Seoul Journal of Business
Volume 10, Number 1 (June 2004)

Benefit of Flexibility in case of Machine Failures*

Ick-Hyun Nam**
College of Business Administration
Seoul National University

Abstract

Flexibility in a queueing system is known to offer faster response time. Two types of flexibility, parallel and serial flexibility, are studied in this paper. As an extension, we deal with the case where there occur server failures. And we show the improvement in waiting time by having flexibility in case of server failures. In deriving the improvement of flexibility, we apply the heavy traffic approximatin method.

(Keywords: flexibility, queueing, machine failure, heavy traffic)

1. Introduction

When we want to model a system where there is stochastic variability, we usually use a queueing system. In a queueing system, we handle random customer arrivals and random service times. Service times are considered to be a random variable in a queueing system. In addition to the stochastic variability in service times, there can happen various random impacts. As one of those impacts, server or machine breakdown can affect a queueing system. In this paper, we consider a queueing system where the server sometimes breaks down. Machine failures are said to occur when the server breaks down

[*] This paper was financially supported by the Institute of Management Research, College of Business Administration, Seoul National University.

[**] Associate Professor of Production and Management Science, Seoul National University(ihnam@snu.ac.kr).

and cannot process customers. As one way to handle the machine breakdown, we can adjust the mean and the variance of service times such that the effective mean and variance are derived incorporating machine failures. But this method is not accurate other than the first and the second moments.

In this paper, we would like to handle the machine failures more directly. We first deal with the modeling technique such that machine failures are considered as a customer class with priority. And then we show the benefit of flexibility in servers where there are machine breakdowns. Since the models we consider are difficult to handle in an exact form, we introduce heavy traffic approximation method which is widely applicable.

2. Heavy Traffic Approximation

When we try to model a general queueing network, it is very difficult to derive a closed form solution. Therefore, it is usually recommended to use an approximation method for modeling a queueing network. One of those approximation methods is the heavy traffic approximation. In the heavy traffic approximation, we use Brownian motion as in [Harrison 1985] under the heavy traffic condition. The heavy traffic condition means that the traffic intensity in a queueing system is approximately one. That is, to apply the heavy traffic approximation, we require the heavy traffic condition, $\rho \approx 1$, where ρ represents traffic intensity. Although the heavy traffic approximation is a powerful modeling technique, the problem lies in the fact that the heavy traffic condition may not be easy to satisfy in general.

We now look at the heavy traffic condition in a processing system which is prone to machine failures. In several cases, the traffic intensity for the customers is far below 1, so we cannot apply heavy traffic approximation. But in some cases, we achieve the necessary heavy traffic condition when we incorporate machine failures. Let us denote type 1 customer as machine failure customers, which represents machine breakdowns. Type 0 customer is a real customer which needs service. For the heavy traffic condition, we do not need

$$\lambda_0 m_0 \approx 1,$$

where λ_i and $m_i(i = 0, 1)$ are arrival rate and mean service time of customers of type i.

Instead we only need

$$\lambda_1 m_1 + \lambda_0 m_0 \approx 1,$$

that is, the system traffic intensity including the machine failure customer is approximately one. Thus in some cases the system traffic intensity may satisfy the heavy traffic condition even though the real customer type alone does not.

3. Priority Scheme

A queueing discipline is a means of choosing which customer in the queue is to be served next. This decision may be based on any or all of the following:

a. some measure related to the relative arrival times for those customers in the queue;

b. some measure of the service time required or the service so far received;

c. some function of group membership.

We call the third case as a priority queueing discipline. Examples of queueing disciplines that depend only upon arrival time are first-come-first-serve(FCFS), last-come-first-serve (LCFS), and random order of service. Discrimination based on service time only may take the following forms: shortest-job-first, longest-job-first, similar rules based on averages, and so on. In Section 5, we will consider the queueing discipline where we give higher priority to the customer class with shorter expected service time. Order of service based on an externally imposed priority class structure may take many forms as, for example, the head-of-the-line system.

We assume that arrival customers belong to one of a set of N different priority classes, indexed by the subscript n ($n = 1, 2, \ldots, N$). We take the assumption that the smaller the value of the index associated with the priority group, the higher is the priority associated with that group; that is, customers from priority group n are given preferential treatment in one form or another on the average over customers from priority group $n + 1$.

We assume that an arriving customer is assigned a set of parameters that determine his relative position in the queue through the decision rule known as the queueing discipline. This position may vary as a function of time owing to the appearance of customers of higher priority in the queue.

If a customer in the process of being served is liable to be ejected from service whenever a customer with a higher priority appears in the queue and returns to the queue afterwards, then we say that the system is a preemptive priority queueing system. If such ejection is not allowed, the system is said to be non-preemptive. If only one customer is allowed in the server at a time, then when there exists a tie between customers, the tie is broken on a first-come-first-serve basis. In the preemptive priority queueing system, we have to consider an additional complexity regarding how a customer recovers when he reenters service after having been preempted. Three cases are usually identified. The first, where a customer picks up from where he left off, is known as preemptive resume. The second and third cases assume that the customer loses credit for all service he has so far received: the second case assumes that a returning customer starts from scratch but with the same total service time requirement as he had upon his earlier visit, and this is known as preemptive repeat without resampling; the third case assumes that a new service time is chosen for our reentering customer and is referred to as preemptive repeat with resampling.

Using the priority scheme, we can deal with the machine failures in a queueing system. We can represent a machine failure as a phantom customer to the corresponding server. The service time of a machine failure is the time required to serve the phantom customer. This is the machine down time, which equals to the machine repair time since a machine is down while it is under repair. We can use the priority scheme in modeling the machine failure. The failure customer is then said to have preemptive highest priority over the other classes of real customers. When a machine breaks down, we can depict the phenomenon as the case where a failure customer arrives into the queueing system. Since a machine failure does not wait until the regular job under processing is completed, a failure customer is said to have preemptive priority.

4. Parallel Flexibility

When there is flexibility in servers, we can derive the benefit of reducing mean throughput time. In Nam[2000], we show the resource pooling effect in a flexible queueing system. Flexibility in a queueing system is defined as the case where servers can process more than one types of customers. Flexibility in servers allows the queueing system administrator to route multiple types of customers such that system idleness is minimized. This resource pooling effect is rather huge in shortening the waiting time of customers. In addition to the resource pooling effect, we can derive some more benefit in a serial queueing system under the flexibility. In a serial queueing system with flexibility, we can have the option of scheduling jobs. By letting jobs near the stage of completion have the higher priority, we can reduce the mean throughput time more.

In this paper, we note that the benefit of flexibility is still valid even in case of machine failures. When we construct a flexible processing system where a type of customers can be served at any one of the multiple servers, we can route a flexible customer to one working machine that is idle when another machine breaks down. We can achieve the resource pooling effect from the fact that in a flexible processing system the connected machines can help each other when some machines are unavailable either because they are down or overloaded. For inflexible systems, even when a machine is down, the dedicated customers cannot be routed to and served by the other working but idle station. In this paper, we deal with two kinds of flexibility: parallel and serial flexibility.

In this section, we consider the parallel flexibility. We now consider the following three queueing systems. In System (i), we have two independent queueing systems. Independency means that each queueing system has its own customer inflows which come from independent distributions and thus are separate. In each queueing system, there are two classes of customers. We denote machine failures as customer class 1 and regular

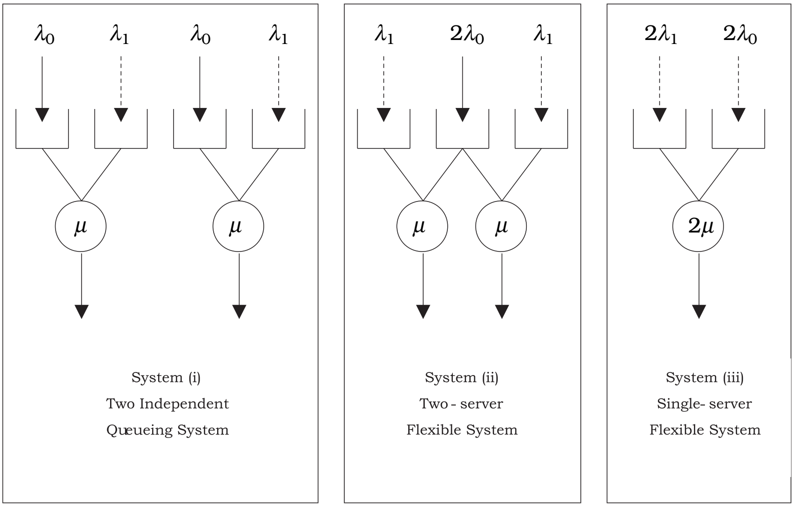


Figure 1.

customers as customer class 0. When we incorporate flexibility into servers such that both servers can help each other in processing regular customers, we get the queueing systems as in System (ii). When we merge the capacity of two servers, we have the queueing system as in System (iii). In System (iii), we have one server whose capacity is twice big as the regular capacity.

We will use the following result in deriving our argument. By extending Loulou[1974] to our multi-customer class models, we have the following proposition.

Proposition 0:

In the heavy traffic limit, the multi-server system and the single server system with correspondingly large service rate have the same unfinished workload and throughput time processes in probability.

[Proposition 0] implies that, in the heavy traffic limit, System (ii) and System (iii) have the same unfinished workload process. And we use System (iii) for our analysis since it offers closed form solution of mean throughput time of regular customer class. In order to compare System (i) with System (iii), we derive the mean throughput time of customer class 0 of each system as follows.

Throughout this paper, we assume the followings for analytical simplicity. We first assume that the inter-arrival time and the service time of each customer follow exponential distributions. Two independent real customers in system (i) have the same inter-arrival time and service time distributions. Also two independent failure customers in system (i) have the same inter-arrival time and service time distributions. In general, two independent customers have distinct inter-arrival time and service time distributions. But this homogeneity assumption makes the formula for mean throughput time of a real customer simple. When we relax this assumption, the ratio of benefit may be a little more complex to derive and different, but the significant improvement should still be valid. Using the formula in page: 125 [Kleinrock 1976, Vol. II], we get the following results. The mean throughput time of regular customers in System (i) is

$$W_1 = \frac{\dfrac{1-\rho_0-\rho_1}{\mu_0} + \dfrac{\lambda_0}{\mu_0^2} + \dfrac{\lambda_1}{\mu_1^2}}{(1-\rho-\rho)(1-\rho)} = \frac{\dfrac{1-\rho_1}{\mu_0} + \dfrac{\rho_1}{\mu_1}}{(1-\rho_0-\rho_1)(1-\rho_1)}.$$

And the mean throughput time of regular customers in System (iii) is

$$W_3 = \frac{\dfrac{1-\rho_0-\rho_1}{2\mu_0} + \dfrac{\lambda_0}{2\mu_0^2} + \dfrac{\lambda_1}{2\mu_1^2}}{(1-\rho_0-\rho_1)(1-\rho_1)} = \frac{\dfrac{1-\rho_1}{2\mu_0} + \dfrac{\rho_1}{2\mu_1}}{(1-\rho_0-\rho_1)(1-\rho_1)}.$$

By comparing those two results, we get the following result.

Proposition 1:

In a two-server parallel queueing system with server breakdowns, we can reduce the mean throughput time into a half when we introduce flexibility among the servers. That is,

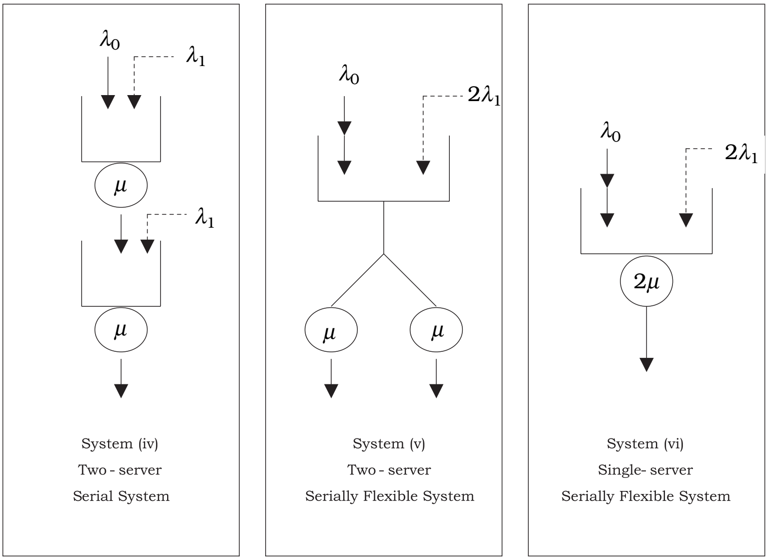
$$\frac{W_1}{W_3} = 2.$$

[Proposition 1] says that we have the same benefit of flexibility as shown in [Nam 2000] even in case where there are machine failures which interrupt processing customers. And ratio of this benefit is the number of servers connected by flexibility.

5. Serial Flexibility

In this section, we deal with a serial queueing system where two queueing systems are attached sequentially. A typical serial queueing system is shown as System (iv). As before, class 1 customers are machine failures which have preemptive high priority. When we introduce flexibility such that both servers can process either the first stage or the second stage jobs, we say that we have serial flexibility. We depict the serial queueing system with flexibility in System (v). We note that the regular customers need to have two stages of service, which is represented as two consecutive arrows. As previously, we introduce a single server flexible system as in System (vi) in order to get closed form solution.

The mean throughput time of a customer in System (iv) is approximately $2W_1$ since an incoming customer needs two



System (iv)
Two - server
Serial System

System (v)
Two - server
Serially Flexible System

System (vi)
Single- server
Serially Flexible System

Figure 2.

stages of service. That is,

$$W_4 \approx 2W_1.$$

In System (vi), we apply the *cμ* rule in scheduling such that the customers waiting for the second stage have high priority. In this case, *cμ* rule implies that the customer class with shorter expected time till completion has higher priority. This implies that the customers get two stages of service consecutively. Using the following notations, we derive the mean throughput time of System (vi).

$$\sigma_0 = \rho_0 + \rho_1 = \frac{\lambda_0}{\mu_0} + \frac{\lambda_1}{\mu_1}, \quad \sigma_1 = \rho_1,$$
$$\bar{x}_0 = 1/\mu_0, \quad \overline{x_0^2} = 3/2\mu_0^2$$
$$\bar{x}_1 = 1/2\mu_1, \quad \overline{x_1^2} = 1/2\mu_1^2,$$

where $\bar{x}$ is the mean service time and $\overline{x^2}$ is the second moments of service time.

Using the same formula in [Kleinrock 1976, vol. II], we get

$$W_6 = \frac{\dfrac{1-\rho_0-\rho_1}{\mu_0} + \dfrac{3\rho_0}{4\mu_0} + \dfrac{\rho_1}{2\mu_1}}{(1-\rho_0-\rho_1)(1-\rho_1)}$$

Proposition 2:

In a two-station serial queueing system, flexibility among the servers induce more than two times reduction in mean throughput time. That is,

$$\frac{W_4}{W_6} > 2 \quad .$$

The proof of [Proposition 2] comes trivially from the inequality, $\frac{\rho_1}{\mu_1} + \frac{\rho_0}{2\mu_0} > 0$. This proposition says that the serial flexibility offers more benefit than the parallel flexibility. And the additional benefit comes from the fact that the serial flexibility allows sequencing among jobs according to *cμ* rule.

We now denote the improvement ratio $r \equiv \frac{W_4}{W_6}$ and analyze the benefit from the serial flexibility. We can easily derive

$$r = \frac{2(\mu_1 - \lambda_1 + \frac{\lambda_1 \mu_0}{\mu_1})}{\mu_1 - \lambda_1 + \frac{\lambda_1 \mu_0}{2\mu_1} - \frac{\mu_1 \lambda_0}{4\mu_0}}.$$

We can extend [Proposition 2] for the n-station serial queueing system. For the n-station serial queueing system, we get the mean throughput time for System (vi) as follows:

$$W_6 = \frac{\frac{1}{\mu_0}(1 - \rho_0 - \rho_1) + \frac{n+1}{2n\mu_0}\rho_0 + \frac{1}{n\mu_1}\rho_1}{(1 - \rho_0 - \rho_1)(1 - \rho_1)}$$

Thus we get the improvement ratio $r(n)$ as follows:

$$r(n) = \frac{n[\frac{1}{\mu_0}(1 - \rho_1) + \frac{1}{\mu_1}\rho_1]}{\frac{1}{\mu_0}(1 - \rho_0 - \rho_1) + \frac{n+1}{2n\mu_0}\rho_0 + \frac{1}{n\mu_1}\rho_1}$$

Proposition 3:

For the n-station serial queueing system, flexibility among the servers induce $r(n)$ times reduction in mean throughput time of a customer, where

$r(n) > n.$

And we now do sensitivity analysis of r as λ_0 and μ_0 increases for the two-station serial queueing system.

Proposition 4:

As λ_0 increases up to $\bar{\mu_0}$, the ratio of serial flexibility, r, increases also.

Proposition 5:

As μ_0 increases from λ_0^+, the ratio of r decreases and then increases.

Proof:

We derive the partial derivative of r with respect to μ_0.

$$\frac{\partial r}{\partial \mu_0} = \frac{(\mu_1 - \lambda_1 + \frac{\lambda_1 \mu_0}{2\mu_1} - \frac{\lambda_0 \mu_1}{4\mu_0})\frac{\lambda_1}{\mu_1} - (\mu_1 - \lambda_1 + \frac{\lambda_1 \mu_0}{\mu_1})(\frac{\lambda_1}{2\mu_1} + \frac{\lambda_0 \mu_1}{4\mu_0^2})}{0.5(\mu_1 - \lambda_1 + \frac{\lambda_1 \mu_0}{2\mu_1} - \frac{\lambda_0 \mu_1}{4\mu_0})^2}$$

$$= \frac{\frac{(\mu_1 - \lambda_1)\rho_1}{2} - \frac{\lambda_0 \mu_1}{2\mu_0}\rho_1 - (\mu_1 - \lambda_1)\frac{\lambda_0 \mu_1}{4\mu_0^2}}{0.5(\mu_1 - \lambda_1 + \frac{\lambda_1 \mu_0}{2\mu_1} - \frac{\lambda_0 \mu_1}{4\mu_0})^2}.$$

Let's denote the nominator of the last equation as $A(\mu_0)$:

$$A(\mu_0) \equiv \frac{(\mu_1 - \lambda_1)\rho_1}{2} - \frac{\lambda_0 \mu_1}{2\mu_0}\rho_1 - (\mu_1 - \lambda_1)\frac{\lambda_0 \mu_1}{4\mu_0^2}.$$

The sign of $\partial r / \partial \mu_0$ is identical to that of A or equivalently $4\mu_0^2 A(\mu_0)$ since the denominator of $\partial r / \partial \mu_0$ is positive. We have

$$4\mu_0^2 A(\mu_0) = 2\lambda_1(1 - \frac{\lambda_1}{\mu_1})\mu_0^2 - 2\lambda_0\lambda_1\mu_0 + \mu_1\lambda_0(\lambda_1 - \mu_1)$$

We derive $4\mu_0^2 A(\mu_0 = \lambda_0)$.

$$4\mu_0^2 A(\mu_0 = \lambda_0) = 2\lambda_1(1 - \frac{\lambda_1}{\mu_1})\lambda_0^2 - 2\lambda_0^2\lambda_1 + \lambda_1\mu_1\lambda_0 - \mu_1^2\lambda_0$$

$$= \frac{\lambda_0}{\mu_1}(\lambda_1\mu_1^2 - \mu_1^3 - 2\lambda_1^2\lambda_0).$$

Since $\lambda_1\mu_1^2 - \mu_1^3 - 2\lambda_1^2\lambda_0 < 0$ for $\mu_1 > \lambda_1$, we note that $A < 0$ at $\mu_0 = \lambda_0$. That is,

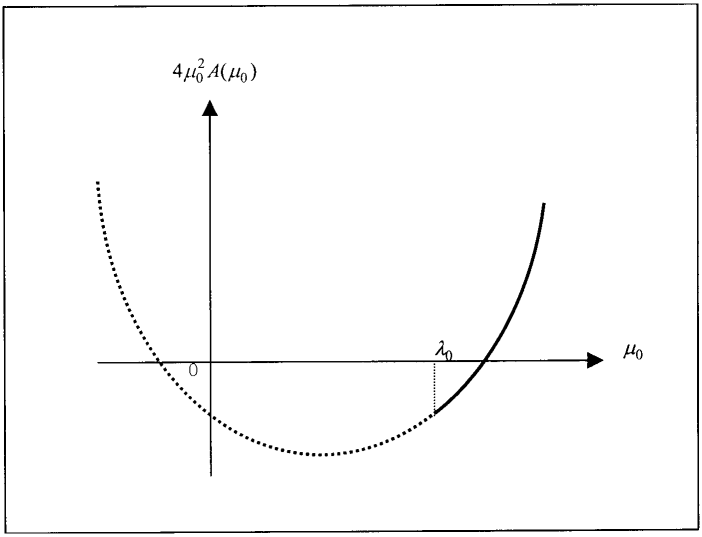


Figure 3.

$$\frac{\partial r}{\partial \mu_0} < 0 \quad \text{at} \quad \mu_0 = \lambda_0.$$

Thus the graph of $4\mu_0^2 A(\mu_0)$ is of the form as in Figure 3.

Thus the sign of $\partial r/\partial \mu_0$ starts from negative and increases to positive as μ_0 increases from λ_0^+. Q.E.D.

This proposition says that the ratio of serial flexibility is rather large when traffic intensity of regular customers is big or small.

7. Conclusion

In this paper, we dealt with a queueing system in which machine failures can occur. When the server in a queueing system is a person, a sudden disruption like absenteeism is the same as the machine failure. In order to model the machine failures, we considered a preemptive priority scheme. And then we derived the benefit of flexibility among servers in a queueing system with machine or server failures. In deriving the benefit of flexibility, we used the heavy traffic approximation to get the closed form solution. Even though the heavy traffic approximation is not exact and thus the improvement ratio in this paper is not accurate, the improvement should be significant.

In showing the benefit of flexibility in a queueing system with machine failures, we considered two types of flexibility. The improvement from flexibility in a parallel queueing system comes mainly from the resource pooling effect. However as in the case where machine failures happen, the resource pooling can be affected since a customer routed to an idle server can be preemptied by a machine failure customer. But it turns out that for a parallel flexible queueing system with machine failures, the improvement ratio is the same as that in a flexible system without machine failures. Thus even in case of machine failures, we can accomplish the same benefit of flexibility in servers as before.

We have another type of flexibility in a queueing system, which is represented as a serial queue. In a serial queueing system, the improvement ratio of flexibility is different from the case without machine failures. We note that the serial flexibility gives us more benefit than the parallel flexibility since we can have sequencing option for customers. By giving the customers nearer to the last stage higher priority in service, we can derive more reduction in mean throughput time. We should note that, in a parallel queueing system, the benefit of flexibility comes from the resource pooling effect. Here resource pooling means that when a machine failure occurs the other working machine can process the customers waiting for the broken-down machine. In analyzing the benefit of flexibility, we should note the additional cost required to have the corresponding flexibility. In order to have a kind of flexibility in servers, it usually requires more investment than non-flexible system. And thus we should compare the cost and the benefit of incorporating flexibility.

We note here some more details which impair the accuracy of heavy traffic approximation. In our modeling technique, we can have more than one machine failures simultaneously. This phenomenon does not describe the actual machine breakdown. And in a serial queueing system, the arrival process to the second stage queueing system is not necessarily Poisson when we allow machine failures. But as mentioned before, the purpose of this paper is not to derive the exact improvement ratio, but to show the significant benefit of flexibility in a queueing system with server failures.

References

Gross, D. and Harris, Carl M. (1998). *Fundamentals of Queueing Theory*, John Wiley & Sons.

Harrison, J.M. (1985). Brownian Models of Queueing Networks with Heterogeneous Customer Populations, *Stochastic Differential Systems, Stochastic Control Theory and Applications*, Springer-Verlag, volume 10, 147-186.

Loulou, R. (1974). On the extension of congestion theorems to multi-channel systems, *Lecture Notes in Economics and Mathematical Systems*, edited by A. B. Clarke, pp.185-198.

Madu, C.N. (1988). A Closed Queueing Maintenance Network with Two Repair Centres, *Journal of Operations Research Society* 39, 959-967.

Nam, I.H. (2000). Improving linear processing system via flexibility, *International Journal of Production Research* 38, 341-352.

Wang, K.H. and Sivazlian, B.D. (1990). Comparative Analysis for the G/G/R Machine Repair Problem, *Computers Industrial Engineering* 18, 511-520.