A Note on the Application of Marie's Method for Queueing Networks with Batch Servers*

Alexander K. Schömig and Matthias Kahnt

Bayerische Julius–Maximilians Universität Würzburg, Institut für Informatik,
Lehrstuhl für Informatik III, Am Hubland, D-97074 Würzburg, Germany
Tel.: +49-931-8885511, Fax: +49-931-8884601,
e-mail: schoemig@informatik.uni-wuerzburg.de

1 Introduction

During the past years much attention has been devoted to just–in–time (JIT) manufacturing systems as an alternative to push systems, traditionally used by European and U.S.–American manufacturers. One way to implement a JIT–system, also called *pull system*, is the *kanban system*. It limits the WIP in a production line in an efficient way: Kanbans circulate within a production stage and their occurence at specific positions indicate the status of this stage.

Batch servers are machines that can process several jobs at a time. Examples include plating baths, drying facilities, heat–treating ovens, e.g. in semiconductor wafer fabrication diffusion and oxidation ovens. Each of these ovens can process a certain number of lots of wafers simultaneously (cf. Johri 1992).

Our study is motivated by the ongoing discussion on whether and how to implement kanban–like pull production control systems in semiconductor fabrication facilities. In this paper we model and analyze a singlecard kanban system with batch servers in order to investigate the impact of buffer allocation. Finally, we compare using our new algorithm numerical approximations with simulations of several configurations. This study belongs to the category of research on queueing networks with restricted capacity as previously presented by DiMascolo et al. (1991, 1992), and Dallery et al. (1990, 1992).

2 Model and Analysis

2.1 Basic Model

DiMascolo et al. (1992) consider in their model a production line consisting of N stages. We only sketch the approach and refer to the original paper for details. Each stage can be described as a queueing network X_k, $k = 1, 2, \ldots, N$. (See Fig. 1 for a example with three stages.) There are C_k kanbans and m_k machines

* This research was supported by the Deutsche Forschungsgemeinschaft (DFG), grant #Tr 257/2–1.

in stage k. Service times are i.i.d. RVs with mean $(\mu_k^i)^{-1}$, $i = 1, \ldots, m_k$. Finished parts are stored in buffers P_k and free kanbans in buffers F_k.

The flow of jobs from stage to stage is controlled according to the kanban principle. There are synchronizing stations, denoted by J_k, at the end of each stage. If and only if there is a part in P_{k-1} and a kanban in F_k at the synchronizing station J_{k-1}, this part may proceed to stage k. The free kanban is removed from F_k and attached to the part. Next, the part sojourns X_k receiving service and waits in buffer X_k until it can enter stage $k+1$. Upon leaving stage k, its kanban is removed and returned to buffer F_k.

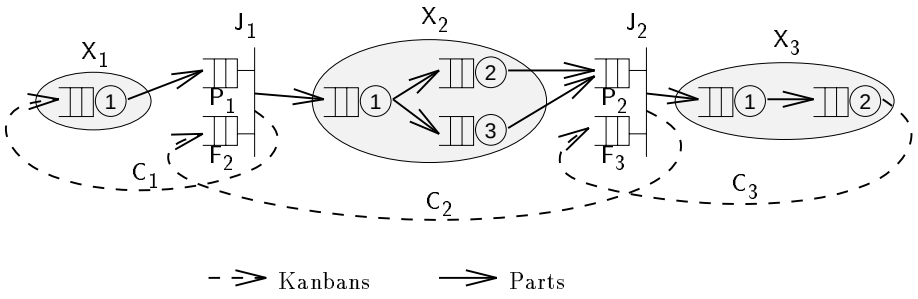

Fig. 1. *Queueing net representation of a kanban system*

For the sake of the reduction of complexity we now focus on the queueing net subsystems Y_k $(k = 1, 2, \ldots, N)$. Subsystems Y_k consists of J_{k-1}, X_k, and J_k. We assume that parts and demands for finished parts, i.e. free kanbans, arrive at Y_k according to state–dependent Poisson processes with rates $\lambda_k^u(n_k^u)$, $0 \le n_k^u \le C_{k-1}$, and $\lambda_k^d(n_k^d)$, $0 \le n_k^d \le C_{k+1}$. The flow of parts and kanbans within Y_k is exactly the same as described above. If we consider kanbans as jobs and parts as external resources, we can treat subsystem Y_k as a closed queueing network. Hence, Y_k can be analyzed approximatively using Marie's Method (1979, 1980). As a result we can calculate the arrival rates λ_k^I and λ_k^O given all parameters.

We now turn back to the overall system by looking at the synchronization stations: Note, that J_k is part of Y_{k-1} *and* Y_k. Subsystem Y_k is fed by Y_{k-1} with parts, subsystem Y_{k+1} demands finished parts. Consequently, the following equations can be derived:

$$\lambda_k^u(n) = \lambda_{k-1}^O(n), \qquad n = 0, 1, \ldots, C_{k-1} - 1, \quad k = 2, 3, \ldots, N \tag{1}$$

$$\lambda_k^d(n) = \lambda_{k+1}^I(n), \qquad n = 0, 1, \ldots, C_{k+1} - 1, \quad k = 1, 2, \ldots, N-1 \ . \tag{2}$$

It is clear that the quantities λ_k^u, $k = 1, 2, \ldots, N-1$, can be obtained by the analysis of subsystems Y_k for $k = 2, 3, \ldots, N$ and quantities λ_k^d of the stages $N, N-1, \ldots, 1$ by analyzing subsystems Y_k for $1, 2, \ldots, N-1$. An iterative procedure can be found that calculates these quantities efficiently. Again, we refer the reader to the original paper for details.

2.2 Adding Batch Servers

In order to analyze a kanban system with batch servers according to the procedure introduced above, we have to adapt Marie's Method. We consider batch servers with batch size $B_k^i = b$ with exponentially distributed service times and mean service rate μ_k^i. We derive the state transition diagram as depicted in Fig. 2 using the notion of *reassembly–units* introduced by Shapiro and Perros (1993). A reassembly–unit assembles parts until a full batch can be formed and queued for service. Only complete batches are served with a mean service rate of μ_k^i. A batch service station can hold at most $w = C_k/b$ batches. The state of a batch service station is defined as (S_t, R_t), $0 \leq S_t \leq w$ and $0 \leq R_t \leq b-1$, where S_t is the number of full batches in the batch service station and R_t the number of parts in the reassembly–unit at time t.

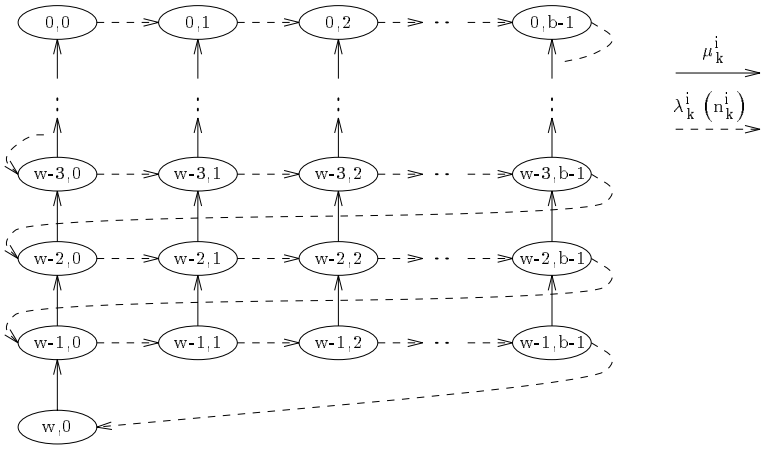

Fig. 2. *Batch service station with* $B_k^i = b$ *and* $C_k = wb$

This state space is equivalent to the state space of an *Erlangian–servers* with b phases, queue length w, and state–dependent arrival rate $\tilde{\lambda}(\tilde{s})$. We denote the state space this Erlangian–server by $(\tilde{s}, \tilde{r})$ and service rate by $\tilde{\mu}_{\tilde{r}}$. These quantities are derived from our original notation using:

$$\tilde{s} \longmapsto s = w - \tilde{s} \ , \tag{3}$$

$$\tilde{r} \longmapsto r = \tilde{r} - 1 \ , \tag{4}$$

and

$$\tilde{\lambda}(\tilde{s}) \longmapsto \mu_k^i \ , \tag{5}$$

$$\tilde{\mu}_{\tilde{r}} \longmapsto \lambda_k^i\left(n_k^i\right) \ . \tag{6}$$

Further, we shall say that

$$n_k^i = sb + r = (w - \tilde{s})b + \tilde{r} - 1 \ . \tag{7}$$

2.3 Adaption of Marie's Method

Using the analogies derived above, we can utilize Marie's approach to analyze Cox–servers (cf. Marie 1980). Unfortunately, we cannot use his equations directly, since they are given for identical distributed service phases only. Equation (6) requires *state–dependent* service rates. Thus we have to adapt Marie's equations. We refer to equations in (Marie 1980) as {nr}.
Marie uses coefficients $c(\tilde{s},\tilde{r}) = p_{\tilde{s},\tilde{r}}/p_{0,0}$, where $p_{\tilde{s},\tilde{r}}$ are the state probabilities for $(\tilde{s},\tilde{r})$. State dependent throughputs $\nu(\tilde{s})$ have to be calculated for all $\tilde{s}$ according to Eqn. {33}:

$$\nu(\tilde{s}) = \lambda(\tilde{s}-1)\,\frac{C(\tilde{s}-1)}{C(\tilde{s})} \quad , \qquad \tilde{s} = 1,2,\ldots,w$$

Coefficients $C(\tilde{s})$ can be calculated as the sum of all $c(\tilde{s},\tilde{r})$, for phases $\tilde{r}$ $(\tilde{r} = 1,2,\ldots,b)$ with a fixed number of jobs $\tilde{s}$ $(\tilde{s} = 1,2,\ldots,w)$. Now we shall calculate $\nu_k^i\,(n_k^i)$ given populations $n_k^i = 1,2,\ldots,C_k$. For each state (s,r) there is a different number of parts present at the station according to Eqn. 7. Hence, we define coefficients $c(w,r)$ as

$$c(w,r) \;=\; \begin{cases} 1, & r = 0 \\ 0, & r = 1,2,\ldots,b-1 \end{cases} , \tag{8}$$

since we have $c(w,0) = p_{w,0}/p_{w,0} = 1$. Note, that $p_{w,r} = 0$, because states (w,r), $r > 0$ are not possible according to our model description.
After having calculated $c(w,r)$, we find $c(s,r)$, and $c(w-1,r)$ using Eqn. {31}:

$$\begin{aligned} c(s,r) = {} & g(s,r)\cdot c(s,b-1) - h(s,r)\cdot c(s+1,b-1) \\ & \text{with } s = w-1,w-2,\ldots,0 \text{ and } r = 0,1,\ldots,b-1 \quad . \end{aligned}$$

We now are able to calculate $\nu_k^i\,(n_k^i)$ using Eqn. {33} in a slightly different form:

$$\nu_k^i\,(n_k^i) = \lambda_k^i\,(n_k^i-1)\,\frac{c\left((n_k^i-1)\ \mathit{DIV}\ B_k^i,\ (n_k^i-1)\ \mathit{MOD}\ B_k^i\right)}{c\left(n_k^i\ \mathit{DIV}\ B_k^i,\ n_k^i\ \mathit{MOD}\ B_k^i\right)} \quad , \tag{9}$$

$$\text{with} \quad n_k^i = 1,2,\ldots,C_k \quad .$$

3 Numerical Results

In the following we compare analytical results with simulations. We observed two performance measures: System throughput ('Thp') and WIP. For validation purposes, 10 independent replications of a simulation run for each parameter set of the system under consideration were done. Each run had a total length of 10,000 time units and statistical data from the initial transient period were discarded. The simulations were coded in MODULA–2 and were run on SUN SPARC workstations. The columns containing the results from these simulation runs also show the lower and upper bound of the 95% confidence interval of the mean value.

From the multitude of possible configurations and parameter sets we choose two exemplary systems. Each system was investigated for 6 and 12 kanbans within a stage. The results are summarized in Tab. 3. For all calculations presented here, we assume exponentially distributed service times, if not mentioned otherwise.

The first system ('A') consists of three stages, where the middle stage includes a network of machines as shown in Fig. 3. System parameters for machine i in stage j are denoted as defined in Sec. 2. Additionally we introduce cv_j^i as the coefficient of variation and q_{i_1,i_2} as the routing probability. Parameters for stages 1 and 3 are $\mu_j^1 = 1.0$ and $B_j^1 = 1, \quad j \in \{1, 3\}$.

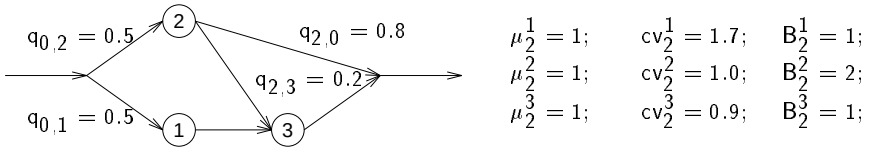

Fig. 3. *System A: Batch service at station # 2*

Figure 4 illustrates the second system. It consists of 9 machines grouped into 6 stages. Mean service rates of all machines except the batch servers are $\mu = 5.0$. The mean service rates of the batch servers are given for case 1 as $\mu_2^2 = 3.0$ and $\mu_5^1 = 2.0$, and for case 2 as $\mu_2^2 = 2.0$ and $\mu_5^1 = 1.0$, respectively. The latter case implies that the batch servers are the bottlenecks of this production line.

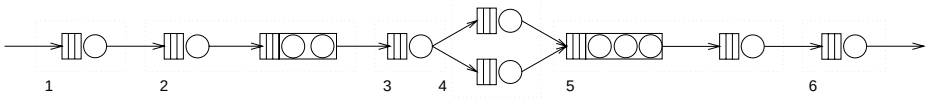

Fig. 4. *System B: Batch service stations in stage # 2 (B=2) and # 5 (B=3)*

The results show for all considered configurations acceptable low relative error margins. Smallest errors are found for system A with 12 kanbans within each stage. We observe the largest deviation of the analytical results from the simulation when the batch service machines are the bottlenecks (System B, Case 2). We explain this effect by the fact that in this case the batch server determinates the overall system performance by a great extent. Note, that the weighted service rate of the bottleneck batch server equals 3.0.
It is clear from our results that the number of kanbans assigned to each stage are crucial for the performance for these systems. This finding is consistent with previous reserach for single server systems. However, further research has to be carried out to investigate how alternative kanban assignments may improve the system performance.

Summing up, we conclude that our approach is well suited for the preformance analysis of queueing network models of manufacturing systems with batch servers.

Table 1. Summary of Simulation and Analytical Results

			Analysis	Simulation			rel. Error
System **A**	6	Thp	0.945	0.921	0.927	0.932	1.9 %
		WIP	14.64	14.72	14.87	15.03	-1.4 %
	12	Thp	0.973	0.967	0.972	0.976	0.1 %
		WIP	29.38	28.50	29.40	30.29	-0.1 %
System **B**, Case 1	6	Thp	3.768	3.633	3.642	3.652	3.5 %
		WIP	28.21	29.43	29.51	29.60	-4.4 %
	12	Thp	4.405	4.382	4.390	4.399	0.3 %
		WIP	56.52	58.90	59.15	59.41	-4.4 %
System **B**, Case 2	6	Thp	2.552	2.507	2.520	2.533	1.3 %
		WIP	29.26	29.74	29.80	29.85	-1.8 %
	12	Thp	2.710	2.899	2.914	2.928	7.0 %
		WIP	59.00	60.43	60.58	60.74	-2.6 %

References

P. K. Johri (1992), "Practical issues in scheduling and dispatching in semiconductor wafer fabrication," *Journal of Manufacturing Systems*, vol. 12, no. 6, pp. 474–485.

M. Di Mascolo, Y. Frein, and Y. Dallery (1992), "Queueing network modeling and analysis of kanban systems," in *Third International Conference On Computer Intergrated Manufacturing*, IEEE / Rensselaer Polytechnic Institute, IEEE Computer Society Press, May 20–22 1992.

M. Di Mascolo, Y. Frein, Y. Dallery, and R. David (1991), "A unified modeling of kanban systems using Petri nets," *The International Journal of Flexible Manufacturing Systems*, vol. 3, pp. 275–307.

Y. Dallery and X.-R. Cao (1992), "Operational analysis of stochastic closed queueing networks," *Performance Evaluation*, vol. 14, pp. 43–61.

Y. Dallery (1990), "Approximate analysis of general open queueing networks," *Performance Evaluation*, vol. 11, pp. 209–222, 1990.

R. A. Marie (1979), "An approximate analytical method for general queueing networks," *IEEE Trans. on Software Engineering*, vol. SE-5, pp. 530–538, Sep. 1979.

R. A. Marie (1980), "Calculating equilibrium probabilities for $\lambda(n)/c_k/1/n$ queues," *ACM Sigmetrics Performance Evaluation Review*, vol. 9, no. 2, pp. 117–125, 1980.

G. W. Shapiro and H. G. Perros (1993), "Nested sliding window protocols with packet fragmentation," *IEEE Trans. on Communication*, vol. 41, pp. 99–110, Jan. 1993.