Modeling Machine Failures in a Queueing System

Ick-Hyun Nam*

《目 次》

Ⅰ. 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 occur another random impacts. As one of those impacts, machine break down 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 and cannot process customers. As one way to handle the machine break down, we can adjust the mean and the variance of service times such that the effective mean and variance be identical. But this method is not accurate other than the first and the second moments. We would like to handle the machine failure more directly.

* College of Business Administration Seoul National University.

Ⅱ. 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 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, $\rho \approx 1$. Although the heavy traffic approximation is a powerful modeling technique, the problem lies in the fact that the heavy traffic condition is not 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 a machine failure customer, which represents machine break-downs. Type 2 customer is a real customer which needs service. For the heavy traffic condition, we do not need

$$\lambda_2 m_2 \approx 1,$$

where λ and m are arrival rate and mean service time respectively.
Instead we only need

$$\lambda_1 m_1 + \lambda_2 m_2 \approx 1,$$

which represents the system traffic intensity including the machine failure customer. Thus in some cases the system traffic intensity may satisfy the heavy traffic condition even though the real customer type alone does not.

Ⅲ. Priority Scheme

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

a. measures related to the relative arrival times for those customers in the queue;
b. measures of the service time required or the service so far received;
c. 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. 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, ···, 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 and returned to the queue whenever a customer with a higher priority appears in the queue, 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.

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 failure 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.

Ⅳ. Three Types of Machine Failures

A machine or a server can be either in a state of up or in a state of down. The up time of a machine can be divided between idle time and busy time. A machine failure's inter-arrival time can be a random variable of several types. We will categorize machine failures according to the inter-arrival time of machine failure customer. If a machine failure's inter-arrival time comes from a distribution that considers the total time, i.e. the inter-arrival time is an

independent random variable, then there can be more than one machine failure customers at a time. This phenomenon can contradict the real world situation. We will call this kind of machine failure as a type 1 machine failure. In this case, the consecutive service time for failure customers, which is busy time to repair them, is the effective down time due to failure. If the machine failure is up time driven, that is, if the inter arrival time of the failure customer comes from a distribution that considers the machine up time, then a failure cannot occur while the machine is down and under repair(while the previous machine failure customer is in service). This is called as a type 2 machine failure. This implies that, in type 2 failures, we cannot have more than one failure customer at a time. In case where the machine failure's inter-arrival time is usage driven, that is, it depends on the time spent for actual service given, we will call the failure as type 3.

For other kinds of machine failures, we can approximate the effective means and variances of the inter-failure times and machine repair times, and then apply the traditional queueing method. To clearly portray the difference among the three kinds of machine failure, we can use the following analogy. A distribution function for a machine failure gives the inter-arrival time x of a machine failure customer. A clock is reset and starts from 0 at the instant a previous failure customer is generated. When the clock reaches point x, it generates a new failure customer, that is, a new machine breakdown. The case where the clock never stops until it hits x is the first failure type(physical time driven). If the clock moves only when the machine is up, that is, the clock stops while the machine is down, then we are in the second type of machine failure inter-arrivals(up time driven). The third case is where the clock moves only when the machine is busy in serving real customers.

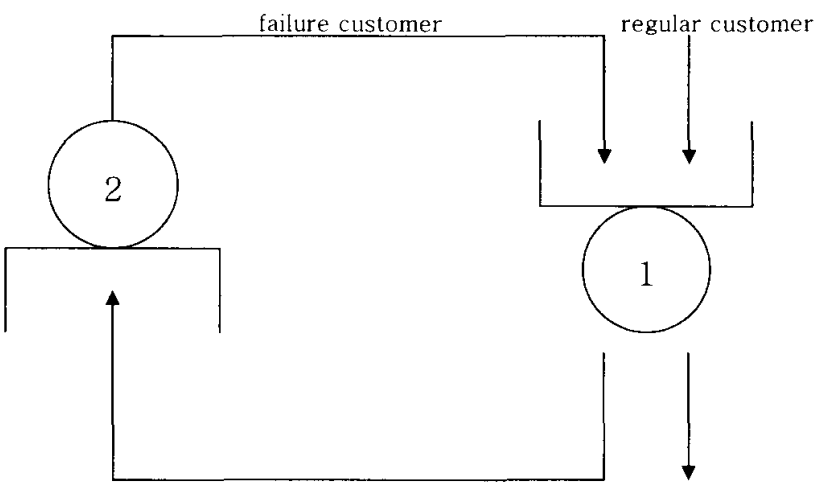
Of these three kinds of machine failures, we can easily apply queueing methodology to the first kind. As mentioned previously, one possible problem with modeling a machine failure as a customer type 1 is that a station may have multiple failure customers. In this case, we have difficulty calculating the

expected delay time for the station since we cannot know whether there are multiple failure customers. We only observe whether the failure customer is under service, that is, whether the machine is down. This problem comes from the characteristics of failure customers in addition to the preemptive priority.

Since the high priority customers vanish in the heavy traffic limit (even though they affect the throughput time of low priority class customer by $\frac{1}{1-\rho_H}U$, where U denotes workload), we do not see any of the high priority customers (that is, machine failures). Therefore, the probability of multiple failure customers is zero in the heavy traffic limit. The variance in addition to the mean of inter arrival times of failure customer affects U. Only the mean via failure customer traffic intensity ρ_H affects the linear relationship coefficient, $\frac{1}{1-\rho_H}$.

Unlike the first case, we need to revise traditional method for the second and the third kinds of failures. Here we consider the second type of machine failure into our model. Since we have one failure customer that circulates the processing network, the representation is modeled as a mixed queueing network. The failure customer gives the closed network component, and the real customer type offers an open network.

〈Figure 1〉 Machine Failures - mixed network representation



Let v_1 and v_2 be the service times of failure customer at stations 1 and 2 respectively. The means and variances are m_1, m_2, s_1^2, and s_2^2. The number of failure customers in the system is one. The service time at a station 1 (v_1) is the machine down time, or equivalently, the machine repair time. The service time at station 2 (v_2) is the inter-failure time of the machine.

Since the failure customer has preemptive priority at station 1, its cycle time is v_1+v_2 with a mean of m_1+m_2 and a variance of $s_1^2+s_2^2$. Although the service time at station 2 (v_2) gives the nominal inter-failure time of the failure customer in the station 1's up time units, the arrival rate of the failure customer to station 1 should be the 1/ cycle time since we should also consider the repair time for station 1. Therefore, the mean arrival rate of the failure customer to the station 1 is $\frac{1}{m_1+m_2}$. This suggests a revision of the current setting. In the closed network element, we substitute failure customer has the mean of m_1+m_2 and the variance of $s_1^2+s_2^2$ to include the cycle time of the failure in the closed network and calculate the failure's inter-arrival time. In the heavy traffic limit, only the mean and the variance are relevant as far as the underlying distributions are concerned. It should account for the special characteristic of a failure customer, which permits only one at a time. As shown in the first case, the high priority customer(failure customer in the current setting) vanishes in the heavy traffic limit, and eliminates the possibility of multiple failure customers.

V. Benefit of 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. Flexibility in servers allows the queueing system administrator to route 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 priority, we can reduce the mean throughput more.

In this paper, we note that the benefit of flexibility is still valid in case of machine failures.

When we construct a flexible processing system where a type of customers can be served at one of the multiple servers, we can route a flexible customer to a working machine that is idle when a machine breaks down. In a flexible processing system, we can utilize the other machine or allow the other machine to process incoming jobs when one machine is down. This resource pooling effect comes 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. In inflexible systems, even when a machine is down, the dedicated customers cannot be routed to or served by the other working station.

REFERENCES

1. Gross, D. and Harris, Carl M.(1998). *Fundamentals of Queueing Theory*, John Wiley & Sons.
2. 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.
3. Madu, C.N.(1988). A Closed Queueing Maintenance Network with Two Repair Centres, *Journal of Operations Research Society* 39, 959-967.
4. Nam, I.H.(2000). Improving linear processing system via flexibility, *International Journal of Production Research* 38, 341-352.
5. Wang, K.H. and Sivazlian, B.D.(1990). Comparative Analysis for the G/G/R Machine Repair Problem, *Computers Industrial Engineering* 18, 511-520.