PRIORITY SCHEDULING IN TANDEM QUEUES

by

Miron Livny and Rajesh Mansharamani

Computer Sciences Technical Report #976

October 1990

Priority Scheduling In Tandem Queues

Miron Livny Rajesh Mansharamani

Computer Sciences Department
University of Wisconsin-Madison.

October 17, 1990

Abstract

Priority scheduling in a system that consists of several resources is not well understood. In this paper, a model of two queues in tandem is used to study the problem of priority placement in a layered system. With the help of simulation we investigate the impact priority scheduling in the first queue has on the waiting time of high and low priority jobs in the other queue. We consider non-preemptive and preemptive scheduling disciplines and show that having priority in the first queue, does benefit high priority jobs in the second queue as well. When the discipline is preemptive we observe that priority scheduling in the first queue may improve the overall system time of low priority jobs. Unlike the case of a single resource system, in a layered system preemptive priority can be used to improve the response time of both high and low priority jobs. These results indicate that a decision to place priority scheduling in one of the resources of a layered system has to take in to account its impact on the waiting time of other resources.

1 Introduction

In modern processing and communication systems different classes of customers are likely to have different response time constraints. For example, in a multimedia communication network, deadlines have to be met for voice packets, but not for data packets. In order to meet these deadlines, 'preferential treatment' has to be given to voice packets, as compared to data packets. The most common method for achieving such treatment, is *priority* scheduling. Jobs that have more stringent timing constraints are given higher priority over ones that don't. In a 'layered system' where a job has to pass through a sequence of layers (like the

ISO seven-layer hierarchy [8]), priority scheduling can be employed by one or more layers. In such systems the response time of a job will be a function of which layers consider priority in their scheduling decisions, and of the scheduling policy each of these layers employs.

Since the objective of priority scheduling is to reduce the response time of high priority jobs, one would naturally opt for priority in all layers. Unfortunately, priority scheduling does not always come for free. Different overheads may be associated with introducing a priority based scheduling discipline [1]. A designer of a layered system may, thus, be faced with a priority placement problem. He will have to select the layers that will provide priority scheduling so that their impact on system performance is maximized. To study the impact of priority scheduling on a layered system, we investigate the effect of priority scheduling on an open queueing system that consists of two queues in tandem. The performance of priority queueing disciplines for a $M/G/1$ queueing system is well understood [3]. Little is known, however, on the performance of priority queues in tandem. In this paper we present the results of our study and show how priority in one queue affects system waiting time. For each of the possible priority placements we used a simulation model to profile the performance of the two queues system for different service time distributions.

Given a system that consists of two queues in tandem and assuming that only one of the two queues can use a priority based scheduling discipline, we are faced with the problem of priority placement. We need to decide whether it is more advantageous to have priority scheduling in the first queue, or the second one. A simple solution to the placement problem is to use priority in the queue that contributes most to the system waiting time of high priority jobs. The problem with this solution is that it is based on the assumption that changes in the scheduling discipline in one queue do not have an impact on the waiting time in the other queue. This assumption, however, does not hold for two queues in tandem. It was shown in [7] that changes in the departure process from the first queue have a significant impact on waiting times in the second queue. Therefore, by introducing a priority based scheduling discipline in the first queue one might increase or decrease the waiting of high priority jobs in the second queue. Such changes may have a significant impact on the system waiting time of high priority jobs, and should thus be considered when priority placement decisions are made.

The arrival pattern to the second queue can be extremely complex on account of priority scheduling in the first queue. Thus, the effect of having a priority queue feed a $FIFO$ queue is not apparent. Previous work in tandem queues mainly deals with the case where both queues are $FIFO$ queues [6] [7]. We are not aware of any results dealing with priority queues in tandem. Owing to the complexity of the departure process from the first queue, we used discrete event simulation to explore the impact priority scheduling in the first queue has on the waiting time of the second queue. In our study we considered both non-preemptive and preemptive scheduling disciplines.

The remainder of the paper is organized as follows : in section 2 we introduce the notation and assumptions. Section 3 focuses on systems with priority in the first queue. It includes a comparison between preemptive and non-preemptive priority systems. In section 4 we consider the issue of priority placement. Finally in section 5 we summarize the work, and present the main conclusions that can be drawn from it. In the attached Appendix we present equations and bounds that are used to the support some of the results presented in the paper.

2 Notation and Assumptions

All the results in this paper are for a queueing system that consists of two queues in tandem. The system and its environment can be completely specified by the arrival process to the first queue, the distribution of the service times for each queue, and finally the queueing disciplines in both queues. In our system two classes of customers arrive viz. high priority customers denoted by 'h', and low priority customers denoted by 'l'. Both of these classes arrive according to an independent Poisson process. Except for a few cases, service times for both queues follow a hyper-exponential [2] distribution. In some of the experiments we assume that all jobs have the same service time for the first queue. The queueing discipline can either be $FIFO$ or head-of-the-line (HOL) priority [3]. When there is priority in a particular queue, it can either be preemptive or non-preemptive.

In our experiments we assume the following: a) both queues have the same service rate. b) both high and low priority classes have the same arrival rate to the system. c) the service times of a job in the two queues are independent random variables. Note that due to the first two assumptions both queues have the same utilization and are also equally utilized by both classes.

Now, that we have described the system and the assumptions, we proceed to describe the solution technique. We used simulation to capture the various queueing disciplines, viz. $FIFO$, HOL-preemption, and HOL-no-preemption. Our simulator was written in the discrete event simulation language DeNet [4]. The service rate of each queue was kept constant at 1.0, so that the total arrival rate could be treated as system utilization. Table 2.1 summarizes the main parameters of the simulator.

Parameter	**Meaning**	**Values**
Queueing Discipline		*FIFO*, priority
λ	Total arrival rate to the system	0.1 – 0.9
C_{x_1}	Coefficient of variation of service times in the first queue	0.0 – 10.0
C_{x_2}	Coefficient of variation of service times in the second queue	1.0 – 7.5
SimTime	Simulation time	1x10^7 – 3x10^7

Table 2.1 Simulation Parameters

The notation used throughout this paper, are presented in Table 2.2. All variables corresponding to high and low priority classes have superscript of 'h' and 'l' respectively. For example $W^h_{2\_NP \rightarrow F}$ is the waiting time of a high priority job in Q_2 in a $NP \rightarrow F$ system. Whenever we refer to response times and waiting times, we imply the average response times and waiting times respectively.

3 The $Priority \rightarrow FIFO$ System

In this section we focus on a two-queue system where the first queue is governed by a priority queueing discipline and the second queue follows the *FIFO* discipline. With the help of the simulation model, we explore the impact that priority scheduling in Q_1 has on the waiting time of high priority jobs in Q_2, W^h_2. Two variants of the *HOL* priority discipline are considered. The first set of results reported in this section profiles the performance of $NP \rightarrow F$ (non-preemptive *HOL* in Q_1) systems. These results show that the impact of priority in Q_1 on W^h_2 is sensitive to changes in system utilization, ρ, the coefficient of variation of the service time in Q_1, C_{x_1}, and the coefficient of variation of the service time in Q_2, C_{x_2}. The second set of results discussed in this section is for $PR \rightarrow F$ (preemptive *HOL* in Q_1) systems. We use these results to analyze the differences between the impact preemptive and non-preemptive scheduling in Q_1 has on the performance of both high and low priority jobs in Q_2.

Notation	**Description**
HOL	Head-of-the-line queueing discipline
NP	Non-preemptive HOL
PR	Preemptive Resume HOL
F	First In First Out queueing discipline ($FIFO$)
Q_i	The i^{th} queue in the system (where i is either 1 or 2)
C_{x_i}	Coefficient of variation of the service times in Q_i
ρ	Utilization of each queue
$x \rightarrow y$	Queueing discipline x in Q_1, queueing discipline y in Q_2
W_i	Average waiting time in Q_i over all classes (the system depends on the context)
$W_{i\_x \rightarrow y}$	Average waiting time in Q_i over all classes in the system with queueing disciplines $x \rightarrow y$
W_i^c	Mean waiting time of a job of class c in Q_i (the system depends on the context)
$W_{x \rightarrow y}^c$	Mean total waiting time of a job of class c in the system with queueing disciplines $x \rightarrow y$
$W_{i\_x \rightarrow y}^c$	Mean waiting time in Q_i of a job of class c in the system with queueing disciplines $x \rightarrow y$
$T_{i\_x \rightarrow y}^c$	Mean response time in Q_i of a job of class c in the system with queueing disciplines $x \rightarrow y$

Table 2.2 Notation

3.1 $NP \rightarrow FIFO$

Figure 1 presents the gain in Q_2 waiting time accrued by high priority jobs as a result of non-preemptive HOL scheduling in Q_1. The gain, G, is defined as

$$G = \frac{W^h_{2\_F\rightarrow F} - W^h_{2\_NP\rightarrow F}}{W^h_{2\_F\rightarrow F}} * 100$$

and is displayed as a function of system utilization for different settings of the simulation parameters. A positive gain means that by introducing priority to the first queue we reduce the waiting time of high priority jobs in **both** queues. It is important to note here that since the service discipline in Q_1 is non-preemptive, what is gained by high priority jobs in Q_2 is lost by low priority jobs. In other words $W^h_{2\_NP\rightarrow F} + W^l_{2\_NP\rightarrow F}$ is equal to $W^h_{2\_F\rightarrow F} + W^l_{2\_F\rightarrow F}$. As is discussed in the Appendix, this is due to the observation that $NP(HOL)$ scheduling does not change the distribution of inter-departure times from Q_1. Therefore, the expected waiting time of jobs [1] in Q_2 in a $NP \rightarrow F$ system, $W_{2\_NP\rightarrow F}$ is the same as that in a $F \rightarrow F$ system. The results presented in Figure 1, are a clear display of the impact priority scheduling in the first queue has on the waiting time of high priority jobs in the second queue. We observe that G is upto 10% for a system where both service times are exponentially distributed ($C_{x_1} = C_{x_2} = 1$), and that it can reach 35% for a system where $C_{x_1} = 10$, and $C_{x_2} = 1$.

In order to understand how priority in Q_1 affects the waiting time of high priority jobs in Q_2 we have to follow the behavior of the $NP \rightarrow F$ system after a job, J, who had a very long service time in Q_1, departs and joins Q_2. While J was served by the server of Q_1, it is very likely that Q_2 served most, if not all, the unfinished work that was in Q_2 just before J's service started. Therefore, when J joins Q_2 it is likely to see a short or even empty queue. The job to follow J into the second queue is also likely to see a small queue when it joins Q_2. Because of the priority scheduling in Q_1 the job that follows J is more likely to be a high priority job than a low priority job. Regardless of the priority of job J, if a high priority job arrives during the service period of J, the HOL scheduling discipline in Q_1 guarantees that a high priority job will follow J into Q_2. Note that, had the scheduling discipline of Q_1 been $FIFO$, it would have been equally likely for low priority and high priority jobs to follow J. Such an increase in the probability that a high priority job follows jobs like J, reduces the the mean queue length seen by high priority jobs upon arrival to Q_2 and thus increases the queue size seen by low priority jobs. This in turn causes a decrease in the waiting time of high

[1] When the class of the job is not specified, the statistics is for ALL jobs regardless of their priority

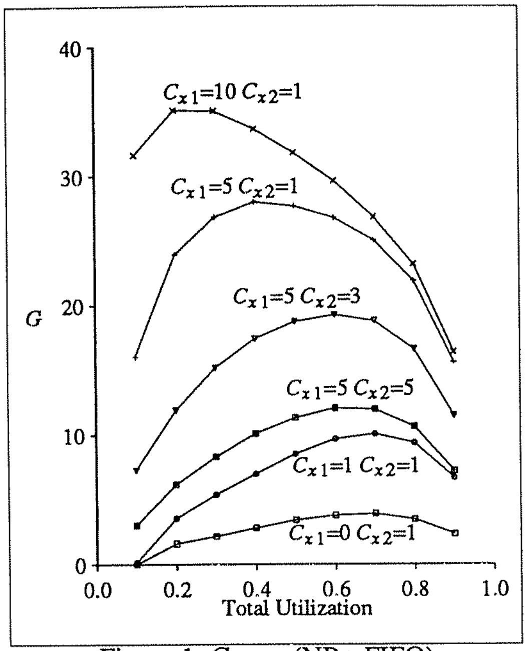

Figure 1. G vs ρ (NP->FIFO)

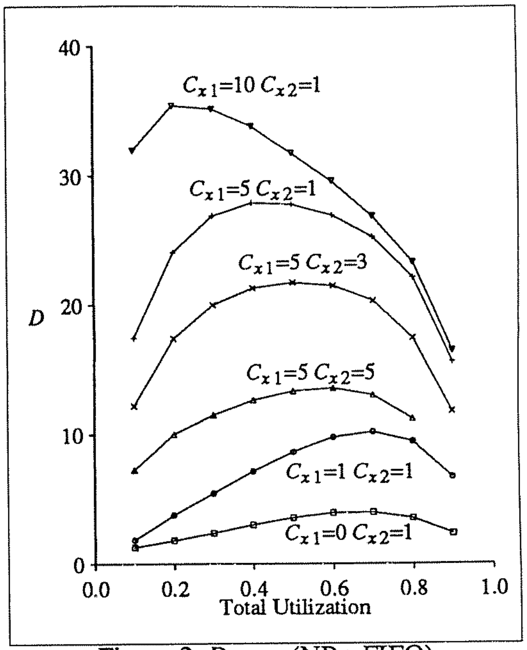

Figure 2. D vs ρ (NP->FIFO)

priority jobs and an increase in the waiting time of low priority jobs.

To verify this explanation we used simulation results to compute

$$D = \frac{L_2|_{Arrival} - L_2^h|_{Arrival}}{L_2|_{Arrival}} * 100$$

where, $L_2^h|_{Arrival}$, is the mean queue size seen by high priority customers, and $L_2|_{Arrival}$ is the mean queue length seen by all jobs, regardless of their priority, upon arrival to Q_2. Figure 2 shows D as a function of ρ for different settings of C_{x_1} and C_{x_2}. D, which can be viewed as the gain in queue size due to priority, has a similar shape and range of values as the gain in waiting time (Figure 1). Note that the queue size as seen by low priority jobs is by definition as far away from the average as the one seen by high priority jobs.

The relative impact of a job, A, which has a long execution time in Q_1 on the high priority job, B, that follows it, depends on the amount of unfinished work in Q_2 at the time Q_1 starts serving A. If this amount is smaller than or about the same as the service time of A, the impact of A on the waiting time of B is significant. If the service time of A is much smaller than this amount, then A will only have a marginal impact on the waiting time of B in Q_2. This explains why we observe a decrease in D when the expected queue size of Q_2 increases. Figure 3 shows how an increase in C_{x_2}, which causes an increase in the expected queue size of Q_2, leads to a decrease in D. As pointed out earlier such a decrease in D entails a decrease

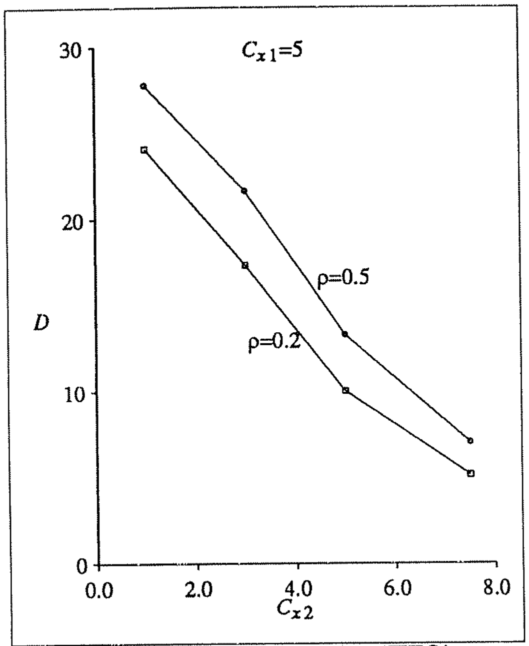

Figure 3. D vs C_{x2} (NP->FIFO)

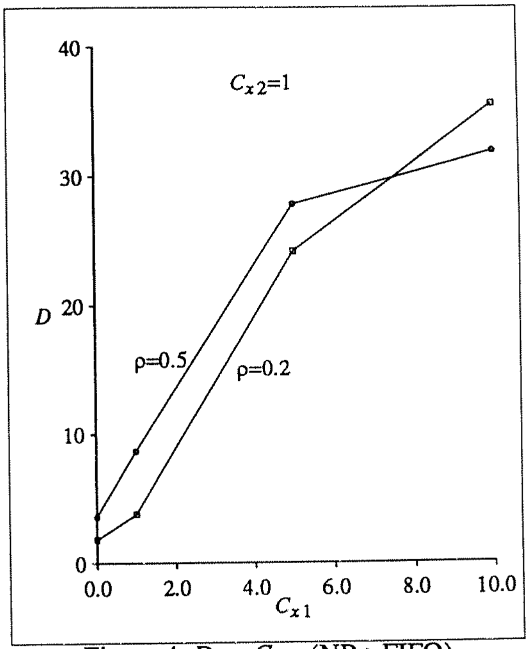

Figure 4. D vs C_{x1} (NP->FIFO)

in G. This behavior can also be observed in Figure 1 and Figure 2 when curves with the same C_{x_1} and different C_{x_2} values are compared. The utilization of the system has also an impact on the mean queue size of Q_2. This explains why we observe in Figure 1 that for the high range of system utilization, an increase in ρ entails a decrease in G.

System utilization has also an impact on the probability that a high priority job immediately follows A into Q_2. Since ρ is determined by the system arrival rate, an increase in ρ implies an increase in the probability that a high priority job will arrive while A is served by the server of Q_1, and thus increases the probability A will be followed by a high priority job. As was discussed earlier the impact of priority scheduling in Q_1 on the waiting time of high priority jobs in the second queue depends upon this probability. The larger the probability, the more significant the impact. This explains why for the low range of system utilizations we observe in Figure 1 and Figure 2 an increase in the gain when ρ increases.

From what we have seen so far, system utilization has an impact on two forces that act in opposite directions as far as the impact of priority scheduling in Q_1 on the waiting time of high priority jobs in Q_2 is concerned. On the one hand, an increase in ρ makes it more likely that a high priority job will follow a long job. On the other hand, the same increase leads to an increase in the expected queue size of Q_2. This explains why for a given C_{x_1} and C_{x_2}, both G (in Figure 1) and D (in Figure 2) initially rise with ρ, but then turn around and start falling. In the lower range of utilizations the first force has the upper hand, whereas in the range of high utilizations the increase in the expected queue length of Q_2 is the dominant force. In Figure 1 we see that as C_{x_1} increases, the point of inflection of G (the utilization after which the gain starts falling) shifts to lower and lower utilizations, as we increase C_{x_1}. As we saw earlier, for a given C_{x_1} the number of customers in Q_2 increased with ρ, and this leads to a decrease in G after the point of inflection. As C_{x_1} increases the mean queue length of Q_2 also increases [7] for a given ρ. This in turn causes D to decrease as discussed above and as demonstrated by the simulation results presented in Figure 2. Therefore the point of inflection of G decreases when C_{x_1} increases.

Now that we have examined the effects of C_{x_2} and ρ on G, we move on to investigating the impact of C_{x_1} on G. We readily observe from Figure 1 that as C_{x_1} is increased, G also increases. The main reason is that as C_{x_1} becomes larger, the mean 'length' of a 'long' job also becomes larger. Moreover, the probability of high priority jobs following a 'long' job also increases with C_{x_1}. The longer is the job in service, the more likely is it that the high priority job following it sees a smaller queue length upon arrival to Q_2, than what is seen by low priority jobs. Therefore, as shown in Figure 4, D increases with C_{x_1} which entails an increase in G.

In the Appendix, we present an alternative explanation for why the waiting time of high priority jobs in Q_2 of a $NP \rightarrow F$ system, $W^h_{2\_NP \rightarrow F}$ is smaller than the waiting time of low priority jobs, $W^l_{2\_NP \rightarrow F}$. This explanation is based on the observation that in a $NP \rightarrow F$ system the mean idle time in Q_2 during the time that a job spends in Q_1, is smaller for high priority jobs than for low priority jobs. As we show in equations A.(2) and A.(3) in the Appendix, this idle time is the only factor that differentiates the mean response times of high priority jobs from those of low priority jobs.

3.2 $PR \rightarrow FIFO$

In the previous subsection it was shown that in a $NP \rightarrow F$ system high priority jobs and low priority jobs spend different amounts of time in the second queue, even though it is a $FIFO$ queue. The average waiting time in the second queue, however, was the same as in a $F \rightarrow F$ system. This is not true any more in the case of a $PR \rightarrow F$ system. By introducing preemptive scheduling in Q_1 we may change the distribution of inter-departure times of jobs from the first queue, and thus change the average waiting time in the second queue, W_2. Therefore, while this distribution is the same for a $F \rightarrow F$ and a $NP \rightarrow F$ system, $PR \rightarrow F$

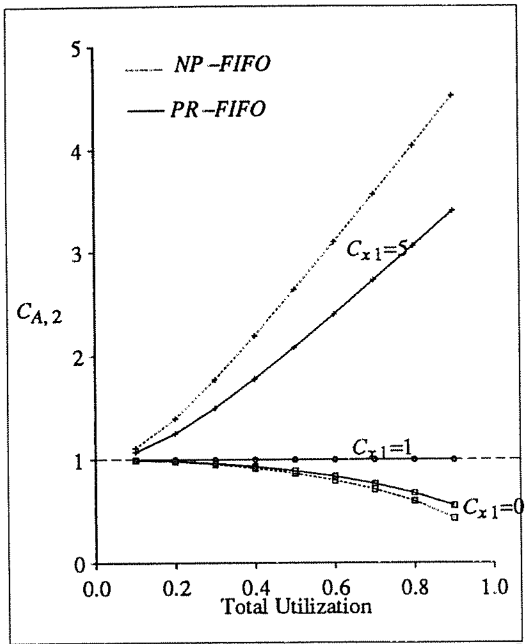

Figure 5. Coeff. of Variation of Arrivals to Q_2

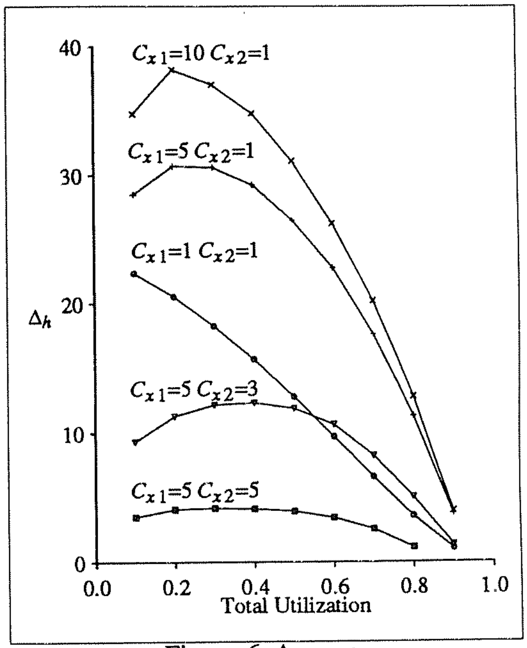

Figure 6. Δ_h vs ρ

may have a different distribution. Preemption in Q_1 does not change the mean inter-departure time. It can however, change higher moments of the inter-departure time distribution. Figure 5 presents the coefficient of variation of the inter-departure times, $C_{A,2}$, for $PR \rightarrow F$ and $NP \rightarrow F$ systems, as was measured by simulation. They are plotted as a function of ρ for different values of C_{x_1}. Note that since non-preemptive priority does not change the departure process of jobs from Q_1, $C_{A,2}$ for a $NP \rightarrow F$ system is the same as that for a $F \rightarrow F$ system. For all the simulations where $C_{x_1} > 1$, the $PR \rightarrow F$ system has a smaller $C_{A,2}$ than the $NP \rightarrow F$ system. By preempting low priority jobs in Q_1, the $PR \rightarrow F$ system reduces the variation in the inter-arrival times to Q_2. In [7] it was shown that a smaller variance in inter-arrivals time to Q_2 for the same mean, reduces the waiting time in Q_2. We would thus expect that a high priority job will spend less time in the second queue of a $PR \rightarrow F$ system than in the second queue of a $NP \rightarrow F$ system. The results presented in Figure 6, depicting

$$\Delta_h = \frac{W^h_{2\_NP \rightarrow F} - W^h_{2\_PR \rightarrow F}}{W^h_{2\_NP \rightarrow F}} * 100$$

versus ρ, meet this expectation. Δ_h which is the percentage by which the waiting time of high priority jobs in Q_2 of a $PR \rightarrow F$ system, $W^h_{2\_PR \rightarrow F}$, differs from the waiting time of the same class of jobs in Q_2 of a $NP \rightarrow F$ system ,$W^h_{2\_NP \rightarrow F}$, is positive for all systems that were simulated. This implies that the $PR \rightarrow F$

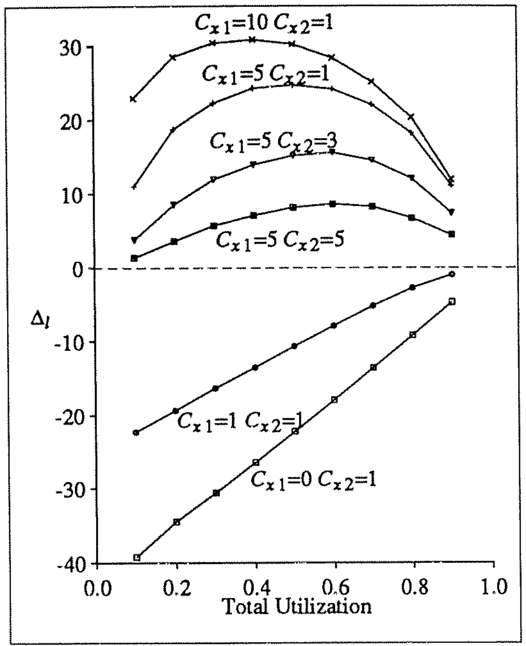

Figure 7. Δ_l vs ρ

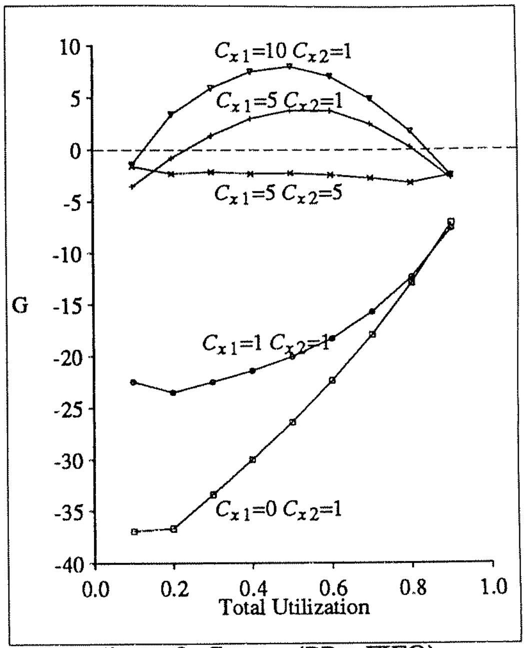

Figure 8. G_l vs ρ (PR->FIFO)

system has a smaller W_2^h than the corresponding $NP \rightarrow F$ system. Even in the cases where preemption did not decrease $C_{A,2}$, (e.g. $C_{x_1} = C_{x_2} = 1$), it did reduce the waiting time of high priority jobs in Q_2. [2]

Now that we have seen that as a result of preemption in Q_1 the waiting time of high priority jobs is smaller in both queues, the next question to ask is: how does preemption in the first queue affect the waiting time of low priority jobs in Q_2? In the previous subsection we saw that in the case of a $NP \rightarrow F$ system, what is gained by high priority jobs in the second queue is lost by low priority jobs. This was due to the observation that priority scheduling in the first queue does not change the average waiting time in the second queue, W_2. In the case of a $PR \rightarrow F$ system it was argued earlier that the total waiting time of Q_2 may be smaller than in a $NP \rightarrow F$ system. This may lead to an interesting situation where in a $PR \rightarrow F$ system, **both** high and low priority jobs have a smaller waiting time in Q_2. The results presented in Figure 7 show that this is actually the case for some of the systems that were simulated. Figure 7 presents

$$\Delta_l = \frac{W^l_{2\_NP \rightarrow F} - W^l_{2\_PR \rightarrow F}}{W^l_{2\_NP \rightarrow F}}$$

versus ρ for different values of C_{x_1} and C_{x_2} as measured by simulation. For certain combinations of C_{x_1} and C_{x_2} values, low priority jobs spend less time in the $FIFO$ queue, when there is preemption in Q_1 than when

[2] The curve for the case $C_{x_1} = 0, C_{x_2} = 1$ is almost the same as the one for $C_{x_1} = C_{x_2} = 1$, and is therefore not included in Figure 5.

there is not. When $C_{x_1} = 10$ and $C_{x_2} = 1$, low priority jobs can reduce their waiting time in Q_2 by more than 25% as a result of preemptive scheduling in the first queue. Although this result seems counter-intuitive, the reason for it is just what was explained above. Preemption in Q_1 reduces the variance of inter-arrival times to Q_2, thus reducing the average queue size. Therefore, as C_{x_1} increases, low priority jobs are more likely to see a smaller Q_2 when there is preemption in Q_1, than when there is not. Since $C_{A,2}$ is the same for the $NP \rightarrow F$ as well as the $F \rightarrow F$ system, it is possible that increasing C_{x_1} causes low priority jobs to perform better in Q_2 than even the $F \rightarrow F$ system. This is what we observe in Figure 8, where the percentage gain

$$G_l = \frac{W^l_{2\_F \rightarrow F} - W^l_{2\_PR \rightarrow F}}{W^l_{2\_F \rightarrow F}} * 100$$

is plotted versus ρ and C_{x_1}.

4 Placement Of Priority

The previous section focused on the performance of high and low priority jobs in the second queue. In this section we shift our attention to overall system performance, and measure the system waiting time of the two-queue system with priority scheduling in either one of the queues. The results of these measurements shed light on the priority placement problem. As was discussed in the introduction, a simple approach to this problem is to place priority in the queue that contributes most to the waiting time of high priority jobs in the $F \rightarrow F$ system. Figure 9 presents the ratio of the waiting times in each queue in a $F \rightarrow F$ system

$$r_w = \frac{W_{1\_F \rightarrow F}}{W_{2\_F \rightarrow F}}$$

as a function of system utilization, for different values of C_{x_1} and C_{x_2}.[3] When $r_w > 1$ a job waits longer, on the average, in Q_1 than in Q_2, and vice versa when $r_w < 1$. The horizontal line at $r_w = 1$, corresponds to the case $C_{x_1} = C_{x_2} = 1$ where the waiting time is the same in both queues. For all the cases that were simulated, this is the only one where r_w does not change as a function of ρ. In all the other cases, r_w is a monotonic decreasing function of ρ. This may lead to a situation, as in the case of $C_{x_1} = 5$ $C_{x_2} = 4$, where $r_w > 1$ in one range of system utilizations and $r_w < 1$ in another range. In order to evaluate this approach to the placement problem, we simulated each of the systems presented in Figure 9 with priority scheduling

[3] Relative queue utilization can also affect r_w but we wanted to study the effects of only C_{x_1} and C_{x_2} on r_w; hence both queues had the same utilization.

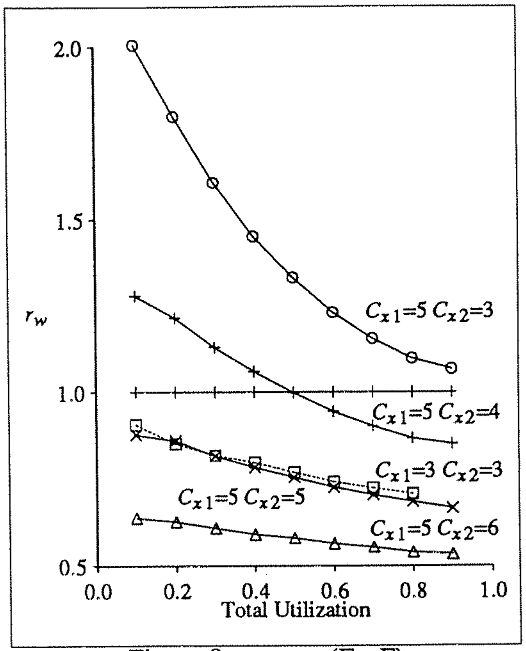

Figure 9. r_w vs ρ (F->F)

in one of the two queues. For each class c, which can be either l or h, and priority discipline x, which can be either NP or PR, the following improvement factors were computed

$$I_{x,1}^{c} = \frac{W_{F\rightarrow F}^{c} - W_{x\rightarrow F}^{c}}{W_{F\rightarrow F}^{c}} * 100$$

and

$$I_{x,2}^{c} = \frac{W_{F\rightarrow F}^{c} - W_{F\rightarrow x}^{c}}{W_{F\rightarrow F}^{c}} * 100$$

The factor $I_{x,j}^{c}$ measures the improvement in the system waiting time of class c jobs when priority discipline x is placed in queue j as compared to their waiting time in a $F \rightarrow F$ system. The larger the value of the improvement factor of class c, the better the performance of class c over the $F \rightarrow F$ system. Note that we expect to see positive values for $I_{x,j}^{h}$, and negative values for $I_{x,j}^{l}$.

Figure 10 displays the improvement factors as a function of ρ when $C_{x_1} = C_{x_2} = 1$. Three observations are apparent. First, for both $x = PR$ and $x = NP$, $I_{x,1}^{h} > I_{x,2}^{h}$ which is in contrast to Figure 9 which led us to believe that priority could be placed in either queue. This is due to the fact that priority in Q_1, benefits high priority jobs in Q_2 too. Second, it can be clearly seen that $I_{PR,j}^{h} > I_{NP,j}^{h}$, which is what we expect. Third, the loss of low priority jobs, is equal to the improvement of their high priority counterparts, that is

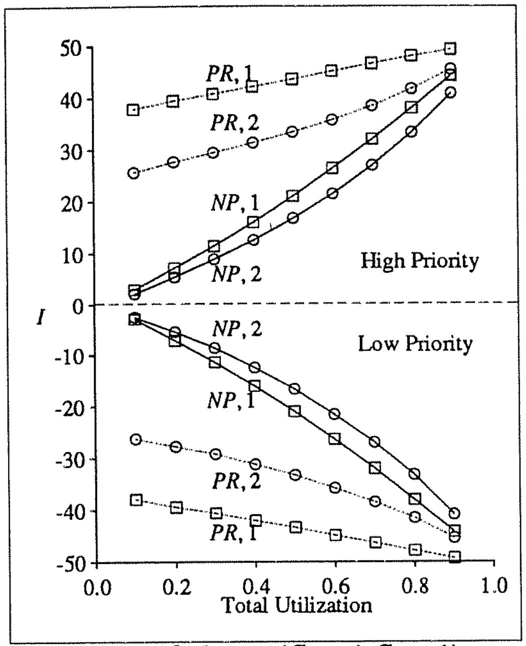

Figure 10. I vs ρ (C_{x1}=1 C_{x2}=1)

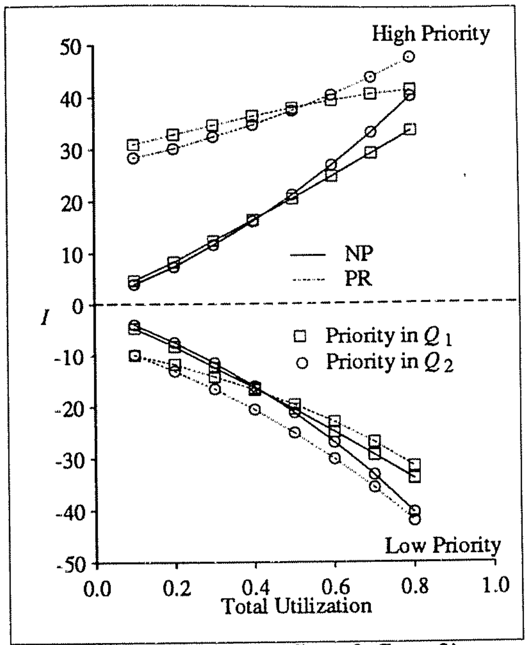

Figure 11. I vs ρ (C_{x1}=3 C_{x2}=3)

$I^l_{x,j} = -I^h_{x,j}$.

Moving onto $C_{x_1} = C_{x_2} = 3$, the naive approach opts for priority in Q_2 for all ρ, as shown in Figure 9. From Figure 11, we find that such is not always the case. At lower utilizations, $I^h_{x,1} > I^h_{x,2}$ for both $x = PR$ and $x = NP$, which implies that priority should be placed in Q_1. As the utilization increases, r_w decreases (Figure 9) to an extent that the gain of high priority jobs in Q_2 is not large enough to compensate for the relative increase in the waiting time in Q_2. Therefore, it is more beneficial to place priority in Q_2 at higher utilizations as shown by the crossover points for high priority in Figure 11. Apart from these crossover points, another observation from Figure 11 is that the curves for low priority are not as spread out as those for high priority. In other words by having PR in one of the queues, high priority jobs gain much more than what low priority jobs lose. It is also interesting to note that when $\rho > 0.4$, $I^c_{PR,1} > I^c_{NP,1}$ for both $c = l$ and $c = h$. This means that by introducing preemption, both classes gain. High priority jobs gain more, and low priority jobs lose less. The improvement curves when $C_{x_1} = C_{x_2} = 5$, as shown in Figure 12, depict a similar behavior to case $C_{x_1} = C_{x_2} = 3$, the main difference being that the crossover points occur at lower utilizations.

The above graphs displayed the behavior of the improvements when $C_{x_1} = C_{x_2}$. In the next two figures we examine cases where $C_{x_1} > C_{x_2}$. Depicted in Figure 13 are the improvement curves when $C_{x_1} = 5$, $C_{x_2} = 3$.

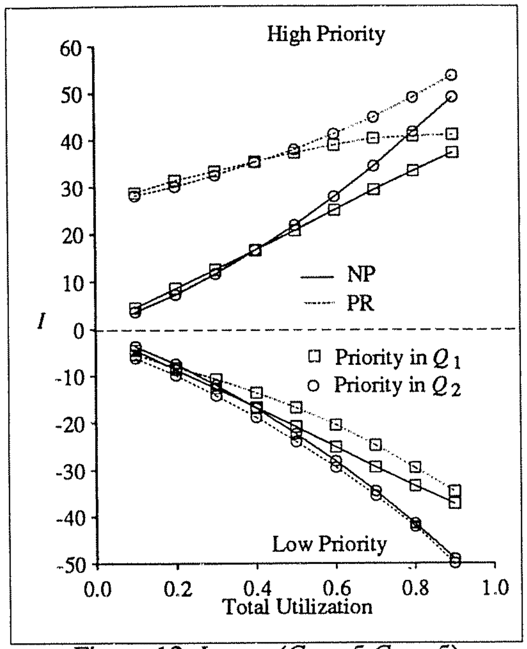

Figure 12. I vs ρ (C_x1=5 C_x2=5)

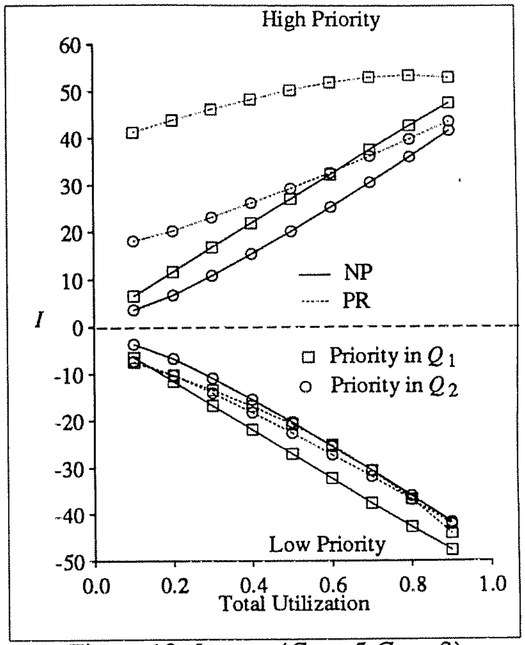

Figure 13. I vs ρ (C_x1=5 C_x2=3)

Figure 9 tells us that $W_{1\_F \rightarrow F} > W_{2\_F \rightarrow F}$. Hence according to the simple rule, priority should be placed in Q_1 which is verified in Figure 13 to be the right placement, as $I^h_{x,1}$ is always higher than $I^h_{x,2}$. However, unlike Figure 9 which depicts that W_1 is close to W_2 at high utilizations, we see that this is not so with regard to the improvement factors. $I^h_{x,1}$ is much farther away from $I^h_{x,2}$ than what one would expect from Figure 9. From Figure 13 it can be further observed that $I^l_{PR,1}$ is higher than $I^l_{PR,2}$ as well as $I^l_{NP,1}$, and at the same time $I^h_{PR,1}$ is also higher than both $I^h_{PR,2}$ and $I^h_{NP,1}$. This seems contradictory, since it implies that both high and low priority jobs are benefiting. But as shown in section 3.2, preemption in the first queue, reduces the average waiting time in the second queue. Thus in this case preemption in Q_1 will give the best performance for both high and low priority jobs. On moving C_{x_2} closer to C_{x_1}, for example $C_{x_1} = 5, C_{x_2} = 4$, the simple approach places priority in Q_1 when $\rho \leq 0.5$, and in Q_2 when $\rho > 0.5$, as shown in Figure 9. On the contrary Figure 14 depicts that the actual crossover point occurs much later, that is, $I^h_{NP,1}$ falls below $I^h_{NP,2}$ only after $\rho = 0.8$, with a similar trend for the preemptive improvements too. We also note that on increasing C_{x_2} from 3 (Figure 13) to 4 (Figure 14), $I^h_{PR,2}$ has come closer to $I^h_{PR,1}$, but $I^l_{PR,2}$ has moved farther away from $I^l_{PR,1}$, which is not in accordance with what one might expect.

We finally review a case where $C_{x_2} > C_{x_1}$, that is $C_{x_1} = 5, C_{x_2} = 6$. Figure 9 suggests that priority be placed in Q_2, and this is exactly what should be done as seen from Figure 15. In Figure 15, $I^h_{x,2}$ is much

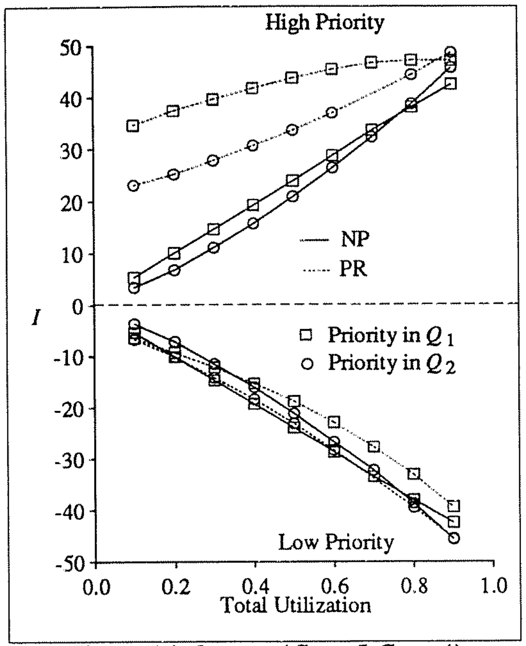

Figure 14. I vs ρ (C_{x1}=5 C_{x2}=4)

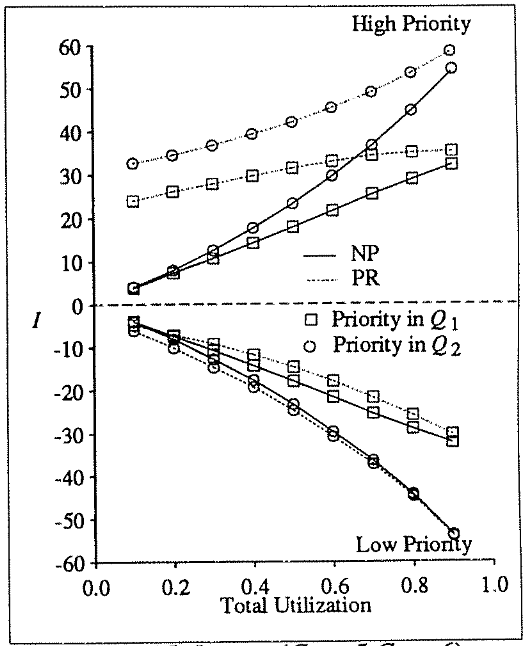

Figure 15. I vs ρ (C_{x1}=5 C_{x2}=6)

higher than $I^h_{x,1}$ and this improvement increases with utilization (which can also be guessed from Figure 9). A further observation is that the curve for $I^h_{PR,1}$ levels off after $\rho = 0.8$, but the one for $I^h_{PR,2}$ keeps increasing with ρ. A similar trend can also be noted for the NP curves.

Summarizing the results of the above experiments we can draw the following conclusions : firstly, using relative waiting times in a $F \rightarrow F$ system as the guideline for priority placement (Figure 9), is biased towards placing priority in Q_2, and hence the curves in Figure 9 need to be shifted upwards by some amount if that guideline is to be used. The reason for the bias towards Q_2 is that the simple approach does not account for the fact that priority scheduling in Q_1 leads to a decrease in the waiting time in Q_2. Secondly, we can conclude that when $C_{x_2} > C_{x_1} > 1$, it is more advantageous to have priority in Q_2 than in Q_1. On the other hand whenever $C_{x_1} > C_{x_2}$, placing priority in Q_1 turns out to be more beneficial in most cases, barring those in which $\rho > 0.8$ and C_{x_2}'close enough' to C_{x_1} (for example $\rho = 0.9$, for $C_{x_1} = 5, C_{x_2} = 4$). Thirdly, when $C_{x_1} = C_{x_2}$, priority should be placed in Q_1 for low utilizations ($\rho < 0.4$), and in Q_2 for high utilizations ($\rho > 0.5$). Finally, we note that though $I^h_{PR,j} > I^h_{NP,j}$, and $I^l_{NP,2} > I^l_{PR,2}$, $I^l_{PR,1}$ is greater than $I^l_{NP,1}$ in many cases. In other words the system waiting time of low priority jobs is smaller with preemption than without.

The last observation drawn above could be restated as follows : given that preemption in the first queue

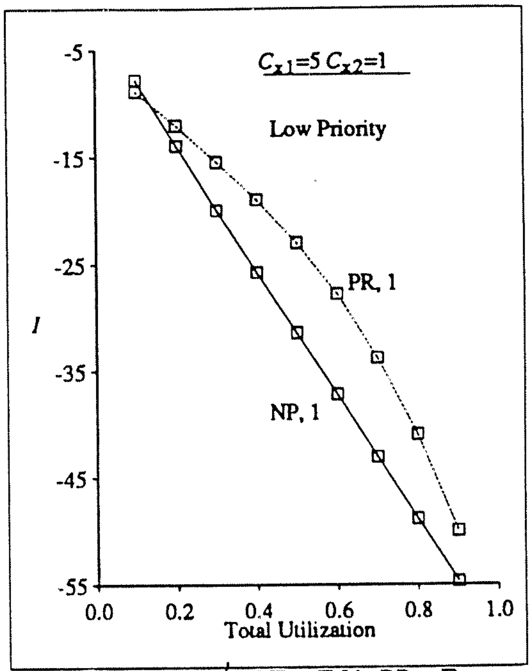

Figure 16. I^l - NP->F Vs PR->F

can reduce the waiting time of low priority jobs in Q_2, this reduction can be big enough to compensate them for the increase in their waiting time in Q_1. This is further exemplified in Figure 16 where we compare $I^l_{PR,1}$ versus $I^l_{NP,1}$, for $C_{x_1} = 5, C_{x_2} = 1$. We observe that $I^l_{PR,1}$ can be upto 10% higher than $I^l_{NP,2}$ for this setting of C_{x_1} and C_{x_2}. More specifically the total waiting time of low priority jobs in the $PR \rightarrow F$ system is considerably smaller than that in the $NP \rightarrow F$ system. This is caused by giving high priority jobs the right to preempt them from the first server. Although this sounds counter-intuitive it can be reasoned in the following way : low priority jobs in the PR queue have the *same* waiting time *before they enter service* in Q_1, as compared to the NP case [3]. As C_{x_1} increases, this component dominates their response time in Q_1, and hence for large C_{x_1}, their response time in Q_1 is almost the same as their non-preemptive counterparts. We have seen in section 3.2 that they perform better in Q_2 as C_{x_1} increases, due to the reduction in the variance of inter-arrival times to Q_2 caused by having preemption in Q_1. Hence they also perform better w.r.t. the total response time in the system. This explains why in a two-queue system both high and low priority jobs can perform better when the priority is preemptive than when it is non-preemptive, which is very much unlike a single queueing system.

5 Conclusion

This paper examined the problem of priority placement in a layered system, which is modeled by a system of two queues in tandem. Simulation was used to study how priority in only one of the two queues, affected the waiting times in the system. We first showed that priority in the first queue does have an impact on the waiting time of high priority jobs in the second queue. Thus, high priority jobs perform better in the second queue, even without explicitly having priority there. With NP scheduling discipline in Q_1 only, high priority jobs benefit in Q_2, and the loss of low priority jobs equals the gain of the high priority jobs. This gain of high priority jobs increases with C_{x_1} and decreases with C_{x_2}. When the scheduling discipline in Q_1 is PR, waiting times of high priority jobs in Q_2 are much smaller than those when NP is used in Q_1. This is true for all values of ρ, C_{x_1}, and C_{x_2}. Perhaps the most counter-intuitive result seen from the $PR \rightarrow F$ system is that low priority jobs benefit because of preemption in Q_1. Their waiting times in Q_2 are smaller than the corresponding waiting times in the $NP \rightarrow F$ as well as $F \rightarrow F$ systems.

The impact that priority in the feeder queue had on the second queue, was used to determine which queue to place priority in. If just the ratio of waiting time in the first queue to that in the second, in a $F \rightarrow F$ system, is used as a guideline to the placement problem, then it is biased towards having priority in Q_2. This is because it does not account for the fact that priority in Q_1 benefits high priority jobs in Q_2. Our simulation results show that when $C_{x_1} > C_{x_2}$ ($C_{x_1} > 1$), it is better to have priority in Q_1, and vice versa when $C_{x_1} < C_{x_2}$. However, when $C_{x_1} = C_{x_2}$ ($C_{x_1} > 1$), then it is better to have priority in Q_1 for low utilizations ($\rho < 0.4$). For higher utilizations ($\rho > 0.4$), the reverse is true.

We view this work as a first step in an ongoing effort to understand the behavior of priority in layered systems. Depending on how stringent are the timing constraints, we might have to place priority in more than one layer. Placement may also be a function of the overhead associated with introducing priority in a given layer, an aspect that was not considered in this paper. Further, service times in different layers can be correlated, and at present the effect of correlation is unknown. Lastly, in real systems we might have more than one queue feeding a single layer. It will be an interesting problem to study the impact of priority in more than one feeder queue, on the lower layer.

Acknowledgements

We are very grateful to Mary Vernon, Thanassis Tsiolis, and Rajeev Agrawal for the very useful discussions on this work. We also thank Randy Nelson for providing us with an excellent reference, Divesh Srivastava for his useful comments, and V. Srinivasan for his help with the document.

References

[1] Carey M., Jauhari R., Livny M., *Priority in DBMS Resource Scheduling*, Proceedings of the Fifteenth International Conference on Very Large Data Bases, Amsterdam 1989, 397-410.

[2] Kleinrock L., *Queueing Systems, Volume I : Theory*, Wiley 1975.

[3] Kleinrock L., *Queueing Systems, Volume II : Computer Applications*, Wiley 1976.

[4] Livny M., *DeNet,* User's Guide, Version 1.0, February 1988.

[5] Mansharamani R., Livny M., *Analysis of a Priority Feeder on a FIFO Server* Technical Report TR-975, Computer Sciences Department, University of Wisconsin-Madison, September 1990.

[6] Niu S. C., *Bounds For The Expected Delays In Some Tandem Queues,* Journal Of Applied Probability 17,831-838 (1980).

[7] Niu S. C., *On The Comparison Of Waiting Times In Tandem Queues,* Journal Of Applied Probability 18,707-714 (1981).

[8] Tanenbaum A., *Computer Networks*, Second Ed., Prentice Hall 1988.

Appendix

Here we present equations and bounds for the $NP \rightarrow F$ system that were derived in earlier work [5]. These are intended to supplement the simulation results presented in this paper, and should not be viewed as an end in themselves. Unfortunately the corresponding expressions for the $PR \rightarrow F$ system were much too complex to draw any inference from them, and hence they are not presented here.

Response time equations

All the notation in this section is w.r.t. the $NP \rightarrow F$ system, and hence the system is not specified in the notation; for example T_2^h is the response time of high priority jobs in Q_2. For the $NP \rightarrow F$ system the first result that we obtained in [5] was that conservation holds for the response times of high and low priority jobs in Q_2. In other words the average response time of the high and low priority jobs in Q_2 is the same as that in a $F \rightarrow F$ system, i.e.

$$E[T_2] = \frac{E[T_2^h] + E[T_2^l]}{2} \tag{1}$$

Although this result has been stated for both classes equally utilizing the system, it can easily be shown for unequal utilizations too. How does this result help us? It proves to be a useful baseline to compare high and low priority response times in Q_2 which are given below.

Equations for the response times in Q_2 of both high and low priority classes, are now presented. Consider a customer J arriving to Q_1. While he is in Q_1 (both waiting and being served), Q_2 might go idle once in a while depending on the arrival pattern of jobs ahead of J. Let us denote by I_2 the total time that Q_2 remains idle, while J is in Q_1. In defining I_2 the class of J is not considered. If we consider only high priority customers however, then let I_2^h denote the total time that Q_2 remains idle while a particular high priority customer is in Q_1. Similarly let I_2^l be the corresponding idle time in Q_2 for low priority customers. Then for high priority and low priority jobs in the $NP \rightarrow F$ system we have

$$E[T_2^h] \quad = \quad \frac{\rho}{1-\rho}E[I_2] + E[I_2^h] + \frac{\rho}{2\mu(1-\rho)}({C_{x_2}}^2 - {C_{x_1}}^2) \tag{2}$$

$$E[T_2^l] \quad = \quad \frac{\rho}{1-\rho}E[I_2] + E[I_2^l] + \frac{\rho}{2\mu(1-\rho)}({C_{x_2}}^2 - {C_{x_1}}^2) \tag{3}$$

We see that the two expressions above are almost identical except for the terms $E[I_2^h]$ and $E[I_2^l]$. Clearly $E[I_2^h] < E[I_2^l]$ since both high and low priority jobs have the same service time distribution in Q_1 but high priority jobs have a smaller response time. Therefore the mean idle time in Q_2 during the response time of a high priority job in Q_1 will be smaller. In order to verify that this is true, we measured $E[I_2^h]$ and $E[I_2^l]$ by simulation for different settings of C_{x_1} and C_{x_2}. Figure 17 presents the results of the simulation, where we plot $E[I_2^h]$ and $E[I_2^l]$ versus ρ. We also verified that the difference in the idle times of high and low priority

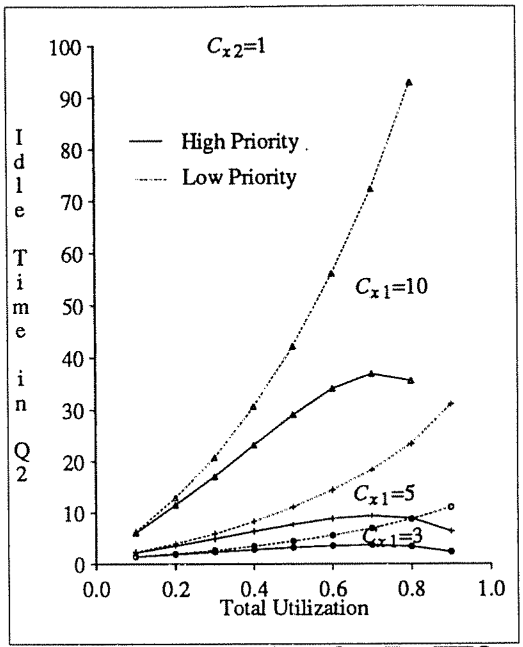

Figure 17 : Q_2 Idle times for NP->FIFO

jobs, was indeed the difference in their response times as seen from equations (2) and (3). times Therefore, we can conclude that high priority jobs gain over low priority ones as stated before. As ρ increases, the contribution of $E[I_2^l]$ and $E[I_2^h]$ to $E[T_2^h]$ and $E[T_2^l]$ respectively, decreases, thereby yielding a smaller gain. We further note that as C_{x_1} increases, the difference between the response times in Q_1, viz. $E[T_1^h]$ and $E[T_1^l]$, increases thereby causing the difference between $E[I_2^h]$ and $E[I_2^l]$ to increase, which can be seen from Figure 17.

Response time bounds

Since the equations in the previous section had unknown terms, we now state the bounds that were obtained in [5] for these response times. For the lower bound of the low priority jobs, we use the same lower bound as the $F \rightarrow F$ case [7]; and for the lower bound of the high priority jobs we use the same lower bound as for the $PR \rightarrow PR$ system. The upper and lower bounds for the response times of high and low priority classes in Q_2 are as follows :

$$\frac{1}{2\mu} + \frac{\rho {C_{x_2}}^2}{2\mu(2-\rho)} \leq E[T_2^h] \leq \frac{1}{\mu} + \frac{\rho({C_{x_1}}^2 + {C_{x_2}}^2 + 2)}{2\mu(1-\rho)} - \frac{\rho(1+{C_{x_1}}^2)}{2\mu(2-\rho)} \qquad (4)$$

$$\frac{1}{2\mu} + \frac{\rho {C_{x_2}}^2}{2\mu(1-\rho)} \quad \leq \quad E[T_2^l] \leq \frac{1}{\mu} + \frac{\rho({C_{x_1}}^2 + {C_{x_2}}^2 + 2)}{2\mu(1-\rho)} + \frac{\rho(1 + {C_{x_1}}^2)}{2\mu(2-\rho)} \tag{5}$$

From equation (5) we see that as C_{x_2} increases the upper and lower bounds become tighter. When $C_{x_2} \gg C_{x_1}$, the mean response time of low priority jobs in Q_2 approaches that of a $M/G/1$ queue in isolation. But this is also what we find for the $F \rightarrow F$ system in [5]. Hence by conservation of response times, it should also be true for high priority jobs. Therefore priority in the feeder (Q_1) has less of an impact as C_{x_2} increases w.r.t. to C_{x_1}, as we saw in section 3.1 of this paper.