An Analytical Method for the Performance Evaluation of Echelon Kanban Control Systems

Stelios Koukoumialos and George Liberopoulos

University of Thessaly, Department of Mechanical and Industrial Engineering, Volos, Greece, Tel: +30 24210 74056, E-mails: skoukoum@mie.uth.gr and glib@mie.uth.gr

January 2004

Abstract

We develop a general purpose analytical approximation method for the performance evaluation of a multi-stage, serial, echelon kanban control system. The basic principle of the method is to decompose the original system into a set of nested subsystems, each subsystem being associated with a particular echelon of stages. Each subsystem is analyzed in isolation using a product-form approximation technique. An iterative procedure is used to determine the unknown parameters of each subsystem. Numerical results show that the method is fairly accurate.

1 Introduction

In this paper, we develop an analytical approximation method for the performance evaluation of an echelon kanban control system, used for the coordination of production in a multi-stage, serial, production/inventory system. We then test the behavior of this method on several numerical examples. The term "echelon kanban" was introduced in [19]. The basic principle of the operation of the echelon kanban control system is very simple: When a part leaves the last stage of the system to satisfy a customer demand, a new part is demanded and authorized to be released into each stage. It is worth noting that the echelon kanban control system is equivalent to the integral control system described in [8]. The echelon kanban control system differs from the conventional kanban control system, which is also referred to as installation kanban control system or policy in [19], in that in the conventional kanban control system, a new part is demanded and authorized to be released into a stage when a part leaves this particular stage and not when a part leaves the last stage, as is the case with the echelon kanban control system. This implies that in the conventional kanban control system, the placement of a demand and an authorization for the production of a new part into a stage

is based on local information from this stage, whereas in the echelon kanban control system, it is based on global information from the last stage. This constitutes a potential advantage of the echelon kanban control system over the conventional kanban control system. Moreover, the echelon kanban control system, just like the conventional kanban control system, depends on only one parameter per stage, the number of echelon kanbans, as we will see later on, and is therefore simpler to optimize and implement than more complicated kanban-type control systems that depend of two parameters per stage, such as the generalized kanban control system [7] and the extended kanban control system [11]. These two apparent advantages of the echelon kanban control system motivated our effort to develop an approximation method for its performance evaluation.

Kanban-type production/inventory systems are often modeled as queueing networks in the literature. Consequently, most of the techniques that have been developed for the analysis of kanban-type production/inventory systems are based on methods for the performance evaluation of queueing networks. Exact analytical solutions exist for a class of queueing networks known as separable, in which the steady-state joint probabilities have a product-form solution. Jackson [18] was the first to show that the steady-state joint probability of an open queueing network with Poisson arrivals, exponential service times, probabilistic routing, and first-come-first-served (FCFS) service disciplines has a product-form solution, where each station of the network can be analyzed in isolation as an M/M/1 queue. For closed queueing networks of the Jackson type, Gordon and Newell [17] showed that an analytical, product-form solution also exists. The performance parameters of such networks can be obtained using efficient algorithms, such as the mean value analysis (MVA) algorithm [22] and the convolution algorithm [9]. The BCMP theorem [1] summarizes extensions of product-form networks that incorporate alternative service disciplines and several classes of customers.

Since the class of queueing networks for which an exact solution is known (separable networks) is too restrictive for modeling and analyzing real systems, much work has been devoted to the development of approximation methods for the analysis of non-separable networks. Whitt [26] presented an approximation method for the analysis of a general open queueing network that is based on decomposing the network into a set of GI/GI/1 queues and analyzing each queue in isolation. In the case of closed queueing networks, the approximation methods are for the most part based on two approaches. The first approach relies on heuristic

extensions of the MVA algorithm (e.g. [23]). The second approach relies on approximating the performance of the original network by that of an equivalent product-form network. Daryanto et al. [13] developed a method that is based on the second approach for a closed-loop, two-indenture, repairable-item system. Interestingly, their system is equivalent to an echelon kanban control system with a finite population of external jobs. Their method aggregates several states of the underlying continuous-time Markov chain and adjusts some service rates using Norton's Theorem for closed queueing networks to obtain a product-form solution. Among the different methods that rely on the second approach, Marie's method [20] has attracted considerable attention. Extensions and comparative studies of Marie's method have been proposed for a variety of queueing networks [2], [3], [4], [5], and [10]. Di Mascolo, Frein and Dallery [14], [16] developed approximation methods based on Marie's method for the performance evaluation of the conventional kanban control system and the generalized kanban control system.

The approximation method that we develop in this paper for the performance evaluation of the echelon kanban control system relies on Marie's method. To develop our method, we first model the system as an open queueing network with synchronization stations. Each stage has associated with it a particular echelon of stages consisting of the stage itself and all its downstream stages. By exchanging the roles of jobs (parts) and resources (echelon kanbans) in the open network we obtain an equivalent, multi-class, nested, closed queueing network, in which the population of each class is equal to the job capacity or number of echelon kanbans of the echelon of stages associated with a particular stage. We then decompose the closed network into a set of nested subsystems, each subsystem being associated with a particular class. This means that we have as many subsystems as the number of the stages. Each subsystem is analyzed in isolation using Marie's method. Each subsystem interacts with its neighboring subsystems in that it includes its downstream subsystem in the form of a single-server station with load-dependent, exponential service rates, and it receives external arrivals from its upstream subsystem. A fixed-point, iterative procedure is used to determine the unknown parameters of each subsystem by taking into account the interactions between neighboring subsystems.

The rest of this paper is organized as follows. In Section 2, we describe the exact operation of the echelon kanban control system by means of a simple example. In Section 3 we present the queueing network model of the echelon kanban control system and the

performance measures of the system that we are interested in evaluating. In Section 4, we describe the decomposition of the original system into subsystems. In Section 5, we present the analysis in isolation of each subsystem, and in Section 6 we develop the analysis of the entire system. In Section 7, we present numerical results on the effects and optimization of the parameters. Finally, in Section 8, we draw conclusions. The analysis of the synchronization stations that appear in the queueing network models of each subsystem is presented in Appendices A and B, and a table of the notation used in the paper is given in Appendix C.

2 Echelon Kanban Control System

In this section, we give a precise description of the operation of the echelon kanban control system by means of a simple example. In this example, we consider a production system that consists of $M = 9$ machines in series, labeled M1 to M9, produces a single part type, and does not involve any batching, reworking or scrapping of parts. Each machine has a random processing time. All parts visit successively machines M1 to M9. The production system is decomposed into $N = 3$ stages. Each stage is a production/inventory system consisting of a manufacturing process and an output buffer. The output buffer stores the finished parts of the stage. The manufacturing process consists of a subset of machines of the original manufacturing system and contains parts that are in service or waiting for service on the machines. These parts represent the work in process (WIP) of the stage and are used to supply the output buffer. In the example, each stage consists of three machines. More specifically, the sets of machines {M1, M2, M3}, {M4, M5, M6} and {M7, M8, M9} belong to stages 1, 2 and 3, respectively. The decomposition of the production system into three stages is illustrated in Figure 1**.**

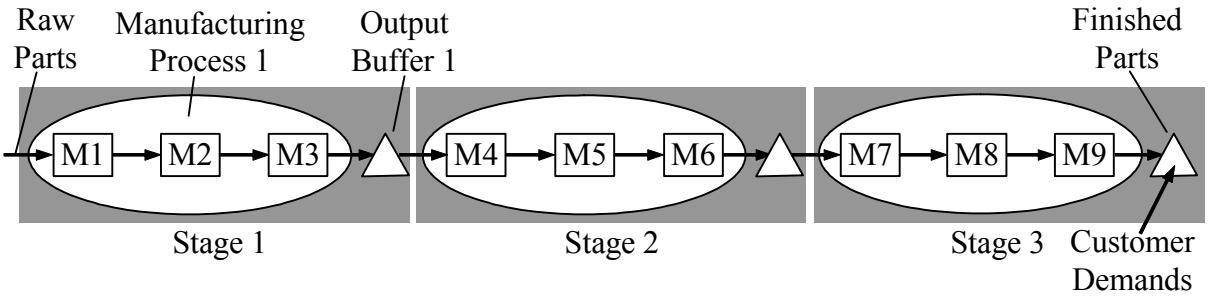

Figure 1: A serial production system decomposed into three stages in series.

Each stage has associated with it a number of echelon kanbans that are used to demand and authorize the release of parts into this stage. An echelon kanban of a particular stage traces a closed path through this stage and all its downstream stages. The number of echelon kanbans of stage i is fixed and equal to K_i. There must be at least one echelon kanban of stage i available in order to release a new part into this stage. If such a kanban is available, the kanban is attached onto the part and follows it through the system until the output buffer of the last stage. Since an echelon kanban of stage i is attached to every part that is in stages i to N, the number of parts in stages i to N is limited by K_i.

Parts that are in the output buffer of stage N are the finished parts of the production system. These parts are used to satisfy customer demands. When a customer demand arrives to the system, a demand for the delivery of a finished part from the output buffer of the last stage to the customer is placed. If there are no finished parts in the output buffer of the last stage, the demand cannot be immediately satisfied and is backordered until a finished part becomes available. If there is at least one finished part in the output buffer of the last stage, this part is delivered to the customer after releasing the kanbans of all the stages (1, 2, and 3, in the example) that were attached to it, in the output buffer of the last stage, hence the demand is immediately satisfied. The released kanbans are transferred upstream to their corresponding stages. The kanban of stage i carries with it a demand for the production of a new stage-i finished part and an authorization to release a finished part from the output buffer of stage i-1 into stage i. When a finished part of stage i-1 is transferred to stage i, the stage-i kanban is attached to it on top of the kanbans of stages 1 to i-1, which have already been attached to the part at previous stages. With this in mind, we can just as well assume that

$$K_i \geq K_{i+1},\ i = 1,\ldots, N-1. \tag{1}$$

3 Queueing Network Model of the echelon Kanban Control System

In order to develop the approximation method for the performance evaluation of the echelon kanban control system, we first model the system as an open queueing network with synchronization stations. Figure 2 shows the queueing network model of the echelon kanban control system with three stages in series considered in Section 2. The manufacturing process of each stage is modeled as a subnetwork in which the machines of the manufacturing process are represented by single-server stations. The subnetwork associated with the manufacturing process of stage i is denoted by L_i, and the single-server stations representing machines

M1,…, M9 are denoted by S_1,…, S_9, respectively. The number of stations of subnetwork L_i is denoted by m_i. In the example, $m_i = 3$, $i = 1, 2, 3$. The echelon kanban control mechanism is modeled via three synchronization stations, denoted by J_i, at the output of each stage i, $i = 1, 2, 3$.

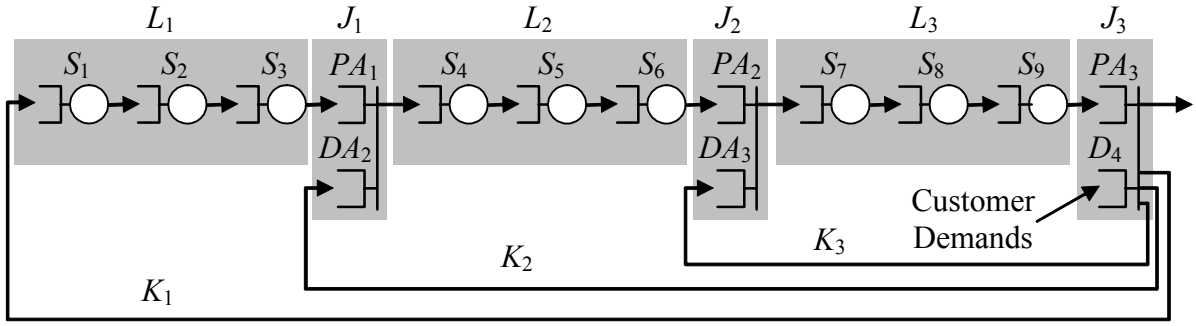

Figure 2: Queueing network model of the echelon kanban control system of Figure 1.

A synchronization station is a tool that is often used to model assembly operations in queueing networks. It can be thought of as a server with instant service times. This server is fed by two or more queues (in our case by two). When there is at least one customer in each of the queues that feed the server, these customers move instantly through and out of the server. This implies that, at any time, at least one of the queues that feed the server is empty. Customers that enter the server immediately exit the server after possibly having been split into more or merged into fewer customers. In our case, the queues in each synchronization station contain either parts or demands combined with kanbans.

To illustrate the operation of the synchronization stations, let us first focus on any synchronization station J_i, except that of the last stage. This synchronization station represents the synchronization between a stage-i finished part and a stage-(i+1) free kanban. Let PA_i and DA_{i+1} denote the two queues of J_i. PA_i represents the output buffer of stage i and contains stage-i finished parts, each of which has attached to it a kanban from each stage from 1 to i. DA_{i+1} contains demands for the production of new stage-(i+1) parts, each of which has attached to it a stage-(i+1) kanban. The synchronization station operates as follows. As soon as there is one entity in each queue PA_i and DA_{i+1}, the stage-i finished part engages the stage-(i+1) kanban without releasing the kanbans from stages 1 to i that were already attached to it, and joins the first station of stage i+1. Note that at stage 1, as soon as a stage-1 kanban is

available, a new part is immediately released into stage 1 since there are always raw parts at the input of the system.

Let us now consider the last synchronization station J_N (J_3 in the example). J_N synchronizes queues PA_N, and D_{N+1}. PA_N represents the output buffer of stage N and contains stage-N finished parts, each of which has attached to it a kanban from each stage from 1 to N. D_{N+1} contains customer demands. When a customer demand arrives to the system, it joins D_{N+1}, thereby demanding the release of a finished part from PA_N to the customer. If there is a finished part in queue PA_N, it is released to the customer and the demand is satisfied. In this case, the finished part in PA_N releases the kanbans that were attached to it, and these kanbans are transferred upstream to queues DA_i (i = 1,…, N). The kanban of stage i carries along with it a demand for the production of a new stage-i (i = 1,…, N) finished part and an authorization for the release of a finished part from queue PA_{i-1} into stage i. If there are no finished parts in queue PA_N, the customer demand remains on hold in D_{N+1} as a backordered demand.

An important special case of the echelon kanban control system in the case where there are always customer demands for finished parts. This case is known as the *saturated* echelon kanban control system. Its importance lies in the fact that its throughput determines the maximum capacity of the system. In the saturated system, when there are finished parts at stage N, they are immediately consumed and an equal number of parts enter the system. As far as the queueing network corresponding to this model is concerned, the synchronization station J_N can be removed since queue D_{N+1} is never empty and can therefore be ignored. In the saturated echelon kanban control system, when the processing of a part is completed at stage N, this part is immediately consumed after releasing the kanbans of stages 1,…, N that were attached to it and sending them back to queues DA_i (i = 1,…, N).

It is worth noting that the echelon kanban control system contains the make-to-stock CONWIP system [23] as a special case. In the make-to-stock CONWIP system, as soon as a finished part leaves the production system to be delivered to a customer, a new part enters the system to begin its processing. An echelon kanban control system with $K_1 \leq K_i$, $i \neq 1$ behaves exactly like the make-to-stock CONWIP system.

The dynamic behavior of the echelon kanban control system depends on the manufacturing processes, the arrival process of customer external demands, and the number of echelon kanbans of each stage. The performance measures that are of particular interest are

the average work in process (WIP) and the average number of finished parts in each stage, the average number of backordered (not immediately satisfied) demands, and the average waiting time and percentage of backordered demands. In the case of the saturated echelon kanban control system, the main performance measure is its production rate, P_r, i.e. the average number of finished parts leaving the output buffer of stage N per unit of time. P_r represents the maximum rate at which customer demands can be satisfied. With this in mind, the average arrival rate of external customer demands in the unsaturated system, say λ_D, must be strictly less than P_r in order for the system to meet all the demands in the long run. In other words, the stability condition for the unsaturated system is

$$\lambda_D < P_r. \tag{2}$$

4 Decomposition of the Echelon Kanban Control System

To evaluate the performance of the multi-stage, serial, echelon kanban control system, we decompose the system into many nested, single-stage subsystems and analyze each system in isolation. The susbsystems are nested in each other in such a way that each subsystem includes its downstream subsystem in the form of a single-server station and receives external arrivals from its upstream subsystem. The first subsystem mimics the original system. To analyze each subsystem, we view it as a closed queueing network and we approximate each station of this network by an exponential-service station with load-dependent service rates. The resulting network is a product-form network. A fixed-point iterative procedure is then used to determine the unknown parameters of each subsystem by taking into account the interactions between neighboring subsystems. A detailed description of the decomposition follows.

Consider the queueing network model of an echelon kanban control system consisting of N stages in series as described in Section 3 (See Figure 2 for $N = 3$). Let us denote the queueing network of the system by R. Our goal is to analyze R by decomposing it into a set of N nested subsystems, R^i, $i = 1,\ldots, N$. This is done as follows (See Figure 3 for $N = 3$).

Subsystem R^N is an open queueing network with restricted capacity consisting of 1) an upstream synchronization station, denoted by I^N, representing J_{N-1} in the original system, 2) the subnetwork of stations, L_N, in the original system, and 3) a downstream synchronization station, denoted by O^N, representing J_N in the original system. Each subsystem R^i, $i = 2,\ldots,$

$N-1$, is an open queueing network with restricted capacity consisting of 1) an upstream synchronization station, denoted by I^i, representing J_{i-1} in the original system, 2) the subnetwork of stations, L_i, in the original system, and 3) a downstream single-server pseudo-station, denoted by $\hat{S}_i$, representing the part of the system downstream of L_i in the original system. Finally, subsystem R^1 is a closed queueing network consisting of 1) the subnetwork of stations, L_1, in the original system, and 2) a downstream single-server pseudo-station, denoted by $\hat{S}_1$, representing the part of the system downstream of L_1 in the original system. Note that pseudo-station $\hat{S}_i$ in subsystem R^i, $i = 1,\ldots, N-1$, is an aggregate representation of subsystem R^{i+1}.

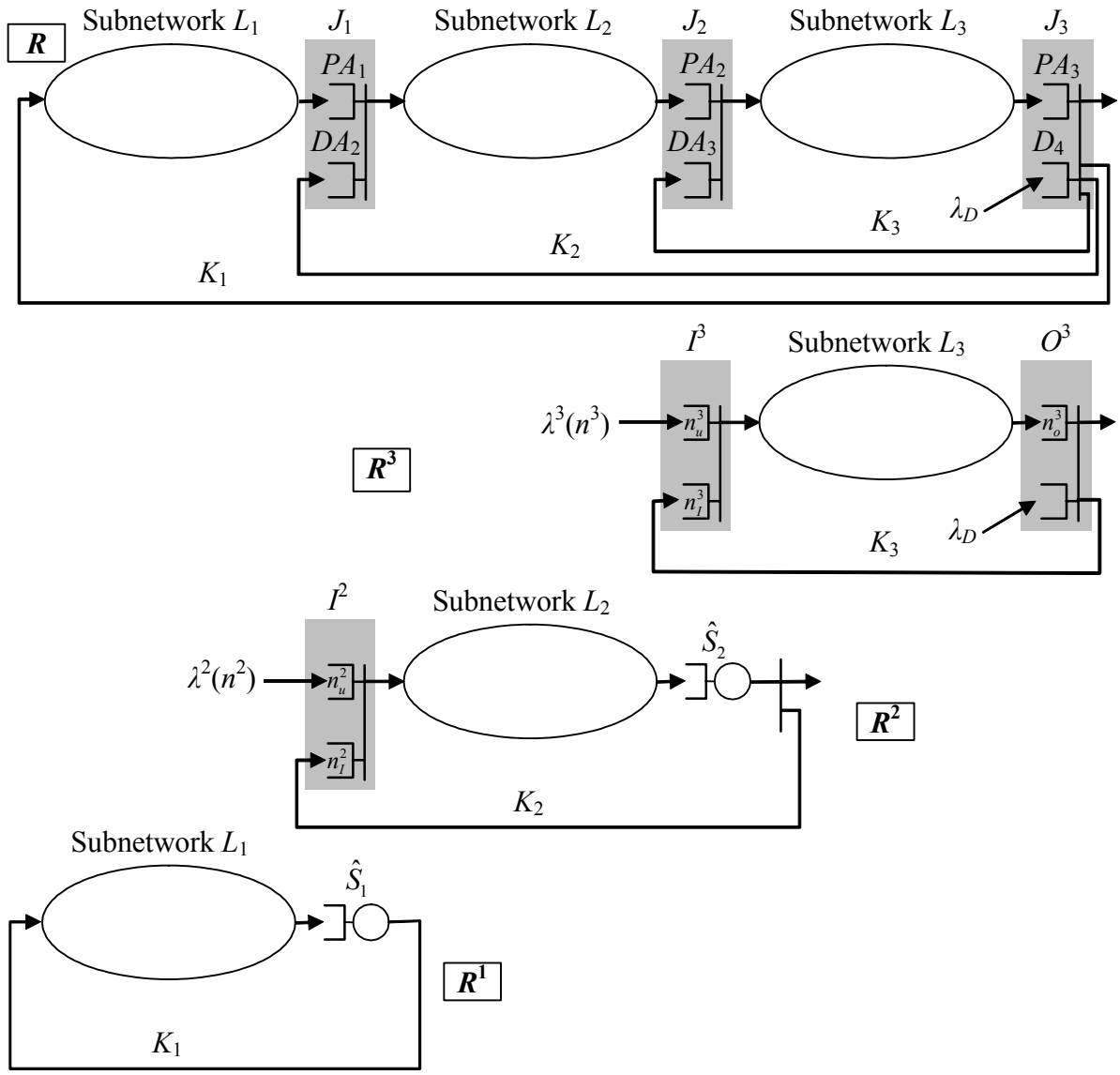

Figure 3: Illustration of the decomposition of a 3-stage echelon kanban control system.

The number of echelon kanbans of subsystem R^i is K_i. Subsystem R^N is synchronized with two external arrival processes, one at synchronization station I^N receiving parts that arrive from subnetwork $L_{N\text{-}1}$, and the other at synchronization station O^N receiving customer demands. Subsystem R^i, $i = 2,\ldots, N-1$, is synchronized with only one external arrival process at synchronization station I^i receiving parts that arrive from subnetwork $L_{i\text{-}1}$. Subsystem R^1 is a closed network; therefore it is not synchronized with any external arrival processes. As can be seen from Figure 3, each synchronization station J_i of the original network R, linking stage i to stage i+1, is represented only once in the decomposition.

To completely characterize each subsystem R^i, $i = 2,\ldots, N-1$, we assume that each of the external arrival processes to R^i is a state-dependent, continuous-time Markov process. Let $\lambda^i(n^i)$ denote the state-dependent arrival rate of stage-i raw parts at the upstream synchronization station I^i of subsystem R^i, where n^i is the state of subsystem R^i and is defined as the number of parts in this subsystem. Let Q_u^i and Q_I^i be the two queues of synchronization station I^i, containing n_u^i and n_I^i customers, respectively, where n_u^i is the number of finished parts of stage i-1 waiting to enter subnetwork L_i, and n_I^i is the number of free stage-i kanbans waiting to authorize the release of stage-(i-1) finished parts into subnetwork L_i. Then, it is clear that the only possible states of the synchronization station are the states $(n_I^i, 0)$, for $n_I^i = 0,\ldots, K_i$, and $(0, n_u^i)$, for $n_u^i = 0,\ldots, K_{i\text{-}1} - K_i$; therefore, the state n^i of subsystem R^i can be simply obtained from n_u^i and n_I^i using the following relation:

$$n^i = \begin{cases} K_i - n_I^i & \text{if } n_I^i \neq 0, \\ K_i + n_u^i & \text{if } n_I^i = 0. \end{cases} \tag{3}$$

The above relation implies that $0 \le n^i \le K_{i\text{-}1}$. Also, since the number of raw parts at the input of stage i cannot be more than the number of stage-(i-1) kanbans, $\lambda^i\left(K_{i-1}\right) = 0$. In subsystem R^N, besides the arrival rate of stage-N raw parts at I^N, $\lambda^N(n^N)$, there is also the external arrival rate of customer demands at O^N, λ_D. Subsystem R^1, as was mentioned above, is a closed network and therefore has no external arrival processes to define.

To obtain the performance of the original network R, the following two problems must be addressed: 1) How to analyze each subsystem R^i, $i = 1,\ldots, N$, assuming that the external arrival rates are known (except in the case of the first subsystem R^1, where there are no

external arrivals), and 2) how to determine the unknown external arrival rates. These two problems are addressed in Sections 5 and 6, respectively. Once these two problems have been solved, the performance of each stage of the original network R can be obtained from the performances of subsystems R^i, $i = 1,\ldots, N$.

5 Analysis of Each Subsystem in Isolation

In this section, we describe how to analyze each subsystem in isolation using Marie's approximate analysis of general closed queueing networks [20]. Throughout this analysis, the state-dependent rates of the external arrival processes, $\lambda^i(n^i)$, $0 \le n^i \le K_{i-1}$, $i = 2,\ldots, N$, are assumed to be known. To analyze each subsystem using Marie's method, we first view the subsystem as a closed queueing network. For subsystems R^i, $i = 2,\ldots, N$, this is done by considering the kanbans of stage i as the customers of the closed network, and the parts and demands (in the case of the last subsystem R^N) as external resources. Note that the queueing network associated with subsystem R^1 is already being modeled as a closed queueing network in the decomposition. Its customers are the kanbans of stage 1.

The closed queueing network associated with subsystem R^N is partitioned into $m_N + 2$ stations, namely the synchronization stations I^N and O^N and the m_N stations of subnetwork L_N. Similarly, the closed queueing network associated with each subsystem R^i is partitioned into $m_i + 2$ stations, namely the synchronization station I^i, the m_i stations of subnetwork L_i, and station $\hat{S}_i$. Finally, the closed queueing network associated with subsystem R^1 is partitioned into $m_1 + 1$ stations, namely the m_1 stations of subnetwork L_1, and station $\hat{S}_1$. Each station is approximated by an exponential-service station with load-dependent service rates. The resulting network associated with each subsystem is a Gordon-Newell, product-form network [17] consisting of K_i customers and $m_i + 2$ stations for subsystem R^i, $i = 2,\ldots, N$, and $m_1 + 1$ stations for subsystem R^1. The stations within each subsystem R^i, $i = 1,\ldots, N$, will be denoted by the index $k \in M_i$, where $M_1 = \{1,\ldots, m_1, \hat{S}_1\}$, $M_i = \{I^i, 1,\ldots, m_i, \hat{S}_i\}$ for $i = 2,\ldots, N$-1, and $M_N = \{I^N, 1,\ldots, m_N, O^N\}$.

Let $\mu_k^i(n_k^i)$ denote the load-dependent service rate of station k in the product-form network of subsystem R^i when there are n_k^i customers in that station. We will show how to determine $\mu_k^i(n_k^i)$, $n_k^i = 1,\ldots, K_i$, for each station $k \in M_i$ within a particular subsystem R^i, $i = 1,\ldots, N$. The method for doing this is the same for all subsystems R^i, $i = 1,\ldots, N$; therefore, for

the sake of notational simplicity we will drop index i that denotes variables associated with subsystem R^i.

Let vector $\mathbf{n} = (n_1,\ldots, n_k,\ldots, n_M)$ be the state of the closed, product-form network, where n_k denotes the number of customers present at station k. Then, the probability of being in stage **n,** $P(\mathbf{n})$, is given by the following product-form solution [11]:

$$P(\mathbf{n}) = \frac{1}{G(K)} \prod_{k \in M} \left[\prod_{n=1}^{n_k} \frac{V_k}{\mu_k(n)} \right], \tag{4}$$

where V_k is the average visit ratio of station k in the original system and is given from the routing matrix of the original system, and $G(K)$ is the normalization constant.

To determine the unknown parameters $\mu_k(n_k)$, $n_k = 1,\ldots, K$, in the product-form solution (4), each station is analyzed in isolation as an open system with a state-dependent, Poisson arrival process, whose rate $\lambda_k(n_k)$ depends on the total number of customers n_k present in the station. Let T_k denote this open system. Assume first that the rates $\lambda_k(n_k)$ are known for $n_k = 1,\ldots, K-1$. The open queueing system T_k can then be analyzed in isolation using any appropriate technique to obtain the steady-state probabilities of having n_k customers in the isolated system, say $P_k(n_k)$. The issue of analyzing in isolation each queueing system T_k will be discussed immediately after Algorithm 1, below. The conditional throughput of this isolated open system when its population is n_k, $v_k(n_k)$, can then be derived using the relation [11],

$$v_k(n_k) = \lambda_k(n_k - 1) \frac{P_k(n_k - 1)}{P_k(n_k)}, \text{ for } n_k = 1,\ldots, K. \tag{5}$$

The load-dependent service rates of the k-th station of the closed product-form network are then set equal to the conditional throughputs of the corresponding open station in isolation, i.e.:

$$\mu_k(n_k) = v_k(n_k), \text{ for } n_k = 1,\ldots, K. \tag{6}$$

Once the rates $\mu_k(n_k)$ have been obtained, the state-dependent arrival rates $\lambda_k(n_k)$ can be obtained from the generalized, product-form solution as [6], [11]:

$$\lambda_k(n_k) = V_k \frac{G_k(K - n_k - 1)}{G_k(K - n_k)}, \text{ for } n_k = 1,\ldots, K-1, \text{ and } \lambda(K) = 0, \tag{7}$$

where $G_k(n)$ is the normalization constant of the closed, product-form network with station k removed (complementary network) and population n. $G_k(n)$ is a function of the parameters $\mu_{k'}(n_{k'})$ for all $k' \neq k$ and $n_{k'} = 1,\ldots, K$, and can be efficiently computed using any computational algorithm for product-form networks [6], [9]. An iterative procedure can then be used to determine these unknown quantities. This procedure is described by the following algorithm.

Algorithm 1: Analysis of a Subsystem in Isolation

Step 0: (Initialization) Set $\mu_k(n_k)$ to some initial value, for $k \in M$ and $n_k = 1,\ldots, K$.

Step 1: For $k \in M$:

Calculate the state-dependent arrival rates $\lambda_k(n_k)$, for $n_k = 0,\ldots, K-1$, using (7).

Step 2: For $k \in M$:

a. Analyze the open queueing system T_k.

b. Derive the steady state probabilities $P_k(n_k)$ of having n_k customers, for $n_k = 1,\ldots, K$.

c. Calculate the conditional throughputs $v_k(n_k)$, for $n_k = 1,\ldots, K$, using (5).

Step 3: For $k \in M$:

Set the load-dependent service rates $\mu_k(n_k)$, for $n_k = 1,\ldots, K$, in the closed, product-form network using (6).

Step 4: Go to Step 1 until convergence of the parameters $\mu_k(n_k)$.

Next, we show how to analyze each open queueing system T_k. To do this, we reintroduce index i denoting subsystem R^i. Step 2a of Algorithm 1 above requires the analysis of the open queueing systems T_k^i for $k \in M_i$ and $i = 1,\ldots, N$. There are four different types of queueing systems: 1) the synchronization station O^N in subsystem R^N, 2) the synchronization stations I^i in subsystems R^i, $i = 2,\ldots, N$, 3) the m_i stations in each subnetwork L_i, $i = 1,\ldots, N$, and 4) the pseudo-stations $\hat{S}_i$ in subsystems R^i, $i = 1,\ldots, N-1$.

First, consider the analysis of synchronization station O^N in subsystem R^N. O^N is a synchronization station fed by a continuous-time Markov arrival process with state-dependent rates, $\lambda_O^N(n_O^N)$, $0 \le n_O^N \le K_N$, and an external Poisson process with fixed rate λ_D. An exact solution for this system is easy to obtain by solving the underlying continuous-time Markov

chain. Namely, the steady-state probabilities $P_O^N(n_O^N)$ of having n_O^N customers in subsystem O^N can be derived, and the conditional throughput $v_O^N(n_O^N)$ can be estimated using (5) (see [10] and Appendix A).

The synchronization station I^i in each subsystem R^i, $i = 2,\ldots, N$, is a synchronization station fed by two continuous-time Markov arrival processes with state-dependent rates, $\lambda_I^i(n_I^i)$, $0 \le n_I^i \le K_i$, and $\lambda^i(n^i)$, $0 \le n^i \le K_{i-1}$. An exact solution for this system is also easy to obtain by solving the underlying continuous-time Markov chain. (see [14] and Appendix B).

The analysis in isolation of any station $k \in \{1,\ldots, m_i\}$ in each subnetwork L_i, $i = 1,\ldots, N$, reduces to the analysis of a $\lambda_k^i(n_k^i)/G_i/1/N$ queue. Classical methods can be used to analyze this queue to obtain the steady-state probabilities $P_k^i(n_k^i)$. For instance, if the service time distribution is Coxian, the algorithms given in [21] may be used. For multiple-server stations, we can use the numerical technique presented in [25]. The conditional throughput $v_k^i(n_k^i)$ can then be derived from the state probabilities using (5). In the special case where the service time is exponentially distributed, the conditional throughput $v_k^i(n_k^i)$ is simply equal to the load-dependent service rate $\mu_k^i(n_k^i)$ [11].

Finally, as was mentioned earlier, pseudo-station $\hat{S}_i$ in subsystem R^i, $i = 1,\ldots, N-1$, is an aggregate representation of subsystem R^{i+1}, which is nested inside subsystem R^i. Therefore, the conditional throughput of pseudo-station $\hat{S}_i$, $v_{\hat{S}}^i(n_{\hat{S}}^i)$, is set equal to the conditional throughput of subsystem R^{i+1}. The conditional throughput of any subsystem R^i, $i = 2,\ldots, N$, is denoted by $v^i(n^i)$ and can be estimated by the following simple expression [3]:

$$v^i(n^i) = \begin{cases} \lambda_I^i(K_i - n^i) & \text{for } 1 \le n^i \le K_i, \\ \lambda_I^i(0) & \text{for } K_i \le n^i \le K_{i-1}. \end{cases} \tag{8}$$

6 Analysis of the Entire Echelon Kanban Control System

In Section 5 we analyzed each subsystem of the decomposition in isolation, given that the arrival rates of the external arrival processes were known. In this section, we show how to determine these arrival rates.

Consider again the queueing network of the original system, R, which has been decomposed into N subsystems (see Figure 3 for $N = 3$). In each subsystem R^i, $i = 2,\ldots, N$, the unknown parameters involved in the decomposition are the arrival rates of raw parts at each upstream synchronization station I^i, $\lambda^i(n^i)$, $0 \le n^i \le K_{i-1}$. Recall that pseudo-station $\hat{S}_{i-1}$ in subsystem R^{i-1} represents subsystem R^i, $i = 2,\ldots, N$; therefore, the external arrival process of raw parts at synchronization station I^i in subsystem R^i should be identical to the arrival process of parts at pseudo-station $\hat{S}_{i-1}$ in subsystem R^{i-1}. The latter process was involved in the analysis of subsystem R^{i-1} in isolation and was characterized by a state-dependent Poisson arrival process with rate $\lambda_{\hat{S}}^{i-1}(n_{\hat{S}}^{i-1})$, $0 \le n_{\hat{S}}^{i-1} \le K_{i-1}$. As a result, the following set of equations holds:

$$\lambda^i(n^i) = \lambda_{\hat{S}}^{i-1}(n^i) \text{ for } 0 \le n^i \le K_{i-1} \text{ and } i = 2,\ldots, N. \tag{9}$$

Equation (9) implies that the unknown parameters $\lambda^i(n^i)$ are the solutions of a fixed-point problem. To determine these quantities we use an iterative procedure. This procedure is described in Algorithm 2 below. Algorithm 2 consists of several forward and backward steps. A forward step from subsystem R^{i-1} to R^i uses new estimates of the arrival rates $\lambda^i(n^i)$ to the upstream synchronization station I^i of subsystem R^i, to resolve R^i using Algorithm 1. A backward step from R^i to R^{i-1} solves R^{i-1} using Algorithm 1, given that the arrival rates $\lambda^i(n^i)$ to the upstream synchronization station I^i of each subsystem R^j, $j = i,\ldots, N$, have converged. The procedure starts with subsystem R^N and moves backwards until it reaches subsystem R^1. Subsystem R^N is analyzed first using Algorithm 1 and current estimates of $\lambda^N(n^N)$. This yields the conditional throughput of R^N, $v^N(n^N)$, which is needed to analyze subsystem R^{N-1}, since it determines the load-dependent exponential-service rates of pseudo-station $\hat{S}_{N-1}$. Subsystem R^{N-1} is analyzed next using Algorithm 1 and current estimates of $\lambda^{N-1}(n^{N-1})$. This yields the conditional throughput of R^{N-1}, $v^{N-1}(n^{N-1})$, and the arrival rates to the pseudo-station $\hat{S}_{N-1}$, $\lambda_{\hat{S}}^{N-1}(n_{\hat{S}}^{N-1})$. If these arrival rates are not equal to the current estimates of the arrival rates $\lambda^N(n^N)$, then the latter rates have not converged. In this case, the current estimates of $\lambda^N(n^N)$ are updated to $\lambda_{\hat{S}}^{N-1}(n_{\hat{S}}^{N-1})$ and subsystem R^N is analyzed again using Algorithm 1 with the new estimates. Otherwise, the arrival rates $\lambda^N(n^N)$ have converged and the procedure moves on to the analysis of subsystem R^{N-2} using Algorithm 1, where the load-dependent exponential-service rates of pseudo-station $\hat{S}_{N-2}$ are set equal to $v^{N-1}(n^{N-1})$. This procedure is repeated for

subsystems R^{N-2}, R^{N-3},…, until the first subsystem, R^1, is reached and all the arrival rates $\lambda^i(n^i)$, $i = 2,..., N$, have converged. All the performance parameters of interest can then be derived.

Algorithm 2: Analysis of a Multi-Stage Echelon Kanban Control System

Step 0: (Initialization) Set the unknown arrival rates of each subsystem R^i to some initial values, e.g. $\lambda^i(n^i) = \lambda_D$, $0 \le n^i \le K_{i-1}$, and $i = 2,\ldots, N$.

Step 1: Computation and convergence of the arrival rates, $\lambda^i(n^i)$, $i = 2,\ldots, N$.

Set $i = N$

While $i \ge 1$

If $i = N$

Solve subsystem R^N using Algorithm 1 and calculate the throughput $v^N(n^N)$, $n^N = 1,\ldots, K_{N-1}$, from (8).

Set $i = i - 1$.

Else

Solve subsystem R^i using Algorithm 1 and calculate the arrival rate $\lambda^i_{\hat{S}}(n^i_{\hat{S}})$, $n^i_{\hat{S}} = 0,\ldots, K_i$, and the throughput $v^i(n^i)$, $n^i = 1,\ldots, K_{i-1}$, from (8).

If $\lambda^{i+1}(n^{i+1}) = \lambda^i_{\hat{S}}(n^{i+1})$, $n^{i+1} = 0,\ldots, K_i$,

Set $i = i - 1$

Else

Set $\lambda^{i+1}(n^{i+1}) = \lambda^i_{\hat{S}}(n^{i+1})$, $n^{i+1} = 0,\ldots, K_i$, and set $i = i + 1$

Endif

Endif

Endwhile

In the case of the saturated echelon kanban control system, we can use the same algorithm. The only difference is in the analysis of subsystem R^N in Algorithm 1, where there is no downstream synchronization station O^N. As far as the convergence properties of Algorithms 1 and 2 are concerned, in all of the numerical examples that we examined (see

Section 7), both algorithms converged. The convergence criterion was that the relative difference between the values of every unknown parameter at two consecutive iterations should be less that 10^{-4}.

Once Algorithm 2 has converged, all the performance parameters of the system can be calculated. Indeed, from the analysis of each subsystem R^i using Algorithm 1, it is possible to derive the performance parameters of stage i in the original network R, especially the throughput and the average length of each queue, including the queues of the synchronization stations. Thus, in the case of the saturated echelon kanban system, we can derive the throughput, the average WIP, the average number of finished parts, and the average number of free echelon kanbans for each stage. In the case of the echelon kanban control system with external demands, some other important performance measures can be derived from the analysis of subsystem R^N, namely, the proportion of backordered demands, p_B, the average number of backordered demands, Q_D, and the average waiting time of backordered demands, W_B. These performance measures can be derived as follows [10], [14]:

$$p_B = P_O^N(0), \qquad Q_D = P_O^N(0)\frac{1}{\frac{\lambda_O^N(0)}{\lambda_D} - 1}, \qquad W_B = \frac{Q_D}{p_B \lambda_D},$$

where $\lambda_O^N(0)$ is the arrival rate of finished parts at synchronization station O^N when there are no finished parts at that station and $P_O^N(0)$ is the steady-state probability of having no finished parts at synchronization station O^N.

7 Numerical Results

In this section, we test the approximation method for the performance evaluation of the echelon kanban control system that we developed in Sections 4-6 on several numerical examples. The approximation method was implemented on an Intel Celeron PC @ 433 MHz, and its results are compared to simulation results obtained using the simulation software ARENA on an AMD Athlon PC @ 1400 MHz. For each simulation experiment we run a single replication. The length of this replication was set equal to the time needed for the system to produce 68 million parts. The initial condition of the system at the beginning of the replication was set to a typical regenerative state, namely the state where all customer demands and demands for the production of new parts at all stages have been satisfied. This

permitted us to set the warm-up period at the beginning of the replication equal to zero. In all simulation experiments we used 95% confidence intervals. The numerical examples are organized into Sections 7.1 and 7.2. In Section 7.1, we study the accuracy and rapidity of the approximation method as well as the influence of some key parameters of the echelon kanban control system on system performance. In Section 7.2, we use the approximation method to optimize the design parameters (echelon kanbans) of the system.

7.1 Influence of Parameters

In this section, we test the accuracy and rapidity of the approximation method with two numerical examples in which we vary the number of stages, the number of kanbans in each stage, and the service-time distributions of the manufacturing process of each stage. For each example, we consider first the case of the saturated system and then the case of the system with external demands. In each example, we compare the performance of the system obtained by the approximation method to that obtained by simulation. We also compare the performance of the echelon kanban control system obtained by the approximation method and by simulation to the performance of the conventional or installation kanban control system obtained by a similar approximation method developed in [14] and by simulation.

Example 1

In Example 1, we consider an echelon kanban system composed of N identical stages, where each stage contains a single machine with exponentially distributed service times with mean equal to 1. In order to compare the echelon kanban control system to the conventional kanban control system, we first set the number of installation kanbans of each stage i in the conventional kanban control system, say K_i^c, equal to some constant K, i.e. $K_i^c = K$. Then, we set the number of echelon kanbans of each stage i in the echelon kanban control system, say K_i^e, equal to the sum of the installation kanbans of stages $i,\ldots, N$, in the conventional kanban control system, i.e. $K_i^e = \sum_{j=i}^{N} K_j^c = (N+1-i)K$.

For the case of the saturated system, the main performance parameter of interest is the throughput of the system, which determines the production capacity of the system. Table 1 shows the throughput of the saturated echelon kanban control system obtained by the approximation method and by simulation, for different values of N and K. The same table also

shows the 95% confidence interval for the simulation results, the percentage of relative error of the approximation method with respect to simulation, and the number of iterations of Algorithm 2 that are needed to reach convergence. Table 2 shows the same results for the conventional kanban control system obtained in [14].

Configuration	Simulation		Approximation		
	Production Capacity	Confidence Interval	Production Capacity	Relative Error	Iterations
1.1: $N = 3$; $K = 1$	0.581	±0.1%	0.571	- 1.8%	7
1.2: $N = 3$; $K = 3$	0.809	±0.1%	0.804	- 0.6%	7
1.3: $N = 3$; $K = 5$	0,877	±0.2%	0.873	- 0.5%	7
1.4: $N = 3$; $K = 10$	0.934	±0.5%	0.933	- 0.1%	7
1.5: $N = 3$; $K = 15$	0.955	±0.6%	0.954	- 0.1%	7
1.6: $N = 5$; $K = 1$	0.522	±0.0009%	0.502	- 4%	16
1.7: $N = 5$; $K = 3$	0,772	±0.1%	0.761	- 1.4%	16
1.8: $N = 5$; $K = 5$	0.85	±0.1%	0.843	- 0.8%	16
1.9: $N = 5$; $K = 10$	0.919	±0.2%	0.916	- 0.3%	16
1.10: $N = 5$; $K = 15$	0.945	±0.0009%	0.942	- 0.3%	16
1.11: $N = 10$; $K = 1$	0.485	±0.0007%	0.456	- 6.4%	56
1.12: $N = 10$; $K = 3$	0.745	±0.5%	0.730	- 2.1%	56
1.13: $N = 10$; $K = 5$	0,831	±0.7%	0.820	- 1.3%	56
1.14: $N = 10$; $K = 10$	0.908	±0.1%	0.902	- 0.7%	56
1.15: $N = 10$; $K = 15$	0.937	±0.1%	0.933	- 0.4%	56

Table 1: Production capacity of the saturated echelon kanban control system (Example 1).

From the results in Table 1, we note that the number of iterations of Algorithm 2 of the approximation method increases with the number of stages, as is expected. Specifically, for $N = 3$, 5 and 10, we have 7, 16, and 56 iterations of Algorithm 2, respectively. As far as the convergence of Algorithm 1 is concerned, we also note that subsystem R^N requires two iterations of Algorithm 1, subsystem R^1 requires one iteration, and all other subsystems require three iterations, irrespectively of the number of stages N, for all the configurations tested. The simulation time is extremely long (over two hours) compared to the time required for the approximation method, which is approximately 1-10 seconds. From Table 1 we note that as the number of echelon kanbans increases, for a given number of stages N, the throughput also increases and asymptotically tends to the production rate of each machine in isolation. Moreover, the throughput seems to be decreasing in the number of stages. The results obtained by the approximation method are fairly accurate when compared to the simulation results. The relative error is very small in general except for the cases where $K = 1$,

where we observe somewhat significant errors. This happens because when the number of echelon kanbans is small, there are strong dependence phenomena among stations and these phenomena are not well captured by the state-dependent, continuous-time, Markov arrival processes assumed in the decomposition method. Comparing the results between Tables 1 and 2, we note that the production capacity of the echelon kanban control system is always higher than that of the conventional kanban control system, given that the two systems have the same value of K.

Configuration	Simulation		Approximation		
	Production Capacity	Confidence Interval	Production Capacity	Relative Error	Iterations
1.1: $N = 3$; $K = 1$	0.562	±0.5%	0.547	- 2.7%	2
1.2: $N = 3$; $K = 3$	0.800	±0.7%	0.792	- 1.0%	2
1.3: $N = 3$; $K = 5$	0.869	±1.3%	0.865	- 0.5%	2
1.4: $N = 3$; $K = 10$	0.926	±0.8%	0.928	+ 0.2%	2
1.5: $N = 3$; $K = 15$	0.952	±1.2%	0.951	- 0.1%	2
1.6: $N = 5$; $K = 1$	0.484	±0.6%	0.449	- 7.0%	4
1.7: $N = 5$; $K = 3$	0.746	±0.8%	0.731	- 2.0%	4
1.8: $N = 5$; $K = 5$	0.833	±0.8%	0.822	- 1.3%	4
1.9: $N = 5$; $K = 10$	0.901	±1.2%	0.904	+ 0.3%	4
1.10: $N = 5$; $K = 15$	0.943	±1.1%	0.934	- 0.9%	4
1.11: $N = 10$; $K = 1$	0.429	±0.5%	0.379	- 11.6%	7
1.12: $N = 10$; $K = 3$	0.704	±0.7%	0.680	- 3.4%	6
1.13: $N = 10$; $K = 5$	0.806	±0.9%	0.786	- 2.6%	5
1.14: $N = 10$; $K = 10$	0.855	±0.5%	0.883	- 3.2%	5
1.15: $N = 10$; $K = 15$	0.917	±1.3%	0.919	+ 0.2%	5

Table 2: Production capacity of the saturated conventional kanban control system (Example 1).

For the system with backordered demands, the main performance parameters of interest are the proportion of backordered demands, p_B, the average number of backordered demands, Q_D, and the average waiting time of backordered demands, W_B, as defined at the end of Section 6. Table 3 shows these performance parameters obtained by the approximation method and by simulation, for the configurations of parameters 1.3, 1.8, and 1.13 of Table 1, i.e. for $K = 5$, and different values of the customer demand rate, λ_D. The same table also shows the 95% confidence interval for the simulation results and the number of iterations of Algorithm 2 that are needed to reach convergence. Table 4 shows the same results for the conventional kanban control system obtained in [14].

Configuration	Q_D	W_B	P_B (%)	Iterations
1.16: $N = 3$; $K = 5$; $\lambda_D = 0.1$ Approximation Simulation	 0.0 0.0	 0.0 0.0	 0.0 0.0	6
1.17: $N = 3$; $K = 5$; $\lambda_D = 0.5$ Approximation Simulation	 0.035 0.034 (±0.9%)	 4.069 2.066 (±1.2%)	 1.729 3.337	7
1.18: $N = 3$; $K = 5$; $\lambda_D = 0.625$ Approximation Simulation	 0.221 0.213 (±0.1%)	 4.594 3.014 (±14.2%)	 7.687 11.32	7
1.19: $N = 3$; $K = 5$; $\lambda_D = 0.8$ Approximation Simulation	 4.176 4.095 (±3.6%)	 10.791 9.755 (±7%)	 48.38 52.47	8
1.20: $N = 5$; $K = 5$; $\lambda_D = 0.1$ Approximation Simulation	 0.0 0.0	 0.0 0.0	 0.0 0.0	16
1.21: $N = 5$; $K = 5$; $\lambda_D = 0.5$ Approximation Simulation	 0.035 0.032 (±0.007%)	 4.070 3.189 (±0.003%)	 1.71 2.03	16
1.22: $N = 5$; $K = 5$; $\lambda_D = 0.8$ Approximation Simulation	 6.774 6.5686 (±0.08%)	 14.440 12.895 (±0.02%)	 58.69 63.67	22
1.23: $N = 10$; $K = 5$; $\lambda_D = 0.1$ Approximation Simulation	 0.0 0.0	 0.0 0.0	 0.0 0.0	20
1.24: $N = 10$; $K = 5$; $\lambda_D = 0.5$ Approximation Simulation	 0.035 0.023 (±0.005%)	 4.070 3.512 (±0.002%)	 1.72 1.28	39
1.25: $N = 10$; $K = 5$; $\lambda_D = 0.77$ Approximation Simulation	 3.817 3.131 (±0.003%)	 10.709 9.064 (±0.001%)	 46.3 49.3	61

Table 3: Average number of backordered demands, average waiting time of backordered demands and proportion of backordered demands for the echelon kanban control system (Example 1).

From the results in Table 3 we note that as the customer demand arrival rate increases, the number of iterations of Algorithm 2 also increases, though not dramatically. As far as the average number of backordered demands, Q_D, is concerned, we note that the analytical method is fairly accurate. This is not true for the average waiting time of backordered demands, W_B, where in some cases the difference between the approximation method and simulation are significant. Comparing the results between Tables 3 and 4, we note that the echelon kanban control system always has a smaller average number of backordered

demands, Q_D, than the conventional kanban control system, given that the two systems have the same value of K. The difference in the average number of backordered demands is more pronounced when the two systems are highly loaded, i.e. when λ_D is close to the production capacity.

Configuration	Q_D	W_B	P_B (%)	Iterations
1.16: $N = 3$; $K = 5$; $\lambda_D = 0.1$ Approximation Simulation	 0.0 0.0	 0.0 0.0	 0.0 0.0	1
1.17: $N = 3$; $K = 5$; $\lambda_D = 0.5$ Approximation Simulation	 0.035 0.033 (±30%)	 2.06 2.16 (±17%)	 3.4 3.1	2
1.18: $N = 3$; $K = 5$; $\lambda_D = 0.625$ Approximation Simulation	 0.222 0.230 (±17%)	 3.00 3.26 (±15%)	 11.82 11.78	3
1.19: $N = 3$; $K = 5$; $\lambda_D = 0.8$ Approximation Simulation	 4.56 4.26 (±19%)	 10.1 10.3(±13%)	 56.3 52.1	4
1.20: $N = 5$; $K = 5$; $\lambda_D = 0.1$ Approximation Simulation	 0.0 0.0	 0.0 0.0	 0.0 0.0	1
1.21: $N = 5$; $K = 5$; $\lambda_D = 0.5$ Approximation Simulation	 0.0353 0.038 (±30%)	 2.07 2.16 (±9%)	 3.40 3.58	2
1.22: $N = 5$; $K = 5$; $\lambda_D = 0.8$ Approximation Simulation	 11.26 8.93 (±22%)	 19.3 17.2 (±15%)	 73.0 65.2	7
1.23: $N = 10$; $K = 5$; $\lambda_D = 0.1$ Approximation Simulation	 0.0 0.0	 0.0 0.0	 0.0 0.0	1
1.24: $N = 10$; $K = 5$; $\lambda_D = 0.5$ Approximation Simulation	 0.0353 0.0368 (±30%)	 2.07 2.18 (±17%)	 3.40 3.38	2
1.25: $N = 10$; $K = 5$; $\lambda_D = 0.77$ Approximation Simulation	 6.89 5.95 (±22%)	 13.9 13.7 (±14%)	 64.2 56.9	11

Table 4: Average number of backordered demands, average waiting time of backordered demands and proportion of backordered demands for the conventional kanban control system (Example 1).

Table 5 shows the results for the average number of finished parts (FP) and the average work-in-process (WIP) at each stage for the configurations of parameters 1.17 and 1.19 in Table 3. Table 6 shows the same results for the conventional kanban control system.

Configuration	Stage 1		Stage 2		Stage 3	
	WIP	FP	WIP	FP	WIP	FP
1.17: $N = 3$; $K = 5$; $\lambda_D = 0.5$						
Simulation	0.988 (±0.1%)	4.039 (±0.09%)	0.978 (±0.1%)	4.022 (±0.1%)	0.961 (±0.1%)	4.011 (±0.1%)
Approximation	0.999	4.031	0.995	4.005	0.969	4.000
Error	+ 1.1%	- 0.2%	+ 1.7%	- 0.4%	+ 0.8%	- 0.3%
1.19: $N = 3$; $K = 5$; $\lambda_D = 0.8$						
Simulation	3.363 (±0.5%)	2.392 (±0.3%)	3.068 (±0.3%)	2.018 (±0.3%)	2.589 (±0.3%)	1.569 (±0.5%)
Approximation	3.479	2.349	3.159	1.902	2.655	1.455
Error	+ 3.3%	- 1.8%	+ 2.9%	- 6.1%	+ 2.5%	- 7.8%

Table 5: Average work in process (WIP) and average number of finished parts (FP) in each stage for the echelon kanban control system (Example 1).

Configuration	Stage 1		Stage 2		Stage 3	
	WIP	FP	WIP	FP	WIP	FP
1.17: $N = 3$; $K = 5$; $\lambda_D = 0.5$						
Simulation	0.94 (±3.2%)	4.06 (±0.7%)	0.95 (±3.1%)	4.02 (±0.7%)	0.94 (±3.2%)	4.04 (±0.8%)
Approximation	0.97	4.03	0.97	4.01	0.97	4.00
Error	+ 3%	- 0.7%	+ 2%	- 0.2%	+ 3%	- 1%
1.19: $N = 3$; $K = 5$; $\lambda_D = 0.8$						
Simulation	2.54 (±3.0%)	2.47 (±4.0%)	2.52 (±3.2%)	1.98 (±5.0%)	2.55 (±3.1%)	1.58 (±6.3%)
Approximation	2.61	2.38	2.58	1.85	2.66	1.40
Error	+ 2.7%	- 3.6%	+ 2.4%	- 6.5%	+ 4%	- 11%

Table 6: Average work in process (WIP) and average number of finished products (FP) in each stage for the conventional kanban control system (Example 1).

Comparing the results between Tables 5 and 6, we note that the echelon kanban control system has slightly higher average WIP and lower FP than the conventional kanban control system, when the two systems are highly loaded (i.e. λ_D is close to P_s), and given that the two systems have the same value of K. When the two systems are not highly loaded, the difference in average WIP and FP between the two systems is very small. Finally, it appears that the difference in average WIP and FP between the echelon kanban control system and the conventional kanban control system is higher in upstream stages than in downstream stages.

Although the above observations hold for the particular configuration of parameters examined, we expect that they should also hold for the other configurations of Table 1 and different values of the customer demand rate, λ_D, because to a large extent they are due to the fact that the echelon kanban control system always responds faster to customer demands than the conventional kanban control system, given that the two systems have the same value of K.

Finally, we should note that the approximation method for the performance evaluation of the conventional kanban control system developed in [14] is also based on decomposing a system of N stages into N subsystems. The total number of the unknown parameter sets (the arrival rates of the external arrival processes to the subsystems) that must be determined for the conventional kanban control system, however, is twice as big as that which must be determined for the echelon kanban control system (i.e. $2(N-1)$ instead of $N-1$ external arrival rates). Yet, for both examples examined, the number of iterations needed for the convergence of the parameters is significantly lower for the conventional kanban control system than for the echelon kanban control system, given the same convergence criterion for the two systems, as can be seen from Tables 1-4. This is due to the fact that the coordination of production is decentralized in the conventional kanban control system, whereas it is centralized in the echelon kanban control system. Nonetheless, this does not seem to constitute a noticeable disadvantage of the approximation method for the echelon kanban control system, since for all the cases examined, the method converges in a matter of 1-10 seconds.

Example 2

In Example 2, we consider an echelon kanban control system consisting of $N = 3$ identical stages, where each stage contains a single machine with identical service-time distribution with mean equal to 1. The number of echelon kanbans at each stage is $K_1 = 15$, $K_2 = 10$, and $K_3 = 5$. Our goal is to investigate the influence of the variability of the service time on the performance of the above system. To this end, we consider three different distributions: a Coxian-2 distribution with squared coefficient of variation $cv^2 = 2.0$, an Erlang-2 distribution with $cv^2 = 0.5$, and an exponential distribution with $cv^2 = 1.0$. Table 7 shows the production capacity for the saturated echelon kanban control system obtained by the approximation method and by simulation, for the three different distributions. Table 8 shows the same results for the conventional kanban control system obtained in [14].

Configuration	Simulation		Approximation		
	Production Capacity	Confidence Interval	Production Capacity	Relative Error	Iterations
2.1: $N = 3$; $K = 5$; $cv^2 = 0.5$	0.929	±0.1%	0.934	+ 0.5%	11
2.2: $N = 3$; $K = 5$; $cv^2 = 1$	0.876	±0.2%	0.873	- 0.3%	7
2.3: $N = 3$; $K = 5$; $cv^2 = 2$	0.813	±0.3%	0.808	- 0.6%	13

Table 7: Production capacity of the echelon kanban control system (Example 2).

Configuration	Simulation		Approximation		
	Production Capacity	Confidence Interval	Production Capacity	Relative Error	Iterations
2.1: $N = 3$; $K = 5$; $cv^2 = 0.5$	0.926	±0.2%	0.932	+ 0.6%	2
2.2: $N = 3$; $K = 5$; $cv^2 = 1$	0.870	±0.1%	0.865	- 0.6%	2
2.3: $N = 3$; $K = 5$; $cv^2 = 2$	0.787	±0.5%	0.786	- 0.2%	2

Table 8: Production capacity of the conventional kanban control system (Example 2).

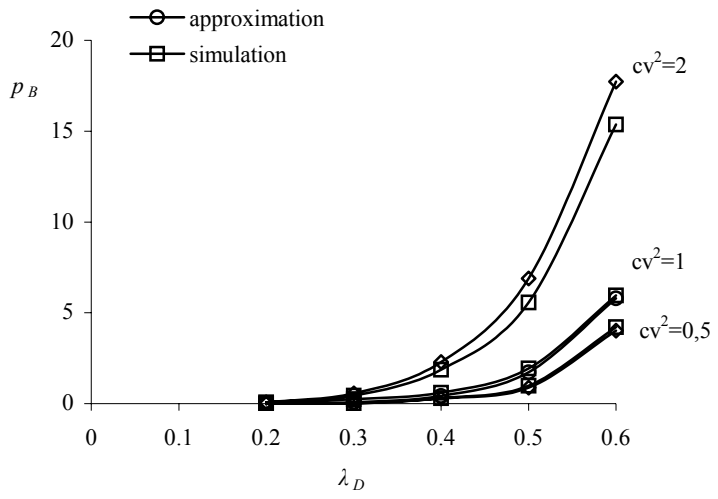

Figure 4: Proportion of backordered demands versus the average arrival rate of demands for different values of the squared coefficient of variation (Example 2).

From the results in Table 7, we note that when the variability of the service time distribution increases, the production capacity decreases, as is expected. The results obtained by the approximation method are fairly accurate when compared to the simulation results. Comparing the results between Tables 7 and 8, we note that for all the service-time distributions, the production capacity of the echelon kanban control system is higher than that of the conventional kanban control system. The results for the analytical solution and

simulation for the case of the echelon kanban system with backordered demands is shown in Figure 4. More specifically, Figure 4 depicts the proportion of backordered demands p_B as a function of the arrival rate of demands λ_D for the three different service time distributions. It appears that as the cv^2 of the service time distribution increases, the difference between simulation and analytical results tends to increase.

7.2 Optimization of Parameters

The main reason for developing an approximation method for the performance evaluation of the echelon kanban control system is to use it to optimize the design parameters of the system. The design parameters of the echelon kanban control system are the number of echelon kanbans for each stage. In order to optimize these parameters, we must define a performance measure of the system. Typical performance measures are those that include the cost of not being able to satisfy the demands on time (i.e. quality of service) and the cost of producing parts ahead of time and, therefore, building up inventory (inventory holding cost). In this paper, we consider an optimization problem where the objective is to meet a certain quality of service constraint with minimum inventory holding cost.

We examine two quality-of-service measures as in [15]. The first measure is the probability that when a customer demand arrives, it is backordered, and the second measure is the probability that when a customer demand arrives, it sees more than n waiting demands, excluding itself. The first measure is denoted by P_{rupt} and concerns the situation where the demands must be immediately satisfied. The second measure is denoted by $P(Q > n)$ and concerns the situation where we have the prerogative to introduce a delay in filling orders, which is equivalent to authorizing demands to wait. Specifically, P_{rupt} is the stationary probability of having no finished parts in the last synchronization station, and can be computed as the marginal distribution of having no finished parts in that station, which is given by (18) in Appendix A. Similarly, $P(Q > n)$ is the stationary probability of having more than n customers waiting and can be computed from the following expression:

$$P(Q > n) = \sum_{x=n+1}^{\infty} P(Q = x) = 1 - \sum_{y=0}^{n} P(Q = y) \tag{10}$$

where $P(Q = n)$ is given by (see Appendix A):

$$P(Q=n)=p_O^N(0,n)=p_O^N(0,0)\left(\frac{\lambda_D}{\lambda_O^N(0)}\right)^n. \tag{11}$$

The stationary distribution $p_O^N(0,0)$ that is needed to evaluate both P_{rupt} and $P(Q > n)$ is given by the following expression:

$$p_O^N(0,0)=\frac{1}{\frac{1}{1-\frac{\lambda_D}{\lambda_O^N(0)}}+\sum_{x=1}^{K_N}\left(\frac{1}{\lambda_D{}^x}\prod_{i=0}^{x-1}\lambda_O^N(i)\right)}. \tag{12}$$

The cost function that we want to minimize is the long-run, expected, average cost of holding inventory,

$$C_{total}=\sum_{i=1}^{N}h_iE\left[WIP_i+FP_i\right] \tag{13}$$

where h_i is the unit cost of holding $WIP_i + FP_i$ inventory per unit time in stage i.

In the remaining of this section, we optimize the echelon kanbans of an echelon kanban control system made up of $N = 5$ stages, where each stage contains a single machine with exponentially distributed service times with mean equal to 1, for different combinations of inventory holding cost rates h_i, $i = 1,\ldots, 5$, and demand arrival rate $\lambda_D = 0.5$. In all cases we assume that there is value added to the parts at every stage so that the inventory holding cost increases as the stage increases i.e. $h_1 < h_2 < \ldots < h_5$. If this were not the case, i.e. if $h_1 = h_2 = \ldots = h_5$, then clearly it would make no sense to block the passage of parts from one stage to another via the use of echelon kanbans, because this would not lower the inventory holding cost but would worsen the quality of service. This implies that if $h_1 = h_2 = \ldots = h_5$, the optimal echelon kanbans satisfy $K_1 \le K_i$, $i = 2,\ldots, 5$, in which case the echelon kanban control system is equivalent to the make-to-stock CONWIP system [23] with a WIP-cap on the total number of parts in the system equal to K_1.

Table 9 shows the optimal design parameters ($K_1,\ldots, K_5$) and associated minimum, long-run, expected, average cost of holding inventory, for $\lambda_D = 0.5$ and different quality of service constraints and inventory holding cost rates $h_1,\ldots, h_5$, where $h_1 < h_2 < \ldots < h_5$. The quality of service constraints that we use are $P_{\text{rupt}} \le 0.02$ and $P(Q > n) \le 0.02$, for $n = 2$, 5, and 10.

Design criterion	K_1	K_2	K_3	K_4	K_5	Cost
$h_1 = 1,\ h_2 = 2,\ h_3 = 3,\ h_4 = 4,\ h_5 = 5$						
$P_{\text{rupt}} \le 0.02$	15	13	12	10	8	55.885
$P(Q > 2) \le 0.02$	13	11	10	8	7	46.555
$P(Q > 5) \le 0.02$	10	8	7	6	2	31.120
$P(Q > 10) \le 0.02$	7	6	5	3	1	20.253
$h_1 = 3,\ h_2 = 8,\ h_3 = 9,\ h_4 = 10,\ h_5 = 12$						
$P_{\text{rupt}} \le 0.02$	15	13	12	10	8	144.314
$P(Q > 2) \le 0.02$	13	11	10	9	6	121.161
$P(Q > 5) \le 0.02$	10	8	7	6	2	84.074
$P(Q > 10) \le 0.02$	7	6	5	3	1	57.360
$h_1 = 1,\ h_2 = 2,\ h_3 = 4,\ h_4 = 11,\ h_5 = 12$						
$P_{\text{rupt}} \le 0.02$	15	14	13	9	8	121.288
$P(Q > 2) \le 0.02$	14	13	10	7	6	98.890
$P(Q > 5) \le 0.02$	10	9	8	5	2	67.383
$P(Q > 10) \le 0.02$	8	6	4	3	1	39.483
$h_1 = 1,\ h_2 = 6,\ h_3 = 11,\ h_4 = 16,\ h_5 = 21$						
$P_{\text{rupt}} \le 0.02$	17	13	11	10	8	218.702
$P(Q > 2) \le 0.02$	15	11	10	8	5	178.162
$P(Q > 5) \le 0.02$	10	8	7	6	2	115.601
$P(Q > 10) \le 0.02$	8	6	5	3	1	76.523
$h_1 = 1,\ h_2 = 11,\ h_3 = 21,\ h_4 = 31,\ h_5 = 41$						
$P_{\text{rupt}} \le 0.02$	17	13	11	10	8	420.405
$P(Q > 2) \le 0.02$	15	11	10	8	5	341.324
$P(Q > 5) \le 0.02$	10	8	7	6	2	221.203
$P(Q > 10) \le 0.02$	8	6	5	3	1	145.047
$h_1 = 1,\ h_2 = 2,\ h_3 = 4,\ h_4 = 8,\ h_5 = 16$						
$P_{\text{rupt}} \le 0.02$	17	15	12	9	7	143.879
$P(Q > 2) \le 0.02$	14	13	11	7	5	112.442
$P(Q > 5) \le 0.02$	10	8	7	6	2	65.843
$P(Q > 10) \le 0.02$	8	6	5	3	1	39.934
$h_1 = 1,\ h_2 = 3,\ h_3 = 9,\ h_4 = 27,\ h_5 = 81$						
$P_{\text{rupt}} \le 0.02$	19	17	14	10	6	633.178
$P(Q > 2) \le 0.02$	17	15	12	8	4	471.867
$P(Q > 5) \le 0.02$	12	10	8	6	1	231.446
$P(Q > 10) \le 0.02$	8	6	5	3	1	139.066

Table 9: Optimal configuration and associated costs for $\lambda_D = 0.5$ and different values of $h_1,\ldots, h_5$, for the echelon kanban control system.

From the results in Table 9, we note that the higher the number of backordered demands, n, the lower the optimal number of echelon kanbans, and hence the inventory holding cost. As the difference between the holding cost rates h_i, $i = 1,\ldots, 5$, increases, the difference between the optimal values of K_i, $i = 1,\ldots, 5$, also increases, since the behavior of

the echelon kanban control system diverts from that of the make-to-stock CONWIP system. When the difference between the holding cost rates h_i, $i = 1,\ldots, 5$, is low, the behavior of the echelon kanban control system tends to that of the make-to-stock CONWIP system.

Table 10 shows the optimal design parameter K_1 and associated minimum inventory holding cost for $\lambda_D = 0.5$ and different quality of service constraints and inventory holding cost rates $h_1,\ldots, h_5$, for the make-to-stock CONWIP system. Comparing the results between Tables 9 and 10, we note that the make-to-stock CONWIP system performs considerably worse than the echelon kanban control system.

Design criterion	K_1	Cost
$h_1 = 1,\ \ h_2 = 6,\ \ h_3 = 11,\ \ h_4 = 16,\ \ h_5 = 21$		
$P_{\text{rupt}} \le 0.02$	14	244.163
$P(Q > 2) \le 0.02$	12	202.415
$P(Q > 5) \le 0.02$	10	161.006
$P(Q > 10) \le 0.02$	8	120.307
$h_1 = 1,\ \ h_2 = 11,\ \ h_3 = 21,\ \ h_4 = 31,\ \ h_5 = 41$		
$P_{\text{rupt}} \le 0.02$	14	474.326
$P(Q > 2) \le 0.02$	12	392.830
$P(Q > 5) \le 0.02$	10	312.012
$P(Q > 10) \le 0.02$	8	232.613
$h_1 = 1,\ \ h_2 = 2,\ \ h_3 = 4,\ \ h_4 = 8,\ \ h_5 = 16$		
$P_{\text{rupt}} \le 0.02$	14	175.160
$P(Q > 2) \le 0.02$	12	143.407
$P(Q > 5) \le 0.02$	10	111.986
$P(Q > 10) \le 0.02$	8	81.260
$h_1 = 1,\ \ h_2 = 3,\ \ h_3 = 9,\ \ h_4 = 27,\ \ h_5 = 81$		
$P_{\text{rupt}} \le 0.02$	14	850.927
$P(Q > 2) \le 0.02$	12	690.358
$P(Q > 5) \le 0.02$	10	531.715
$P(Q > 10) \le 0.02$	8	377.102

Table 10: Optimal configuration and associated costs for $\lambda_D = 0.5$ and different values of $h_1,\ldots, h_5$, for the CONWIP system.

8 Conclusions

We developed an analytical, decomposition-based approximation method for the performance evaluation of the echelon kanban control system and tested it on several numerical examples. The numerical examples showed that the method is quite accurate in most cases. They also showed that the echelon kanban control system has some advantages

over the conventional kanban control system. Specifically, when the two systems have the same value of K, the production capacity of the echelon kanban control system has higher production capacity, lower average number of backordered demands, but only slightly higher average WIP and either slightly higher or slightly lower FP than the conventional kanban control system. The numerical results also showed that as the variability of the service time distribution increases, the production capacity of the echelon kanban control system and the accuracy of the approximation method decrease. Finally, we know that the optimized echelon kanban control system always performs at least as well as the optimized make-to-stock CONWIP system since the latter system is a special case of the first system. The numerical results showed that in fact the superiority in performance of the echelon kanban control system over that of the make-to-stock CONWIP system can be quite significant, particularly when the increase in inventory holding costs from one stage to its downstream stage becomes large.

References

[1] Baskett, F., K.M. Chandy, R.R. Muntz and F. Palacios-Gomez (1975) "Open, Closed and Mixed Networks of Queues with Different Classes of Customers," *Journal of ACM*, 22, 248-260.

[2] Baynat, B. and Y. Dallery (1993) "A Unified View of Product-Form Approximation Techniques for General Closed Queueing Networks," *Performance Evaluation*, 18 (3), 205-224.

[3] Baynat, B. and Y. Dallery (1993) "Approximate Techniques for General Closed Queueing Networks with Subnetworks Having Population Constraints," *European Journal of Operational Research*, 69, 250-264.

[4] Baynat, B. and Y. Dallery (1996) "A Product-Form Approximation Method for General Closed Queueing Networks with Several Classes of Customers," *Performance Evaluation*, 24, 165-188.

[5] Baynat, B., Y. Dallery and K. Ross (1994) "A Decomposition Approximation Method for Multiclass BCMP Queueing Networks with Multiple-Server Stations," *Annals of Operations Research*, 48, 273-294.

[6] Bruell, S.C. and G. Balbo (1980) *Computational Algorithms for Closed Queueing Networks*, Elsevier North-Holland, Amsterdam.

[7] Buzacott, J.A. (1989) "Queueing Models of Kanban and MRP Controlled Production Systems," *Engineering Costs and Production Economics*, 17, 3-20.

[8] Buzacott, J.A. and J.G. Shanthikumar (1993) *Stochastic Models of Manufacturing Systems*, Prentice-Hall, Englewood Cliffs, NJ.

[9] Buzen, J.P. (1973) "Computational Algorithms for Closed Queueing Networks with Exponential Servers," *Comm. ACM*, 16 (9), 527-531.

[10] Dallery, Y. (1990) "Approximate Analysis of General Open Queueing Networks with Restricted Capacity," *Performance Evaluation*, 11 (3), 209-222.

[11] Dallery, Y. and X. Cao (1992) "Operational Analysis of Stochastic Closed Queueing Networks," *Performance Evaluation*, 14 (1), 43-61.

[12] Dallery, Y. and G. Liberopoulos (2000) "Extended Kanban Control System: Combining Kanban and Base Stock," *IIE Transactions*, 32 (4), 369-386.

[13] Daryanto, A. J.C.W van Ommeren and W.H.M. Zijm (2003) "A Closed-Loop Two-Indenture Reparable Item System," in *Proceedings of the Fourth Aegean International Conference on Analysis of Manufacturing Systems,* Samos Island, Greece, July 1-4, Ziti Publishing, Thessaloniki, Greece, 191-201.

[14] Di Mascolo, M., Y. Frein and Y. Dallery (1996) "An Analytical Method for Performance Evaluation of Kanban Controlled Production Systems," *Operations Research*, 44 (1), 50-64.

[15] Duri, C., Y. Frein and M. Di Mascolo (2000) "Comparison among Three Pull Control Policies: Kanban, Base Stock and Generalized Kanban," *Annals of Operations Research*, 93, 41-69.

[16] Frein, Y., M. Di Mascolo and Y. Dallery (1995) "On the Design of Generalized Kanban Control Systems," *International Journal of Operations and Production Management*, 15 (9), 158-184.

[17] Gordon, W.J. and G.F. Newell (1967) "Closed Queueing Networks with Exponential Servers," *Operations Research*, 15, 252-267.

[18] Jackson, J.R. (1963) "Jobshop-Like Queueing Systems," *Management Science*, 10 (1), 131-142.

[19] Liberopoulos, G. and Y. Dallery (2002) "Comparative Modeling of Multi-Stage Production-Inventory Control Policies with Lot Sizing," *International Journal of Production Research*, 41 (6), 1273-1298.

[20] Marie, R. (1979) "An Approximate Analytical Method for General Queueing Networks," *IEEE Transactions on Software Engineering*, 5 (5), 530-538.

[21] Marie, R. (1980) "Calculating Equilibrium Probabilities for $\lambda(n)/C_k/1/N$ Queues," *Performance Evaluation Revue*, 9, 117-125.

[22] Reiser, M. and S.S. Lavenberg (1980) "Mean Value Analysis of Closed Multichain Queueing Networks," *Journal of ACM*, 27 (2), 313-322.

[23] Schweitzer, P.J. (1979) "Approximate Analysis of Multiclass Closed Networks of Queues," *Proceedings of the International Conference on Stochastic Control and Optimization*, Amsterdam.

[24] Spearman, M.L., D.L. Woodruff and W.J. Hopp (1990) "CONWIP: A Pull Alternative to Kanban," *International Journal of Production Research*, 28, 879-894.

[25] Stewart, W.J. and R. Marie, (1980) "A Numerical Solution for the $\lambda(n)/C_k/r/N$ Queue," *European Journal of Operational Research*, 5, 56-68.

[26] Whitt, W. (1983) "The Queueing Network Analyser," *Bell Systems Technology Journal*, 62 (9), 2779-2815.

Appendix A - Analysis of synchronization station O^N

O^N is a synchronization station fed by a continuous-time Markov arrival process with state-dependent arrival rate $\lambda_O^N(n_O^N)$, $0 \le n_O^N < K_N$, and an external Poisson process with rate λ_D. The underlying continuous-time Markov chain is shown in Figure 5. The state of this Markov chain is (n_O^N, n_D), where n_O^N is the number of engaged kanbans and n_D, $n_D \ge 0$, is the number of external resources (customer demands) currently present in subsystem O^N. Let $p_O^N(n_O^N, n_D)$ be the steady-state probabilities of the Markov chain. These probabilities are solution of the following balance equations:

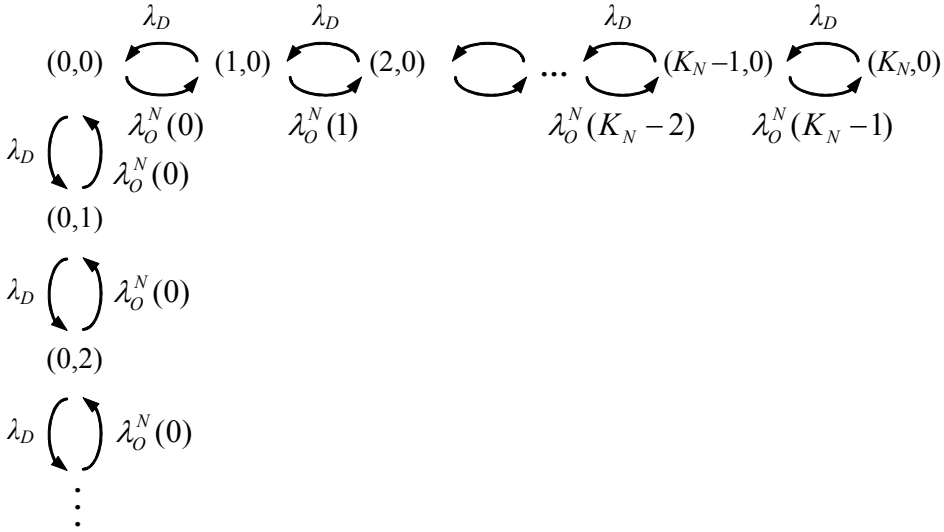

Figure 5: Continuous-time Markov chain describing the state (n_0^N, n_D) of synchronization station O^N.

$$p_O^N(n_O^N,0)\lambda_D = p_O^N(n_O^N-1,0)\lambda_O^N(n_O^N-1) \quad \text{for } n_O^N = 1,...,K_N\,, \tag{14}$$

$$p_O^N(0,n_D)\lambda_O^N(0) = p_O^N(0,n_D-1)\lambda_D \quad \text{for } n_D > 0\,. \tag{15}$$

The marginal probabilities $P_O^N(n_O^N)$ are then simply given by

$$P_O^N(n_O^N) = p_O^N(n_O^N,0) \quad \text{for } n_O^N = 1,...,K_N\,, \tag{16}$$

$$P_O^N(0) = \sum_{n_D=0}^{\infty} p_O^N(0,n_D)\,. \tag{17}$$

From (15) and (17) we get

$$P_O^N(0) = \sum_{n_D=0}^{\infty} p_O^N(0,0)\left(\frac{\lambda_D}{\lambda_O^N(0)}\right)^{n_D} = p_O^N(0,0)\frac{1}{1-\dfrac{\lambda_D}{\lambda_O^N(0)}}\,. \tag{18}$$

The conditional throughputs of subsystem O^N are obtained from (5), (14) and (16), as follows:

$$v_O^N(n_O^N) = \lambda_D \quad \text{for } n_O^N = 2,...,K_N \tag{19}$$

From (5), (14), (16) and (18), we also get

$$v_O^N(1) = \frac{1}{\frac{1}{\lambda_D} - \frac{1}{\lambda_O^N(0)}}. \tag{20}$$

Appendix B - Analysis of synchronization station I^i

I^i, $i = 2,\ldots, N$, is a synchronization station fed by two continuous-time Markov arrival processes with state-dependent arrival rates: $\lambda_I^i(n_I^i)$, $0 \le n_I^i \le K_i$, and $\lambda^i(n^i)$, $0 \le n^i \le K_{i\text{-}1}$. The underlying continuous-time Markov chain is shown in Figure 6. The state of this Markov chain is (n_I^i, n_u^i), where n_I^i is the number of free kanbans and n_u^i is the number of external resources (finished parts of stage i-1) currently present in subsystem I^i. Recall that n^i can be obtained from n_u^i and n_I^i using (3). The steady-state probabilities $p_I^i(n_I^i, n_u^i)$ can be derived as solutions of the underlying balance equations and are given by:

$$p_I^i(n_I^i, 0) = \left[\prod_{n=1}^{n_I^i} \frac{\lambda_I^i(n-1)}{\lambda^i(K_i - n)}\right] p_I^i(0,0), \tag{21}$$

$$p_I^i(0, n_u^i) = \frac{\prod_{n=1}^{n_u^i} \lambda^i(K_i + n - 1)}{\left[\lambda_I^i(0)\right]^{n_u^i}} p_I^i(0,0). \tag{22}$$

The marginal probabilities, $P_I^i(n_I^i)$, can then be derived by summing up the probabilities above as follows:

$$P_I^i(n_I^i) = \left[\prod_{n=1}^{n_I^i} \frac{\lambda_I^i(n-1)}{\lambda^i(K_i - n)}\right] p_I^i(0,0) \text{ for } n_I^i = 1,\ldots, K_i, \tag{23}$$

$$P_I^i(0) = \left[1 + \sum_{n_u^i=1}^{K_{i-1}-K_i} \frac{\prod_{n=1}^{n_u^i} \lambda^i(K_i + n - 1)}{\left[\lambda_I^i(0)\right]^{n_u^i}}\right] p_I^i(0,0). \tag{24}$$

The estimation of the conditional throughputs of subsystem I^i can then be obtained by substituting the above probabilities into (5), as follows:

$$v_I^i(n_I^i) = \lambda^i(K_i - n_I^i) \quad \text{for } n_I^i = 2,\ldots, K_i, \tag{25}$$

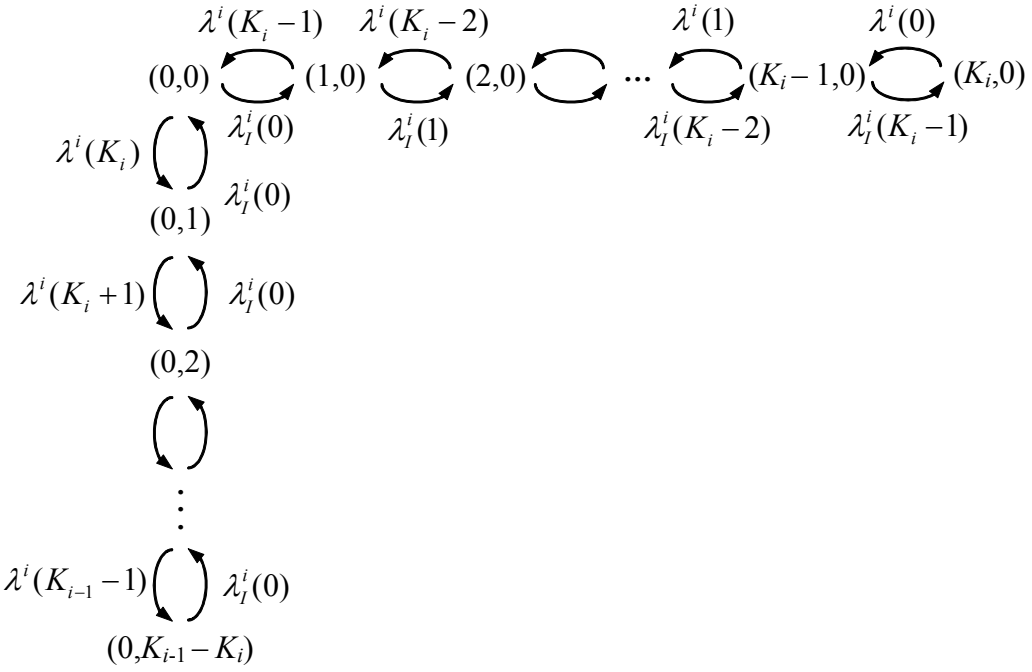

Figure 6: Continuous-time Markov chain describing the state (n_I^i, n_u^i) of queueing network I^i.

$$v_I^i(1) = \lambda^i(K_i-1)\left[1+\sum_{n_u^i=1}^{K_{i-1}-K_i}\frac{\prod_{n=1}^{n_u^i}\lambda^i(K_i+n-1)}{\left[\lambda_I^i(0)\right]^{n_u^i}}\right]. \tag{26}$$

Appendix C – Table of Notation

N	Number of stages
K_i	Number of echelon kanbans of stage i
L_i	Subnetwork associated with the manufacturing process of stage i
m_i	Number of stations of subnetwork L_i
J_i	Synchronization station at the output of stage i
λ_D	Average arrival rate of external customer demands in the unsaturated system
P_r	Maximum rate at which customer demands can be satisfied
R	Queueing network of the echelon kanban control system

R^i Subsystem associated with stage i

I^i Upstream synchronization station of subsystem R^i

O^N Downstream synchronization station of subsystem R^N

$\hat{S}_i$ Downstream single-server pseudo-station of subsystem R^i

n^i State of subsystem R^i

$\lambda^i(n^i)$ State-dependent arrival rate of stage-i raw parts at the upstream synchronization station I^i of subsystem R^i

$v^i(n^i)$ Conditional throughput of subsystem R^i

$k \in M_i$ Index denoting the stations within subsystem R^i, where $M_1 = \{1,\ldots, m_1, \hat{S}_1\}$, $M_i = \{I^i, 1,\ldots, m_i, \hat{S}_i\}$ for $i = 2,\ldots, N\text{-}1$, and $M_N = \{I^N, 1,\ldots, m_N, O^N\}$

n_k^i State of station k in subsystem R^i

$\mu_k^i(n_k^i)$ Load-dependent service rate of station k in subsystem R^i

$\mu_k(n_k)$ Same as $\mu_k^i(n_k^i)$ with index i dropped

T_k^i Open system representing station k in subsystem R^i

T_k Same as T_k^i with index i dropped

$\lambda_k^i(n_k^i)$ Rate of state-dependent Poisson arrival process at T_k^i

$\lambda_k(n_k)$ Same as $\lambda_k^i(n_k^i)$ with index i dropped

$v_k^i(n_k^i)$ Conditional throughput of T_k^i

$v_k(n_k)$ Same as $v_k^i(n_k^i)$ with index i dropped

$P_k^i(n_k^i)$ Steady-state probability of T_k^i

p_B Proportion of backordered demands

Q_D Average number of backordered demands

W_B Average waiting time of backordered demands