

**Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596**

**PERFORMANCE OF LCFS QUEUEING SYSTEMS
WITH IMPATIENT CUSTOMERS**

Chialin Chang

**June 1991
CSD-910016**

UNIVERSITY OF CALIFORNIA
Los Angeles

Performance of LCFS Queueing Systems with Impatient Customers

A thesis submitted in partial satisfaction of the
requirements for the degree Master of Science
in Computer Science

by

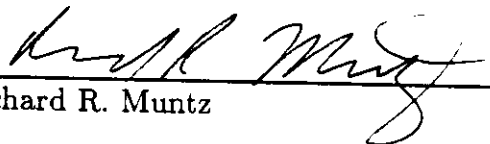
Chialin Chang

1991

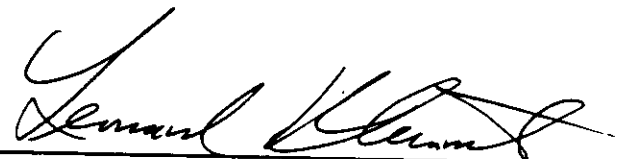
The thesis of Chialin Chang is approved.



Mario Gerla



Richard R. Muntz



Leonard Kleinrock, Committee Chair

University of California, Los Angeles

1991

TABLE OF CONTENTS

1	Introduction	1
2	Model Description and The Optimal Scheduling Policy	4
3	The LCFS-TO Policy for M/G/1 Queues	8
	3.1 Model Assumptions	8
	3.2 Analysis	8
	3.3 The Conditional Waiting Time Density Function	9
	3.4 The Probability That The System Is Busy	12
	3.5 The Final Results	14
	3.6 An Example : The M/M/1 Queue	16
4	The LCFS-TO Policy for M/M/m Queues	28
	4.1 Model Analysis and Policy Description	28
	4.2 Analysis	29
	4.3 The Waiting Time Density Function	31
	4.4 The Probability of Being In The Fully-Loaded State	34
	4.5 Numerical Results	42
5	Conclusion	45
	References	46

LIST OF FIGURES

1	An example of a concave function.	5
2	The tagged job arrives in a busy period.	11
3	The validation of the approximation for P_B ($\mu = 1$).	14
4	The conditional density function $w_d(t \text{busy})$ with $\mu = 1$	18
5	The three concave deadline distribution functions.	20
6	The goodputs of the three systems at $\lambda = 0.9$, and $\mu = 1$	20
7	The fraction of jobs that are discarded ($\lambda = 0.9$, $\mu = 1$).	22
8	The fraction of jobs that are served unsuccessfully ($\lambda = 0.9$, $\mu = 1$).	22
9	The goodput of system 1 with different system loads.	23
10	The optimal goodputs of the three systems.	24
11	The optimal thresholds T_0^* corresponding to Figure 10.	25
12	The optimal goodput of system 1 (under the LCFS-TO queueing policy) and the goodput of the STE queueing policy ($\mu = 1$).	26
13	The state transition diagram of an M/M/m queueing system under the LCFS-TO policy.	30
14	The tagged job arrives in the fully-loaded state.	33
15	The validation of the approximation for P_{FL} ($\mu = 1$, $T_0 = 2$).	40
16	The probability that the system is in the fully-loaded state obtained via simulation and via approximation ($\mu = 1$, $m = 2$).	40
17	The probability that the system is in the fully-loaded state obtained via simulation and via approximation ($\mu = 1$, $\lambda = 4$).	41
18	The goodputs of an M/M/m queueing system with $\mu = 1$ and $\lambda = 0.9$	42
19	The probability that a job is discarded ($\lambda = 4$, $\mu = 1$).	43
20	The probability that a job is served unsuccessfully ($\lambda = 4$, $\mu = 1$).	43

ACKNOWLEDGEMENTS

I wish to thank Dr. Leonard Kleinrock, who is my advisor and committee chair, for his great help and constructive comments in this thesis. I am grateful as well to Shiou-Pyn Shen and Jonathan Lu for all those discussions that solved some of the problems. I also wish to acknowledge Doris Sublette and Lily Chien for all the help they provided.

Last of all, I wish to express my appreciation to Tsung-Yuan Tai, Fu-Chung Wang, Mi-Sui Ling and my family, for their continuing support and encouragement.

ABSTRACT OF THE THESIS

Performance of LCFS Queueing Systems with Impatient Customers

by

Chialin Chang

Master of Science in Computer Science

University of California, Los Angeles, 1991

Professor Leonard Kleinrock, Chair

In many applications, jobs that arrive in a queueing system have real-time constraints to their waiting times, and these jobs should begin their service before their respective deadlines expire. Otherwise, the jobs are considered lost. Therefore, it is desired to schedule jobs such that a fraction of jobs that begin their service within their respective deadlines is maximized. In this thesis, we consider a queueing policy, known as LCFS-TO, in a system where only the distribution of jobs' deadlines, rather than the exact deadline of each arriving job, is available to the server. Based on the waiting times of the queued jobs, the policy decides the job service order and also which job(s) to discard, since jobs are unaware of their deadlines and therefore, even if their deadlines have expired, they have no idea about the expiration and do not leave the system automatically. We build an approximate model to analyze the performance of the LCFS-TO policy for $M/G/1$ and $M/M/m$ non-preemptive queueing models.

1 Introduction

In many applications, jobs that arrive in a queueing system have real-time constraints, and these jobs should be served before their respective deadlines. The deadlines can be the limiting constraint either on the jobs' waiting times or on the jobs' sojourn times (namely, waiting time + service time). For some systems, it is unacceptable for any job to miss its deadline. In these systems, which are referred as *hard* real-time systems, job service demands are usually well understood and much work has been done on the development and evaluation of scheduling policies [Liu 73, Mok 78]. Other systems consist of jobs for which it is not critical that all jobs meet their deadlines. In this thesis, we will focus on the latter model with a limiting constraint on a job's *waiting time* (i.e. time in queue). Typical examples of this model are impatient customers that give up their connections in a telecommunication network before the connections are completely connected, hospital emergency rooms handling critical patients, and the operation of radar screens of air defense systems. The common feature of these applications is that if any job begins its service after its deadline expires, the job is considered lost, and any service that it received is considered useless. Thus, it is desirable to schedule the jobs such that the fraction of jobs that begin service within their respective deadlines is maximized. This fraction is usually referred as the *goodput*.

There are usually two kinds of scenarios in these queueing systems. The first one is that the server is aware of the exact deadline of each arriving job. The application of radar screens of air defense systems fits in this scenario. Once the signal received by the radar is not processed within some known (and maybe fixed) interval of time, the signal is no longer useful. In this scenario, the server can discard the jobs whose waiting times exceed their associated deadlines, and

therefore no service work is useless. It can be shown that under certain conditions, the best policies when the deadlines are available to the server belong to the class of policies that choose to serve the job closest to its deadline (STE and STEI) [Tows 88].

The other scenario is that the server only knows the deadline *distribution* of the arriving jobs, rather than the exact deadline of each job. This often happens when the deadlines are not available to the server, and the only time we learn that the deadlines expire is when the results of the service are returned back to the users who submitted the jobs. For example, in a telecommunication network, the server (i.e. the switching box) is never able to know exactly how impatient each customer is, but may know the distribution of customers' impatience through some statistical investigation. Compared to the first scenario, this is a model with reduced information. Without knowing the exact deadline of every specific job, the control action is to decide, at appropriate decision instants, which job to serve and which job(s) to reject. A rejection scheme is necessary since the jobs are unaware of their deadlines in our model and, even if their deadlines have expired, they have no idea of the expiration and thus do not leave the queue automatically. Therefore a job could be either served in an order decided by a service discipline, or discarded by a rejection scheme. And due to the unawareness of the respective deadline, a job that gets served may or may not have met its deadline. We call a job that gets served and meets its deadline a *successful* job, and one that gets served after its deadline expires an *unsuccessful* job. Thus, a job in the system can either be discarded, successful, or unsuccessful. We will also use the term queueing policy in this thesis to refer the combination of the service discipline and the rejection scheme in a queueing system. Note that some server work may be useless due to the unawareness of the expiration of the jobs' deadlines. It can also be shown that under certain conditions, the last-come-first-served policy

with a time-out rejection mechanism (LCFS-TO) [Zhao 91] is the optimal policy in this scenario. The objective of this thesis is to build a model of the LCFS-TO policy, and using the model to solve for the goodput of the system.

2 Model Description and The Optimal Scheduling Policy

We consider a non-preemptive M/G/1 queueing system with an infinite number of buffers. The distribution of the arrival times is Poisson with parameter λ , and the distribution of the service times is an arbitrary function. All the service times are independent and identically distributed. Upon being generated, each job randomly selects a deadline, which is only known to the user who generated this job. This deadline is the maximum acceptable waiting time (in queue). All jobs draw their deadlines from a common distribution function $F_d(\cdot)$ on the set of positive real numbers. That is,

$$F_d(t) = \Pr\{\text{deadline} \leq t\}$$

Note that $F_d(\cdot)$ is a non-decreasing function. It is assumed that the server only knows the deadline distribution function $F_d(\cdot)$, and is unaware of the exact deadline of each job.

The deadline of a job may expire while waiting in the queue. As mentioned above, if a job with an expired deadline gets served, its service is considered unsuccessful. Therefore, an optimal scheduling policy would maximize the fraction of jobs that begin their service before their deadlines expire, namely, this policy maximizes the system's goodput. Zhao *et al.* [Zhao 91] showed that if $F_d(\cdot)$ is a concave function (for example, see Figure 1), that is, if jobs are more likely to have short deadlines rather than long deadlines, then the following two theorems hold.

Theorem 1 *For a non-preemptive M/G/1 queue, if the jobs' deadlines are independent and identically distributed with a concave function $F_d(\cdot)$, there exists an optimal scheduling policy which does not reject a job with a given waiting time while another job present in the buffer with a larger waiting time get served later,*

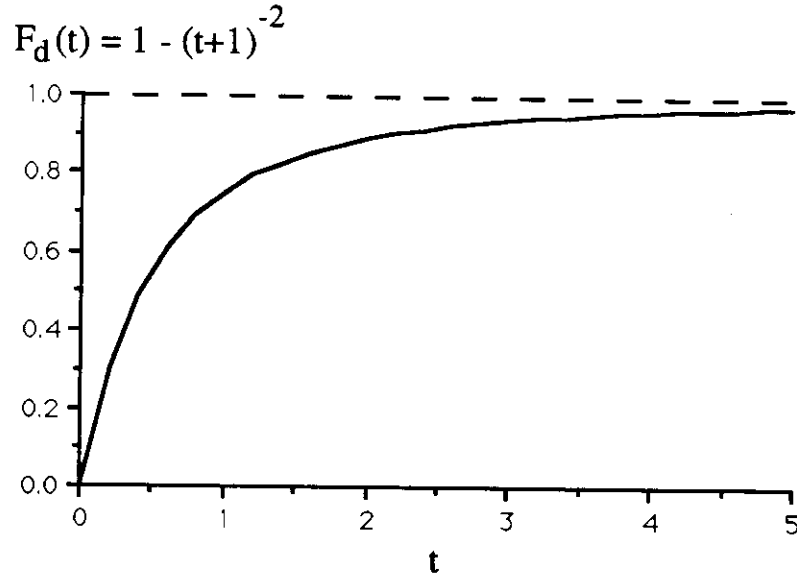


Figure 1: An example of a concave function.

and will not allow unforced idle times (that is, the server is allowed to become idle only when the system is empty).

The intuition behind this theorem is that since all jobs have the same service time distribution and all tend to have short deadlines, serving the job with a shorter waiting time would always have a higher probability to produce successful service than serving one with a longer waiting time. Furthermore, when the queueing policy decides that some job in the queue should get served, it should be served at once, without further delay. Inserting unforced idle times only decreases the probability that the service is successful.

However, some jobs may wait in the queue for a long time. For example, when the arrival rate to a queueing system is larger than the service rate of the server, the queue will build up and some jobs may never reach the server if a rejection scheme is not applied to the system. These jobs stay in the buffer for ever which is equivalent to being rejected. In general, a job waiting in the buffer for a very long time will have an expired deadline, with a probability approaching

one. Since serving these jobs could very possibly result in useless server work, it is worth rejecting them and serving another job with shorter waiting time, or even waiting for a new arrival and serving it.

Theorem 2 *Consider a non-preemptive $M/G/1$ queue where the waiting times of queued jobs are available. If the jobs' deadlines are independent and identically distributed with a concave function $F_d(\cdot)$, there exists an optimal stationary queueing policy, the LCFS-TO policy, which is described below.*

1. *the service discipline is last-come-first-served.*
2. *every arriving job that joins the buffer will get discarded if its waiting time exceeds some threshold T_0 – a time-out rejection mechanism.*
3. *unforced idle times are not allowed.*

Imagine how the system works under the LCFS-TO scheduling policy with some waiting time threshold T_0 . Every job that arrives in the system will join the buffer and be time-stamped with its arriving time. Whenever the server becomes idle, it selects from the buffer the job with the newest arriving time if the buffer is not empty, or waits for a new arriving job if the buffer is empty and selects it immediately upon its arrival. This selection process implements the last-come-first-served service discipline. Now the server has to check the selected job's waiting time, which can be obtained by subtracting the selected job's arriving time, or, its time-stamp, from the server's current system time. If the waiting time is not greater than T_0 , the selected job is served, and hopefully the service will be successful. If the waiting time exceeds T_0 , the selected job is discarded by the time-out mechanism. In fact, if the process overhead for selecting a job and calculating the waiting time is negligible and thus no arrival occurs between the time the job is selected and the time it is discarded, the server can not only throw away the selected job, but also discard all the jobs waiting in the buffer.

This is because with last-come-first-served service discipline, all jobs waiting in the buffer will always have waiting times longer than that of the selected job, and certainly longer than the threshold T_0 .

It can easily be realized that the waiting time threshold T_0 plays an important role in the LCFS-TO queueing policy. The value of T_0 decides how often does the policy discard queued jobs. If T_0 is too small, many jobs will be discarded, even though they could be successful with a high probability. If T_0 is too large, many queued jobs with long waiting times will receive service which would turn out to be unsuccessful with a high probability. In both cases, the fraction of successful jobs would decrease, and the performance of the queueing system is degraded. In the next section, we will build an approximate model to analyze the performance of the LCFS-TO queueing policy for an M/G/1 queueing model. The extended model for m servers will be presented in section 4.

3 The LCFS-TO Policy for M/G/1 Queues

In this section, we analyze the performance of the LCFS-TO queueing policy under an M/G/1 non-preemptive model, and find the goodput of the queueing system.

3.1 Model Assumptions

For our M/G/1 model, the arrival process is assumed to be Poisson with mean arrival rate λ . The jobs' service times are independent and identically distributed with an arbitrary distribution, whose Laplace transform is denoted as $B^*(s)$. The mean service time is \bar{x} . The jobs' deadlines are also independent and identically distributed with distribution function $F_d(\cdot)$. As mentioned in the previous section, if $F_d(\cdot)$ is a concave function, then the LCFS-TO queueing policy is known to be the optimal policy. Our investigation of the LCFS-TO policy does not require $F_d(\cdot)$ to be concave. It is also assumed that the processing overhead to enforce the LCFS-TO queueing policy is negligible.

3.2 Analysis

The performance metric that we are interested in is the fraction of jobs that begin their service before their respective deadlines expire. When the system reaches equilibrium, all jobs statistically have the same behavior. Specifically, they all have the same waiting time distribution. And with the same deadline distribution, the fraction of successful jobs for a system in equilibrium is the same as the probability that a job is successful. Let P_s be the probability that an individual job is successful. It can be expressed as follows.

$$P_s = \int_0^{T_0} [1 - F_d(t)] w_d(t) dt \quad (1)$$

where $w_d(w)$ is the probability density function of the job's waiting time.

Realizing that a job which arrives in an idle period, namely a period of time during which the system is idle (empty), will always be successful, we can rewrite equation (1) by conditioning on whether the system is idle or not when the job arrives. Define

$$P_B \triangleq \Pr\{\text{a job arrives in a busy period}\}$$

Then, we have

$$P_s = (1 - P_B) + P_B \int_0^{T_0} [1 - F_d(t)] w_d(t|\text{busy}) dt \quad (2)$$

where $w_d(t|\text{busy})$ is the conditional probability density function of the job's waiting time, given that the job arrives in a busy period.

Now, all we need to know is $w_d(t|\text{busy})$ and P_B . We will find them in the next two subsections.

3.3 The Conditional Waiting Time Density Function

Consider a tagged job that arrives at the system during a busy period. Let W_T be the random variable of the waiting time of our tagged job. With the time-out mechanism, our tagged job may get discarded when its waiting time exceeds the waiting time threshold T_0 . Let's define W_T to be ∞ when this happens. This means that the density function of W_T would be some kind of continuous curve in the range $(0, T_0]$, with an impulse at $W_T = \infty$.

Note that under the LCFS-TO queueing policy, the tagged job will always be selected by the service discipline before any other job that arrives before the tagged job and is still waiting in the buffer. Therefore, in addition to the job in service, only jobs that arrive after our tagged job arrives will affect the tagged job's waiting time W_T . Furthermore, whenever some job that arrives after our tagged job arrives is selected by the service discipline but discarded by the time-out mechanism, our tagged job will automatically be discarded too. In other

words, if our tagged job eventually enters the server and receives its service, none of the jobs that arrive within the waiting time of our tagged job are discarded, and the system appears to our tagged job as a simple LCFS queueing system without any constraint on the waiting time. That is, if W_S is the random variable of a job's waiting time in the simple LCFS queueing model, the distribution of W_T would be exactly identical to that of W_S in the range of $(0, T_0]$, which is exactly the range that we need for equation (2). Therefore, we temporarily forget about T_0 , and imagine that our tagged job now arrives in the simple LCFS queueing system instead. Note that the waiting time of our tagged job is now W_S .

Let $w_B(t)$ be the density function of W_S with Laplace transform $W_B^*(s)$. Furthermore, define the following random variables (see Figure 2).

X = the service time of the job in the server when our tagged job arrives

Y = the residual life of the service time X after our tagged job arrives

N = the number of jobs that arrive during the residual service time Y

G = the duration of a busy period in an M/G/1 queueing system, with probability density function $g(\cdot)$, and its Laplace transform $G^*(s)$.

From Figure 2, we can see that the waiting time of our tagged job consists of two parts: the residual service time Y , and the N sub-busy periods generated by all the jobs that arrive after our tagged job. Thus we have the following relation.

$$W_S = Y + G_N + G_{N-1} + \dots + G_2 + G_1 \quad (3)$$

where G_i is the duration of the sub-busy period generated by job C_i . Note that each of G_i has the same distribution as that of G , the duration of a busy period in an M/G/1 queueing system.

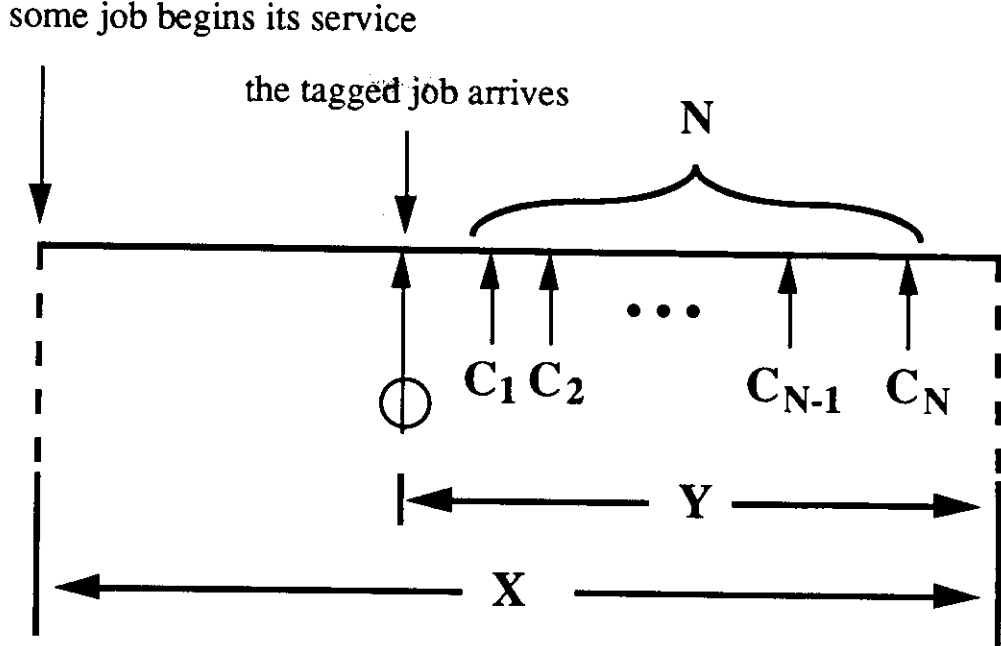


Figure 2: The tagged job arrives in a busy period.

Therefore,

$$\begin{aligned}
 E[e^{-sW_s} | X = x, Y = y, N = n] &= E[e^{-s(y+G_n+G_{n-1}+\dots+G_2+G_1)}] \\
 &= e^{-sy} E[e^{-sG_n} e^{-sG_{n-1}} \dots e^{-sG_2} e^{-sG_1}]
 \end{aligned}$$

Since the sub-busy periods have durations that are independent of each other, we may write this last as

$$E[e^{-sW_s} | X = x, Y = y, N = n] = e^{-sy} [G^*(s)]^n$$

For Poisson arrivals with mean arrival λ , the probability to have n arrivals during an interval of y is $e^{-\lambda y} (\lambda y)^n / n!$. Therefore,

$$\begin{aligned}
 E[e^{-sW_s} | X = x, Y = y] &= \sum_{n=0}^{\infty} e^{-sy} [G^*(s)]^n \frac{(\lambda y)^n}{n!} e^{-\lambda y} \\
 &= e^{-[s+\lambda-\lambda G^*(s)]y}
 \end{aligned}$$

Knowing that the joint density function of X and Y is [Klei 75]

$$\Pr\{y < Y \leq y + dy, x < X \leq x + dx\} = \frac{dy dB(x)}{\bar{x}} \quad (4)$$

We can uncondition on X and Y over the appropriate ranges, and obtain the following result.

$$\begin{aligned}
E[e^{-sW_S}] &= \int_{x=0}^{\infty} \int_{y=0}^x e^{-[s+\lambda-\lambda G^*(s)]y} \frac{dB(x)dy}{\bar{x}} \\
&= \int_{x=0}^{\infty} \frac{1 - e^{-[s+\lambda-\lambda G^*(s)]x}}{[s + \lambda - \lambda G^*(s)]\bar{x}} dB(x) \\
\text{or } W_B^*(s) &= \frac{1 - B^*[s + \lambda - \lambda G^*(s)]}{[s + \lambda - \lambda G^*(s)]\bar{x}} \tag{5}
\end{aligned}$$

This is the known result for the conditional waiting time transform as found, for example, in [Klei 76]. Since λ , \bar{x} , and $B^*(s)$ are known, and $G^*(s)$ can be obtained from the following relation [Klei 75],

$$G^*(s) = B^*[s + \lambda - \lambda G^*(s)]$$

we can get the density function of W_S by applying inverse Laplace transform to the result in equation (5). Namely,

$$w_d(t|\text{busy}) = \mathbf{L}^{-1}\{W_B^*(s)\} \tag{6}$$

where $\mathbf{L}^{-1}\{\cdot\}$ stands for inverse Laplace transform.

3.4 The Probability That The System Is Busy

In this subsection, we will find P_B , the probability that a job arrives in a busy period. We know that for Poisson arrivals, the system states found by arrivals always have the same distribution as that of the real system states [Klei 75]. Therefore, the probability that an arrival sees a busy period is the same as the probability that the system is busy, or the fraction of time that the system in a busy period.

Since the system passes through alternating cycles of busy periods and idle periods, P_B can be obtained from the following expression.

$$P_B = \frac{E[\text{duration of a busy period}]}{E[\text{duration of a busy period}] + E[\text{duration of an idle period}]}$$

For the Poisson arrivals with mean arrival rate λ ,

$$\bar{T} \triangleq E[\text{duration of an idle period}] = \frac{1}{\lambda}$$

However, due to the time-out rejection scheme, the duration of a busy period is not easy to obtain directly. Define r to be the probability that a job which arrives in a busy period will eventually receive its service. Note that r is also the fraction of arrivals during a busy period that actually receive service. Clearly r depends on T_0 . r can be obtained from the following expression.

$$\begin{aligned} r &= \Pr\{\text{waiting time} \leq T_0 | \text{the job arrives in a busy period}\} \\ &= \int_0^{T_0} w_d(t | \text{busy}) dt \end{aligned} \quad (7)$$

Realizing that only a fraction of arrivals during a busy period will receive service and therefore contribute to the busy period, we consider a new M/G/1 non-preemptive queueing model, as opposed to the *original* model we had before. We call this new model the *approximate* model. The approximate model differs from our original model in two ways. First, there is no time-out rejection mechanism, namely, all arrivals will eventually receive their service. Second, the mean arrival rate of its Poisson arrival process is $r\lambda$. The first modification makes our analysis easier to handle, and the second modification tries to capture the characteristic of the rejection scheme in our original model.

Since only a fraction r of the arrivals during a busy period in the original model contribute to its busy period, the duration of a busy period in the approximate model with a reduced arrival rate $r\lambda$, should roughly be the same as that of the original model. Let

$$\bar{g}_A = E[\text{duration of a busy period in the approximate model}]$$

Then for an M/G/1 model, we have [Klei 75]

$$\bar{g}_A = \frac{\bar{x}}{1 - r\lambda\bar{x}}$$

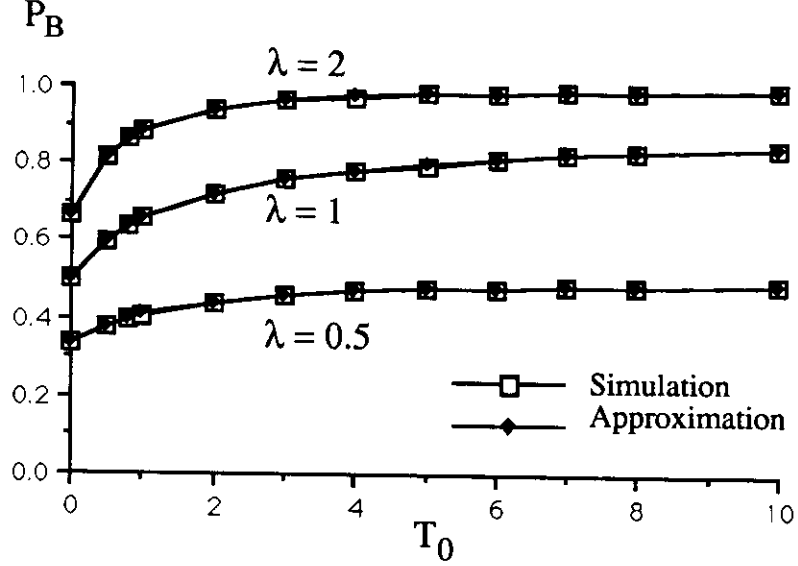


Figure 3: The validation of the approximation for P_B ($\mu = 1$).

Approximating the mean duration of a busy period in the original model with \bar{g}_A in the approximate model, we have P_B as follows.

$$\begin{aligned}
 P_B &\approx \frac{\bar{g}_A}{\bar{g}_A + \bar{I}} \\
 &= \frac{\lambda \bar{x}}{1 + (1 - r)\lambda \bar{x}}
 \end{aligned} \tag{8}$$

Figure 3 shows the results of P_B obtained via approximation and via simulation. It can be seen that the approximation is very close to the simulation.

3.5 The Final Results

Substituting for P_B , we can finally obtain the goodput from equation (2).

$$\begin{aligned}
 P_s &\approx \frac{1 - r\lambda \bar{x}}{1 + (1 - r)\lambda \bar{x}} + \\
 &\quad \frac{\lambda \bar{x}}{1 + (1 - r)\lambda \bar{x}} \int_0^{T_b} [1 - F_d(t)] w_d(t|busy) dt
 \end{aligned} \tag{9}$$

In addition, knowing $w_d(t|\text{busy})$ and P_B from equation (5) and (6), we can also obtain the following relations.

$$\begin{aligned}
P_d &\triangleq \Pr\{\text{a job gets discarded}\} \\
&\approx \frac{\lambda\bar{x}}{1 + (1-r)\lambda\bar{x}} \left[1 - \int_0^{T_0} w_d(t|\text{busy})dt\right] \\
&= \frac{(1-r)\lambda\bar{x}}{1 + (1-r)\lambda\bar{x}} \tag{10}
\end{aligned}$$

$$\begin{aligned}
P_u &\triangleq \Pr\{\text{a job is unsuccessful}\} \\
&\approx \frac{\lambda\bar{x}}{1 + (1-r)\lambda\bar{x}} \int_0^{T_0} F_d(t)w_d(t|\text{busy})dt \tag{11}
\end{aligned}$$

To achieve the maximum goodput, T_0 must be chosen such that the derivative of P_s with respect to T_0 is equal to zero. But first, let's find the derivative of r with respect to T_0 .

$$\begin{aligned}
\frac{dr}{dT_0} &= \frac{d}{dT_0} \int_0^{T_0} w_d(t|\text{busy})dt \\
&= w_d(T_0|\text{busy})
\end{aligned}$$

Now, differentiating equation (9) with respect to T_0 , we get

$$\begin{aligned}
\frac{dP_s}{dT_0} &= \frac{d}{dT_0} \left(\frac{1 - r\lambda\bar{x}}{1 + (1-r)\lambda\bar{x}} \right) + \\
&\quad \left[\frac{d}{dT_0} \left(\frac{\lambda\bar{x}}{1 + (1-r)\lambda\bar{x}} \right) \right] \int_0^{T_0} [1 - F_d(t)]w_d(t|\text{busy})dt + \\
&\quad \left(\frac{\lambda\bar{x}}{1 + (1-r)\lambda\bar{x}} \right) \frac{d}{dT_0} \int_0^{T_0} [1 - F_d(t)]w_d(t|\text{busy})dt \\
&= \frac{(\lambda\bar{x})^2 w_d(T_0|\text{busy})}{[1 + (1-r)\lambda\bar{x}]^2} \left\{ \int_0^{T_0} [1 - F_d(t)]w_d(t|\text{busy})dt - 1 \right\} + \\
&\quad \frac{\lambda\bar{x}}{1 + (1-r)\lambda\bar{x}} [1 - F_d(T_0)]w_d(T_0|\text{busy}) \tag{12}
\end{aligned}$$

The optimal threshold, T_0^* , must satisfy the following equation.

$$\left. \frac{dP_s}{dT_0} \right|_{T_0=T_0^*} = 0$$

3.6 An Example : The M/M/1 Queue

Here we apply the results we obtained in the previous subsection to an M/M/1 queueing system, in which the jobs' service times are exponentially distributed with mean service time $\bar{x} = 1/\mu$.

3.6.1 Analysis

For an M/M/1 system, we have the following relations [Klei 75].

$$\begin{aligned} B^*(s) &= \frac{\mu}{s + \mu} \\ G^*(s) &= B^*[s + \lambda - \lambda G^*(s)] \\ &= \frac{\mu}{s + \lambda - \lambda G^*(s) + \mu} \end{aligned}$$

Namely,

$$\lambda[G^*(s)]^2 - (s + \lambda + \mu)G^*(s) + \mu = 0$$

Solving for $G^*(s)$ and restricting our solution to the required (stable) case, for which $|G^*(s)| \leq 1$ for $\text{Re}(s) \geq 0$, gives

$$G^*(s) = \frac{s + \lambda + \mu - \sqrt{(s + \lambda + \mu)^2 - 4\lambda\mu}}{2\lambda}$$

This equation may be inverted (by referring to transform tables) to obtain the probability density function for the busy period, namely,

$$g(t) = \frac{1}{t\sqrt{\rho}} e^{-(\lambda+\mu)t} I_1 \left[2t\sqrt{\lambda\mu} \right] \quad (13)$$

where $\rho = \lambda/\mu$, and I_1 is the modified Bessel function of the first kind of order one.

Plugging $G^*(s)$ into equation (5) and substituting \bar{x} with $1/\mu$, we have

$$W_B^*(s) = \frac{1 - G^*(s)}{[s + \lambda - \lambda G^*(s)]\bar{x}}$$

$$\begin{aligned}
&= \frac{\mu \left(2 - \frac{s+\lambda+\mu-\sqrt{(s+\lambda+\mu)^2-4\lambda\mu}}{\lambda} \right)}{2s+2\lambda-s-\lambda-\mu+\sqrt{(s+\lambda+\mu)^2-4\lambda\mu}} \\
&= \frac{\mu \left(\lambda-s-\mu+\sqrt{(s+\lambda+\mu)^2-4\lambda\mu} \right)}{\lambda \{s+\lambda-\mu+\sqrt{(s+\lambda+\mu)^2-4\lambda\mu}\}}
\end{aligned}$$

Note that $\sqrt{(s+\lambda+\mu)^2-4\lambda\mu} = \sqrt{(s+\lambda-\mu)^2+4\mu s}$. Hence

$$\begin{aligned}
W_B^*(s) &= \frac{\mu}{\lambda} \left\{ 1 - \frac{2s}{s+\lambda-\mu+\sqrt{(s+\lambda-\mu)^2+4\mu s}} \right\} \\
&= \frac{\mu}{\lambda} \left\{ 1 + \frac{2s[(s+\lambda-\mu)-\sqrt{(s+\lambda-\mu)^2+4\mu s}]}{4\mu s} \right\} \\
&= \frac{s+\lambda+\mu-\sqrt{(s+\lambda+\mu)^2-4\lambda\mu}}{2\lambda} \\
&= G^*(s)
\end{aligned}$$

The equation that we just derived says the distribution of the waiting time W_S in the simple LCFS M/M/1 queueing system is statistically the same as that of the duration of a busy period in an M/M/1 queueing system. The reason is that in an M/M/1 system, the exponential distribution of the jobs' service times is memoryless. This implies that the residual service time Y has the same distribution as that of the service time X . Therefore, the waiting time W_S of our tagged job in equation (3) can be rewritten as

$$W_S = X + G_N + G_{N-1} + \dots + G_2 + G_1$$

This sum is exactly the same as the duration of a busy period [Klei 75]. Thus, in an M/M/1 queueing system, the waiting time W_S always has the same distribution as that of the duration of a busy period.

From equation (13), we have the result of the inverse Laplace transform of $W_B^*(s)$ for free.

$$w_d(t|\text{busy}) = \frac{1}{t\sqrt{\rho}} e^{-(\lambda+\mu)t} I_1 \left[2t\sqrt{\lambda\mu} \right]$$

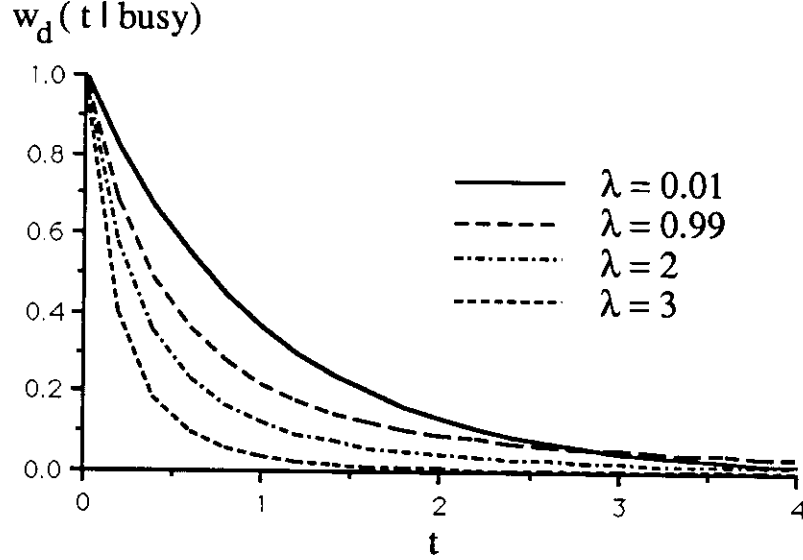


Figure 4: The conditional density function $w_d(t|\text{busy})$ with $\mu = 1$.

Figure 4 plots this function with different values of λ 's.

Plugging the result above into equation (7), we have r , which then defines P_B from equation (8). Namely,

$$r = \int_0^{T_0} \frac{1}{t\sqrt{\rho}} e^{-(\lambda+\mu)t} I_1 \left[2t\sqrt{\lambda\mu} \right] dt$$

$$P_B \approx \frac{\lambda}{\mu + (1-r)\lambda}$$

Hence, the goodput can be obtained by plugging the results above into equation (9).

$$P_s \approx \frac{\mu - r\lambda}{\mu + (1-r)\lambda} + \frac{\lambda}{\mu + (1-r)\lambda} \int_0^{T_0} [1 - F_d(t)] \frac{1}{t\sqrt{\rho}} e^{-(\lambda+\mu)t} I_1 \left[2t\sqrt{\lambda\mu} \right] dt \quad (14)$$

Furthermore, knowing $w_d(t|\text{busy})$ and P_B , the probability that a job is discarded and the probability that a job is unsuccessful can also be obtained from equation (10) and (11).

$$P_d \approx \frac{\lambda}{\mu + (1-r)\lambda} \int_0^{T_0} \frac{1}{t\sqrt{\rho}} e^{-(\lambda+\mu)t} I_1 \left[2t\sqrt{\lambda\mu} \right] dt$$

$$P_u \approx \frac{\lambda}{\mu + (1-r)\lambda} \int_0^{T_0} F_d(t) \frac{1}{t\sqrt{\rho}} e^{-(\lambda+\mu)t} I_1 \left[2t\sqrt{\lambda\mu} \right] dt$$

From equation (12), we can find as well the equation that defines the optimal threshold T_0^* .

$$\frac{\lambda^2 e^{-(\lambda+\mu)T_0} I_1 \left[2T_0 \sqrt{\lambda\mu} \right]}{[\mu + (1-r)\lambda]^2 T_0 \sqrt{\rho}} \left\{ \int_0^{T_0} [1 - F_d(t)] \frac{1}{t\sqrt{\rho}} e^{-(\lambda+\mu)t} I_1 \left[2t\sqrt{\lambda\mu} \right] dt - 1 \right\} + \frac{\lambda}{\mu + (1-r)\lambda} [1 - F_d(T_0)] \frac{1}{T_0 \sqrt{\rho}} e^{-(\lambda+\mu)T_0} I_1 \left[2T_0 \sqrt{\lambda\mu} \right] = 0$$

3.6.2 Numerical Results and Discussion

In this subsection, we present and discuss some numerical results for the LCFS-TO queueing policy in an M/M/1 queueing system. We use the following three concave functions as examples of the deadline distribution functions.

$$\begin{aligned} F_1(t) &= 1 - \frac{1}{(t+1)^2} \\ F_2(t) &= 1 - \frac{4}{(t+2)^2} \\ F_3(t) &= 1 - \frac{9}{(t+3)^2} \end{aligned}$$

Consider three M/M/1 queueing systems, each of which has one of the $F_i(t)$ functions as its deadline distribution. In the rest of this section, we shall refer to the queueing system with deadline distribution $F_i(t)$ as *system i* ($i = 1, 2, 3$). Figure 5 shows the three functions graphically. Note that $F_1(t)$ approaches 1 much faster than the others, and $F_2(t)$ approaches 1 faster than $F_3(t)$. This means that, with the same waiting time, a job served in system 1 has the lowest probability to be successful, while a job served in system 3 has the highest probability to be successful. In other words, jobs in system 1 tend to have stricter deadlines than those in system 2, which tend to have stricter deadlines than those in system 3.

Figure 6 shows the goodputs of the three systems with $\lambda = 0.9$ and $\mu = 1$, obtained from equation (14). When $T_0 = 0$, namely, when only the jobs that arrive in idle periods are served, all jobs that get served have waiting time 0. Therefore, the deadline distribution functions have no effect on the goodput,

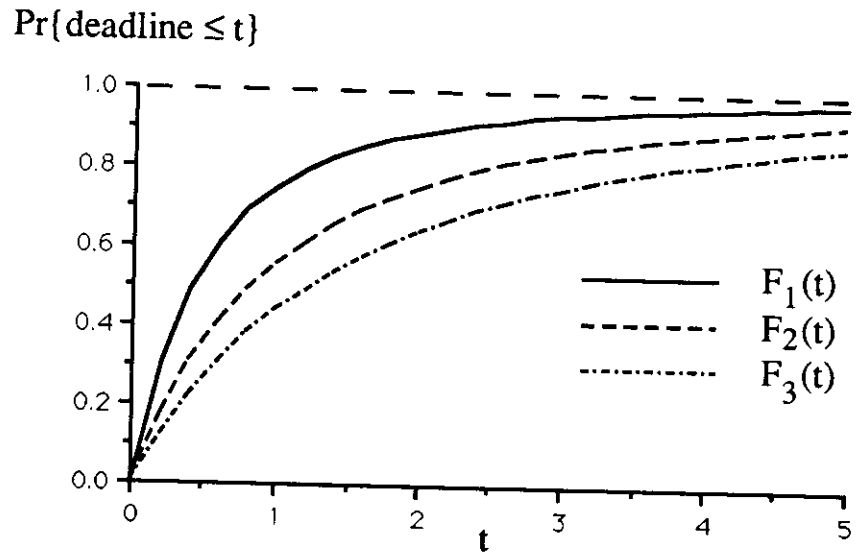


Figure 5: The three concave deadline distribution functions.

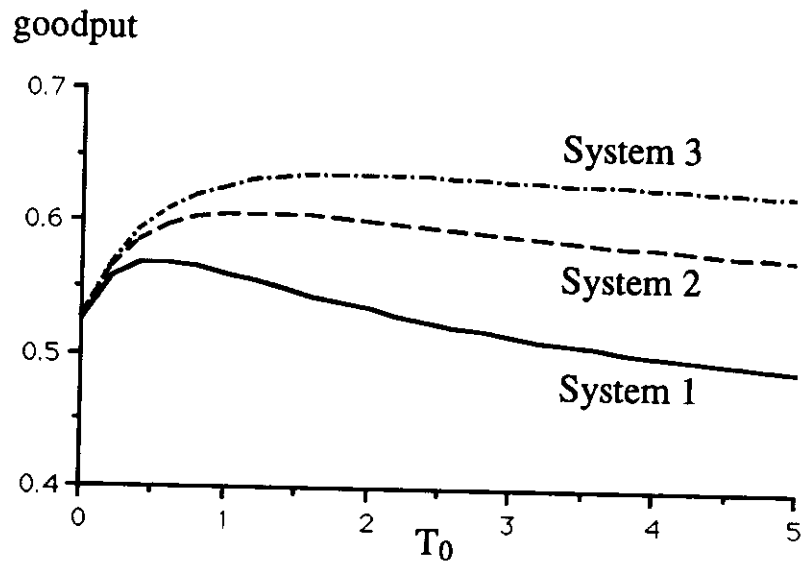


Figure 6: The goodputs of the three systems at $\lambda = 0.9$, and $\mu = 1$.

and the three systems behave identically and all have the same goodput. As T_0 increases, more jobs tend to get served. When T_0 , though increasing, still remains small enough so that jobs that get served have very short waiting times and hence very high probabilities to be successful, the goodput increases. System 3 has the most relaxed deadline distribution, and not surprisingly its goodput increases faster than the others.

However, if T_0 becomes too large, more jobs with long waiting times will get served. The service times of these old jobs not only have higher probabilities to become useless, but also block the new arrivals and force them to wait in the buffer. This decreases the probabilities for those new arrivals to become successful. Hence, the system performance is degraded and the goodput decreases. System 1 has the strictest deadline distribution, and thus its goodput deteriorates more significantly than the others. On the other hand, the deadline distribution of system 3 is so relaxed that even with a large T_0 , the jobs that get served can still be successful with high probabilities. Therefore, the goodput of system 3 only has insignificant decay as T_0 increases.

When T_0 becomes even larger, the probability that a job with a waiting time comparable to T_0 becomes very small, and very few jobs really have to wait in the queue for T_0 before they are served. Therefore the speed of degradation slows down, and as T_0 approaches ∞ , the system approaches an M/M/1 queueing system with last-come-first-served service discipline and no time-out rejection scheme. Note that when the deadline distribution is strict enough, the goodput at a large T_0 could be worse than that at $T_0 = 0$ (e.g. system 1 in Figure 6).

Figure 7 and Figure 8 show the fraction of jobs that are discarded (P_d), and the fraction of jobs that are served unsuccessfully (P_u). The deadline distribution function has no effect on P_d , and all three systems have the same P_d , which decreases as T_0 increases. System 1 has the strictest deadline distribution, there-

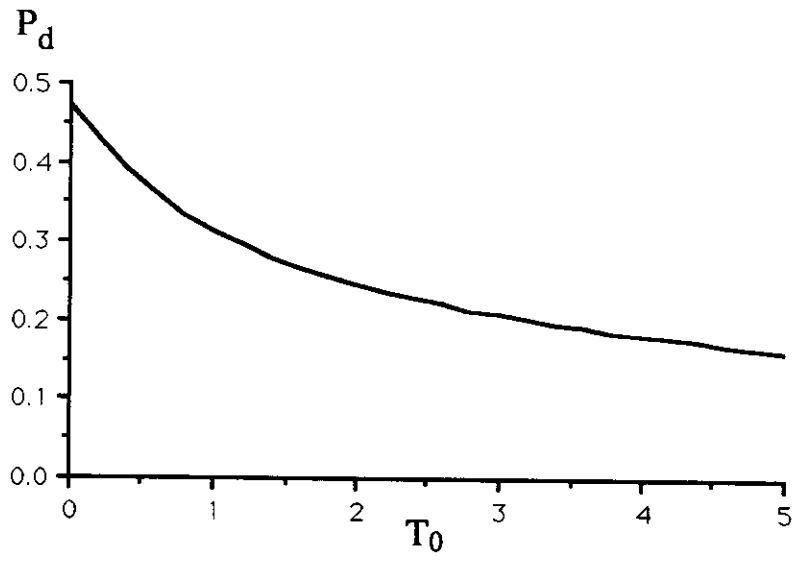


Figure 7: The fraction of jobs that are discarded ($\lambda = 0.9, \mu = 1$).

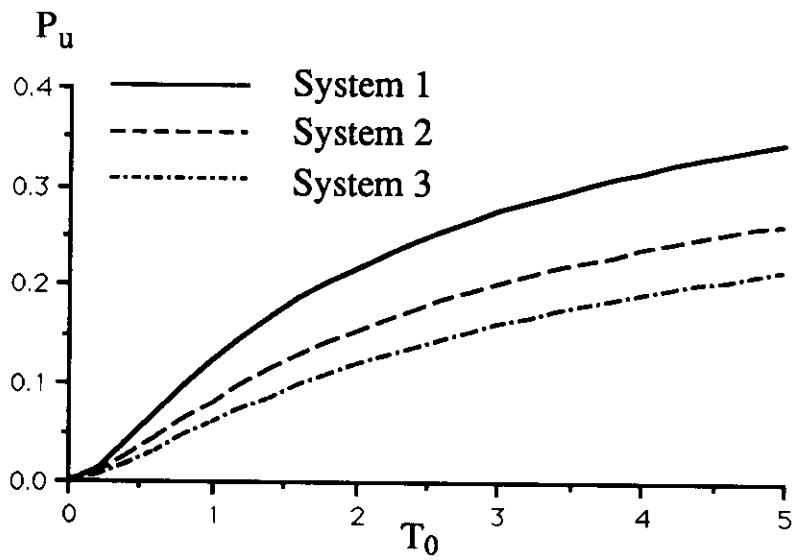


Figure 8: The fraction of jobs that are served unsuccessfully ($\lambda = 0.9, \mu = 1$).

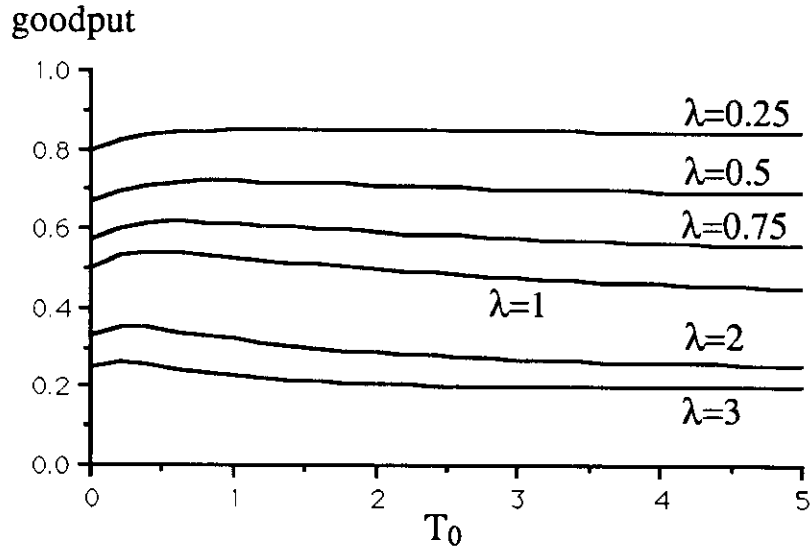


Figure 9: The goodput of system 1 with different system loads.

fore jobs that are served in system 1 have the highest probabilities to become unsuccessful.

Figure 9 shows the goodputs of system 1 with different mean arrival rates. Since the mean service rate is fixed at $\mu = 1$, different λ 's correspond to different system loads. Note that with the time-out rejection scheme, λ/μ is not the effective system load, and the system remains stable even if λ exceeds μ . From the figure, we see that the goodput decreases significantly as the system load increases. This is because the heavier the system load, the longer the waiting times, and thus the less the number of jobs that get served.

When $T_0 = 0$, the goodput is determined by the probability that the system is idle, which is closely related to the mean arrival rate λ . As λ increases, the probability that the system is idle decreases, and therefore more jobs are rejected and the goodput decreases. Note that when system load is heavy (e.g. $\lambda = 3$), the goodput at large T_0 is even worse than that at $T_0 = 0$.

In order to achieve the maximal goodput, the optimal value of the waiting

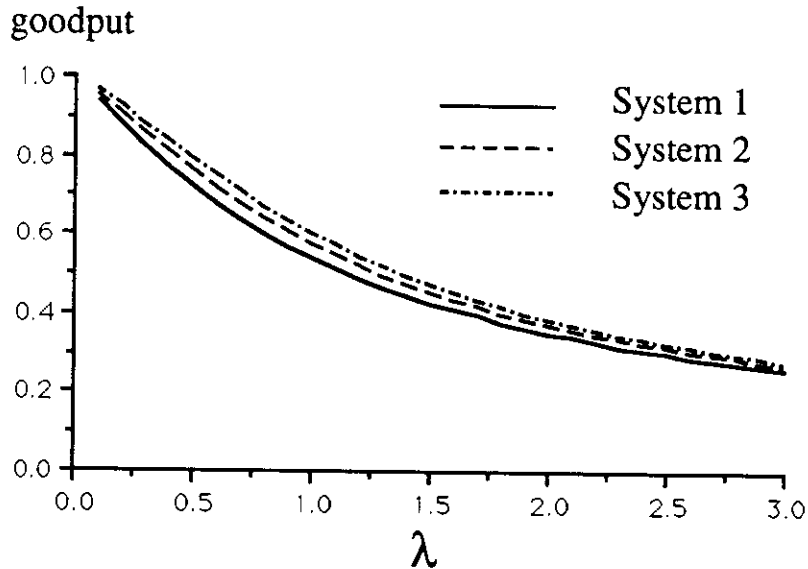


Figure 10: The optimal goodputs of the three systems.

time threshold T_0 must be chosen for each set of given system parameters. Figure 10 shows the optimal goodputs for the three systems over a range of λ 's, and Figure 11 shows the corresponding optimal threshold T_0^* . We can see that when system load is light, T_0^* is large. This is because that with light system load, the probability for having a new arrival during the service time is very small and thus serving an old job is very unlikely to do any harm. Hence the LCFS-TO queueing policy tends to serve each job that arrives in the system. However, as the system load becomes heavy, T_0^* decreases, and the LCFS-TO queueing policy tends to reject old jobs waiting in the buffer and wait to serve new arriving jobs. This is because new arriving jobs have higher probabilities to become successful, and with large λ , the new jobs arrive so frequent that it is worthwhile for the server to wait for a new job instead of serving an old job in the buffer. System 3 has the most relaxed deadline distribution function and thus has the best goodput.

Note that the difference among the goodputs of the three systems are small when the system load becomes very light (e.g. $\lambda < 0.1$) or very heavy (e.g. $\lambda > 2$).

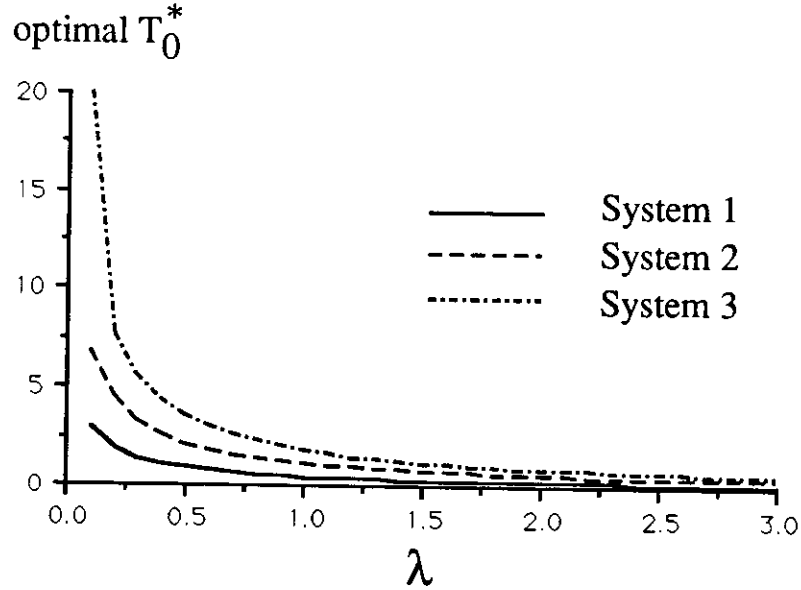


Figure 11: The optimal thresholds T_0^* corresponding to Figure 10 .

This is because when system load is very light, most of the arriving jobs will find the system empty and are served immediately without further waiting. Therefore, even the optimal threshold T_0 is large, very few jobs would actually have to wait in the buffer. On the other hand, when the system load becomes very heavy, the optimal threshold T_0 tends to approach 0, and each arriving job is only allowed to wait for a very short interval in the buffer. If its waiting time becomes a little larger, it is discarded. In both situations, the waiting times of the jobs that are served are very small, either due to the light system load when λ is small, or due to the small T_0^* when λ is large. Thus the strictness of the deadline distribution function has little effect on the goodput, and the three systems tend to have similar goodputs.

We mentioned before that when the exact deadline of each job is available to the server, the STE policy – the policy that serves the job with the closest deadline to expire – is the optimal policy. Note that jobs are either discarded or successful under the STE policy, and no jobs are unsuccessful. However, when

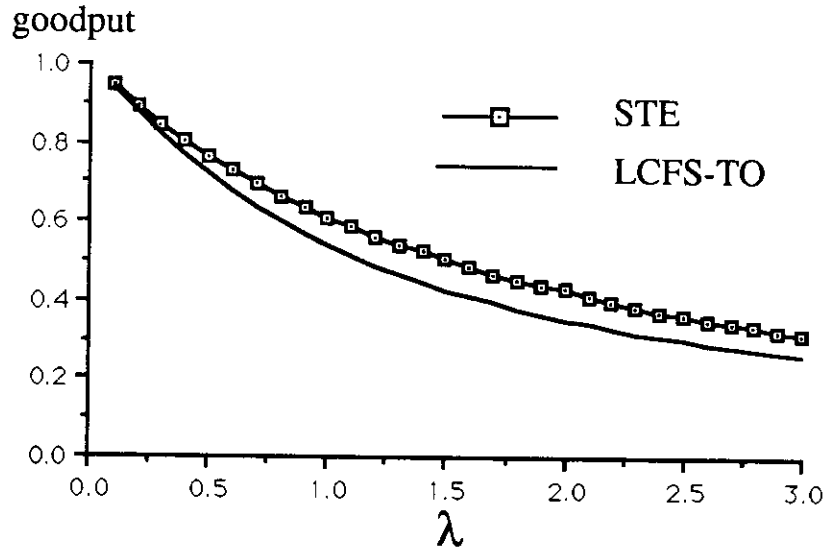


Figure 12: The optimal goodput of system 1 (under the LCFS-TO queueing policy) and the goodput of the STE queueing policy ($\mu = 1$).

the exact deadlines are not available to the server, the LCFS-TO policy has to "guess" which job to serve and which job to discard, based on the job's waiting time and the deadline distribution. Note that LCFS-TO is using less information than STE does, and therefore any wrong decision degrades the system's goodput. This brings up an interesting question: how much penalty does one have to pay for not knowing the exact deadlines? Figure 12 shows the optimal goodput of system 1 and the goodput of the corresponding system under the STE queueing policy. In this figure, both systems have the same mean service rate and the same deadline distribution. The data here for the STE policy is obtained via simulation. Note how well the LCFS-TO policy compares to the optimal (STE) policy. It can be seen that when the system load is very light, most arrivals find the system empty, and begin their service immediately. These jobs surely will succeed. Since the LCFS-TO policy always serves jobs with waiting time zero when possible (i.e. when the server is idle), it seldom has a chance to make wrong

decisions when the system load is light. This makes the goodput of system 1 very close to that under the STE policy for small λ . However, as λ increases, more jobs have to wait before getting served. In general, without the exact deadlines, it is not easy to decide whether the deadline of a job with a non-zero waiting time has expired. Any wrong decision made by the LCFS-TO policy to decide to serve a job whose deadline has expired or to discard a job whose deadline has not expired, degrades the goodput of system 1, and thus makes the goodput of the LCFS-TO policy inferior to the goodput of a system under the STE queueing policy.

We can see from Figure 12 that in the case of system 1, LCFS-TO still performs quite close to STE, though the former uses less information. In fact, the performance of LCFS-TO is closely related to the variance of the deadline distribution function. If the variance of the deadline distribution function $F_d(\cdot)$ is small, then $F_d(\cdot)$ provides more accurate information. This improves the correctness of the decisions made by the LCFS-TO policy, and thus makes the goodput of LCFS-TO closer to that of STE.

Since the LCFS-TO policy is much easier to implement than the STE policy, it seems to be a good alternative policy when its performance is close to STE, namely, when a queueing system is operating at light system load or has a low variance deadline distribution.

4 The LCFS-TO Policy for M/M/m Queues

In this section, we generalize the previous results to a queueing system with m servers. Since the time-dependent behavior of an M/G/m queueing model is intractable, we assume that the service times here are exponentially distributed.

4.1 Model Analysis and Policy Description

We assume that there are m identical servers in the M/M/m model. The mean arrival rate and the mean service rate of each server is denoted as λ and μ respectively. The deadline distribution is again an arbitrary function $F_d(\cdot)$.

The LCFS-TO queueing policy in the M/M/m model basically behaves the same as it did in the M/G/1 model. When a job arrives and finds some server(s) idle, it would randomly be assigned to one of the idle servers, which are all identical, and begins its service immediately. If a job arrives and finds no idle server, it would join the buffer, and waits for its service. Every arrival is time-stamped with its arriving time, and whenever a server becomes idle, it selects the job from the buffer with the most recent arriving time — the last-come-first-served service discipline. Again, servers are allowed to become idle only when there is no job waiting in the buffer. When a job enters an idle server, its waiting time is first compared to the waiting time threshold, T_0 . If its waiting time does not exceed T_0 , the job is served. If its waiting time exceeds T_0 , then the job is discarded. Since the processing overhead to enforce the queueing policy is assumed to be negligible, whenever a job is discarded, all the other jobs waiting in the buffer are also discarded from the system. This is because that under the last-come-first-served queueing policy, all the jobs waiting in the buffer always have waiting times longer than that of the job that is selected by the service discipline, and therefore longer than the threshold T_0 . Note that jobs that arrive

and find some idle servers always get served. In other words, the system will discard jobs only when all the m servers are busy. This means when jobs waiting in the buffer are discarded by some server, the rest of the $m - 1$ servers must still be busy serving some jobs, and thus $m - 1$ jobs are left in the system after the discarding. Figure 13 shows the state transition diagram of the LCFS-TO queueing policy. In this diagram, the system state is defined by the number of jobs in the system. In state k , all the k jobs will be in the servers if $k \leq m$, and m of the k jobs will be in the servers if $k > m$. The transitions that go from state k to state $k + 1$ and vice versa correspond to arrival and departure events in the queueing system, while the transitions that go from state k ($k > m$) to state $(m - 1)$ correspond to the case when jobs are discarded from the system.

As defined before, jobs that are served before their deadlines are said to be *successful*, and those that are served after their deadlines are said to be *unsuccessful*.

4.2 Analysis

Just like the M/G/1 queueing model, the performance metric that we are interested in is the fraction of jobs that begin their service before their respective deadlines (the goodput). Or, equivalently, the probability that a job is successful, which is denoted as $P_{s,m}$.

$$P_{s,m} = \int_0^{T_0} [1 - F_d(t)] w_m(t) dt$$

where $w_m(t)$ is the density function of the job's waiting time in the M/M/m queueing system.

Realizing the fact that all arrivals that find some idle server(s) are always successful, we can break down the system states into two large states: the *lightly-loaded state* (LL), which is a set of states in which some server(s) in the system

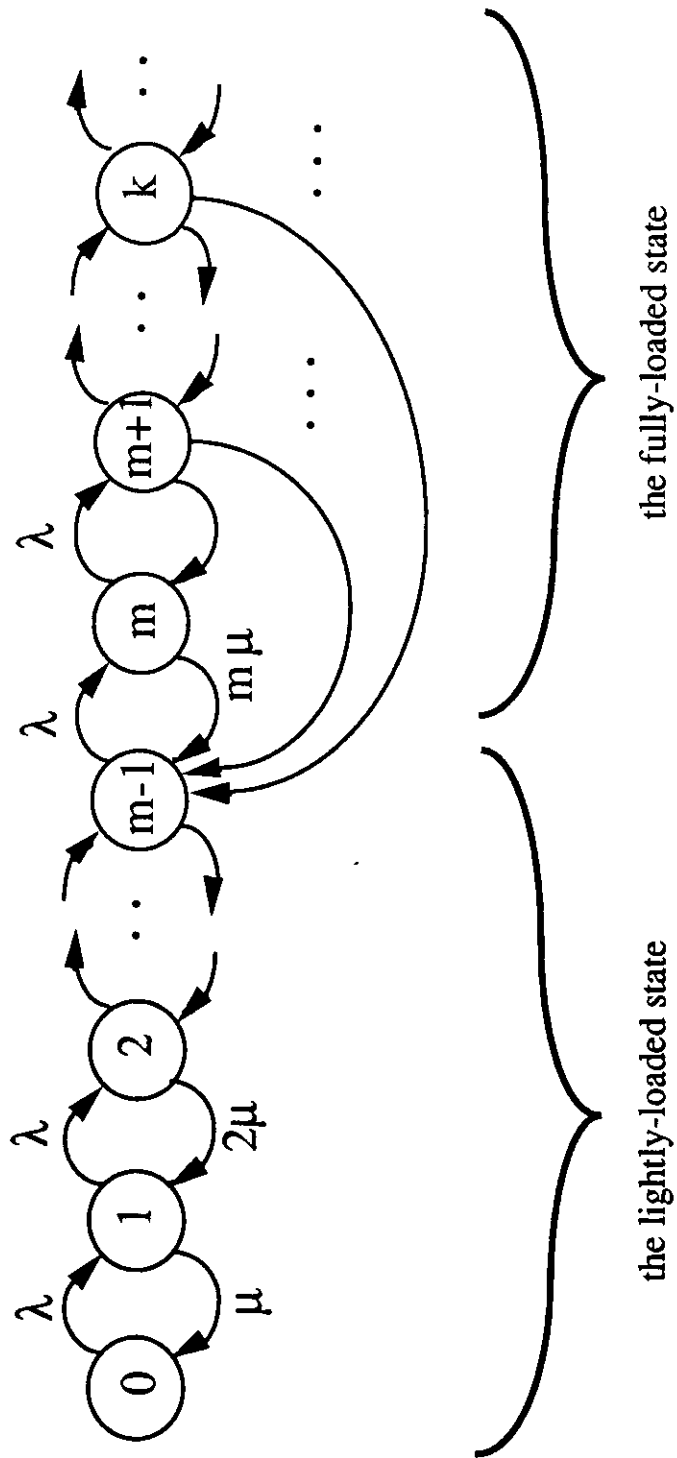


Figure 13: The state transition diagram of an M/M/m queueing system under the LCFS-TO policy.

is (are) idle, and the *fully-loaded state* (FL), which is a set of states in which all the m servers are busy. That is, the system is in the lightly-loaded state when there are less than m jobs in the system, and in the fully-loaded state when there are at least m jobs in the system. As shown in Figure 13, the lightly-loaded state consists of state $0, 1, \dots, m - 1$, and the fully-loaded state consists of the others.

Therefore, conditioning on whether a job arrives in the lightly-loaded state or the fully-loaded state, we have

$$P_{s,m} = \Pr\{\text{a job arrives in LL}\} + \Pr\{\text{a job arrives in FL}\} \int_0^{T_0} [1 - F_d(t)] w_{FL}(t) dt$$

where $w_{FL}(t)$ is the conditional waiting time density function, given that a job arrives in the fully-loaded state.

We will find the expressions for the probability that a job arrives in the fully-loaded state, and the conditional density function for the waiting times, given that a job arrives in the fully-loaded state. In the rest of the section, we use the symbol $M/M/m-(A,B,C)$ to represent an $M/M/m$ queueing system with mean arrival rate A , mean service rate for each of the m servers B , and queueing policy C , which may be last-come-first-served with no rejection scheme (LCFS), or last-come-first-served with time-out rejection scheme (LCFS-TO). For example, the model that we are discussing here is $M/M/m-(\lambda, \mu, \text{LCFS-TO})$.

4.3 The Waiting Time Density Function

Consider a tagged job that arrives in the fully-loaded state of the $M/M/m$ queueing model under the LCFS-TO queueing policy. Since no server is idle in the fully-loaded state, the arrival will join the buffer and waits for its service. Let $W_{T,m}$ be the waiting of our tagged job. The tagged job may be discarded if its waiting time exceeds T_0 . Let $W_{T,m}$ be ∞ if it is discarded. This means that the

density function of $W_{T,m}$ would be a continuous curve in the range $(0, T_0]$, with an impulse at $W_{T,m} = \infty$.

Suppose that our tagged job eventually receives its service. Under the LCFS-TO queueing policy, all the jobs that arrive before our tagged job and are present in the buffer when the tagged job arrives will be scheduled after the tagged job starts its service, and therefore have no effect on the waiting time $W_{T,m}$. Only the m jobs in the servers when the tagged arrives, and the jobs that arrive during the tagged job's waiting time $W_{T,m}$ will affect $W_{T,m}$. Note that servers are allowed to become idle only when there is no job waiting in the buffer, thus the system will surely stay in the fully-loaded state until the tagged job begins its service. With exponential servers, we know that as long as the system stays in the fully-loaded state, the m servers behave wholly as a single large server with mean service rate $m\mu$. And since no jobs will be discarded before our tagged job begins its service, the system looks just like an $M/M/1-(\lambda, m\mu, LCFS)$ system to the tagged job.

Now, imagine that our tagged job arrives in an $M/M/1-(\lambda, m\mu, LCFS)$ system instead, and let W_{FL} be the waiting time. Note that W_{FL} has the same distribution as that of $W_{T,m}$ in the range of $(0, T_0]$. Let random variables Y, N be the residual life of the service time, and the number of arrivals within Y (see Figure 14). Parallel to the relations we have for the $M/G/1$ case, we have the following relations.

$$E[e^{-sW_{FL}}|Y = y, N = n] = e^{-sy}[G_m^*(s)]^n$$

where $G_m^*(s)$ is the Laplace transform of the duration of a busy period in the $M/M/1-(\lambda, m\mu, LCFS)$ queueing system. Unconditioning on N , we have

$$\begin{aligned} E[e^{-sW_{FL}}|Y = y] &= \sum_{n=0}^{\infty} e^{-sy}[G_m^*(s)]^n \frac{(\lambda y)^n}{n!} e^{-\lambda y} \\ &= e^{-(s+\lambda)y} \sum_{n=0}^{\infty} \frac{[\lambda y G_m^*(s)]^n}{n!} \\ &= e^{-[s+\lambda-\lambda G_m^*(s)]y} \end{aligned}$$

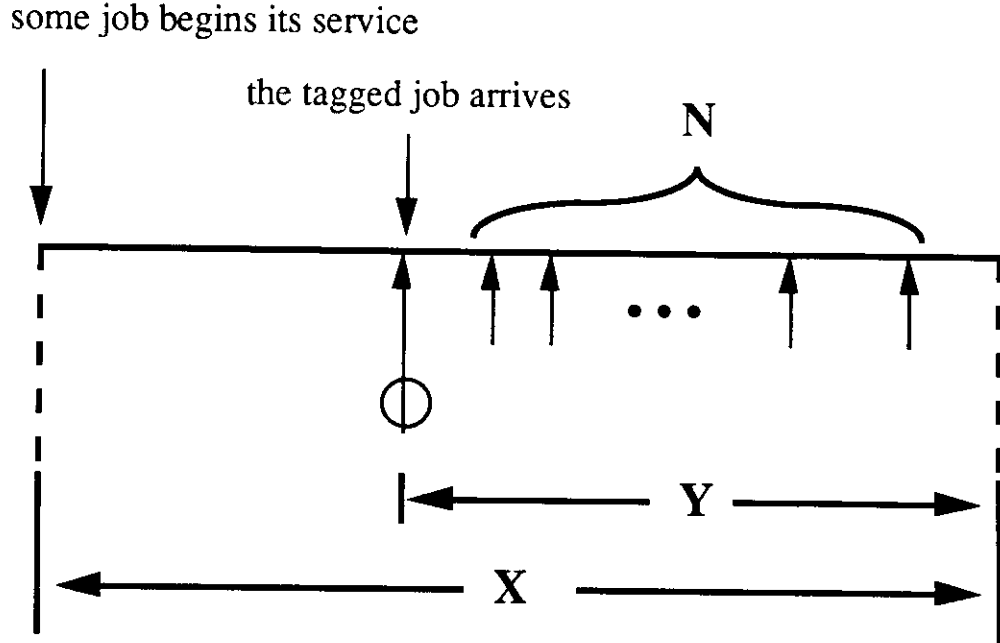


Figure 14: The tagged job arrives in the fully-loaded state.

But we know that with exponential service time, the distribution of the residual service time is the same as that of the service time, $B(x)$. Therefore,

$$\begin{aligned}
 E[e^{-sW_{FL}}] &= \int_{y=0}^{\infty} e^{-[s+\lambda-\lambda G_m^*(s)]y} dB(y) \\
 &= B^*[s + \lambda - \lambda G_m^*(s)] \\
 &= G_m^*(s)
 \end{aligned} \tag{15}$$

Again we find that due to the memoryless property of the M/M/1 queueing system, the distribution of the waiting time is always equal to that of the duration of a busy period. For M/M/1-($\lambda, m\mu, LCFS$), we know that [Klei 75]

$$B^*(s) = \frac{m\mu}{s + m\mu}$$

Hence,

$$\begin{aligned}
 G_m^*(s) &= B^*[s + \lambda - \lambda G_m^*(s)] \\
 &= \frac{m\mu}{s + \lambda - \lambda G_m^*(s) + m\mu}
 \end{aligned}$$

Solving for $G_m^*(s)$ and restricting our solution to the required (stable) case, for which $|G_m^*(s)| \leq 1$ for $\text{Re}(s) \geq 0$, gives

$$G_m^*(s) = \frac{s + \lambda + m\mu - \sqrt{(s + \lambda + m\mu)^2 - 4\lambda m\mu}}{2\lambda} \quad (16)$$

Let $W_{FL}^*(s)$ be the Laplace transform of the density function of W_{FL} , i.e.

$$E[e^{-sW_{FL}}] = W_{FL}^*(s)$$

From equation (15) and (16), we have

$$W_{FL}^*(s) = G_m^*(s) = \frac{s + \lambda + m\mu - \sqrt{(s + \lambda + m\mu)^2 - 4\lambda m\mu}}{2\lambda}$$

This equation can also be inverted (by referring to transform tables) to obtain the probability density function.

$$w_{FL}(t) = \frac{1}{t\sqrt{\rho_m}} e^{-(\lambda+m\mu)t} I_1 \left[2t\sqrt{\lambda m\mu} \right] \quad (17)$$

where $\rho_m = \lambda/m\mu$, and I_1 is the modified Bessel function of the first kind of order one.

4.4 The Probability of Being In The Fully-Loaded State

In this subsection, we will find the probability that a job arrives in the fully-loaded state, denoted as P_{FL} . Let P_{LL} be the probability that a job arrives in the lightly-loaded state. Surely we have,

$$P_{LL} + P_{FL} = 1 \quad (18)$$

Since the arrival process is again Poisson, like the M/G/1 queueing system, the system states of an M/M/m queueing system found by the arrivals still have the same distribution as that of the real system states. Therefore, the probability that an arrival finds the system in the fully-loaded state is equal to the fraction of

time that the system is in the fully-loaded state. We refer the interval of time that the system stays in the lightly-loaded state before transferring to the fully-loaded state as the lightly-loaded period, and the interval of time that the system stays in the fully-loaded state before transferring to the lightly-loaded period as the fully-loaded period. Realizing that the system passes through alternating cycles of the lightly-loaded periods and the fully-loaded periods, we have the following relation.

$$\frac{P_{LL}}{P_{FL}} = \frac{E[\text{duration of a lightly-loaded period}]}{E[\text{duration of a fully-loaded period}]} \quad (19)$$

Let r_m be the probability that a job who arrives in the fully-loaded period is eventually served. Namely,

$$\begin{aligned} r_m &= \Pr\{\text{waiting time} \leq T_0 \mid \text{arriving in the fully-loaded period}\} \\ &= \int_0^{T_0} w_{FL}(t) dt \end{aligned}$$

Before obtaining P_{FL} , let's consider three queueing systems. The first one is the *old model* that we have been discussing. That is, the M/M/m queueing system with the mean arrival rate λ and the mean service rate for each server μ . The queueing policy is LCFS-TO. The second queueing system, referred as *model A*, is another M/M/m queueing system which is almost identical to the old model. The only difference between model A and the old model is that the queueing policy in model A is last-come-first-served with no time-out rejection scheme. All arrivals in model A are served. The last queueing system, referred as *model B*, is an M/M/1 queueing system. The mean arrival rate is $r_m\lambda$, and the mean service rate is $m\mu$. The queueing policy in model B is also last-come-first-served with no time-out rejection scheme. In short,

the old model is M/M/m-(λ, μ , LCFS-TO)

model A is M/M/m-($\lambda, m\mu$, LCFS)

model B is M/M/1-($\lambda, m\mu, \text{LCFS}$)

Let p_k be the probability that model A has k jobs in the system. From the analysis of the classic M/M/ m model [Klei 75], we know that

$$p_k = \begin{cases} p_0 \frac{(m\rho_m)^k}{k!} & k \leq m \\ p_0 \frac{(\rho_m)^k m^m}{m!} & k \geq m \end{cases} \quad (20)$$

where

$$p_0 = \left[\sum_{k=0}^{m-1} \frac{(m\rho_m)^k}{k!} + \frac{(m\rho_m)^m}{m!} \left(\frac{1}{1-\rho_m} \right) \right]^{-1}$$

and

$$\rho_m = \frac{\lambda}{m\mu} \quad (21)$$

Realizing that model A goes through alternating cycles of the lightly-loaded periods and the fully-loaded periods, we have

$$\begin{aligned} & \frac{\text{E}[\text{duration of an LL period} \mid \text{model A}]}{\text{E}[\text{duration of an FL period} \mid \text{model A}]} \\ &= \frac{\text{Pr}\{\text{system in LL} \mid \text{model A}\}}{\text{Pr}\{\text{system in FL} \mid \text{model A}\}} \\ &= \frac{p_0 + p_1 + \dots + p_{m-1}}{\sum_{k=m}^{\infty} p_k} \end{aligned} \quad (22)$$

Furthermore, when model A is in the fully-loaded state, it behaves just like an M/M/1-($\lambda, m\mu, \text{LCFS}$) queueing system in a busy period. Hence

$$\begin{aligned} & \text{E}[\text{duration of an FL period} \mid \text{model A}] \\ &= \text{E}[\text{duration of a busy period} \mid \text{M/M/1-(}\lambda, m\mu)] \\ &= \frac{\frac{1}{m\mu}}{1 - \frac{\lambda}{m\mu}} \\ &= \frac{1}{m\mu - \lambda} \end{aligned}$$

Under model B, we have the mean duration of its busy period from the analysis of M/M/1 models [Klei 75].

$$\begin{aligned} E[\text{duration of a busy period} \mid \text{model B}] &= \frac{\frac{1}{m\mu}}{1 - \frac{r_m\lambda}{m\mu}} \\ &= \frac{1}{m\mu - r_m\lambda} \end{aligned}$$

Comparing the mean duration of a busy period in model B to the mean duration of a fully-loaded period in model A, we have the following relation.

$$\frac{E[\text{duration of a busy period} \mid \text{model B}]}{E[\text{duration of an FL period} \mid \text{model A}]} = \frac{m\mu - \lambda}{m\mu - r_m\lambda}$$

Or

$$\begin{aligned} E[\text{duration of a busy period} \mid \text{model B}] &= \\ E[\text{duration of an FL period} \mid \text{model A}] &\left(\frac{m\mu - \lambda}{m\mu - r_m\lambda} \right) \end{aligned} \quad (23)$$

Now let's go back to the old model. What we want is the probability that the system is in the lightly-loaded state, P_{LL} , and the probability that the system is in the fully-loaded state, P_{FL} . Observe that when the old system is in the lightly-loaded state, it behaves just like model A in the lightly-loaded state, since no time-out rejection scheme can be activated during the lightly-loaded state. Therefore, they must have the same duration of a lightly-loaded period.

$$\begin{aligned} E[\text{duration of a LL period} \mid \text{old model}] &= \\ E[\text{duration of a LL period} \mid \text{model A}] & \end{aligned} \quad (24)$$

Next, consider the case when the old model is in the fully-loaded state. Due to the rejection scheme, only a fraction r_m of the jobs that arrive during a fully-loaded period are served and thus contribute to the full-loaded period. With the similar argument that we used for the M/G/1 case, we can approximate the duration of

a fully-loaded period in the old model with that in model B, an M/M/1 queueing system with reduced arrival rate $r_m\lambda$. That is,

$$\begin{aligned}
& \text{E}[\text{duration of a FL period} \mid \text{old model}] \\
& \approx \text{E}[\text{duration of a busy period} \mid \text{model B}] \\
& = \frac{1}{m\mu - r_m\lambda}
\end{aligned} \tag{25}$$

Plugging equation (24) and (25) into equation (19), we have

$$\begin{aligned}
\frac{P_{LL}}{P_{FL}} &= \frac{\text{E}[\text{duration of a LL period} \mid \text{old model}]}{\text{E}[\text{duration of a FL period} \mid \text{old model}]} \\
&\approx \frac{\text{E}[\text{duration of a LL period} \mid \text{model A}]}{\text{E}[\text{duration of a busy period} \mid \text{model B}]}
\end{aligned}$$

Using equation (22) and (23), we can rewrite the equation above.

$$\begin{aligned}
\frac{P_{LL}}{P_{FL}} &\approx \frac{\text{E}[\text{duration of a LL period} \mid \text{model A}]}{\text{E}[\text{duration of a FL period} \mid \text{model A}]^{\frac{m\mu - \lambda}{m\mu - r_m\lambda}}} \\
&= \frac{p_0 + p_1 + \dots + p_{m-1}}{\sum_{k=m}^{\infty} p_k} \left(\frac{m\mu - r_m\lambda}{m\mu - \lambda} \right)
\end{aligned}$$

Substituting p_k with equation (20) and using equation (21), we have

$$\begin{aligned}
\frac{P_{LL}}{P_{FL}} &\approx \frac{\sum_{k=0}^{m-1} p_k}{\sum_{k=m}^{\infty} p_k} \left(\frac{m\mu - r_m\lambda}{m\mu - \lambda} \right) \\
&= \frac{\sum_{k=0}^{m-1} p_0 \frac{(m\rho_m)^k}{k!}}{\sum_{k=m}^{\infty} p_0 \frac{(\rho_m)^k m^m}{m!}} \left(\frac{m\mu(1 - r_m\rho_m)}{m\mu(1 - \rho_m)} \right) \\
&= \frac{(1 - r_m)\rho_m m!}{(m\rho_m)^m} \sum_{k=0}^{m-1} \frac{(m\rho_m)^k}{k!}
\end{aligned}$$

Note that the pole $(1 - \rho_m)$ is eliminated, and thus this equation is still valid even when $\rho_m = 1$.

Combining equation (18) with the result above, we have

$$\begin{aligned} P_{FL} &= 1 - P_{LL} \\ &\approx 1 - \left[\frac{(1 - r_m)m!}{(m\rho_m)^m} \sum_{k=0}^{m-1} \frac{(m\rho_m)^k}{k!} \right] P_{FL} \end{aligned}$$

Let

$$C = \frac{m!}{(m\rho_m)^m} \sum_{k=0}^{m-1} \frac{(m\rho_m)^k}{k!}$$

Note that C is independent of T_0 . Then, we have

$$P_{FL} \approx [1 + (1 - r_m)C]^{-1} \quad (26)$$

Figure 15, Figure 16 and Figure 17 show P_{FL} obtained via approximation with that obtained via simulation. If we let μ to be fixed at 1, P_{FL} depends on three variables, λ , m and T_0 . In order to plot the results on a 2-dimension graph, we have to make one of the three variables fixed each time. In Figure 15, the threshold T_0 is fixed at 2, and P_{FL} is plotted for the cases that $m = 2$ and $m = 4$. Similarly, in Figure 16, m is fixed at 2, while in Figure 17, λ is fixed at 4. The simulation shows that the approximation is very close to the real queueing system under the LCFS-TO queueing policy.

Obtaining $w_{FL}(t)$ and P_{FL} from equation (17) and (26), we can now get the probability that a job is successful.

$$P_{s,m} \approx \frac{(1 - r_m)C}{1 + (1 - r_m)C} + \frac{1}{1 + (1 - r_m)C} \int_0^{T_0} \frac{e^{-(\lambda+m\mu)t}}{t\sqrt{\rho_m}} I_1 \left[2t\sqrt{\lambda m \mu} \right] dt \quad (27)$$

We can also obtain the the probability that a job is discarded and the probability that a job is unsuccessful.

$$\Pr\{\text{a job is discarded}\} \approx P_{FL} \int_{T_0}^{\infty} \frac{e^{-(\lambda+m\mu)t}}{t\sqrt{\rho_m}} I_1 \left[2t\sqrt{\lambda m \mu} \right] dt \quad (28)$$

$$\Pr\{\text{a job is unsuccessful}\} \approx P_{FL} \int_0^{T_0} F_d(t) \frac{e^{-(\lambda+m\mu)t}}{t\sqrt{\rho_m}} I_1 \left[2t\sqrt{\lambda m \mu} \right] dt \quad (29)$$

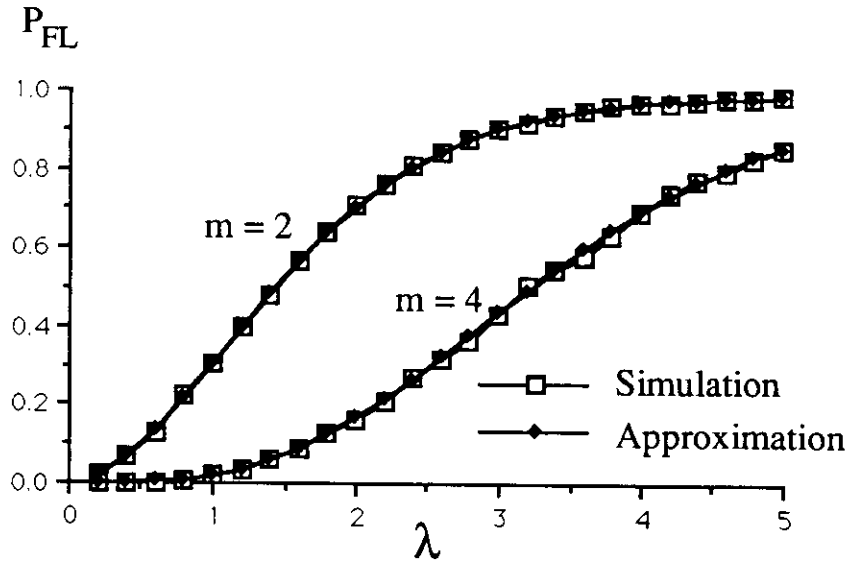


Figure 15: The validation of the approximation for P_{FL} ($\mu = 1, T_0 = 2$).

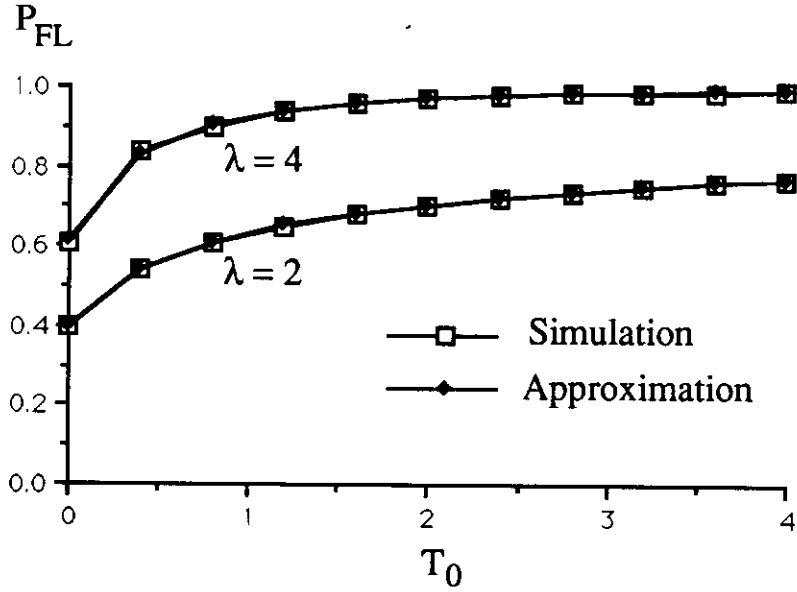


Figure 16: The probability that the system is in the fully-loaded state obtained via simulation and via approximation ($\mu = 1, m = 2$).

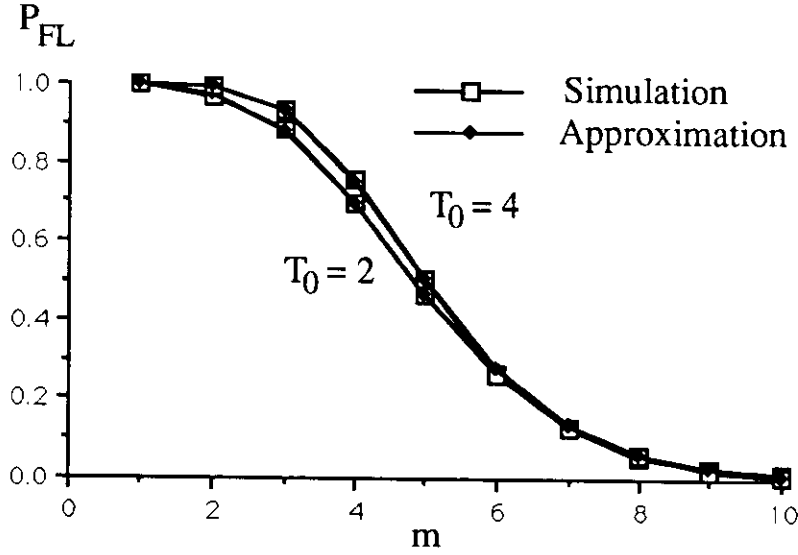


Figure 17: The probability that the system is in the fully-loaded state obtained via simulation and via approximation ($\mu = 1$, $\lambda = 4$).

Differentiating equation (27) with respect to T_0 and setting it to zero gives us the equation for the optimal threshold T_0^* . But first let's find the derivative of r_m with respect to T_0 .

$$\begin{aligned} \frac{dr_m}{dT_0} &= \frac{d}{dT_0} \int_0^{T_0} \frac{1}{t\sqrt{\rho_m}} e^{-(\lambda+m\mu)t} I_1 \left[2t\sqrt{\lambda m \mu} \right] dt \\ &= \frac{1}{T_0\sqrt{\rho_m}} e^{-(\lambda+m\mu)T_0} I_1 \left[2T_0\sqrt{\lambda m \mu} \right] \end{aligned}$$

Now, the derivative of $P_{s,m}$ with respect to T_0 can be obtained.

$$\begin{aligned} \frac{dP_{s,m}}{dT_0} &= \frac{[1 - F_d(T_0)]e^{-(\lambda+m\mu)T_0}}{[1 + (1 - r_m)C]T_0\sqrt{\rho_m}} I_1 \left[2T_0\sqrt{\lambda m \mu} \right] + \\ &\quad \frac{C e^{-(\lambda+m\mu)T_0} I_1 \left[2T_0\sqrt{\lambda m \mu} \right]}{T_0\sqrt{\rho_m}[1 + (1 - r_m)C]^2} \left\{ \int_0^{T_0} \frac{1}{t\sqrt{\rho_m}} e^{-(\lambda+m\mu)t} I_1 \left[2t\sqrt{\lambda m \mu} \right] dt - 1 \right\} \end{aligned}$$

The optimal threshold, T_0^* , must satisfy the following relation.

$$\left. \frac{dP_{s,m}}{dT_0} \right|_{T_0=T_0^*} = 0$$

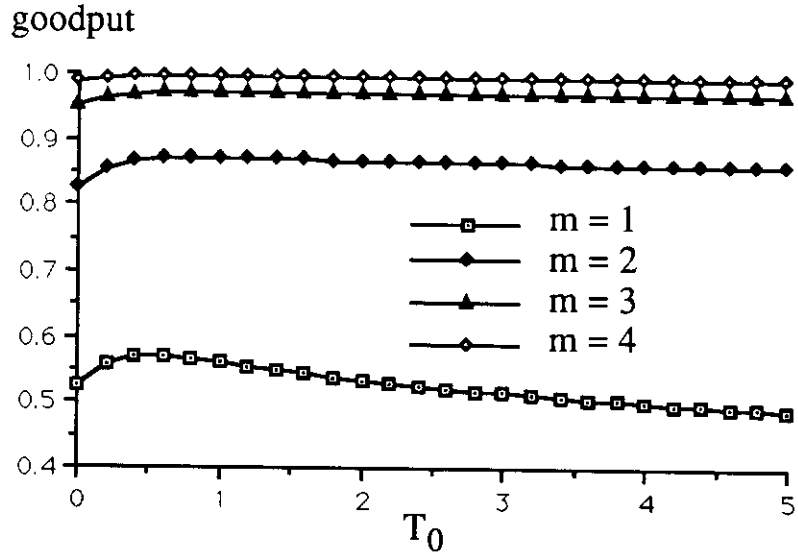


Figure 18: The goodputs of an M/M/m queueing system with $\mu = 1$ and $\lambda = 0.9$.

4.5 Numerical Results

In the following, we show some numerical results for the M/M/m queueing system. As an example, we use function

$$F_1(t) = 1 - \frac{1}{(t+1)^2}$$

which is defined in the previous section as the deadline distribution function. Figure 18 plots the goodputs for the M/M/m system with $\lambda = 0.9$ and $\mu = 1$. Note that when $m = 1$, we have system 1 defined in the previous section. One can find that as m increases, the effective system load decreases, and therefore the goodput increases. When m becomes larger, the effective system load becomes so small that most of the arrivals can find an idle server and begin their service immediately. This reduces the effect of T_0 on the goodput. Therefore, the curve in the figure for $m = 4$ almost stays flat over a wide range of T_0 's.

Figure 19 and Figure 20 show the probability that a job is discarded (P_d) and the probability that a job is served unsuccessfully (P_u). As m increases, not surprisingly P_d decreases. What is interesting is how P_u is related to T_0 . When

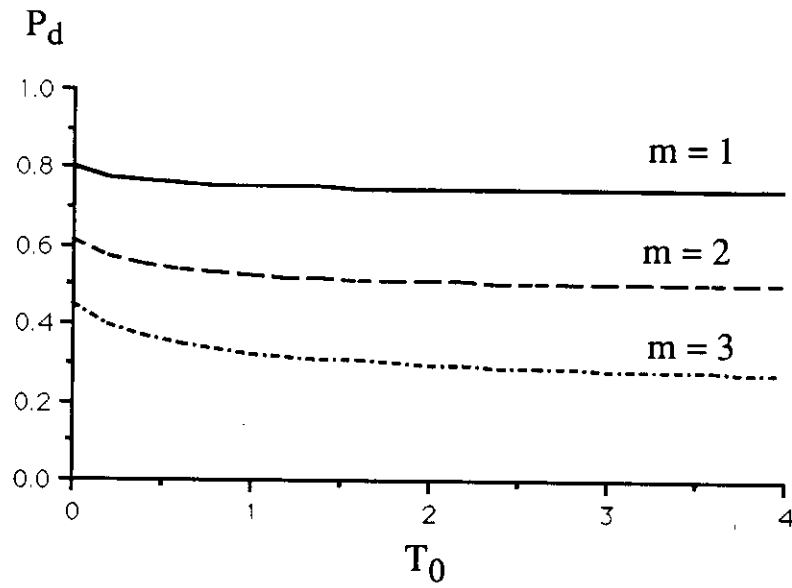


Figure 19: The probability that a job is discarded ($\lambda = 4, \mu = 1$).

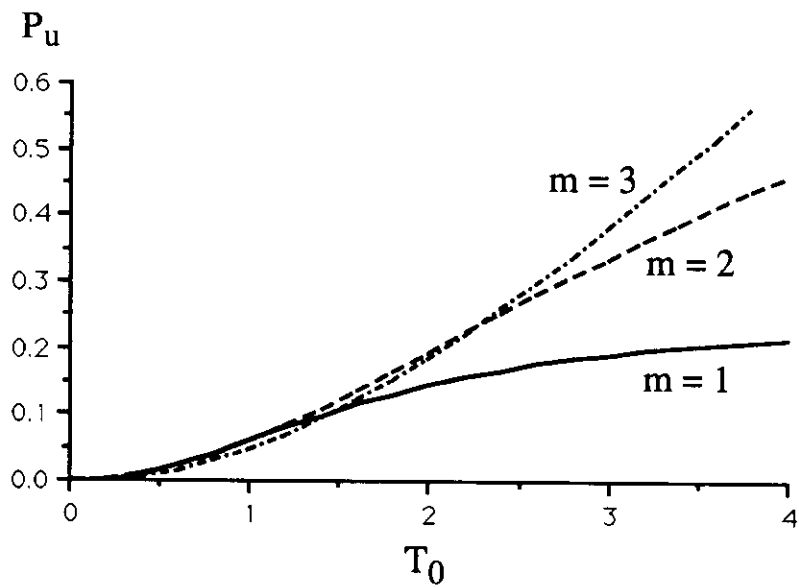


Figure 20: The probability that a job is served unsuccessfully ($\lambda = 4, \mu = 1$).

T_0 is small, P_u 's for the three cases, $m = 1, 2, 3$, are quite close, with the P_u for the case $m = 3$ slightly smaller than that for the others. As T_0 increases, P_u for the case $m = 3$ exceeds the others and has the largest value. This shows that although the system with $m = 3$ is able to serve the largest number of jobs, yet it suffers the highest percentage of useless work when T_0 is large.

5 Conclusion

We considered a real-time queueing system where jobs have randomly selected deadline constraints on their waiting times. We assume that only the deadline *distribution* is available to the server, rather than the exact deadline of each arriving job. If the deadline distribution function is concave, the LCFS-TO queueing policy is known to be the optimal policy which maximizes the fraction of jobs that begin service before their respective deadlines expire.

We developed the approximate models for the goodput of the LCFS-TO queueing policy and obtained closed form solutions for the M/G/1 queueing system and the M/M/m queueing system. Simulations showed that the approximation was very close to the real system. The effect of the deadline distribution function on the system's goodput was also discussed.

Comparisons between the STE queueing policy and the LCFS-TO policy were shown for the M/M/1 queueing model and the M/M/m queueing model. We pointed out that when the system is operating at light system load and has a low variance deadline distribution, the LCFS-TO queueing policy performs very close to the STE policy, and is a good alternative policy due to the simplicity of its implementation.

Bibliography

- [Barr 57] D. Y. Barrer, "Queueing with Impatient Customers and Ordered Service", *Operations Research*, Vol 5, 1957.
- [Cohc 69] J.W. Cohen, "Single Server Queues with Restricted Accessibility", *Journal of Engineering Mathematics*, Vol 3, NO. 4, October 1969.
- [Klei 75] L. Kleinrock, *Queueing Systems, Volumn I: Theory*, Wiley-Interscience, 1975.
- [Klei 76] L. Kleinrock, *Queueing Systems, Volumn II: Computer Applications*, Wiley-Interscience, 1976.
- [Liu 73] C. Liu and J. Layland, "Scheduling Algorithms for Multi-Programming in a Hard-Real-Time Environment", *J. ACM*, Vol.20, 1973.
- [Mok 78] A. Mok and M. Dertouzos, "Multiprocessor Scheduling in a Hard Real-Time Environment", *Proc. 7th Texas Conf. on Comp. Syst.*, Nov. 1978.
- [Tows 88] D. Towsley and S.S. Panwar, "On the Optimality of the STE Rule for Multiple Server Queues that Serve Customers with Deadlines", *COINS Technical Report 88-81*, Department of Computer and Information Science, University of Massachusetts, July 1988.
- [Zhao 91] Z. Zhao, S.S. Panwar and D. Towsley, "Queueing Performance with Impatient Customers", *INFOCOM*,1991.