Properties of Fork/Join Queueing Networks with Blocking under Various Operating Mechanisms

Yves Dallery, Zhen Liu, Don Towsley

Abstract— **Queueing networks with fork/join mechanisms and finite-capacity buffers are of interest because they are suited to modeling and evaluating the performance of a large class of discrete event systems such as manufacturing systems (e.g., manufacturing flow lines, assembly lines, kanban controlled manufacturing systems). In a recent paper, Dallery, Liu and Towsley [14] considered a special class of queueing networks with fork/join mechanisms and finite-capacity buffers called Basic Fork/Join Queueing Networks with Blocking (B-FJQN/B). For this class of networks, they established duality, reversibility, symmetry and concavity properties. However, in order to be able to accurately model the various operating mechanisms (blocking, loading and unloading mechanisms) encountered in manufacturing systems, it is necessary to consider a larger class of networks that will be referred to as Fork/Join Queueing Networks with Blocking (FJQN/B). The purpose of this paper is to introduce this class of queueing networks and investigate its properties. The approach is to first show that any FJQN/B can be equivalently represented as a B-FJQN/B and then use results derived in [14] for the underlying B-FJQN/B to establish the properties of the model under consideration. This approach is also used to compare the behavior of two models having different operating mechanisms. The usefulness of these results for performance evaluation and optimization of manufacturing systems is illustrated through a few examples.**

Keywords— **discrete event systems, manufacturing systems performance evaluation, fork/join queueing networks, finite buffers, blocking mechanisms, loading and unloading policies, throughput properties.**

I. Introduction

Queueing networks with fork/join mechanisms and finite-capacity buffers are of interest because they are suited to modeling and evaluating the performance of a large class of discrete event systems such as manufacturing systems. Tandem or closed tandem queueing networks with finite-capacity buffers can be used to model manufacturing flow lines (also referred to as production lines or transfer lines) [13]. Join operations can be used to model assembly systems [18], [20]. In this case, a join operation corresponds to the assembly of two or more subcomponents into a single component. Fork operations are useful in modeling disassembly operations. Consider, for instance, a manufacturing system where, at a certain stage, parts need to be fixed onto pallets. The resulting part/pallet pair then visits a set of machines which perform different operations on the part. Upon completion of these operations, the part is unloaded and the pallet is released. This unloading operation is a fork operation for which an item (the part/pallet pair) is split into two items (the part and the pallet). Fork and join operations are used in modeling kanban controlled manufacturing systems [19]. In such systems, a part is processed at a given stage of the manufacturing system only if a kanban associated with this stage is available. The kanban is held while the part is processed throughout this stage. It is released when the part is consumed by the next stage, i.e., when a kanban of the next stage is available. Thus, kanbans act as production orders. In this setting, a join operation corresponds to the assembly of a part and a kanban, and a fork operation corresponds to the disassembly of the part/kanban pair.

The purpose of this paper is to introduce and study the behavior of a class of queueing networks with fork/join mechanisms and finite-capacity buffers that can be used to model such manufacturing systems. A wide variety of mechanisms that exist in manufacturing systems are provided in these networks. For example, both blocking-after-service and blocking-before-service blocking mechanisms (cf. [29]) are considered. In the case of assembly and disassembly, both independent and simultaneous loading and unloading mechanisms are modeled. Here an independent loading mechanism allows an assembly machine to load different subcomponents independent of each other whereas a simultaneous loading mechanism requires that all subcomponents be present and loaded at the same time. Independent and simultaneous unloading mechanisms operate in an analogous manner at disassembly. Other features accounted for in these queueing networks are described later on.

We primarily focus on the behavior of the throughput of these networks. Specifically, we study properties of reversibility, symmetry, monotonicity and concavity. Our approach is to transform these queueing models into queueing networks belonging to a smaller class of networks first introduced in [2] and studied in [14] which we will refer to as Basic Fork/Join Queueing Networks with Blocking (B-FJQN/Bs). For such networks we established duality, reversibility, symmetry, and concavity properties in [14]. Except for the concavity property, these results were obtained under very weak assumptions on the sequences of service times (which include sequences of service times that are i.i.d. random variables). The concavity property is restricted to a class of distributions called PERT distributions [4].

Y. Dallery is with Laboratoire MASI, Universite Pierre et Marie Curie, 4 Place Jussieu, 75252 Paris Cedex 05, FRANCE. Z. Liu is with INRIA Centre Sophia Antipolis, 2004 route des Lucioles, 06565 Valbonne, FRANCE. D. Towsley is with Department of Computer Science, University of Massachusetts, Amherst, MA 01003, U.S.A. E-mail: dallery@masi.ibp.fr, liu@sophia.inria.fr, towsley@cs.umass.edu

The work of D. Towsley was supported in part by the National Science Foundation under grant ASC 88-8802764. This work was performed in part while the author was visiting Laboratoire MASI, UPMC, Paris.

We first establish the equivalence between the larger class of networks of interest to us in this paper and the class of B-FJQN/Bs. Using this equivalence, we extend some of the previously mentioned properties exhibited by B-FJQN/Bs to the larger class of FJQN/Bs. We also make some comparisons between different operating mechanisms by studying the B-FJQN/Bs that equivalently model these mechanisms. Some of the results established in this paper were already obtained in some special cases (mainly for tandem and closed tandem queueing networks). However, the results presented in this paper not only generalize and unify these results but also offer a simple way of transferring results obtained for the class of B-FJQN/B networks to a large variety of queueing networks with fork/join mechanisms and finite buffers. Besides obtaining more general results than those reported in the literature, our approach has the advantage of avoiding the aggravations that would typically accompany the establishment of these properties on a case-by-case basis.

Another approach for deriving properties of certain classes of queueing networks recently proposed by Glasserman and Yao [21] exploits comparison results established by these authors for specially structured generalized semi-Markov processes (GSMP's) [22], [23]. It requires that, among other things, one establish that the GSMP (or more precisely the so-called generalized semi-Markov scheme) exhibit properties such as *strong permutability* and *min-closure*. This approach was recently used, for example, by Rajan and Agrawal [31] to establish properties of a class of cyclic networks with so-called general blocking and starvation mechanisms.

The paper is organized as follows. In Section II, the class of Fork/Join Queueing Networks with Blocking (FJQN/Bs) considered in this paper is introduced. It is then shown how any model of this class can equivalently be represented as a B-FJQN/B. This equivalence is used in Section III to establish the properties of FJQN/Bs. Some applications of these results to models of manufacturing systems are described in Section IV and their usefulness for performance evaluation and optimization of manufacturing systems is discussed in Section V. Finally, extensions of our approach to handle a larger class of models is briefly discussed in Section VI.

II. Fork/Join Queueing Networks with Blocking under Various Operating Mechanisms

A. Definition

A Fork/Join Queueing Network with Blocking (FJQN/B) is a queueing network consisting of a set of servers and a set of buffers such that each buffer has exactly one upstream server and one downstream server. In addition, each server may have several input buffers and/or several output buffers. There is at most one buffer between any pair of servers. There may be some servers that have either no input buffers or no output buffers. A FJQN/B is structurally characterized by a directed graph whose nodes represent the servers and whose directed edges represent the buffers that connect the servers. Let V denote the set of servers and E the set of directed edges. An edge $(i,j) \in E$ indicates that there is a buffer connecting server i to server j and that jobs flow through this buffer from server i to server j. Let $G = (V,E)$ denote the underlying directed graph, which is assumed to be connected.

Let $u(i)$ be the set of upstream buffers (or input buffers) of server i and $d(i)$ be the set of downstream buffers (or output buffers) of server i:

$$u(i) = \{(j,i) \in E, j \in V\}, \qquad i \in V,$$

$$d(i) = \{(i,j) \in E, j \in V\}, \qquad i \in V.$$

Let $p(i)$ be the set of immediate server predecessors of server i and $s(i)$ be the set of immediate server successors of server i:

$$p(i) = \{j \in V \mid (j,i) \in E\}, \qquad i \in V,$$

$$s(i) = \{j \in V \mid (i,j) \in E\}, \qquad i \in V.$$

A server of a FJQN/B is generally referred to as an *assembly/disassembly* server. Consider server $i \in V$; if it has no upstream buffers, i.e., $u(i) = \emptyset$, it is referred to as a *source* and, if it has no downstream buffers, i.e., $d(i) = \emptyset$, it is referred to as a *sink*. If server i has at most one upstream buffer and at most one downstream buffer, i.e., $|u(i)| \leq 1$ and $|d(i)| \leq 1$, it is referred to as a *simple server*. If server i has several upstream buffers and at most one downstream buffer, i.e., $|u(i)| > 1$ and $|d(i)| \leq 1$, it is referred to as an *assembly server*; on the other hand if i has several downstream buffers and at most one upstream buffer, i.e., $|u(i)| \leq 1$ and $d(i)| > 1$, it is referred to as a *disassembly server*.

In the context of manufacturing systems, a source can be interpreted as an input server which has an infinite supply of raw parts in front of it. Similarly, a sink can be interpreted as an output server which has an infinite storage area downstream of it.

An example of a FJQN/B is given in Figure 1. This FJQN/B has 10 servers and 13 buffers. Servers 1 and 2 are sources and servers 6 and 10 are sinks; servers 2, 3, 6 and 9 are simple servers; server 10 is an assembly server; servers 1 and 8 are disassembly server; servers 4, 5 and 7 are assembly/disassembly servers.

Each buffer (i,j) has a finite capacity $B_{i,j} \in \mathbb{N}^+$. Let $\boldsymbol{B}$ denote the buffer capacity vector. In addition, some servers may have buffer space to accommodate jobs in service. For the sake of simplicity, we only consider the following two cases: either a server has no buffer space to accommodate jobs in service, or a server has space to accommodate all of the jobs in service, i.e., a job from each of its upstream buffers. The first type of server will be referred to as an *unbuffered server* or *U-server* while a server of the second type will be referred to as a *buffered server* or *B-server*. B-servers are naturally encountered in the modeling of manufacturing systems because they represent machines that usually have space to accommodate the parts currently in process. On the other hand, U-servers are useful because they correspond to the basic class of FJQN/Bs studied in [14] and used in this paper as a building block.

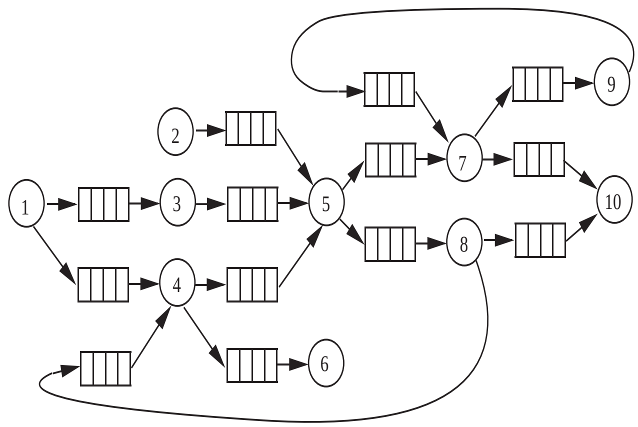

Fig. 1. Example of a FJQN/B.

In order to characterize the behavior of a FJQN/B, we first need to define the behavior of the servers. Consider first a U-server, say i. Server i initiates a *service period* (or *service activity*) whenever there resides at least one job in each of the buffers in $u(i)$ and there is space for at least one job in each of the buffers in $d(i)$. Jobs remain in the buffers in $u(i)$ throughout the service period. At the completion of the service period, a job is removed from each of the buffers in $u(i)$ and a job is immediately placed in each of the buffers in $d(i)$.

We now define the behavior of B-servers. There are many different types of B-servers that differ from each other according to the loading and unloading policies as well as the blocking mechanism. For simplicity, we will only consider several special cases which cover most practical applications. We note however that the methodology used in this paper can be applied to servers with other operating mechanisms.

General behavior. All of the B-servers considered in this paper share the following behavior.
- Any time a B-server performs an operation, it consumes one job from each of its upstream buffers and places one job in each of its downstream buffers. (Note that this is similar to the behavior of U-servers.)
- A B-server cannot begin its service activity unless all of the jobs (one from each of its upstream buffers) have been loaded onto the server.
- No new job can be loaded onto the server before all of the jobs produced during the previous service activity have been unloaded.

Loading policy. We consider two loading policies referred to as the *independent loading* (IL) and *simultaneous loading* (SL) policies.
- In the case of the IL policy, the jobs coming from each of the upstream buffers can be loaded onto the server independently of the other jobs.
- In the case of the SL policy, jobs stay in the upstream buffers until one job in each of the upstream buffers is available. At this instant, all of the jobs are simultaneously (and instantaneously) loaded onto the server.

Unloading policy. We consider two unloading policies referred to as the *independent unloading* (IU) and *simultaneous unloading* (SU) policies.
- In the case of the IU policy, jobs can be unloaded independently of each other into the downstream buffers.
- In the case of the SU policy, jobs remain on the server until space becomes available in each of the downstream buffers. At the instant that space is available for all jobs, they are simultaneously (and instantaneously) unloaded into the downstream buffers.

Blocking mechanism. We consider three blocking mechanisms referred to as the *blocking-after-service* (BAS), *blocking-before-service* (BBS), and *blocking-before-service with conditional loading* (BBS-CL) mechanisms.
- In the case of the BAS mechanism, a server initiates a service activity whenever a job from each of its upstream buffers has been loaded onto the server. The server becomes blocked if at the end of the service activity, at least one job cannot be unloaded (because a downstream buffer is full). It remains blocked until all the jobs have been unloaded.
- In the case of the BBS mechanism, a server initiates a service activity whenever a job from each of its upstream buffers has been loaded onto the server and there is space for at least one job in each of its downstream buffers. A server is blocked if at least one of its downstream buffers is full.
- The BBS-CL mechanism is identical to the BBS mechanism except that no job can be loaded onto the server while the server is blocked. If server i operates under the BBS-CL mechanism, we assume that $B_{i,j} > 1, \forall j \in s(i)$.

Remark. Note that, in the case of the BBS and BBS-CL mechanisms, the unloading policy is irrelevant since a buffer space in each of the downstream buffers is, by definition, available at the end of any service activity.

Combining these different operating mechanisms and taking the preceding remark into account yields 8 different types of B-servers:
- BAS/IL/IU server: a B-server with a blocking-after-service mechanism, independent loading policy and independent unloading policy;
- BAS/SL/SU server: a B-server with a blocking-after-service mechanism, simultaneous loading policy and simultaneous unloading policy;
- BAS/IL/SU server: a B-server with a blocking-after-service mechanism, independent loading policy and simultaneous unloading policy;
- BAS/SL/IU server: a B-server with a blocking-after-service mechanism, simultaneous loading policy and independent unloading policy;
- BBS/IL server: a B-server with a blocking-before-service mechanism and independent loading policy;
- BBS/SL server: a B-server with a blocking-before-service mechanism and simultaneous loading policy;
- BBS-CL/IL server: a B-server with a blocking-before-service with conditional loading mechanism and independent loading policy;
- BBS-CL/SL server: a B-server with a blocking-before-service with conditional loading mechanism and simulta-

neous loading policy.

Let T_i denote the type of server i, either a U-server or one of the above 8 types of B-servers. Let $\boldsymbol{T}$ denote the server type vector for the network.

Remark. The operating mechanisms described above can be associated with any general assembly/disassembly B-server (referred to as a server from now on). However, in some cases, a server will exhibit identical behavior under two different mechanisms. For example, the loading and unloading policies are irrelevant in the case of a simple server. There are actually only three different types of operating mechanisms in this case: BAS, BBS and BBS-CL. (Note that in the case of a simple server, BBS and BBS-CL have alternatively been referred to as blocking-before-service with place occupied (BBS-PO) and blocking-before-service with place non-occupied (BBS-PNO) in [29].) In the case of a disassembly server (and a source), the loading policy is irrelevant, in the case of an assembly server (and a sink), the unloading policy is irrelevant, and in the case of a sink, the blocking mechanism is irrelevant.

In general, the behavior of a FJQN/B depends on its initial condition. This initial condition includes the number of jobs present in each buffer at time $t = 0$ as well as the information pertaining to the initial condition of each B-server. Let $M_{i,j}$ denote the total number of jobs present at time $t = 0$ in buffer (i, j). Note that $0 \leq M_{i,j} \leq B_{i,j}$. Also, $B_{i,j} - M_{i,j}$ represents the number of holes, i.e., the available space, in the buffer. In the case of a B-server, we also need to specify its initial condition. Consider B-server i with any blocking mechanism (BAS, BBS or BBS-CL). Let $L_{i,j}$ denote whether a job from buffer (j, i) is currently loaded on the server ($L_{i,j} = 1$) or not ($L_{i,j} = 0$), $j \in p(i)$. In the case of a simultaneous loading policy, $L_{i,j} = L_i$ does not depend on the predecessor node $j \in p(i)$. Note that in the case of a server with a BBS-CL mechanism, this initial condition has to be consistent with the blocking mechanism. Indeed, the server space cannot be occupied if one of the downstream buffers is full. In the case of a B-server with the BAS mechanism, it may be of interest to consider an initial condition for which the server is blocked. Let $U_{i,k} = 1$ if server i has completed its service activity but the job going to buffer (i, k) cannot be unloaded because this buffer is full, and let $U_{i,k} = 0$ otherwise, $k \in s(i)$. For a simultaneous unloading policy, $U_{i,k} = U_i$ does not depend on the successor node $k \in s(i)$. Note that the quantities L_i, or $L_{i,j}, j \in p(i)$, and U_i, or $U_{i,k}, k \in s(i)$, have to be consistent with the condition that no new job can be loaded onto the server before all of the jobs produced during the previous service activity have been unloaded. Last, we arbitrarily assign $L_{i,j} = U_{i,k} = 0$ in the case of U servers, and $U_{i,k} = 0$ in the case of type BBS B-servers.

Let $\boldsymbol{M}$ be the initial marking vector of the buffers. Let $\boldsymbol{L}$ and $\boldsymbol{U}$ be the initial marking vectors of the servers. A FJQN/B is further characterized by the durations of service times which are random variables (r.v.'s). The durations of the service periods at server i are given by a sequence of non-negative service times, $\{\sigma_{i,n}\}_{n\geq 1}$, $i \in V$. The complete characterization of a FJQN/B, say $\mathcal{S}$, is thus given by $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M}, \boldsymbol{T}, \boldsymbol{L}, \boldsymbol{U})$.

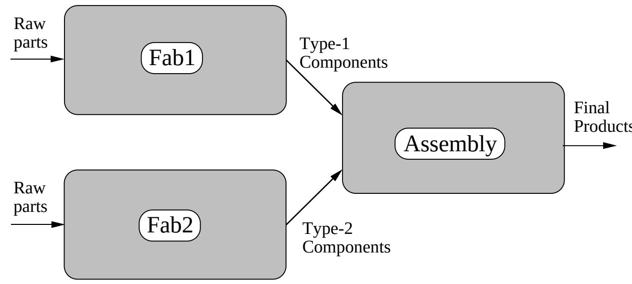

Fig. 2. Structure of the fabrication/assembly system.

Let $D_{i,n}$ denote the completion time of the n-th service period of server i. Denote by $\theta_i(\mathcal{S})$ the (asymptotic) throughput of server $i \in V$, given by:

$$\theta_i(\mathcal{S}) = E\left[\left(\lim_{n\to\infty} \frac{D_{i,n}}{n}\right)^{-1}\right], \qquad i \in V, \tag{1}$$

provided the limit exists. In Section III we establish properties of the throughput of FJQN/Bs.

Remark. The class of B-FJQN/Bs studied in [14] is a subclass of the class of FJQN/Bs defined above. Indeed, a B-FJQN/B is a FJQN/B in which all of the servers are U-servers. The characterization of a B-FJQN/B is simply given by $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M})$.

B. Example of the modeling of manufacturing systems by FJQN/Bs.

The purpose of this section is to illustrate the usefulness of the class of FJQN/Bs for modeling manufacturing systems. For that purpose, we consider an example of a kanban controlled fabrication/assembly system [17]. This system consists of two fabrication stages and an assembly stage (see Figure 2). Each fabrication stage produces one basic component. The final products are then obtained by the assembly of the components produced by each of the fabrication stages. The first fabrication stage (Fab1) consists of a series of 3 machines, M_1, M_2 and M_3, while the second fabrication stage (Fab2) consists of a series of 2 machines, M_4 and M_5. The assembly stage consists of 3 machines, M_6, M_7 and M_8. M_8 is the station that assembles the two components. M_6 (resp. M_7) is a machine that performs a preliminary assembly operation on type-1 (resp. type-2) components, for instance the insertion of some external parts.

This fabrication/assembly system is controlled by a multi-stage kanban system [9], [19], [17]. There are three kanban loops, each controlling one of the stages. The first and second kanban loops control the release of raw materials to Fab1 and Fab2, respectively. Let K_1 (resp. K_2) be the number of kanbans associated with Fab1 (resp. Fab2).

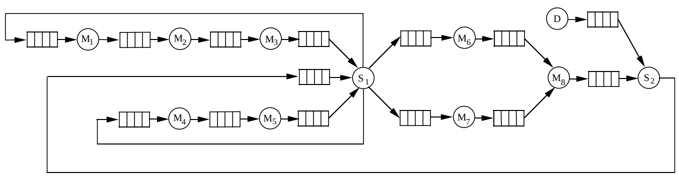

Fig. 3. FJQN/B model of a kanban controlled fabrication/assembly system.

As soon as a type-1 (resp. type-2) kanban is available, a raw part is released to Fab1 (resp. Fab2)[1]. Upon completion of processing at the different machines of a fabrication stage, the component (to which is attached the kanban) is stored in an output buffer. The third kanban loop controls the entry of the basic components into the assembly stage. Let K_3 be the number of kanbans associated with the assembly stage. As soon as a type-3 kanban as well as one component of each type are available, the components are released to the assembly stage and a type-1 (resp. type-2) kanban is returned to the input of Fab1 (resp. Fab2). The system is driven by actual demands. When a demand arrives to the system, a finished product is consumed and the corresponding type-3 kanban is returned to the input of the assembly stage. If no finished part is available when a demand arrives, the demand is backordered.

This kanban controlled assembly system can be modeled as a FJQN/B [17]. The set of servers are the machines, M_1,..., M_8, two synchronization stations, say S_1 and S_2, and an additional server modeling the arrival of external demands, say D. Synchronization station S_1 models the synchronization between the fabrication stages and the assembly stage. Synchronization station S_2 models the synchronization between the assembly stage and the demand process. Both synchronization stations have zero processing times. The demand process is modeled as an additional server, D, whose service times corresponds to the interarrival times of the demand process. All buffers pertaining to Fab1 (resp. Fab2) have capacity K_1 (resp. K_2). All buffers pertaining to the assembly stage have capacity K_3. The backlog buffer (i.e., the buffer between server D and synchronization station S_2) has infinite capacity. The initial condition is that the output buffers of Fab1, Fab2 and the assembly stage contain K_1, K_2 and K_3 jobs at time 0, respectively, and all other buffers are empty. Servers S_1, S_2 and D are U-servers, while the other servers are B-servers. The assumptions regarding the behavior of the B-servers depend on the specific system under consideration. For instance the blocking mechanism in the fabrication stages could be of BAS-type, while the blocking mechanism within the assembly stage could be of BBS-type. The loading mechanism would be of IL or SL type, depending on whether the components are loaded independently or simultaneously on the assembly machine, M_8. The FJQN/B model of the above fabrication/assembly system is shown in Figure 3.

[1]Infinite supplies of raw parts are assumed to be present at the inputs of Fab1 and Fab2.

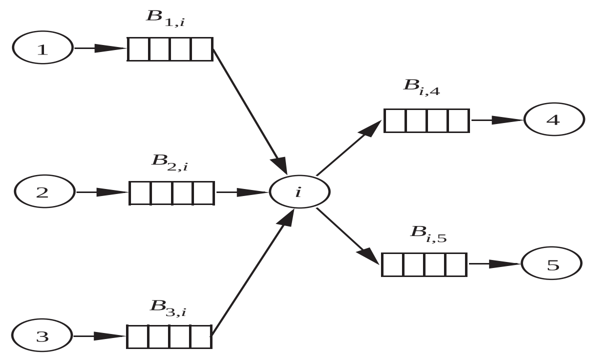

Fig. 4. The original FJQN/B.

Note that there are two ways of implementing a kanban control mechanism for assembly systems as described in [17]. The one considered above is the so-called kanban control system with *simultaneous release*. In the other one, the so-called kanban control system with *independent release*, the components may be released to the assembly stage independently of one another, given that there is a type-3 kanban available. It should be emphasized that under the assumption of independent release, the system can also be described as a FJQN/B; see [17] for details.

C. Modeling a FJQN/B as a B-FJQN/B

In this section, we describe how to transform a FJQN/B $\mathcal{S}$ into an *equivalent* B-FJQN/B $\mathcal{S}^b$ such that, given any sequences of service times, $\{\sigma_{i,n}\}_{n\geq 1}$, $i \in V$ and equivalent initial conditions, the sample path behaviors of the two networks are identical. In particular, the instants of the beginning and completion of the n-th service activity of each server i are the same in both networks, $\forall i \in V, n \geq 1$, and there is a one to one correspondence between the buffer contents of both networks. The equivalent B-FJQN/B $\mathcal{S}^b$ is obtained by transforming each server in $\mathcal{S}$, one at a time, into a subnetwork consisting solely of U-servers.

Let $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M}, \boldsymbol{T}, \boldsymbol{L}, \boldsymbol{U})$ be an arbitrary FJQN/B. Consider any B-server of $\mathcal{S}$, say i. We transform $\mathcal{S}$ into a second FJQN/B $\mathcal{S}' = (V', E', \boldsymbol{B}', \boldsymbol{M}', \boldsymbol{T}', \boldsymbol{L}', \boldsymbol{U}')$ that differs from $\mathcal{S}$ only in that server i is replaced by a subnetwork consisting of U-servers. We describe below how to construct $\mathcal{S}'$ for the different types of B-servers. For the sake of simplicity, this is done using the 6 server FJQN/B shown in Figure 4. Servers 1 to 5 are arbitrary servers and server i, is the B-server that will be transformed into a subnetwork of U-servers. Servers 1, 2 and 3 are sources while servers 4 and 5 are sinks. We have $p(i) = \{1, 2, 3\}$ and $s(i) = \{4, 5\}$.

Since the transformation focuses on server i and not on the rest of the FJQN/B, we first characterize the overall transformation of $\mathcal{S}$ into $\mathcal{S}'$ without giving details regarding the transformation of server i. This overall characterization is given once as it does not depend on the server type of i. Then we describe the transformation of server i for each of the different B-server types. This is done by showing the B-FJQN/B obtained after transformation of the FJQN/B in Figure 4. For the case of a BAS/IL/IU server, we also provide a mathematical description of the transformation. Similar mathematical descriptions for the other cases are given in [15]. In each case it is easy to check that the FJQN/B $\mathcal{S}'$ obtained from $\mathcal{S}$ by transforming server i exhibits the same behavior as the original FJQN/B $\mathcal{S}$. A formal statement of equivalence and its proof can be found in [15].

Overall transformation. The equivalent FJQN/B $\mathcal{S}'$ is:

$$
\begin{array}{lcll}
V' & = & V + V_a, & \\
E' & = & E - \{(j,i), j \in p(i)\} & \\
 & - & \{(i,k), k \in s(i)\} + E_a, & \\
B'_{j,k} & = & B_{j,k}, & j,k \in V,\ j,k \neq i, \\
M'_{j,k} & = & M_{j,k}, & j,k \in V,\ j,k \neq i, \\
L'_{j,k} & = & L_{j,k}, & j,k \in V,\ k \neq i, \\
U'_{j,k} & = & U_{j,k}, & j,k \in V,\ j \neq i, \\
T'_j & = & T_j, & j \in V,\ j \neq i, \\
\sigma'_{j,n} & = & \sigma_{j,n}, & j \in V,\ j \neq i,\ n \geq 1,
\end{array}
$$

where V_a and E_a are the servers and edges within the subnetwork that will represent server i within $\mathcal{S}'$. These, as well as the buffer sizes and initial markings pertaining to the subnetwork, will now be described for each type of server i.

Transformation of a BAS/IL/IU server. The subnetwork transformation of server i is defined as follows:

$$
\begin{array}{lcl}
V_a & = & \{l_{i,j}, j \in p(i)\} + \{u_{i,k}, k \in s(i)\}, \\
E_a & = & \{(j, l_{i,j}), j \in p(i)\} + \{(l_{i,j}, i), j \in p(i)\} \\
 & & +\{(u_{i,k}, k), k \in s(i)\} + \{(i, u_{i,k}), k \in s(i)\} \\
 & & + \{(l_{i,j}, u_{i,k}), j \in p(i), k \in s(i)\},
\end{array}
$$

$$
\begin{array}{lclcll}
B'_{j,l_{i,j}} & = & B_{j,i}, & M'_{j,l_{i,j}} & = & M_{j,i}, \quad j \in p(i), \\
B'_{u_{i,k},k} & = & B_{i,k}, & M'_{u_{i,k},k} & = & M_{i,k}, \quad k \in s(i), \\
B'_{l_{i,j},i} & = & 1, & M'_{l_{i,j},i} & = & L_{i,j}, \quad j \in p(i), \\
B'_{i,u_{i,k}} & = & 1, & M'_{i,u_{i,k}} & = & U_{i,j}, \quad k \in s(i), \\
B'_{l_{i,j},u_{i,k}} & = & 1, & \multicolumn{3}{l}{j \in p(i),\ k \in s(i),} \\
M'_{l_{i,j},u_{i,k}} & = & \multicolumn{4}{l}{L_{i,j} + U_{i,k},\ j \in p(i), \quad k \in s(i),}
\end{array}
$$

$$
\begin{array}{lcll}
\sigma'_{i,n} & = & \sigma_{i,n}, & n \geq 1, \\
\sigma'_{l_{i,j},n} & = & 0, & j \in p(i), \quad n \geq 1, \\
\sigma'_{u_{i,k},n} & = & 0, & k \in s(i), \quad n \geq 1.
\end{array}
$$

The B-FJQN/B equivalent to the FJQN/B shown in Figure 4 is given in Figure 5 for the case that server i is a BAS/IL/IU server. The shaded part models the behavior of server i. All the buffers within this shaded area have capacity one. The service activity of any server $l_{i,j}, j \in p(i)$, represents the loading of a job from buffer (j,i) to server i. The service activity of any server $u_{i,k}, k \in s(i)$, represents the unloading of a job from server i to buffer (k,i). Buffers $(l_{i,j}, u_{i,k}), j \in p(i), k \in s(i)$, ensure that no new job is loaded onto the server before all the jobs produced during the previous service activity have been unloaded.

BAS/SL/SU server. The B-FJQN/B equivalent to the FJQN/B shown in Figure 4 is given in Figure 6 for the case that server i is a BAS/SL/SU server. The shaded part models the behavior of server i. All of the buffers within this shaded area have capacity one. The service activity of server l_i represents the simultaneous loading of the jobs. The service activity of server u_i represents the simultaneous unloading of the jobs. Buffer (l_i, u_i) ensures that no new job is loaded onto the server before all of the jobs produced during the previous service activity have been unloaded.

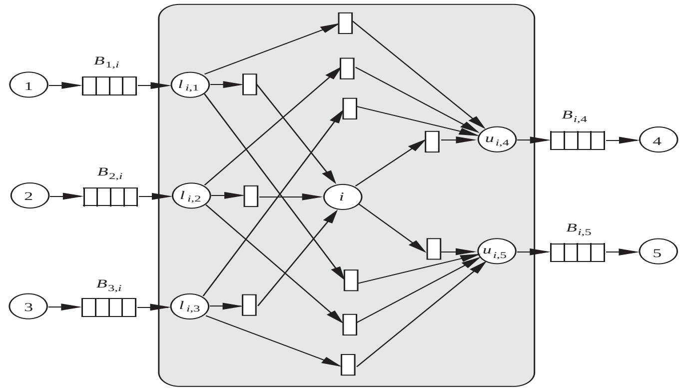

Fig. 5. Illustration of the transformation of a BAS/IL/IU server.

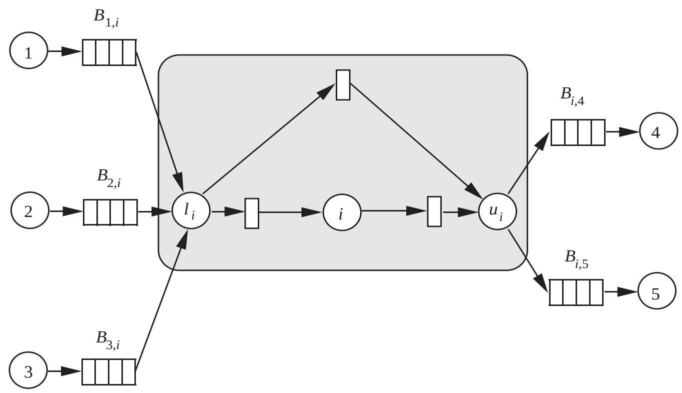

Fig. 6. Illustration of the transformation of a BAS/SL/SU server.

BAS/IL/SU server. The B-FJQN/B equivalent to the FJQN/B shown in Figure 4 is given in Figure 7 for the case that server i is a BAS/IL/SU server. The shaded part models the behavior of server i. All of the buffers within this shaded area have capacity one. The service activity of any server $l_{i,j}, j \in p(i)$, represents the loading of a job from buffer (j,i). The service activity of any server u_i represents the simultaneous unloading of the jobs. Buffers $(l_{i,j}, u_i), j \in p(i)$ ensure that no new job is loaded onto the server before all of the jobs produced during the previous service activity have been unloaded.

BAS/SL/IU server. The B-FJQN/B equivalent to the FJQN/B shown in Figure 4 is given in Figure 8 for the case that server i is a BAS/SL/IU server. The shaded part models the behavior of server i. All of the buffers within this shaded area have capacity one. The service activity of server l_i represents the simultaneous loading of the jobs. The service activity of server $u_{i,k}, k \in s(i)$, represents the unloading of a job into buffer (k,i). Buffer $(l_i, u_{i,k}), k \in s(i)$, ensures that no new job is loaded onto

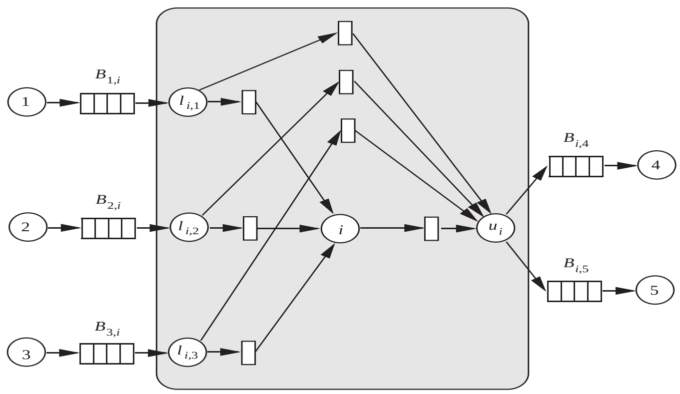

Fig. 7. Illustration of the transformation of a BAS/IL/SU server.

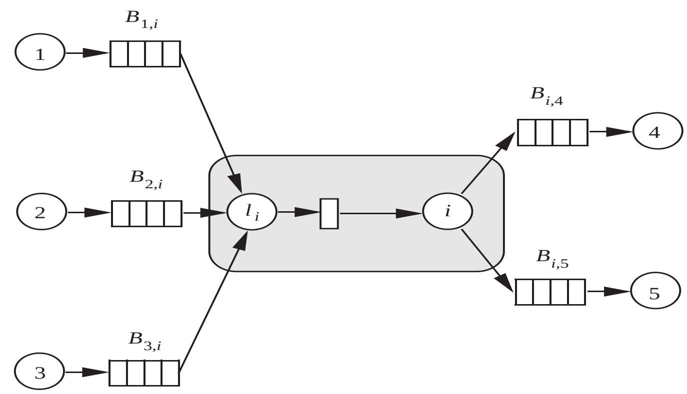

Fig. 10. Illustration of the transformation of a BBS/SL server.

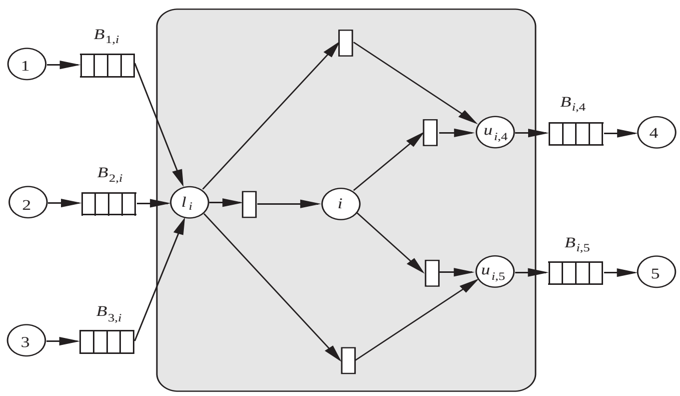

Fig. 8. Illustration of the transformation of a BAS/SL/IU server.

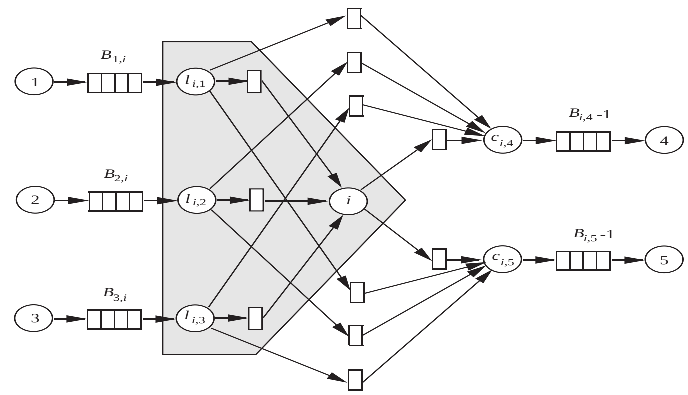

Fig. 11. Illustration of the transformation of a BBS-CL/IL server.

the server before all of the jobs produced during the previous service activity have been unloaded.

BBS/IL server. The B-FJQN/B equivalent to the FJQN/B shown in Figure 4 is given in Figure 9 for the case that server i is a BBS/IL server. The shaded part models the behavior of server i. All of the buffers within this shaded area have capacity one. The service activity of server $l_{i,j}, j \in p(i)$, represents the loading of a job from buffer (j, i).

BBS/SL server. The B-FJQN/B equivalent to the FJQN/B shown in Figure 4 is given in Figure 10 for the case that server i is a BBS/SL server. The shaded part models

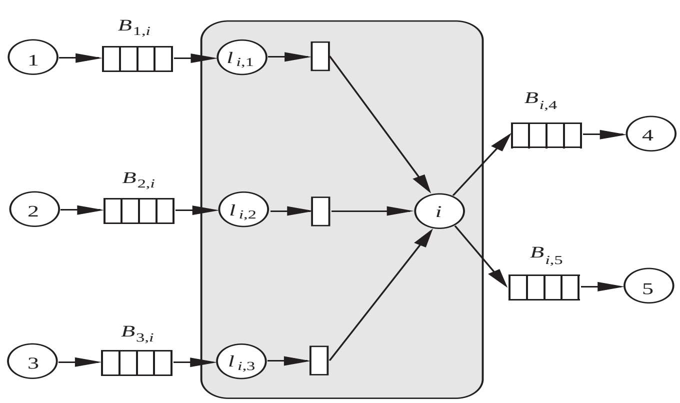

Fig. 9. Illustration of the transformation of a BBS/IL server.

the behavior of server i. The buffer within this shaded area has capacity one. The service activity of server l_i represents the simultaneous loading of the jobs.

BBS-CL/IL server. The B-FJQN/B equivalent to the FJQN/B shown in Figure 4 is given in Figure 11 for the case that server i is a BBS-CL/IL server. The shaded part models the behavior of server i. All of the buffers within this shaded area have capacity one. The service activity of server $l_{i,j}, j \in p(i)$, represents the loading of a job from buffer (j, i). Each buffer $(i, k), k \in s(i)$ is split into two buffers, $(i, c_{i,k})$ and $(c_{i,k}, k)$, separated by a server, $c_{i,k}$. These buffers have capacities $B_{i,k} - 1$ and 1, respectively. Server $c_{i,k}$ has zero service time. The service activity of any server $c_{i,k}, k \in s(i)$, represents the transfer of a job from the last position of buffer (i, k) to the last but one position. Buffer $(l_{i,j}, c_{i,k}), j \in p(i), k \in s(i)$, having capacity one, ensures that no new job is loaded onto the server until the last position of buffer (i, k) is unoccupied.

BBS-CL/SL server. The B-FJQN/B equivalent to the FJQN/B shown in Figure 4 is given in Figure 12 for the case that server i is a BBS-CL/SL server. The shaded part models the behavior of server i. The buffer within this shaded area has capacity one. The service activity of server l_i represents the simultaneous loading of the jobs. Buffer $(i, k), k \in s(i)$, is split into two buffers, $(i, c_{i,k})$ and $(c_{i,k}, k)$, separated by a server, $c_{i,k}$. These buffers have capacity $B_{i,k} - 1$ and 1, respectively. Server $c_{i,k}$ has zero service time. The service activity of server $c_{i,k}, k \in s(i)$, represents

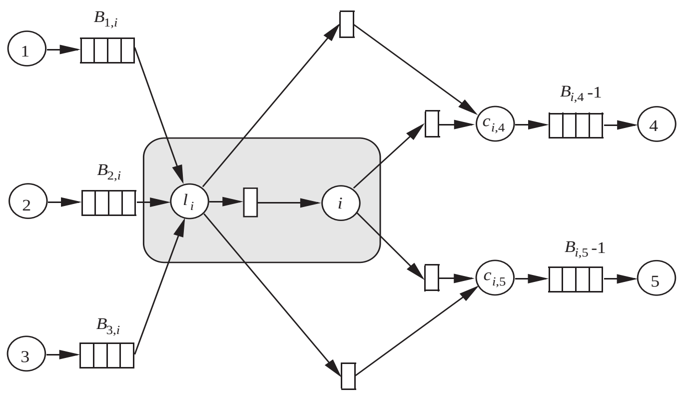

Fig. 12. Illustration of the transformation of a BBS-CL/SL server.

the transfer of a job from the last position of buffer (i,k) to the last but one position. Buffer $(l_i, c_{i,k}), k \in s(i)$, having capacity one, ensures that no new job is loaded onto the server until the last position of buffer (i,k) is unoccupied.

III. Properties of FJQN/Bs

In this section, we establish properties of FJQN/Bs. These properties pertain to the asymptotic throughput of B-FJQN/Bs and are obtained by converting an arbitrary FJQN/B into an equivalent B-FJQN/B as described in Section II-C together with the properties of B-FJQN/Bs established in [14]. They require one or more of the following assumptions on the sequences of service times.

Assumption A1. The service times form jointly stationary and ergodic sequences of integrable random variables (r.v.'s).

Assumption A2. The sequences of service times at different servers are mutually independent.

Assumption A3. The sequences of service times are jointly reversible.

Assumption A4. The service times form mutually independent sequences of independent and identically distributed (i.i.d.) r.v.'s having PERT distributions [4], [14].

It is worth emphasizing that the first three assumptions include the case where the service times are i.i.d. r.v.'s as a special case. Assumption A4 is more restrictive but includes exponential and Erlang distributions as special cases.

Special cases of some of the results presented here were obtained in earlier papers [1], [3], [6], [24], [25], [26], [28], [32], [35]. Most of these results pertain to tandem and closed tandem queueing networks.

For simplicity, unless otherwise stated, we assume throughout this section that the initial condition of a FJQN/B is such that all of the B-servers are empty so that all of the jobs reside in the buffers, i.e., all of the components of $\boldsymbol{L}$ and $\boldsymbol{U}$ are zero. In this case, a FJQN/B $\mathcal{S}$ is simply characterized by $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M}, \boldsymbol{T})$. This will allow us to keep the notation as simple as possible. However, all of the results that we present generalize to other initial conditions.

Throughout this section, we only consider deadlock-free FJQN/Bs. Whether or not a FJQN/B is deadlock-free can be checked by verifying the conditions given in [14] on the equivalent B-FJQN/Bs.

A. Throughput

Consider a FJQN/B $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M}, \boldsymbol{T})$. The following result establishes the existence of the asymptotic throughputs of the servers of a FJQN/B, as well as their equality. It is a consequence of Theorem 5.2 in [14] and the equivalence of FJQN/Bs and B-FJQN/Bs.

Theorem 1: If $\mathcal{S}$ is a FJQN/B, then, under Assumption A1, the asymptotic throughputs exist and satisfy the following relation:

$$\theta_i(\mathcal{S}) = \theta(\mathcal{S}), \quad \forall i \in V.$$

Henceforth, $\theta(\mathcal{S})$ will be referred to as the throughput of $\mathcal{S}$.

Let C be a cycle in (V, E) and let $E(C)$ denote the set of edges of C. Let us define an arbitrary orientation of this cycle. $E(C)$ can be partitioned into two subsets with respect to this reference orientation. Let $E^+(C)$ be the subset of edges oriented according to the reference orientation and $E^-(C)$ be the subset of edges oriented in the reverse direction. Let $I_C^+(\boldsymbol{M})$ be the total number of jobs in all buffers corresponding to the reference direction plus the total number of holes in all buffers corresponding to the reverse direction, i.e.,

$$I_C^+(\boldsymbol{M}) = \sum_{(i,j)\in E^+(C)} M_{i,j} + \sum_{(i,j)\in E^-(C)} (B_{i,j} - M_{i,j}) \quad (2)$$

Last, an elementary cycle is defined as a cycle that contains no subcycle.

Definition 2: Let (V, E) contain exactly n_c distinct elementary cycles $C_1, \cdots, C_{n_c}$. Let

$$\boldsymbol{I}(\boldsymbol{M}) = (I_1^+(\boldsymbol{M}), I_2^+(\boldsymbol{M}), \cdots, I_{n_c}^+(\boldsymbol{M})).$$

Then markings $\boldsymbol{M}^1$ and $\boldsymbol{M}^2$ are equivalent (written $\boldsymbol{M}^1 \sim \boldsymbol{M}^2$) iff $\boldsymbol{I}(\boldsymbol{M}^1) = \boldsymbol{I}(\boldsymbol{M}^2)$.

The following result establishes the independence of the throughput of a FJQN/B with respect to equivalent initial markings. It is a consequence of Theorem 5.3 in [14] and the equivalence of FJQN-Bs and B-FJQN/Bs.

Theorem 3: Let $\boldsymbol{M}^1$ and $\boldsymbol{M}^2$ be two initial markings of a FJQN/B $\mathcal{S}$. Under Assumptions A1 and A2, if $\boldsymbol{M}^1 \sim \boldsymbol{M}^2$, we have:

$$\theta(\boldsymbol{M}^1) = \theta(\boldsymbol{M}^2).$$

B. Monotonicity and Concavity

The following result establishes the monotonicity of the throughput of a FJQN/B with respect to the buffer capacities. It is a consequence of corollaries 2.1 and 2.2 in [15] and the equivalence of FJQN-Bs and B-FJQN/Bs.

Theorem 4: Consider a FJQN/B $\mathcal{S}$ with two different buffer capacity vectors and initial marking vectors $(\boldsymbol{B}, \boldsymbol{M})$ and $(\boldsymbol{B}+\boldsymbol{k}, \boldsymbol{M}+\boldsymbol{j})$. Then, under Assumption A1, we have:

$$\theta(\boldsymbol{B}+\boldsymbol{k}, \boldsymbol{M}+\boldsymbol{k}) \geq \theta(\boldsymbol{B}, \boldsymbol{M}), \ \forall \boldsymbol{k} \geq \boldsymbol{j} \geq \boldsymbol{0}.$$

Special cases of the above result were previously obtained by Shanthikumar and Yao [32] in the case of a closed tandem queueing network and by Adan and Van Der Waal [1] in the case of an assembly network.

The following result establishes the concavity of the throughput of a FJQN/B with respect to both the buffer capacities and the initial marking. It is a consequence of Theorem 8.1 in [14] and the equivalence of FJQN-Bs and B-FJQN/Bs.

Theorem 5: Let $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M}, \boldsymbol{T})$ be a FJQN/B. Then, under Assumption A4, $\theta(\mathcal{S})$ is a concave function of $\boldsymbol{B}$ and $\boldsymbol{M}$.

Special cases of the above result were previously obtained by Anantharam and Tsoucas [3] and Meester and Shanthikumar [25] in the case of tandem queueing networks and by Shanthikumar and Yao [32] in the case of a closed tandem queueing networks, all under the assumption of exponential service times.

C. Comparison of Various Operating Mechanisms

Consider a particular B-server, say i, of a FJQN/B $\mathcal{S}$. We are interested in comparing the behavior of $\mathcal{S}$ under different operating mechanisms of server i. We first establish an equivalence between the BBS-CL and BAS blocking mechanisms. Let $\boldsymbol{B}$ be a buffer capacity vector and let $\boldsymbol{B}^{i+}$ be the buffer capacity vector obtained from $\boldsymbol{B}$ as:

$$B^{i+}_{j,k} = \begin{cases} B_{i,k} + 1, & \forall k \in s(i), \quad j = i, \\ B_{j,k}, & \forall j, k \in V, \quad j \neq i. \end{cases}$$

Theorem 6: Let $\mathcal{S}^1$ be a FJQN/B with buffer capacity vector $\boldsymbol{B}^1 = \boldsymbol{B}$ in which server i is a BAS/IL/IU (resp. BAS/SL/IU) server. Let $\mathcal{S}^2$ be a FJQN/B identical to $\mathcal{S}^1$ except that it has buffer capacity vector $\boldsymbol{B}^2 = \boldsymbol{B}^{i+}$ and server i is a BBS-CL/IL (resp. BBS-CL/SL) server. Then, $\mathcal{S}^2$ has exactly the same behavior as $\mathcal{S}^1$ in the sense that they have the same equivalent B-FJQN/B.

Proof. Consider for instance the case where server i is a BAS/IL/IU server in $\mathcal{S}^1$ and a BBS-CL/IL server in $\mathcal{S}^2$. (The proof for the case where server i is a BAS/SL/IU server in $\mathcal{S}^1$ and a BBS-CL/SL server in $\mathcal{S}^2$ is similar and therefore omitted.) Consider the two B-FJQN/Bs, $\mathcal{S}^{1b}$ and $\mathcal{S}^{2b}$, equivalent to $\mathcal{S}^1$ and $\mathcal{S}^2$, respectively. Let us concentrate on the transformations of server i as the remaining portions of the two networks remain unchanged. After applications of the transformations described in Section II-C to server i in both networks, we observe that the resulting B-FJQN/Bs are identical. Indeed, server $l_{i,j}$ in $\mathcal{S}^{2b}$ plays the same role as server $l_{i,j}$ in $\mathcal{S}^{1b}$, $\forall j \in p(i)$, and server $c_{i,k}$ in $\mathcal{S}^{2b}$ plays the same role as server $u_{i,k}$ in $\mathcal{S}^{1b}$, $\forall k \in s(i)$. Moreover the capacity of buffer $(c_{i,k}, k)$ in $\mathcal{S}^{2b}$ is $B^2_{i,k} - 1$ and the capacity of buffer $(u_{i,k}, k)$ in $\mathcal{S}^{1b}$ is $B^1_{i,k}$ and are both equal to $B_{i,k}$, $\forall k \in s(i)$. It is easy to check that the rest of the two B-FJQN/Bs are identical. ∎

Thus, the BBS-CL blocking mechanism is equivalent to the BAS blocking mechanism with a reduction in the buffer capacity of one in each of the downstream buffers. Consequently, we no longer deal with servers having BBS-CL blocking mechanisms. This equivalence between BAS and BBS-CL mechanisms was first reported by Onvural and Perros [28] for the special case of tandem queueing networks.

The following result compares the loading and unloading policies of a server with a BAS blocking mechanism.

Theorem 7: Let $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M}, \boldsymbol{T})$ be a FJQN/B. Let $\theta(BAS/IL/IU)$, $\theta(BAS/SL/SU)$, $\theta(BAS/IL/SU)$ and $\theta(BAS/SL/IU)$, denote the throughputs of this network when server i is a BAS/IL/IU server, a BAS/SL/SU server, a BAS/IL/SU server and a BAS/SL/IU server, respectively. Then, under Assumption A1, we have:

$$\theta(BAS/IL/IU) \geq \theta(BAS/IL/SU) \geq \theta(BAS/SL/SU),$$

and

$$\theta(BAS/IL/IU) \geq \theta(BAS/SL/IU) \geq \theta(BAS/SL/SU).$$

Proof. We prove that

$$\theta(BAS/IL/SU) \geq \theta(BAS/SL/SU).$$

The other inequalities are easily obtained using similar arguments. Let $\mathcal{S}^1$ (resp. $\mathcal{S}^2$) denote the FJQN/B $\mathcal{S}$ in the case where server i is a BAS/IL/SU (resp. BAS/SL/SU) server. Let $\mathcal{S}^{1b}$ (resp. $\mathcal{S}^{2b}$) be the B-FJQN/Bs equivalent to $\mathcal{S}^1$ and $\mathcal{S}^2$, respectively (see Figures 7 and 6). Consider the B-FJQN/B, say $\mathcal{S}^m$, obtained from $\mathcal{S}^{1b}$ as follows. For every $j \in p(i)$, add a buffer in between every server $k \in p(i), k \neq j$ and server $l_{i,j}$ having the same buffer capacity and initial marking as the buffer already existing between server j and server $l_{i,j}$, that is: $B_{j,i}$ and $M_{j,i}$. It then appears that all of the servers $l_{i,j}, j \in p(i)$, in $\mathcal{S}^m$ are connected exactly in the same way to the rest of the network and, as a result, behave identically. Moreover, they behave exactly in the same way as server l_i in $\mathcal{S}^{2b}$. They can actually be viewed as $|p(i)|$ replications of server l_i in $\mathcal{S}^{2b}$. In other words, the addition of the buffers to the B-FJQN/B $\mathcal{S}^{1b}$ has transformed the independent loading policy into a simultaneous loading policy. As a result, we have $\theta(\mathcal{S}^m) = \theta(\mathcal{S}^{2b})$. Now, $\mathcal{S}^{1b}$ is a subnetwork of $\mathcal{S}^m$ and thus, from Corollary 2.3 in [15], we get $\theta(\mathcal{S}^{1b}) \geq \theta(\mathcal{S}^m)$. As a result, we have $\theta(\mathcal{S}^{1b}) \geq \theta(\mathcal{S}^{2b})$ and, thus, $\theta(BAS/IL/SU) \geq \theta(BAS/SL/SU)$. ∎

This theorem states that an independent loading (resp. unloading) policy achieves a higher throughput than a simultaneous loading (resp. unloading) policy. The result pertaining to the ordering of the loading policies was stated (however not proved) in [6] in the case of an assembly network. (In that paper, the independent and simultaneous loading policies were called push and pull modes).

The following result compares the performance of a server using the BBS blocking mechanism under different loading policies.

Theorem 8: Let $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M}, \boldsymbol{T})$ be a FJQN/B. Let $\theta(BBS/IL)$ and $\theta(BBS/SL)$ denote the throughputs of this network when server i is a BBS/IL server and a BBS/SL server, respectively. Then, under Assumption A1, we have:

$$\theta(BBS/IL) \geq \theta(BBS/SL).$$

Proof. The proof is similar to that of Theorem 7 and is therefore omitted.

The last result is concerned with the comparison of BAS and BBS blocking mechanisms.

Theorem 9: Consider a FJQN/B $\mathcal{S}$ in which server i may operate under different operating mechanisms. Then, under Assumption A1, we have:

$$\theta(BBS/IL, \boldsymbol{B}) \leq \theta(BAS/IL/SU, \boldsymbol{B}) \leq \theta(BAS/IL/IU, \boldsymbol{B}) \leq \theta(BBS/IL, \boldsymbol{B}^{i+}),$$

and

$$\theta(BBS/SL, \boldsymbol{B}) \leq \theta(BAS/SL/SU, \boldsymbol{B}) \leq \theta(BAS/SL/IU, \boldsymbol{B}) \leq \theta(BBS/SL, \boldsymbol{B}^{i+}).$$

Proof. The proof can be found in [15].

The above result implies that the throughput of a FJQN/B with BAS servers and buffer capacity vector $\boldsymbol{B}$ can be bounded from below and from above by the throughputs of two FJQN/Bs with BBS servers and buffer capacity vectors $\boldsymbol{B}$ and $\boldsymbol{B}+\boldsymbol{1}$, respectively. Correspondingly, the result also implies that the throughput of a FJQN/B with BBS servers and buffer capacity vector $\boldsymbol{B}$ can be bounded from below and from above by the throughputs of two FJQN/Bs with BAS servers and buffer capacity vectors $\boldsymbol{B}-\boldsymbol{1}$ and $\boldsymbol{B}$, respectively.

D. Reversibility Properties

In this section, we establish reversibility properties of FJQN/Bs.

Definition 10: Let $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M}, \boldsymbol{T})$ be a FJQN/B. The FJQN/B $\mathcal{S}^r = (V^r, E^r, \boldsymbol{B}^r, \boldsymbol{M}^r, \boldsymbol{T}^r)$ is the reverse of $\mathcal{S}$ if:

$$\begin{aligned} V^r &= V, \\ E^r &= \{(i,j) | (j,i) \in E\}, \\ B^r_{i,j} &= B_{j,i}, \qquad (j,i) \in E, \\ M^r_{i,j} &= M_{j,i}, \qquad (j,i) \in E, \\ \boldsymbol{T}^r &= \boldsymbol{T}. \end{aligned}$$

Observe that $\mathcal{S}^r$ differs from $\mathcal{S}$ in that all of the buffers are reversed. The nodes remain the same type as in the original FJQN/B.

Theorem 11: Let $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M}, \boldsymbol{T})$ be a FJQN/B in which all servers are either BAS/IL/IU or BAS/SL/SU B-servers, or U servers. Let $\mathcal{S}^r = (V^r, E^r, \boldsymbol{B}^r, \boldsymbol{M}^r, \boldsymbol{T}^r)$ be the reverse of $\mathcal{S}$. If $\mathcal{S}$ and $\mathcal{S}^r$ have the same (joint distribution of the) sequences of service times, then under Assumptions A1 and A3, we have:

$$\theta(\mathcal{S}^r) = \theta(\mathcal{S}). \tag{3}$$

Proof. It is easy to check from the transformations described in Section II-C that the equivalent B-FJQN/B of $\mathcal{S}^r$ is the reverse of the equivalent B-FJQN/B of $\mathcal{S}$. The proof then follows from Theorem 6.2 in [14]. ∎

Special cases of the above result were previously obtained by Yamazaki and Sakasegawa [34] and Muth [26] in the case of tandem queueing networks and by Liu [24] in the case of closed tandem queueing networks.

Remark: There exists an alternate definition for the reverse FJQN/B which only applies to FJQN/Bs consisting of U-servers and/or BAS type B-servers. Within the reverse network, a BAS type B-server is given a loading mechanism and an unloading mechanism corresponding respectively to the unloading mechanism and loading mechanism for that server in the original network (e.g., a BAS/IL/SU server in the original network is transformed into a BAS/SL/IU server in the reverse network). Theorem 11 can be proven for this alternate form of a reverse FJQN/B as well without any restrictions on the loading and unloading mechanisms associated with BAS type B-servers.

Next, we establish a reversibility property of FJQN/Bs with BBS/IL servers. On the other hand, there is no reversibility property for FJQN/Bs with BBS/SL servers. The proof of the reversibility property of FJQN/Bs with BBS/IL servers relies on the following lemma.

Lemma 12: Let $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M}, \boldsymbol{T})$ be a FJQN/B in which all servers are BBS/IL servers. Then, the B-FJQN/B $\mathcal{S}^b = (V, E, \boldsymbol{B}^+, \boldsymbol{M})$ is equivalent to $\mathcal{S}$.

Proof. Consider the transformation of any server $i \in V$ of $\mathcal{S}$ as defined in Section II-C. Each buffer $(j,i), j \in p(i)$, is replaced by a series of two buffers: buffer $(j, l_{i,j})$ with capacity $B_{j,i}$ and buffer $(l_{i,j}, i)$ with capacity one, separated by server $l_{i,j}$. Since $l_{i,j}$ has zero service time, this series of two buffers can be replaced by a single buffer (j,i) with capacity equal to the sum of the capacity of the two buffers, that is $B_{j,i}+1$. Thus, the transformation of a BBS-IL server can simply be obtained by increasing the capacity of each of its upstream buffers by one, and replacing it by a U-server. ∎

Theorem 13: Let $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M}, \boldsymbol{T})$ be a FJQN/B in which all servers are BBS/IL servers. Let $\mathcal{S}^r = (V^r, E^r, \boldsymbol{B}^r, \boldsymbol{M}^r, \boldsymbol{T}^r)$ be the reverse of $\mathcal{S}$. If $\mathcal{S}$ and $\mathcal{S}^r$ have the same (joint distribution of the) sequences of service times, then under Assumptions A1 and A3, we have:

$$\theta(\mathcal{S}^r) = \theta(\mathcal{S}). \tag{4}$$

Proof. The proof follows simply from Lemma 12 and Theorem 6.2 in [14]. ∎

E. Symmetry Properties

In this section, we investigate whether or not a symmetry property similar to that of B-FJQN/Bs can be established for FJQN/Bs. We first establish the following result pertaining to FJQN/Bs with BBS servers.

Theorem 14: Consider a FJQN/B $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M}, \boldsymbol{T}, \boldsymbol{L})$ in which all servers are BBS servers (either BBS/IL or BBS/SL servers). Consider the FJQN/B $\mathcal{S}^s(V, E, \boldsymbol{B}, \boldsymbol{M}^s, \boldsymbol{T}, \boldsymbol{L}^s)$ which is identical to $\mathcal{S}$ except that: $\boldsymbol{M}^s = \boldsymbol{B} - \boldsymbol{M}$ and $\boldsymbol{L}^s = \boldsymbol{1} - \boldsymbol{L}$. Then, under Assumptions

A1 and A3, we have:

$$\theta(\mathcal{S}^s) = \theta(\mathcal{S}) \quad (5)$$

Proof. The proof simply follows by applying Theorem 7.2 in [14] to the equivalent B-FJQN/Bs of $\mathcal{S}$ and $\mathcal{S}^s$. ∎

The following corollary follows from Theorems 14 and 3.

Corollary 15: Consider a FJQN/B $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M}, \boldsymbol{T})$ in which all servers are BBS servers (either BBS/IL or BBS/SL servers) with empty initial loadings. Assume that $\boldsymbol{M} \geq \boldsymbol{1}$. Let $\boldsymbol{M}^s$ be the initial marking defined as: $\boldsymbol{M}^s = \boldsymbol{B} - \boldsymbol{M} + \boldsymbol{1}$. Then, under Assumptions A1 and A3, we have

$$\theta(\mathcal{S}^s) = \theta(\mathcal{S}) \quad (6)$$

for $\mathcal{S}^s = (V, E, \boldsymbol{B}, \boldsymbol{M}^s, \boldsymbol{T})$.

On the other hand, there is no symmetry property for FJQN/Bs in which servers have BAS blocking mechanisms. Let us briefly explain why. Consider a simple case of a FJQN/B $\mathcal{S} = (V, E, \boldsymbol{B}, \boldsymbol{M})$ in which server i is, for instance, a BAS/SL/SU server. Again, assume that the initial condition of server i is given by $L_i = U_i = 0$. Let $\mathcal{S}^b$ be the B-FJQN/B equivalent to $\mathcal{S}$. Its initial marking is such that: $M^b_{l_i,i} = M^b_{i,u_i} = M^b_{l_i,u_i} = 0$. Theorem 7.2 in [14] can be applied to $\mathcal{S}^b$. Let $\boldsymbol{M}^s$ denote the symmetrical marking of $\boldsymbol{M}^b$, i.e., $\boldsymbol{M}^s = \boldsymbol{B}^b - \boldsymbol{M}^b$, and let $\mathcal{S}^s$ denote the resulting B-FJQN/B. In particular, we have: $M^s_{l_i,i} = M^s_{i,u_i} = M^s_{l_i,u_i} = 1$. Unfortunately, the $\mathcal{S}^s$ cannot be interpreted as the equivalent B-FJQN/B of a FJQN/B where server i has BAS/SL/SU operating mechanism. Actually, it is equivalent to a FJQN/B where server i operates as follows: after completion of a service activity, the jobs cannot be unloaded before new jobs have been loaded onto the server.

IV. Applications to Manufacturing Systems

The purpose of this section is to illustrate the application of the results derived in this paper to various problems encountered in the design and operation of manufacturing systems.

A. Simple Assembly Systems

Consider first a simple assembly system consisting of three machines, M_1, M_2 and M_3. Machines M_1 and M_2 produce two parts that are then assembled by machine M_3 to obtain the final product. There is a finite buffer of capacity $B_{1,3}$ (resp. $B_{2,3}$) between M_1 (resp. M_2) and M_3. Machine M_3 has an independent loading policy, i.e., the parts coming from M_1 and M_2 can be loaded on this machine independently of one another. One important parameter of this system is its production capacity, which is defined as its throughput assuming that raw parts are always available in front of machines M_1 and M_2.

This system is considered in [7]. In that paper, the authors note that there is no equivalence between this assembly system and a three-machine serial line with a BAS blocking mechanism. This is indeed true. However, using our results, we can show that there is an equivalence between this assembly system and a three-machine serial line

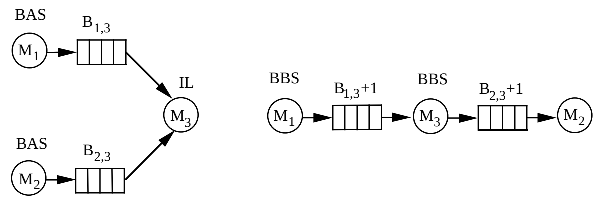

Fig. 13. A three machine assembly system and its equivalent serial line.

with BBS blocking. This line consists of machines M_1, M_3 and M_2 in series separated by two buffers. The first buffer has capacity $B_{1,3} + 1$, while the second one has capacity $B_{2,3} + 1$. The production capacity of this serial line is defined as its throughput assuming that raw parts are always available at in front of machine M_1. These two systems are illustrated in Figure 13.

Theorem 16: A three machine assembly system with BAS blocking mechanism, independent loading on the assembly machine and buffer capacities $B_{1,3}$ and $B_{2,3}$ is equivalent to a three-machine line consisting of machines M_1, M_3 and M_2 in series, separated by buffers of capacity $B_{1,3}+1$ and $B_{2,3}+1$. In particular, these two systems have the same production capacity.

Proof. Let us first transform the FJQN/Bs of the assembly system and the serial line into B-FJQN/Bs using the transformation presented in Section II-C. The B-FJQN/B equivalent to the assembly system and that equivalent to the serial line both have buffer capacities $B_{1,3} + 2$ and $B_{2,3} + 2$. Therefore, according to the duality property established in [14], these two models are equivalent and the proof follows. ∎

Note that, because of the symmetrical role of machines M_1 and M_2, the assembly system is also equivalent to a three-machine line consisting of machines M_2, M_3 and M_1 in series, separated by buffers of capacity $B_{2,3} + 1$ and $B_{1,3} + 1$, and BBS mechanism.

B. Closed-Loop Production Lines

Consider a production line consisting of L machines in series, M_1, M_2,..., M_L, separated by finite capacity buffers. In order to be processed by the different machines of the line, parts need to be fixed onto pallets. Each part is first loaded onto a pallet and is then carried by the pallet during its sojourn in the production line. When the last operation has been performed, the part is unloaded and the pallet is available to carry a new part. The (feedback) buffer that contains the available pallets also has a finite capacity so that the last machine may be blocked if this buffer is full. The production capacity of such a closed-loop production line is defined as the throughput of this system assuming that raw parts are always available at the input of the production line. An issue of high interest is to determine the optimal number of pallets that maximize this production capacity.

This closed-loop production line can be modeled as a closed tandem queueing network having L servers. Servers i models machines M_i, $i = 1, \ldots, L$. The capacity of buffer $(i, i+1)$, $B_{i,i+1}$, is equal to the maximum number of parts that can be stored between machines M_i and M_{i+1}, $i = 1, \ldots, L-1$. The capacity of buffer $(L, 1)$, $B_{L,1}$, equals to the maximum number of pallets that can be stored in the feedback buffer. Let C denote the total storage capacity of the system including the buffer space on each machine, i.e.:

$$C = \sum_{i=1}^{L-1} B_{i,i+1} + B_{L,1} + L \quad (7)$$

Assume the production line operates under the BBS blocking mechanism, i.e., a machine is not allowed to start processing a part unless a space is available in the next buffer. Also, assume that the processing times at each machine are i.i.d. random variables. The next result follows from Theorems 14, 1 and 3:

Theorem 17: The production capacity of a closed-loop production line with BBS blocking mechanism is a symmetric function of the number of pallets N, i.e., $\theta(C-N) = \theta(N)$, $\forall N = 1, ..., C-1$.

If one can further show the existence of an integer N^*, $1 \le N^* \le C-1$, such that $\theta(N)$ is a nondecreasing function of N for $N = 1, ..., N^*$, and $\theta(N)$ is a nonincreasing function of N for $N = N^*, ..., C-1$, then Theorem 17 implies that the optimal number of pallets is equal to half of the total storage capacity of the closed-loop production lines. Such an integer exists when the throughput is a concave function of N, which is the case when the processing time distributions are of PERT-type (Assumption A4). Thus, the next result follows from Theorems 17 and 5.

Theorem 18: The production capacity of a closed-loop production line with BBS blocking mechanism and PERT-type processing time distributions is maximized when the number of pallets N is such that $N = \lfloor C/2 \rfloor$ or $N = \lceil C/2 \rceil$.

Although this result is proved only under the assumption that the processing time distributions are PERT distributions (Assumption A4), we conjecture that it holds for more general processing time distributions.

The above results extend to the case of a closed-loop assembly line. A closed-loop assembly line consists of a main closed-loop in which production and assembly operations are performed. The assembly operations requires components produced by feeder lines. An example of a closed-loop assembly line is shown in Figure 14. In this example, the main closed-loop consists of machines M_1 through M_5; there are two feeder lines, the first one consisting of machines M_6 and M_7 and the second one consisting of machines M_8, M_9 and M_10. The component produced by the first (resp. second) feeder line is assembled on the main product at machine M_3 (resp. M_5).

It is easy to check that Theorems 17 and 18 still hold in

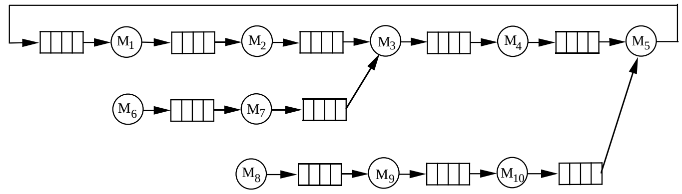

Fig. 14. An example of a closed-loop assembly line.

the case of closed-loop assembly line under the assumptions that the blocking is BBS for all the machines. Note that for the assembly machines, the loading mechanism (independent loading or simultaneous loading) does not matter. In particular, the production capacity is still maximized with a number of pallets equal to half of the total capacity of the loop.

C. Kanban Controlled Production Lines

Consider a production line operating using a kanban control mechanism; see e.g. [9], [19]. The production line consists of a series of L machines, M_1, M_2,..., M_L, decomposed into N stages. The j-th stage consists of machines $M_{l_{j-1}+1}, \ldots, M_{l_j}$, $j = 1, \ldots, N$ where $0 = l_0 < l_1 < \cdots < l_{N-1} < l_N = L$. Associated with each stage j is a number of kanbans, say $K_j, j = 1, ..., N$. Each stage j has an output buffer which contains parts that have already been processed on all of the machines M_1 up to M_{l_j}. A part located in the output buffer of stage j can be transferred to stage $j+1$ only if a kanban (production order) for stage $j+1$ is available. The kanban is then associated with the part during its stay in stage $j+1$. The kanban is released only when the part is transferred to stage $j+2$.

The production capacity of such a kanban controlled production line is defined as the throughput of this system assuming that raw parts are always available at the input of the first stage and that external demands are always present at the output of the last stage. This is a performance measure of great interest since it corresponds to the maximum (average) rate of external demands that can be satisfied [9], [19]. In other words, it provides the stability condition of the kanban controlled production line.

This kanban controlled production line can be modeled as a FJQN/B [19]. The set of servers are the L machines and the $N-1$ synchronization stations. The j-th synchronization station represents the synchronization of a part completed by stage j and a kanban of stage $j+1$. All buffers pertaining to stage j have capacity K_j. The initial condition is that the buffer upstream of the j-th synchronization station contains K_j jobs at time 0, $j = 1, ..., N$, and all other buffers are empty. We assume that the processing times at each machine are i.i.d. random variables. All synchronization stations have zero processing times. The FJQN/B model of a kanban controlled production line consisting of $L = 6$ machines, $N = 3$ stages, each stage consisting of 2 machines, is shown in Figure 15.

The following property follows from the results of Section III-B.

Theorem 19: The production capacity of a kanban con-

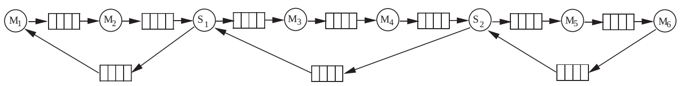

Fig. 15. The FJQN/B model of a kanban controlled production line.

trolled production line is a monotone and (under Assumption A4) concave function of the number of kanbans of the different stages.

Consider now the following reverse system. It consists of the series of L machines in reverse order, M_L, M_{L-1},..., M_1, again decomposed into N stages. Stage 1 consists of machines M_L up to $M_{l_{N-1}+1}$, stage 2 consists of machines $M_{l_{N-1}}$ up to $M_{l_{N-2}+1}$,..., stage N consists of machines M_{l_1} up to M_1. The number of kanbans associated with stage j is $K_{N-j+1}, j = 1, ..., N$. In words, the j-th stage of the reverse system is identical to the K_{N-j+1}-th stage of the original system.

Now, it is easy to check that the FJQN/B model of the reverse kanban controlled production line is the reverse of the FJQN/B model of the original kanban controlled production line. The following property then follows from the results derived in Section III-D.

Theorem 20: The production capacity of a kanban controlled production line is the same as that of the reverse kanban controlled production line.

The above results (monotonicity and reversibility) were derived in [10], [11], [21], [33] in the special case where each stage consists of a single machine. Our results show that they hold regardless of the number of machines in each stage. Moreover, it is easy to check that they also hold if the buffers in between consecutive machines within a stage have finite capacity.

Variations around this kanban control mechanism have been described in the literature see e.g. [9], [11]. In each case, it is possible to model the resulting system as a FJQN/B and apply the results derived in this paper. For instance, the general blocking mechanism introduced in [11] gives rise to a FJQN/B similar to the one described above with one machine per stage, except that the input and output buffers of each machine have finite capacity.

V. Performance Evaluation and Optimization

The purpose of this section is to illustrate the usefulness of the results derived in this paper to the performance evaluation and optimization of FJQN/B models. Before doing so, let us briefly describe some general results on the performance evaluation of FJQN/Bs for some special classes of FJQN/Bs. As is usually the case with discrete event systems, most analytical techniques for performance evaluation rely on Markov chain analysis. In order to use Markov chain analysis, service time distributions must be represented by exponential distributions or mixtures of exponential distributions such as phase-type (PH) distributions [27]. Unfortunately, FJQN/Bs do not have close-form solutions, even under the assumption of exponentially distributed service times. For small or moderate size systems, the performance parameters can be derived by numerically solving the underlying Markov chain. For large systems (whose Markov chains have too large a state space to be solved numerically), the only tractable approach is to use approximation methods. Most approximation methods are based on decomposition and considerable work has been devoted to developing such methods for special classes of FJQN/Bs, in particular tandem queueing networks and closed tandem queueing networks; see [12], [13] and the references therein. For a special class of FJQN/Bs where networks have a single source with exponentially distributed service times, the mean waiting times can be obtained by a Taylor series expansion [5], and a closed-form expression exists when the service times of other servers are deterministic.

Let us illustrate the usefulness of the results presented in this paper for addressing performance evaluation. For that purpose, consider again the simple assembly system presented in Section IV-A. In [7], Baker et al. use an exact solution of the underlying Markov chain to calculate performance parameters. However, they only consider the simple case with no intermediate buffers and exponentially distributed service times at the three machines. For systems with intermediate buffers, this may no longer be possible, in particular if more general PH distributions are used to represent service time distributions. Thus, a tractable analysis needs to rely on an approximation method. Now, a very accurate decomposition method for tandem queueing networks with finite buffers and blocking before service was proposed in [8]. The equivalence result of Theorem 16 allows us to use this method to analyze the equivalent serial line and then transpose the results obtained to the assembly system.

The results presented in this paper are also useful for optimization purposes. When addressing design or operations issues in manufacturing systems, the problem can often be stated as: find the configuration of the system that achieves a given performance, for instance a given production capacity, at a minimal cost. The approach used to solve optimization problems is to use a "performance evaluation building block" coupled with a search procedure. The performance evaluation building block addresses the following question: for a given configuration, what is the production capacity of the system. The search procedure starts from an initial configuration and step by step moves towards the optimal configuration. At each step, the performance evaluation building block is used to determine in what direction to move, for instance by calculating gradient information.

Consider first the closed-loop production (or assembly) line considered in Section IV-B. A typical design problem would be the following: determine the capacity of the buffers and the number of pallets that achieve a given production capacity at a minimal cost. If the costs of pallets are insignificant (compared with buffer costs), then the results of Section IV-B simply state that for any given set of buffer capacities, the number of pallets that should be chosen is equal to half of the total capacity of the main loop, i.e., $N = \lfloor C/2 \rfloor$. This drastically reduces the complexity of the optimization search procedure. On the other hand, if the cost of the pallets has to be taken into account, the results of Section IV-B are still useful because they imply that the optimal number of pallets falls in the range $1, \ldots, \lfloor C/2 \rfloor$.

Consider now an optimization problem pertaining to the kanban controlled production line considered in Section IV-C. A typical optimization problem would be to determine the number of kanbans that achieve a given production capacity at a minimal cost, the cost being for instance a linear function of the numbers of kanbans at each stage. For that purpose, we could use the analytical method developed in [19] as the performance evaluation building block. This building block would address the issue of calculating the production capacity for given values of the number of kanbans. The building block should then be embedded into a search procedure. Again the results obtained in Section IV-C can be used to would be useful an efficient search procedure. Indeed, according to Theorem 19, the production capacity of a kanban controlled production line is an increasing and (under Assumption A4) concave function of the number of kanbans of the different stages. Let $\theta(K_1, K_2, \ldots, K_N)$ denote the capacity of the production line with $K_1, K_2, \ldots, K_N$ kanbans in stages $1, 2, \ldots, N$. Then, for any fixed production capacity γ that the production line should achieve, the set Φ of feasible kanbans

$$\Phi = \{(K_1, K_2, \ldots, K_N) \mid \theta(K_1, K_2, \ldots, K_N) \geq \gamma\}$$

is a convex subset of $\mathbb{N}^{\mathrm{N}}$. Thus, a simple gradient-based search algorithm can solve the problem of minimizing any linear function of $K_1, K_2, \ldots, K_N$ within Φ.

VI. Concluding Remarks

For simplicity, we have restricted our attention to a specific class of queueing networks with finite capacity buffers and fork/join mechanisms. Although this class is likely to include many models encountered in practical applications, it may be of interest to deal with models that do not belong to this class. For each such model, the approach presented in this paper can be used to derive its properties. Extensions of the class of FJQN/Bs considered in this paper include: incorporating non-zero loading and/or unloading times; dealing with more general loading and unloading policies; combining different blocking mechanisms with respect to different buffers of a given server; handling assembly operations that can be decomposed into several tasks that have to be performed according to a given precedence graph. A detailed discussion of these extensions can be found in [15].

References

[1] I. Adan, J. Van Der Wal, "Monotonicity of the Throughput in Single Server Production and Assembly Networks with Respect to the Buffer Sizes," in *Queueing Networks with Blocking*, H.G. Perros, T. Altiok (Ed.s), North-Holland, 325-344, 1989.

[2] M. H. Ammar, S. B. Gershwin, "Equivalence Relations in Queuing Models of Fork/Join Queueing Networks with Blocking", *Performance Evaluation* **10**, pp. 233-245, 1989.

[3] V. Anantharam, P. Tsoucas, "Stochastic Concavity of Throughput in Series of Queues with Finite Buffers," *Adv. Appl. Prob.*, **22**, 761-763, 1990.

[4] F. Baccelli, Z. Liu, "Comparison Properties of Stochastic Decision Free Petri Nets", *IEEE Trans. on Automatic Control*, **37**, 1905-1920, 1992.

[5] F. Baccelli, V. Schmidt, "Taylor Expansions for Poisson Driven (max, +)-Linear Systems", INRIA RR No. 2494, 1995.

[6] K. R. Baker, S. G. Powell, D. F. Pyke, "Buffered and Unbuffered Assembly Systems with Variable Processing Times", *Journal of Manufacturing and Operations Management*, **3**, 200-223, 1990.

[7] K. R. Baker, S. G. Powell, D. F. Pyke, "Optimal Allocation of Work in Assembly Systems", *Management Science*, **39**, 1, 101-106, 1993.

[8] A. Bouhchouch, Y. Frein, Y. Dallery, "A Decomposition Method for the Analysis of Tandem Queueing Networks with Blocking Before Service", in *Queueing Networks with Finite Capacity*, R.O. Onvural and I.F. Akyildiz (Eds.), Elsevier Science Publishers, 1993.

[9] J. A. Buzacott, J. G. Shanthikumar, "Stochastic Models of Manufacturing Systems", Prentice Hall, Englewood Cliffs, N. J., 1993.

[10] D. W. Cheng, "Line Reversibility of Tandem Queues with General Blocking", *Management Science*, **41**, 5, 864-873, 1995.

[11] D. W. Cheng, D. D. Yao, "Tandem Queues with General Blocking: A Unified Model and Comparison Results," *Journal of Discrete Event Dynamical Systems (DEDS)*, **2**, 207-234, 1993.

[12] Y. Dallery, Y. Frein, "On Decomposition Methods for Tandem Queueing Networks with Blocking", *Operations Research*, **41**, 2, 386-399, 1993.

[13] Y. Dallery, S. B. Gershwin, "Manufacturing Flow Line Systems: a Review of Models and Analytical Results," *Queueing Systems Theory and Applications (QUESTA)*, **12**, 3-94, 1992.

[14] Y. Dallery, Z. Liu, D. Towsley, "Equivalence, Reversibility, Symmetry and Concavity in Fork/Join Queueing Networks with Blocking," *Journal of the ACM*, **41**, 5, 903-942, 1994.

[15] Y. Dallery, Z. Liu, D. Towsley, "Properties of Fork/Join Queueing Networks with Blocking under Various Operating Mechanisms," Tech. Report COINS, 1992.

[16] Y. Dallery, D. Towsley, "Symmetry Property of the Throughput in Closed Tandem Queueing Networks with Finite Buffers", *Operations Research Letters* **10**, 9, 541-547, 1991.

[17] M. Di Mascolo, Y. Dallery, "Performance Evaluation of Kanban Controlled Assembly Systems", *Symposium on Discrete Events and Manufacturing Systems of the Multiconference IMACS-IEEE/SMC CESA'96*, Lille, France, July 1996.

[18] M. Di Mascolo, R. David, Y. Dallery, "Modeling and Analysis of Assembly Systems with Unreliable Machines and Finite Buffers", *IIE Transactions* **23**, 4, 315-330, 1991.

[19] M. Di Mascolo, Y. Frein, Y. Dallery, "An Analytical Method for Performance Evaluation of Kanban Controlled Production Systems", *Operations Research*, Special Issue on New Directions in Operations Management, **44**, 1, 50-64, 1996.

[20] S. B. Gershwin, "Assembly/Disassembly Systems: an Efficient Decomposition Algorithm for Tree-Structured Network", *IIE Transactions* **23**, 4, 1991.

[21] P. Glasserman, D. D. Yao, "Structured Buffer-Allocation Problems in Production Lines," *Discrete Event Dynamic Systems*, to appear.

[22] P. Glasserman, D. D. Yao, "Monotonicity in Generalized Semi-Markov Processes," *Math. of Operations Research* **17**, 2, May 1992.

[23] P. Glasserman, D. D. Yao, "Generalized Semi-Markov Processes: Antimatroid Structure and Second-Order Properties," *Math. of Operations Research* **17**, 2, May 1992.

[24] X.-G. Liu, "Toward Modeling Assembly Systems: Applications of Queueing Networks with Blocking," Ph. D. Thesis, Department of Management Sciences, University of Waterloo, Waterloo, 1990.

[25] L.E. Meester, J.G. Shanthikumar, "Concavity of the throughput of tandem queueing systems with finite buffer storage space," *Adv. Appl. Prob.*, **22**, 764-767, 1990.

[26] E. G. Muth. "The reversibility property of production lines", *Management Science*, **25**, 2, 152-158, 1979.

[27] M.F. Neuts, "Matrix-geometric solutions in stochastic models: an algorithmic approach", The John Hopkins Univ. Press, Baltimore. 1981.

[28] R. O. Onvural, H. G. Perros. "On Equivalencies of Blocking Mechanism in Queueing Networks with Blocking", *Operations Research Letters*, **5-6**, 293-298, 1986.

[29] H. G. Perros, "Approximation Algorithms for Open Queueing Networks with Blocking", *Stochastic Analysis of Computer and Communication Systems*, (ed. H. Takagi), North-Holland, Amsterdam, Netherlands, 1989.

[30] B. Philippe, Y. Saad, W. J. Stewart, "Numerical methods in Markov chain modelling, *Operations Research*, **40**, 6, 1156-1179, 1992.

[31] R. Rajan, R. Agrawal, "Cyclic Networks with General Blocking and Starvation," to appear in *Queueing Systems Theory and Applications (QUESTA)*.

[32] J.G. Shanthikumar, D.D. Yao, "Monotonicity and Concavity Properties in Cyclic Queueing Networks with Finite Buffers," in *Queueing Networks with Blocking*, H.G. Perros, T. Altiok (Ed.s), North-Holland, 325-344, 1989.

[33] S. Tayur, "Structural Results and a Heuristic for Kanban Controlled Serial Lines", Tech. Report No. 934, Cornell University, December 1991.

[34] G. Yamazaki, H. Sakasegawa, "Properties of Duality in Tandem Queueing Systems", *Annals of the Institute of Statistical Mathematics*, **27**, 201-212, 1975.

[35] G. Yamazaki, T. Kawashiha, H. Sakasegawa, "Reversibility of Tandem Blocking Queueing Systems", *Management Science*, **31**, 1, 78-83, 1985.

Yves Dallery received his Ph.D. and the degree of *Habilitation Diriger des Recherches* from the *Institut National Polytechnique de Grenoble* (INPG) in 1984 and 1989, respectively. He is currently *Chargé de Recherche* at the *Centre National de la Recherche Scientifique* (CNRS) and the Associate Director of Laboratoire MASI. In 1984-1985, he was a post-doctoral fellow at Harvard University. In 1991-1992, he was a visiting scientist at M.I.T. and in 1992-1993 he was an Associate Professor of Manufacturing Engineering at Boston University. His research interests are in stochastic models of discrete event systems, exact and approximation techniques for queueing network analysis, modeling, performance evaluation and control of manufacturing systems, and operations management.

Zhen Liu received the Ph.D degree and the *Habilitation à diriger des recherches*, both in Computer Science, from the University of Orsay (Paris XI), France, in 1989 and 1993, respectively. He is a research scientist at the national research center on information and automation (INRIA), France. His current research interests are in performance analysis and control of discrete event systems, queueing systems and Petri nets, with applications to parallel and distributed systems, communication networks and manufacturing systems.

Don Towsleyreceived the B.A. degree in physics and the Ph.D. degree in Computer Science from University of Texas in 1971 and 1975 respectively. From 1976 to 1985 he was a member of the faculty of the Department of Electrical and Computer Engineering at the University of Massachusetts, Amherst. He is currently a Professor of Computer Science at the University of Massachusetts. During 1982-1983, he was a Visiting Scientist at the IBM T.J. Watson Research Center, Yorktown Heights, NY. and during the year 1989-1990, he was a Visiting Professor at the Laboratoire MASI, Paris, France. His research interests include high speed networks and multimedia systems.

He currently serves on the Editorial board of *Performance Evaluation* and has previously served on several editorial boards including those of the *IEEE Transactions on Communications* and *IEEE/ACM Transactions on Networking.* He was a Program Co-chair of the joint ACM SIGMETRICS and PERFORMANCE '92 conference. He is a member of ACM, ORSA and the IFIP Working Groups 6.3 and 7.3.

He has received two best conference paper awards from ACM SIGMETRICS and has been elected Fellow of both the ACM and IEEE.