ is

$$\bar{q}(\boldsymbol{a}, \boldsymbol{a}'; \boldsymbol{n} + \boldsymbol{a}^+) = \alpha(\boldsymbol{n}, \boldsymbol{a}) p(\boldsymbol{a}, \boldsymbol{a}'; \boldsymbol{n} + \boldsymbol{a}^+) \tag{31}$$

resulting in a transition from state $\boldsymbol{n}$ to state $\boldsymbol{n}' \equiv \boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+$, provided that $\boldsymbol{n}' \in \mathcal{S}$. Note that for this network, the network state first changes from $\boldsymbol{n}$ to base state $\boldsymbol{n} + \boldsymbol{a}^+$ and then changes from $\boldsymbol{n} + \boldsymbol{a}^+$ to $\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+$, yielding an arrival first network.

The following example illustrates the transition rates in a kanban system.

**Example 5.1.** To illustrate these rates, consider the part of a network depicted in Figure 1. Requests for a batch $a(1)$ of completed items are generated at station 1 at rate $\alpha(\boldsymbol{n}, a(1)\boldsymbol{e}_1)$, where $a(1)\boldsymbol{e}_1 = (0, a(1), 0, 0, 0)$. To produce $a(1)$ parts at station 1, we require either $a'(2)$ parts from station 2 and $a'(3)$ parts from station 3, or just $a'(4)$ parts from station 4, which occurs with respective probabilities $p(a(1)\boldsymbol{e}_1, a'(2)\boldsymbol{e}_2 + a'(3)\boldsymbol{e}_3; \boldsymbol{n} + a(1)\boldsymbol{e}_1)$ and $p(a(1)\boldsymbol{e}_1, a'(4)\boldsymbol{e}_4; \boldsymbol{n} + a(1)\boldsymbol{e}_1)$. Thus station 1 pulls parts from stations 2 and 3 with probability

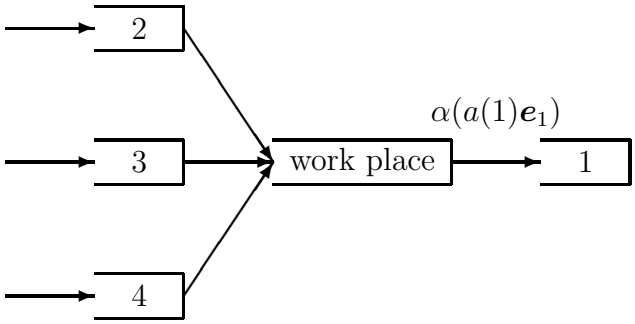

Figure 1.

$$p(a(1)\boldsymbol{e}_1, a'(2)\boldsymbol{e}_2 + a'(3)\boldsymbol{e}_3; \boldsymbol{n} + a(1)\boldsymbol{e}_1)$$

and from station 4 with probability

$$p(a(1)\boldsymbol{e}_1, a'(4)\boldsymbol{e}_4; \boldsymbol{n} + a(1)\boldsymbol{e}_1).$$

The transition rates for these events are

$$\begin{aligned} q(\boldsymbol{n}, \boldsymbol{n} + a(1)\boldsymbol{e}_1 - a'(2)\boldsymbol{e}_2 - a'(3)\boldsymbol{e}_3) &= \alpha(\boldsymbol{n}, a(1)\boldsymbol{e}_1) p(a(1)\boldsymbol{e}_1, a'(2)\boldsymbol{e}_2 + a'(3)\boldsymbol{e}_3; \boldsymbol{n} + a(1)\boldsymbol{e}_1), \\ q(\boldsymbol{n}, \boldsymbol{n} + a(1)\boldsymbol{e}_1 - a'(4)\boldsymbol{e}_4) &= \alpha(\boldsymbol{n}, a(1)\boldsymbol{e}_1) p(a(1)\boldsymbol{e}_1, a'(4)\boldsymbol{e}_4; \boldsymbol{n} + a(1)\boldsymbol{e}_1). \end{aligned}$$

Note that the arrows in Figure 1 describe actual flows of items in the system, and that requests for items are routed in the opposite direction. If this system has requests from the outside, they can be described in a similar way. For instance, if there is a demand for $a(0)$ with rate $\alpha(\boldsymbol{n}, a(0)\boldsymbol{e}_0)$, and it is assembled by $a(1)$ parts from station 1 with probability $p(a(0)\boldsymbol{e}_0, a(1)\boldsymbol{e}_1; \boldsymbol{n})$, this means that $a(1)$ items are removed at the buffer of station 1 to satisfy the request, resulting in a state change $\boldsymbol{n} \to \boldsymbol{n} - a(1)\boldsymbol{e}_1$. Note that this is not the same as a service completion at station 1, since $\alpha(\boldsymbol{n}, a(0)\boldsymbol{e}_0)$ is not the departure rate at station 1.

The resulting network process has arrival first dynamics, with

$$\begin{aligned} \alpha(\boldsymbol{n}, \boldsymbol{a}) &= \mu(\boldsymbol{n}) b(\boldsymbol{n}, \boldsymbol{a}), \\ r_a((\boldsymbol{n}, \boldsymbol{a}), (\boldsymbol{n}', \boldsymbol{a}')) &= p(\boldsymbol{a}, \boldsymbol{a}'; \boldsymbol{n} + \boldsymbol{a}), \end{aligned}$$

but it does not have a linear traffic equation. For a linear traffic equation additional assumptions handling empty stations must be imposed. To this end, in Section 5.3 below we will take a closer look at the kanban protocol. First we consider arrival functions that are consistent with arrival first dynamics.

## 5.2 Arrival functions

The arrival function $\alpha(\boldsymbol{n}, \boldsymbol{a})$ determines the rate at which batches are requested when the network is in state $\boldsymbol{n}$. This section provides possible choices for the arrival function that are consistent with kanban production. We will focus on arrival functions as introduced in Remark 3.3.

If the outside generates requests for finished parts with state independent rate, then the functions $g(\cdot)$ of Remark 3.3 or $\Psi(\cdot)$ of Example 3.4 are reduced to $g = \Phi$ or $\Psi = \Phi$, in correspondence with a similar observation in departure first dynamics, see [8, 14]. The resulting arrival function is

$$\alpha(\boldsymbol{n}, \boldsymbol{a}) = \frac{\Phi(\boldsymbol{n} + \boldsymbol{a}^+)}{\Phi(\boldsymbol{n})}\theta(\boldsymbol{a}). \tag{32}$$

For $\alpha(\boldsymbol{n}, \boldsymbol{a}) = \bar{\alpha}(\boldsymbol{a})$ the network has single server semantics: the rate at which batches are requested is independent of the network state. This can be modelled using $\Phi(\boldsymbol{n}) = 1$ for all $\boldsymbol{n}$. As a consequence, steering of the buffer contents cannot be achieved using such arrival function.

For networks operating under kanban protocols, it is desirable to avoid empty buffers, as well as large buffer contents. Therefore, the request rate might be increased when the buffer content decreases. A typical choice for the arrival function that avoids a large buffer content is

$$\Phi(\boldsymbol{n})^{-1} = \prod_{k=1}^{N} n_k!$$

resulting in an arrival function

$$\frac{\Phi(\boldsymbol{n} + \boldsymbol{a}^+)}{\Phi(\boldsymbol{n})} = \left[\prod_{k=1}^{N}\prod_{j=1}^{a_k}(n_k + j)\right]^{-1} \tag{33}$$

with maximum value 1, which does not avoid empty buffers, but makes large buffer contents relatively unlikely.

Large buffer contents may be excluded via truncation of $\Phi(\boldsymbol{n})$. To this end, buffers exceeding level $U(i)$ at station $i$, $i = 1, \ldots, N$, can be avoided by setting

$$\alpha(\boldsymbol{n}, \boldsymbol{a}) = \mathbf{1}[n(i) + a(i) \leq U(i),\ i = 1, \ldots, N],$$

which can be modelled via

$$\Phi(\boldsymbol{n}) = \mathbf{1}[n(i) \leq U(i),\ i = 1, \ldots, N].$$

The resulting arrival function can also be combined with the arrival function obtained from (33).

Empty buffers can be avoided by further increasing the arrival function for empty buffers. In principle, the arrival function for requests at station $i$ can be increased to $\infty$ when $n(i) = 0$. However, although the resulting Markov chain can be analysed and has a linear traffic equation, the intricate problem of infinite transition rates can be avoided using the 'roving for requests protocol' presented below.

## 5.3 Blocking protocols

The rate at which requests are generated is determined by the 'down-stream' stage. Following the kanban protocol, it is assumed that, upon request, the stage providing the parts has completed production to immediately satisfy demand. This implies that a batch must be in production in each stage. Indeed, due to the assumption that requests are generated at exponential rate, the time until the next request is exponentially distributed with the same rate.

### 5.3.1 Andon-blocking

When a stage cannot satisfy a request (i.e., it does not contain enough items) the production line will be blocked. Under kanban production, a worker that cannot continue production signals to the production manager. In principle, the manager will assist to solve the problem, and in serious cases the production line is *stopped*. This is referred to as the andon-principle. The corresponding blocking protocol was used in [3] and referred to as *andon-blocking*. The andon-blocking protocol *stops production at all stages but the stages feeding into a stage that cannot satisfy demand.* The resulting transition rates are

$$\bar{q}(\boldsymbol{a}, \boldsymbol{a}'; \boldsymbol{n} + \boldsymbol{a}^+) = \mu(\boldsymbol{n}) b(\boldsymbol{n}, \boldsymbol{a}) r_a(\boldsymbol{a}, \boldsymbol{a}') \kappa(\boldsymbol{n} + \boldsymbol{a}^+), \tag{34}$$

where $\kappa$ takes into account the andon-blocking protocol:

$$\kappa(\boldsymbol{n}) = \prod_{\boldsymbol{a} \in \mathcal{A}} \mathbf{1}[\boldsymbol{n} \geq \boldsymbol{a}^+],$$

as it restricts the transition rates to allow only those transitions in which after a request is generated all stations contain at least one part. The blocking protocol is a generalisation of a similar protocol for a network with batches of size 1 introduced in [3]. Observe that the andon protocol requires the set of batches $\mathcal{A}$ to be small and bounded. This is a natural requirement in applications, where requests usually have limited size.

Assume that a non-negative function $H$ exists satisfying (recall (32))

$$H(\boldsymbol{n}) \sum_{\boldsymbol{a}' \in \mathcal{A}} \theta(\boldsymbol{a}) r_a(\boldsymbol{a}, \boldsymbol{a}') = \sum_{\boldsymbol{a}' \in \mathcal{A}} H(\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+) \theta(\boldsymbol{a}') r_a(\boldsymbol{a}', \boldsymbol{a}), \tag{35}$$

then the network has a linear traffic equation, and the stationary distribution is given by

$$\pi(\boldsymbol{n}) = C^{-1} \Phi(\boldsymbol{n}) H(\boldsymbol{n}).$$

To illustrate the role of the blocking function $\kappa$, consider the backward local balance equations (14)

$$C \sum_{\boldsymbol{a}' \in \mathcal{A}} \Big\{ \pi(\boldsymbol{n}) \mu(\boldsymbol{n}) b(\boldsymbol{n}, \boldsymbol{a}) r_a(\boldsymbol{a}, \boldsymbol{a}') \kappa(\boldsymbol{n} + \boldsymbol{a}^+) \\ - \pi(\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+) \mu(\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+) b(\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+, \boldsymbol{a}') r_a(\boldsymbol{a}', \boldsymbol{a}) \kappa(\boldsymbol{n} + \boldsymbol{a}^+) \Big\}$$

$$
\begin{aligned}
&= \sum_{\boldsymbol{a}' \in \mathcal{A}} \Big\{ H(\boldsymbol{n}) \Phi(\boldsymbol{n} + \boldsymbol{a}^+) \theta(\boldsymbol{a}) r_a(\boldsymbol{a}, \boldsymbol{a}') \kappa(\boldsymbol{n} + \boldsymbol{a}^+) \\
&\qquad - H(\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+) \Phi(\boldsymbol{n} + \boldsymbol{a}^+) \theta(\boldsymbol{a}') r_a(\boldsymbol{a}', \boldsymbol{a}) \kappa(\boldsymbol{n} + \boldsymbol{a}^+) \Big\} \\
&= \sum_{\boldsymbol{a}' \in \mathcal{A}} \Big\{ H(\boldsymbol{n}) \theta(\boldsymbol{a}) r_a(\boldsymbol{a}, \boldsymbol{a}') - H(\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+) \theta(\boldsymbol{a}') r_a(\boldsymbol{a}', \boldsymbol{a}) \Big\} \Phi(\boldsymbol{n} + \boldsymbol{a}^+) \kappa(\boldsymbol{n} + \boldsymbol{a}^+),
\end{aligned}
$$

due to the blocking function $\kappa$ being a function of $\boldsymbol{n} + \boldsymbol{a}^+$ only. Thus, the traffic equation (35) involving only the state-independent routing function $r_a(\boldsymbol{a}, \boldsymbol{a}')$ is sufficient for invoking Theorem 4.1 to conclude that the network has a linear traffic equation.

### 5.3.2 Spare parts for starved stations

Under andon-blocking as characterised above, the production line is stopped to solve empty buffer problems. Alternatively, when a buffer becomes empty requests for parts might by satisfied using suitable spare parts, that is, if buffer $k$ is empty and station $j$ requests for parts from buffer $k$ then these parts are provided from an outside stock of spare parts (emergency buffer).

Formalising the blocking protocol described above, consider the network with rate for a transition from state $\boldsymbol{n}$ to state $\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+$ determined by

$$
\bar{q}(\boldsymbol{a}, \boldsymbol{a}'; \boldsymbol{n} + \boldsymbol{a}^+) = \begin{cases}
\alpha(\boldsymbol{n}, \boldsymbol{a}) r_a(\boldsymbol{a}, \boldsymbol{a}') \mathbf{1}[\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+ \in \mathcal{S}] & \boldsymbol{a}, \boldsymbol{a}' \neq \mathbf{0} \\
\alpha(\boldsymbol{n}, \boldsymbol{a}) \{ r_a(\boldsymbol{a}, \mathbf{0}) + \sum_{\boldsymbol{a}''} r_a(\boldsymbol{a}, \boldsymbol{a}'') \mathbf{1}[\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}'')^+ \notin \mathcal{S})\} & \boldsymbol{a}' = \mathbf{0} \\
\alpha(\boldsymbol{n}, \mathbf{0}) r_a(\mathbf{0}, \boldsymbol{a}'] & \\
\quad + \sum_{\boldsymbol{a}''} s(\boldsymbol{a}', \boldsymbol{a}'') \mathbf{1}[\boldsymbol{n} - (\boldsymbol{a}')^+ \in \mathcal{S},\ \boldsymbol{n} - (\boldsymbol{a}'')^+ \notin \mathcal{S}] & \boldsymbol{a} = \mathbf{0}
\end{cases}
$$

Comparison of these rates with the rates (34) shows that requests for batch $\boldsymbol{a}$ are replenished using spare parts. The total rate of absorption of spare parts is

$$
\sum_{\boldsymbol{a}''} r_a(\boldsymbol{a}, \boldsymbol{a}'') \mathbf{1}[\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}'')^+ \notin \mathcal{S}]
$$

corresponding to all batches $\boldsymbol{a}''$ that cannot be delivered by stations of the network. In compensation, to fill the extra buffer of spare parts, at rate

$$
s(\boldsymbol{a}', \boldsymbol{a}'') \mathbf{1}[\boldsymbol{n} - (\boldsymbol{a}')^+ \in \mathcal{S},\ \boldsymbol{n} - (\boldsymbol{a}'')^+ \notin \mathcal{S}]
$$

when requests for batches $\boldsymbol{a}''$ can be issued that cannot be satisfied by the buffer contents at the stations, batches $\boldsymbol{a}'$ are removed from the network to be modified into batches $\boldsymbol{a}''$ and stored in the emergency buffer. This extra rate must be such that the emergency buffer of spare parts is capable of handling requests for spare parts. As a consequence,

$$
s(\boldsymbol{a}', \boldsymbol{a}'') = h(\boldsymbol{a}'') \theta(\boldsymbol{a}'') r_a(\boldsymbol{a}'', \boldsymbol{a}') \tag{36}
$$

compensating for the mean request rate of batches $\boldsymbol{a}''$ that cannot be satisfied by the network, where it is assumed that $h$ exists such that

$$
\frac{h(\boldsymbol{a}')}{h(\boldsymbol{a})} = \frac{H(\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+)}{H(\boldsymbol{n})}, \tag{37}
$$

where $H$ solves (35). The compensating rate $s$ is complementary to redial rates introduced for mobile networks in [5]. It can readily be shown that the conditions (36), (37) are sufficient for the traffic equation (23) to have solution $H(\boldsymbol{n})$. Applying Theorem 4.1 yields the equilibrium distribution $\pi(\boldsymbol{n}) = C^{-1}\Phi(\boldsymbol{n})H(\boldsymbol{n})$.

### 5.3.3 Roving for requests

The blocking protocols presented above either stop the production line or introduce an emergency buffer containing spare parts. As an alternative, a request for a batch $\boldsymbol{a}$ that cannot be satisfied might be satisfied using an emergency completion of the required batch $\boldsymbol{a}$. This is in correspondence with the kanban protocol, where the manager can assist to solve production problems. Typically, if batch $\boldsymbol{a}$ is not available in the buffers upon request, using a batch $\boldsymbol{a}'$ with probability $p(\boldsymbol{a}, \boldsymbol{a}')$ the batch $\boldsymbol{a}$ is immediately completed, and the request for the batch $\boldsymbol{a}$ is satisfied. If the batch $\boldsymbol{a}'$ is not available, then also this batch will be completed via an emergency completion. As a consequence, upon request for batch $\boldsymbol{a}$, the network starts roving for batches needed to complete $\boldsymbol{a}$.

Formalising the roving for requests blocking protocol, for $\boldsymbol{a}, \boldsymbol{a}'$ let

$$[R_a(\boldsymbol{m})]_{\boldsymbol{a},\boldsymbol{a}'} = r_a(\boldsymbol{a}, \boldsymbol{a}')\mathbf{1}[\boldsymbol{m} - \boldsymbol{a}^+ \notin \mathcal{S}]$$

be the routing function for a request $\boldsymbol{a}$ that is not available in the buffers of the stations, i.e., the request $\boldsymbol{a}$ is replenished via an emergency completion using the batch $\boldsymbol{a}'$. Observe that $R_a(\boldsymbol{m})$ has rows containing only 0's for all $\boldsymbol{a}$ such that $\boldsymbol{m} - \boldsymbol{a}^+ \in \mathcal{S}$, i.e., for all batches that are available in base state $\boldsymbol{m}$.

Let $R_a = [r_a(\boldsymbol{a}, \boldsymbol{a}')]$ denote the operator containing the routing function. Implementing roving for requests, the rate for a transition from state $\boldsymbol{n}$ to state $\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+$ is determined by

$$\bar{q}(\boldsymbol{a}, \boldsymbol{a}'; \boldsymbol{n} + \boldsymbol{a}^+) = \alpha(\boldsymbol{n}, \boldsymbol{a}) \sum_{j=0}^{\infty} [R_a R_a^j(\boldsymbol{n} + \boldsymbol{a}^+)]_{\boldsymbol{a},\boldsymbol{a}'}$$

where $R_a^0(\boldsymbol{m}) = I$, the identity operator. Observe that these rates coincide with the rates (31) for all $\boldsymbol{a}'$ for which $\boldsymbol{n} + \boldsymbol{a}^+ - (\boldsymbol{a}')^+ \in \mathcal{S}$.

Note that this blocking protocol is similar to the jump over blocking defined for departure first networks: whenever there is a transition under which the state of the network goes out of the state space, the transition jumps over that state and continues the transition process until the first moment the process lands in a feasible state. By analogy with [4, 9] it can be shown that $H(\boldsymbol{n})$ as defined in (35) also solves the traffic equation (23). Thus, invoking Theorem 4.1 the resulting equilibrium distribution is

$$\pi(\boldsymbol{n}) = C^{-1}\Phi(\boldsymbol{n})H(\boldsymbol{n}),$$

and $\pi$ satisfies backward group local balance (14).

# References


[1] 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 (1975) 248-260.

[2] R.J. Boucherie, Product forms based on backward traffic equations, Journal of Applied Probability 32 (1995) 508-518.

[3] R.J. Boucherie, On the arrival theorem for queueing networks operating under a just-in-time protocol, Performance Evaluation 34 (1998) 109-121.

[4] R.J. Boucherie, Batch routing queueing networks with jump-over blocking, Probability in the Engineering and Informational Sciences 10 (1996) 287-297.

[5] R.J. Boucherie, M. Mandjes, Estimation of performance measures for product form cellular mobile communications networks, Telecommunication Systems 10 (1998) 321-354.

[6] R.J. Boucherie, N.M. van Dijk, Product forms for queueing networks with state-dependent multiple job transitions, Advances in Applied Probability 23 (1991) 152-187.

[7] J.A. Buzacott, J.G. Shantikumar, Stochastic models of manufacturing systems, Prentice Hall, New Jersey, 1993.

[8] K.M. Chandy, A.J. Martin, A characterization of product-form queueing networks, Journal of the ACM 30 (1983) 286–299.

[9] X. Chao, M. Miyazawa, On truncation properties of finite-buffer queues and queueing networks, Probability in the Engineering and Informational Sciences 14 (2000) 409-323.

[10] X. Chao, M. Miyazawa, M. Pinedo, Queueing networks: customers, signals and product form solutions, John Wiley & Sons, Chichester, 1999.

[11] H. Chen, J.M. Harrison, A. Mandelbaum, A. van Ackere, L.M. Wein, Empirical evaluation of a queueing network model for semiconductor wafer fabrication, Operations Research 36 (1988) 202-215.

[12] W.J. Gordon, G.F. Newell, Cyclic queueing systems with restricted length queues, Operations Research 15 (1967) 266–277.

[13] W. Henderson, P.G. Taylor, Product form in networks of queues with batch arrivals and batch services, Queueing Systems 6 (1990) 71-88.

[14] W. Henderson, P.G. Taylor, Some new results on queueing networks with geometric release probabilities, Journal of Applied Probability 28 (1991) 409-421.

[15] J.R. Jackson, Networks of waiting lines, Operations Research 5 (1957) 518-521.

[16] F.P. Kelly, Reversibility and stochastic networks, John Wiley & Sons, New York, 1979.

[17] M. Miyazawa, On the characterization of departure rules for discrete-time queueing networks with batch movements and its applications, Queueing Systems 18 (1994) 149-166.

[18] M. Miyazawa, Structure-reversibility and departure functions of queueing networks with batch movements and state dependent routing, Queueing Systems 25 (1997) 45-75.

[19] M.L. Spearman, M.A. Zazanis, Push and pull production systems: issues and comparisons, Operations Research 40 (1992) 521-532.

[20] N.M. van Dijk, Queueing networks and product forms: A systems approach, John Wiley & Sons, Chichester, 1993.