International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:10, No:7, 2016

Performance Evaluation of a Prioritized, Limited Multi-Server Processor-Sharing System That Includes Servers with Various Capacities

Yoshiaki Shikata, Nobutane Hanayama

Abstract**—**We present a prioritized, limited multi-server processor sharing (PS) system where each server has various capacities, and N (≥2) priority classes are allowed in each PS server. In each prioritized, limited server, different service ratio is assigned to each class request, and the number of requests to be processed is limited to less than a certain number. Routing strategies of such prioritized, limited multi-server PS systems that take into account the capacity of each server are also presented, and a performance evaluation procedure for these strategies is discussed. Practical performance measures of these strategies, such as loss probability, mean waiting time, and mean sojourn time, are evaluated via simulation. In the PS server, at the arrival (or departure) of a request, the extension (shortening) of the remaining sojourn time of each request receiving service can be calculated by using the number of requests of each class and the priority ratio. Utilising a simulation program which executes these events and calculations, the performance of the proposed prioritized, limited multi-server PS rule can be analyzed. From the evaluation results, most suitable routing strategy for the loss or waiting system is clarified.

Keywords**—**Processor sharing, multi-server, various capacity, N priority classes, routing strategy, loss probability, mean sojourn time, mean waiting time, simulation.

I. Introduction

PROCESSOR sharing (PS) is important for evaluating the performance of a variety of resource allocation mechanisms. In PS, if there are n (> 0) requests in a single-server system, each request receives 1/n of the server capacity; no arriving request has to wait for service because it is served promptly, even if the service rate becomes slow. Thus, the sojourn time of each request that receives service in the server is n times the service time. The PS paradigm emerged as an idealization of Round-Robin (RR) scheduling algorithms in time-shared computer systems. A PS discipline with a priority structure has already been proposed, wherein a larger service ratio is allocated to requests with higher-priority. To prevent an increase in the sojourn time of each request in a prioritized single-server PS paradigm, and to realize a realistic sharing model, a method for limiting the number of requests that receive service is essential. In the prioritized, limited PS server, a high-priority request is allocated a service ratio that is m (called the priority ratio) times greater than that of a low-priority request. Moreover, the sum of the number of requests that receive service is restricted to a fixed value. An arriving request that cannot receive service due to this restriction is queued in the waiting room (waiting system) or rejected (loss system).

Communication services, such as web server farms, database systems, and grid computing clusters, routinely employ multi-server systems to provide a range of services to their customers. An important issue in such multi-server systems is to determine which server an arriving request should be routed to in order to optimize a given performance criterion. Therefore, in this paper, we propose a prioritized, limited multi-server PS system where each server can have various capacities, and N (≥2) priority classes are allowed in each PS server. Routing strategies of such prioritized, limited multi-server PS systems that take the capacity of each server into account are also proposed. The performance evaluation procedure of these strategies is discussed, and practical performance measures such as loss probability, mean waiting time in the service waiting queue, and mean sojourn time are evaluated via simulation. Based on the evaluation results, we discern the most suitable routing strategies of the prioritized, limited PS system that includes multi-servers with various capacities and requests of N priority classes.

Under the PS rule, when a request either arrives or departs from the PS server, the remaining sojourn time of other requests currently receiving service is extended or reduced, respectively. This extension or reduction of the sojourn time is calculated by using the number of requests of each class and the priority ratio. Employing a simulation program to execute these events and calculations allow us to analyze the prioritized, limited multi-server PS rule, which is realistic in a time-sharing system (TSS) with a sufficiently small time slot.

The PS rule, an idealization of quantum-based RR scheduling at the limit where quantum size becomes infinitesimal, has been the subject of many papers [1]-[4]. A limited PS system and a prioritized, limited PS system, in which the number of requests receiving service is restricted to a fixed value, have been proposed, and the performance of these systems has been analyzed [5], [6]. Load-balancing strategies for multi-class multi-server PS systems with a Poisson input stream and heterogeneous service rates have also been investigated [7]-[12]. However, routing strategies for prioritized, limited multi-server PS systems where each server can have various capacities, and N (≥2) priority classes are

Y. Shikata (Prof. Dr.) is with the Faculty of Informatics for Arts, Shobi University, Kawagoe, Saitama 350-1110, Japan (corresponding author, phone:81-49-246-2700; fax: 81-49-246-2531; e-mail: shikata@ictv.ne.jp).
N. Hanayama (Prof. Dr.) is with the Faculty of Informatics for Arts, Shobi University, Kawagoe, Saitama 350-1110, Japan.

allowed in each PS server are scarce; a performance evaluation procedure for these strategies have also not been investigated.

II. Prioritized, Limited Multi-Server PS System Including Servers with Various Capacities

A. System Concept

In a prioritized, limited multi-server PS system, an arriving request enters the dispatcher, which routes the request to each prioritized, limited PS server according to a predetermined strategy (Fig. 1). Suppose that there are N classes, and an arriving request, which is routed to server h, encounters n_{hj} class-j requests (including the arriving request). Furthermore, let m_j (≥ 1) denote the priority ratio of the class-j request, SFC ($\leq \infty$) denote the service facility capacity, and C_h denote the capacity ratio of server h to the reference server. According to the proposed prioritized, limited multi-server PS rule, if $(\sum_{j=1}^{N} m_j * n_{hj}) / C_h \leq SFC$, an arriving class-k request individually and simultaneously receives $m_k / \sum_{j=1}^{N} m_j * n_{hj}$ of the capacity of server h. When a server satisfying $(\sum_{j=1}^{N} m_j * n_{hj}) / C_h \leq SFC$ does not exist, the arriving request is queued in the corresponding class waiting room prepared in the dispatcher or rejected. The service time of a request in server h is given by the service time of the request in the server with capacity ratio one (the reference server) divided by C_h. In the waiting system, when the service for a request ends in one of the servers, another request is obtained from the service waiting queue and is routed to this server.

B. Routing Strategies

The following three routing strategies are considered, and their performances are compared.

1. Remaining Service Time Strategy

In this strategy, at the arrival of a request, the sum of the remaining service time (RST) of each class request currently receiving service for each server is evaluated. An arriving request is routed to the server that satisfies $(\sum_{j=1}^{N} m_j * n_{hj}) / C_h \leq SFC$ and has the smallest RST sum.

2. Normalized Service Capacity Strategy

In this strategy, at the arrival of a class-k request, the value of $C_h * m_k / (\sum_{j=1}^{N} m_j * n_{hj})$ for server h, which is called the normalized service capacity (NSC) that can be allocated to this request, is evaluated. An arriving request is routed to the server that satisfies $(\sum_{j=1}^{N} m_j * n_{hj}) / C_h \leq SFC$ and has the largest NSC value.

3. Normalized Number of Requests Strategy

In this strategy, at the arrival of a request, the value of $(\sum_{j=1}^{N} n_{hj}) / C_h$ for server h, which is called the normalized number of requests (NNR) that receive service, is evaluated. An arriving request is routed to the server that satisfies $(\sum_{j=1}^{N} m_j * n_{hj}) / C_h \leq SFC$ and has the smallest NNR value.

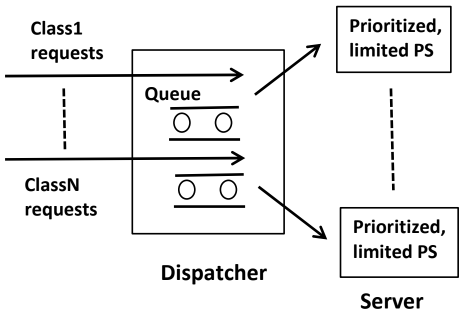

Fig. 1 Evaluation model

For these three strategies, when plural servers with the same RST sum, NSC, or NNR exist, a server is chosen from these servers with the same probability to route new arriving requests.

C. Extension or Reduction of the Remaining Sojourn (or Service) Time

At the arrival of a request to server h, the remaining sojourn time of this request is determined based on the service time of this request and the server capacity given to it. The service time is inversely proportional to the capacity ratio of server h (see Section A). For example, when a class-k request arrives at server h, if n_{hj} class-j requests (including the arriving request) are served in this server, $m_k / \sum_{j=1}^{N} m_j * n_{hj}$ of this server capacity is given to the request from this time, until the arrival (or departure) of the next request. The sojourn time of an arriving class-k request S_{ak} is then given by:

$$S_{ak} = (S_{rk} / C_h) * \sum_{j=1}^{N} m_j * n_{hj} / m_k \tag{1}$$

where S_{rk} represents the service time of an arriving class-k request in the reference server.

Moreover, at the arrival of a request, the server capacity given to each request currently receiving service decreases owing to the increase in the number of requests that share the server capacity. The ratio of the sojourn time of each request before and after the arrival of a request is equal to the inverse ratio of the server capacity given to each request before and after the request's arrival. For example, when a class-k request arrives $m_i / \{\sum_{j=1}^{k-1} m_j * n_{hj} + m_k * (n_{hk} - 1) + \sum_{j=k+1}^{N} m_j * n_{hj}\}$ of the server capacity is given to the class-i request that receives service by this time, but from this time until the arrival (or departure) of the next request, $m_i / (\sum_{j=1}^{N} m_j * n_{hj})$ is given to a class-i request. Therefore, the remaining sojourn time S_{ni} of each class-i request after this class-k request arrives is then extended as:

$$S_{ni} = S_{oi} * (\sum_{j=1}^{N} m_j * n_{hj}) / \{\sum_{j=1}^{k-1} m_j * n_{hj} + m_k * (n_{hk} - 1) + \sum_{j=k+1}^{N} m_j * n_{hj}\} \tag{2}$$

where S_{oi} is the remaining sojourn time of a class-i request immediately before this class-k request arrives. Similarly, at the end of the sojourn time of a request, the server capacity given to requests currently receiving service increases owing to the decrease in the number of requests that share the server capacity. The ratio of the sojourn time of each request before and after the departure of a request is also equal to the inverse ratio of the server capacity given to each request before and after the request's departure. For example, for the end of the sojourn time of a class-k request, the remaining sojourn time S_{ni} of each class-i request after this class-k request departs is reduced as:

$$S_{ni} = S_{oi} * \{\sum_{j=1}^{k-1} m_j * n_{hj} + m_k * (n_{hk} - 1) + \sum_{j=k+1}^{N} m_j * n_{hj}\} / (\sum_{j=1}^{N} m_j * n_{hj}) \quad (3)$$

where S_{oi} is also the remaining sojourn time of a class-i request immediately before this class-k request departs, and n_{hk} does not include a departing request. By executing these events and calculations in a simulation program, the performance of the prioritized, limited multi-server PS system can be analyzed. In the simulation program, the variable time increment method, where the simulation time is omitted until the next event that causes a change in the system state occurs, such as the arrival or departure of a request mentioned above, is used in order to shorten the simulation execution time.

At the arrival of a class-k request in server h, RST for this request is calculated as S_{rk} / C_h. Then, RST reduction for this request at the outbreak of each event mentioned above can be evaluated by the duration of the omitted time from the outbreak of the previous event by $m_k / \left(\sum_{j=1}^{N} m_j * n_{hj}\right)$.

D. Simulation Flow

Simulation flow of the prioritized, limited multi-server PS system is shown in Fig. 2. In the simulation program, the simulation clock is controlled by the arrival timer or service timer of each request that is receiving service. At the arrival of each class request, the time duration until the next arrival of the request is set into the arrival timer according to the predetermined arrival time distribution. Further, the service time (e.g., remaining sojourn time) of each arriving request calculated using (1) is set into the service timer. Moreover, the arrival time of each request is memorized in the corresponding variable. The sojourn time of each request in the server is evaluated by using these data and service end time. In addition, the waiting time in the service-waiting queue is evaluated by using these data and service start time. In the while loop of this simulation program, the extension or shortening of the remaining sojourn time of each request is calculated using (2) or (3) on the expiry of one of the arrival timers or service timers. Moreover, the service timer or arrival timer with the next smallest value is detected, and the time duration of this timer is subtracted from all the remaining timers. Therefore, in the next while loop this timer expires. Simultaneously, the simulation clock is pushed forward by this time duration in order to skip the insignificant simulation clock. The while loop is repeated until the number of arriving requests attains a predetermined value.

III. Evaluation Results

In the evaluation, class-1 (m_1=4) requests, class-2 (m_2=3) requests, and class-3 (m_3=2) requests were assumed to be served in each server. The arrival rate, or mean service time, of each priority class request was assumed to be the same value. The two-stage Erlang inter-arrival time distribution and the exponential service time distribution were considered. Evaluation results were obtained from the average of ten simulation results. Approximately 140000 requests were produced for each class in each run. In each figure that shows an evaluation result, A_r represents the arrival rate and S represents the mean service time.

A. Loss System

Fig. 3 shows the relationship between the mean sojourn times of class-1 requests (round markers), class-2 requests (square markers), and class-3 requests (cross markers) and the service facility capacities for the NSC (straight lines), RST (dashed lines), and NNR (dotted lines) strategies. Three servers were prepared, and each server had the capacity ratios of 1, 0.8, and 0.6. In Fig. 3, the range of the markers includes 95% of the reliability intervals obtained from the ten simulation runs. Notice that the mean sojourn time increases as the service facility capacity increases. The mean sojourn times of class-1 requests and class-2 requests of the NSC strategy are smaller than their corresponding values obtained by the RST and NNR strategies. On the other hand, the mean sojourn times of class-3 requests of the NNR strategy are smaller than their corresponding values obtained by the NSC and RST strategies. In other words, the NSC strategy extends the difference of the mean sojourn time between each class request, and the NNR strategy reduces it.

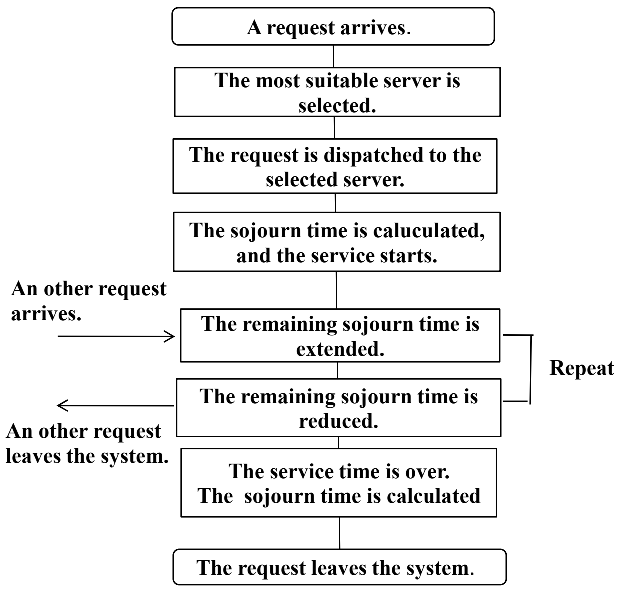

Fig. 2 Simulation flow

Fig. 4 compares the loss probabilities of each class request for the NSC, RST, and NNR strategies. The marker and line styles are the same as those used in Fig. 3. The logarithm of the loss probability increases linearly as the service facility capacity decreases. The loss probabilities of class-1 and class-2 requests for the NSC, RST, and NNR strategies are almost identical. On the other hand, the loss probability value of class-3 requests for the NSC strategy is less than that of the RST and NNR strategies.

The NSC value and NNR value can be calculated easily by using the number of each class request that is served in each server. On the other hand, the RST value has to be calculated by the service time of each request minus the amount of service time that has actually been assigned to the request (see Section II-D. Accordingly, the calculation algorithm of the NSC and NNR is easier to compute than that of the sum of the RST. Based on the aforementioned results, the NSC strategy, which realizes higher performance and an easier calculation algorithm than that of the other strategies, is the most suitable routing strategy for the loss system of the prioritized, limited multi-server PS system.

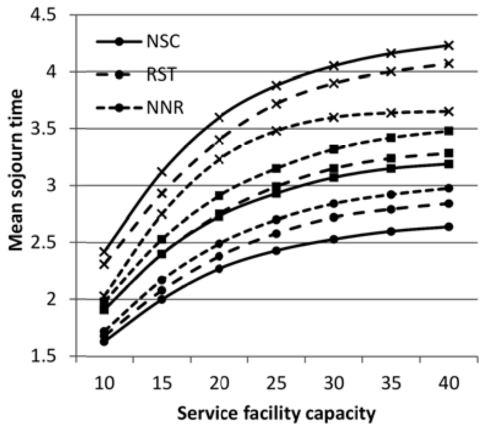

Fig. 3 Comparison of the mean sojourn time (Ar=0.7, S=1)

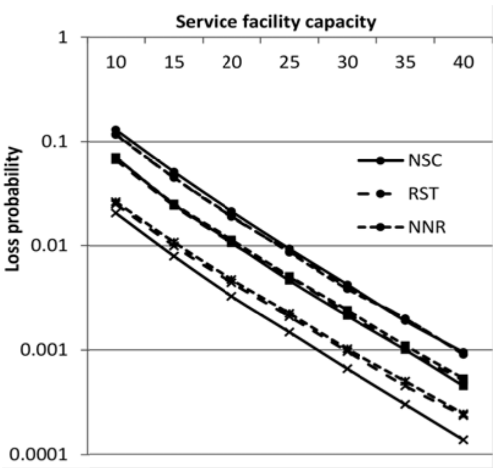

Fig. 4 Comparison of the loss probability (Ar=0.7, S =1)

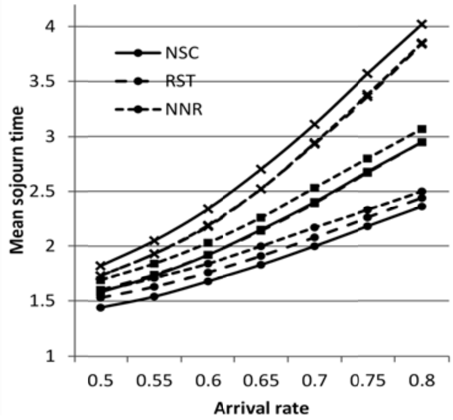

Fig. 5 Comparison of the mean sojourn time (S=1, SFC=15)

Tables I-III list the ratios when each arriving class request is dispatched into each server with various capacities, also known as the routing ratio. The arrival rate, mean service time of each priority class request, and SFC are assumed to be 0.7, 1, and 30, respectively. For the NSC strategy, almost half of the class-1 requests are dispatched into the server with the highest capacity. Conversely, for the NNR strategy, 60% of class-3 requests are dispatched into the server with the highest capacity. For the RST strategy, each class request is dispatched into each server at the same ratio. These ratios may be the reason that the sojourn times of class-1 requests in the NSC strategy and class-3 requests in the NNR strategy are smaller than those of the other strategy.

TABLE I
ROUTING RATIO (NSC)

Capacity ratio	Class-1	Class-2	Class-3
1	49%	42%	37%
0.8	32%	35%	35%
0.6	19%	23%	28%

TABLE II
ROUTING RATIO (RST)

Capacity ratio	Class-1	Class-2	Class-3
1	41%	41%	41%
0.8	34%	34%	33%
0.6	25%	25%	26%

TABLE III
ROUTING RATIO (NNR)

Capacity ratio	Class-1	Class-2	Class-3
1	32%	32%	58%
0.8	36%	36%	28%
0.6	32%	32%	14%

Fig. 5 shows the relationship between the mean sojourn times of class-1 requests, class-2 requests, and class-3 requests and the arrival rates for the NSC, RST, and NNR strategies. The marker and line styles are the same as those used in Fig. 3. The mean sojourn time of class-3 requests is easily affected by the number of arriving requests, more than the corresponding values of class-1 requests and class-2 requests. Fig. 6 also shows the relationship between the loss probability of class-1 requests, class-2 requests, and class-3 requests and the arrival rates for NSC, RST, and NNR strategies. The marker and line styles are the same as those used in Fig. 3. Note that the loss probability of class-1 requests is easily affected by the number of arriving requests, more than the corresponding values of class-2 requests and class-3 requests. Fig. 7 compares the mean sojourn times of class-1 requests (round markers), class-2 requests (square markers), and class-3 requests (cross markers) when the capacity sum of each server is 2 and the capacity ratio of each server is 1:1 (straight lines), 1:0.7:0.3 (dashed lines), and 1:0.5:0.5 (dotted lines). In this evaluation, the NSC strategy is considered. The mean sojourn times of each class request for capacity ratio 1:1 are smaller than their corresponding values when the capacity ratio is 1:0.7:0.3 and 1:0.5:0.5. The mean sojourn times of class-1 requests and classs-2 requests when the capacity ratio is 1:0.7:0.3 are slightly less than their corresponding values when the capacity ratio is 1:0.5:0.5.

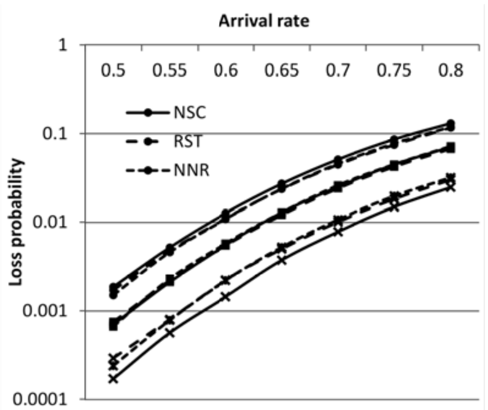

Fig. 6 Comparison of the loss probability (S=1, SFC=15)

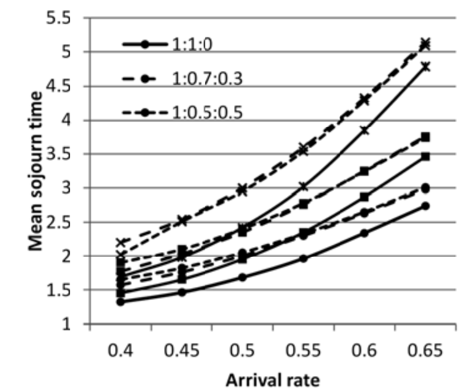

Fig. 7 Comparison of the mean sojourn time (S=1, SFC=20)

Fig. 8 also compares the loss probability of each class request using the same conditions as Fig. 7. The marker and line styles are the same as those used in Fig. 7. The loss probabilities of class-1 requests and class-2 requests when the capacity ratio is 1:1 are smaller than their corresponding values when the capacity ratio is 1:0.7:0.3 and 1:0.5:0.5. The loss probabilities of class-3 requests for a multi-server with capacity ratios of 1:1, 1:0.7:0.3, and 1:0.5:0.5 achieve approximately the same values Based on these results, we conclude that a multi-server system with fewer servers realizes higher performance when the same capacity sum of each server is used.

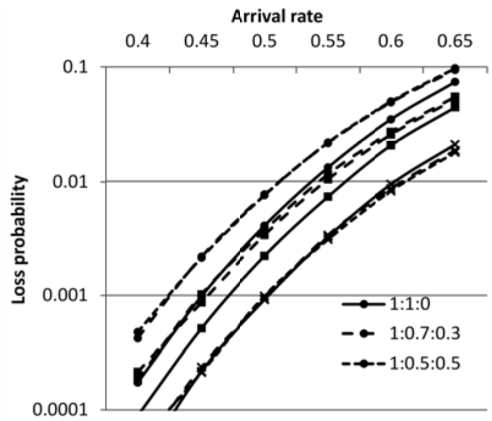

Fig. 8 Comparison of the loss probability (S=1, SFC=20)

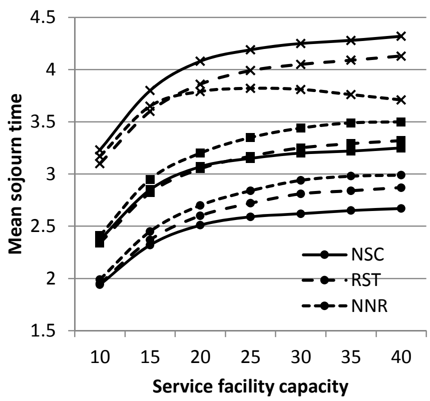

Fig. 9 Comparison of the mean sojourn time (S=1, SFC=15)

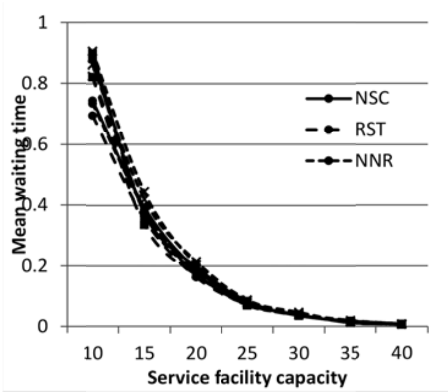

Fig. 10 Comparison of the mean sojourn time (Ar=0.7, S=1)

B. Waiting System

Fig. 9 shows the relationship between the mean sojourn times of class-1 requests (round markers), class-2 requests (square markers), and class-3 requests (cross markers) and the service facility capacities for NSC (straight lines), RST (dashed lines), and NNR (dotted lines) strategies. Three servers were prepared, and each server had the capacity ratios of 1, 0.8, and 0.6. Notice that the mean sojourn times of class-1 and class-2 requests increase as the service facility capacities increase. On the other hand, the mean sojourn time of class-3 requests for the NNR strategy is maximized when SFC=20. The mean sojourn times of class-1 and class-2 requests of the NSC strategy are smaller than those of the RST and NNR strategies. Furthermore, the mean sojourn time of class-3 requests of the NNR strategy is smaller than that of the NSC and RST strategies.

Fig. 10 compares the mean waiting times in the service waiting queue in the waiting systems of class-1, class-2, and class-3 requests for the NSC, RST, and NNR strategies. The marker and line styles are the same as those used in Fig. 9. As the SFC decreases, the mean waiting time for the RST strategy becomes smaller than that of the NSC and NNR strategies. However, the differences in the mean waiting times between these strategies are much smaller than the differences between their sojourn times. Based on these results, the NSC and NNR strategies that realize higher performances and easier algorithm calculations than the RST strategy are the most suitable routing strategies for the waiting system of the prioritized, limited multi-server PS system.

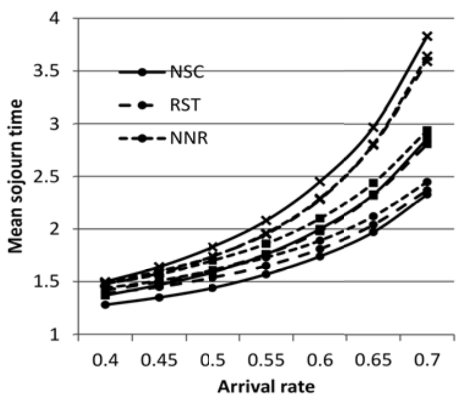

Fig. 11 Comparison of the mean sojourn time (S=1, SFC=15)

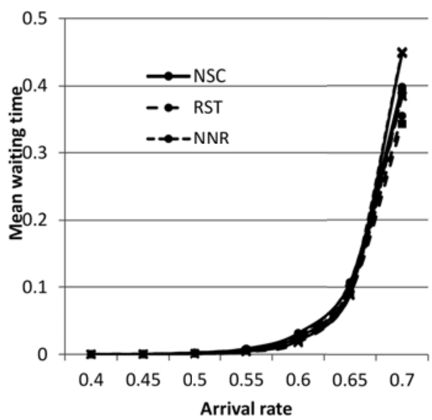

Fig. 12 Comparison of the mean waiting time (S=1, SFC=15)

Fig. 11 shows the relationship between the mean sojourn times of class-1, class-2, and class-3 requests and the arrival rates for the NSC, RST, and NNR strategies. The marker and line styles are the same as those used in Fig. 9. Note that the mean sojourn time of class-3 requests is affected more by the number of arriving requests than class-1 and class-2 requests.

Fig. 12 compares the mean waiting times in the service waiting queue in the waiting systems of class-1, class-2, and class-3 requests for the NSC, RST, and NNR strategies. The marker and line styles are the same as those used in Fig. 9. When an arrival rate exceeds a certain value, the waiting times rapidly increase.

IV. Conclusion

A prioritized, limited multi-server PS system where each server can have various capacities and N (≥2) priority classes are allowed in each PS server was proposed. Routing strategies for this system that factors in the capacity of each server was also proposed, and performance evaluation procedures for these strategies were discussed. Practical performance measures for these strategies, such as the loss probability, mean waiting time in the service waiting queue, and mean sojourn time, were evaluated via simulation.

In the loss system of the prioritized, limited multi-server PS system, the NSC strategy extended the difference of the mean sojourn time between each class request, while the NNR strategy reduced it. Furthermore, the NSC strategy that realizes higher performance and an easier calculation algorithm than the other strategies was determined to be the most suitable routing strategy for the loss system of the prioritized, limited multi-server PS system. Moreover, the NSC and NRR strategies that realize higher performance and easier calculation algorithms than the RST strategy are suitable routing strategies for the waiting system of the prioritized, limited multi-server PS system.

In the future, we plan to study the influence of the service time distribution, or inter-arrival time distribution, on the routing strategy of a prioritized, limited multi-server PS system and the routing strategies of a prioritized, limited multi-server RR system.

References

[1] L. Kleinrock, "Time-Shared Systems: A Theoretical Treatment", J.A.C.M Vol.1, No.14, 242-261 (1967).

[2] G.Fayolle and I.Mitrani, "Sharing a Processor Among many Job Classes", J.A.C.M Vol.27, No.3, July 1980. Pp519-532

[3] E. Altman, K. Avrachenkov and U. Ayesta, "A survey on processor sharing", Queueing Syst (2006) 53:53-63

[4] M. Haviv and J. Val, "Mean sojourn times for phase-type discriminatory processor sharing system", European Journal of Operational Research, 189(2008), 375-386

[5] G. Yamazaki and H. Sakasegawa, "An optimal design problem for limited sharing systems, Management Science", vol.33(8), pp.1010--1019 (1987).

[6] Y. Shikata, W. Katagiri, and Y. Takahashi, "Prioritized Limited Processor-Sharing System with its Performance Analysis", International Conference on Operations Research, August 30 - 1, 2011 Zurich.

[7] H.L. Chen, J. Marden, and A. Wierman, "The effect of local scheduling in load balancing designs", In Proceedings of IEEE INFOCOM, 2009.

[8] V. Gupta, M. Harchol-Balter, K. Sigman, and W. Whitt, "Analysis of join-the-shortest-queue routing for web server farms", In Proceedings of

Performance, page 180, 2007.
[9] E. Altman1, U. Ayesta, and B.J. Prabhu, “Load Balancing in Processor Sharing Systems”, Telecommunication Systems, June 2011, Volume 47, Issue 1-2, pp 35-48.
[10] M. Haviv and T. Roughgarden. “The price of anarchy in an exponential multi-server”, Operations Research Letters, 35:421–426, 2007.
[11] H. Kameda, E. Altman, O. Pourtallier, J. Li, and Y. Hosokawa. “Paradoxes in performance optimization of distributed systems”, In Proceedings of SSGRR 2000 Computer and ebusiness conference, 2000.
[12] D. Starobinski and T. Wu. “Performance of server selection algorithms for content replication networks”, In IFIP Networking, 2005.