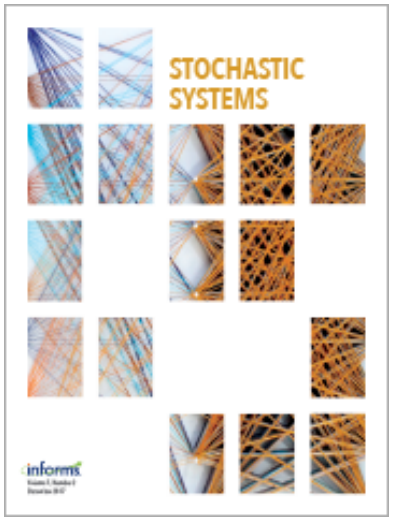
This article was downloaded by: [216.73.217.74] On: 15 July 2026, At: 04:23
Publisher: Institute for Operations Research and the Management Sciences (INFORMS)
INFORMS is located in Maryland, USA



Stochastic Systems

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Open Problem—Regarding Static Priority Scheduling for Many-Server Queues with Reneging

Amy R. Ward

To cite this article:
Amy R. Ward (2019) Open Problem—Regarding Static Priority Scheduling for Many-Server Queues with Reneging. Stochastic Systems 9(3):313-314. https://doi.org/10.1287/stsy.2019.0048

This work is licensed under a Creative Commons Attribution 4.0 International License. You are free to copy, distribute, transmit and adapt this work, but you must attribute this work as "*Stochastic Systems*. Copyright © 2019 The Author(s). https://doi.org/10.1287/stsy.2019.0048, used under a Creative Commons Attribution License: https://creativecommons.org/licenses/by/4.0/."

Copyright © 2019, The Author(s)

Please scroll down for article—it is on subsequent pages

With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org

http://pubsonline.informs.org/journal/stsy

STOCHASTIC SYSTEMS
Vol. 9, No. 3, September 2019, pp. 313–314
ISSN 1946-5238 (online)

Special Section: Open Problems in Applied Probability

Open Problem—Regarding Static Priority Scheduling for Many-Server Queues with Reneging

Amy R. Ward[a]

[a] The University of Chicago Booth School of Business, Chicago, Illinois 60637

Contact: amy.ward@chicagobooth.edu, https://orcid.org/0000-0003-2744-2960 (ARW)

Published Online in Articles in Advance: September 17, 2019

https://doi.org/10.1287/stsy.2019.0048

Copyright: © 2019 The Author(s)

History: This paper was accepted for the *Stochastic Systems* Special Section on Open Problems in Applied Probability, presented at the 2018 INFORMS Annual Meeting in Phoenix, Arizona, November 4–7, 2018.

Open Access Statement: This work is licensed under a Creative Commons Attribution 4.0 International License. You are free to copy, distribute, transmit and adapt this work, but you must attribute this work as "*Stochastic Systems*. Copyright © 2019 The Author(s). https://doi.org/10.1287/stsy.2019.0048, used under a Creative Commons Attribution License: https://creativecommons.org/licenses/by/4.0/."

Keywords: stochastic networks • scheduling control • optimization

Downloaded from informs.org by [216.73.217.74] on 15 July 2026, at 04:23 . For personal use only, all rights reserved.

We study the many-server queue shown in Figure 1(a). Customers from class $j \in \{1,2\}$ arrive according to renewal processes having rate $\lambda_j > 0$ and request processing. There are $N \in \{1,2,\ldots\}$ servers, each working at rate one. All servers are fully flexible; that is, every server can serve customers from both classes. Each customer has a randomly distributed patience time that may depend on the class and reneges (abandons the system without receiving service) if service does not begin before the patience time expires. The system incurs the penalty $p_1 > 0$ when a class 1 customer reneges and the penalty $p_2 > 0$ when a class 2 customer reneges. Upon arrival, each class j customer independently samples from the distribution determined by cumulative distribution function (cdf) G_j^r having mean $1/\theta_j < \infty$ to find its patience time and from the distribution determined by cdf G_j^s having mean $1/\mu_j < \infty$ to determine its service time (that is, the required processing time).

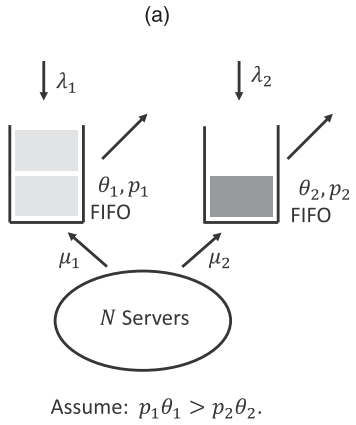
The scheduling policy determines what to do when a server is available. In particular, when customers from multiple classes are waiting, the scheduling policy must specify which class the server should next serve. Within the class selected, the customer who has waited the longest is served; this customer is called the head-of-line customer. If a server becomes available to find no waiting customers, then the server necessarily idles. We would like to find a scheduling policy π (within a suitably defined class of admissible scheduling policies) that minimizes the long-run average cost

$$\mathcal{C}(\pi) := \limsup_{T\to\infty} \frac{1}{T} \mathbb{E}\left[\sum_{j=1}^{J} p_j R_j(T,\pi)\right],$$

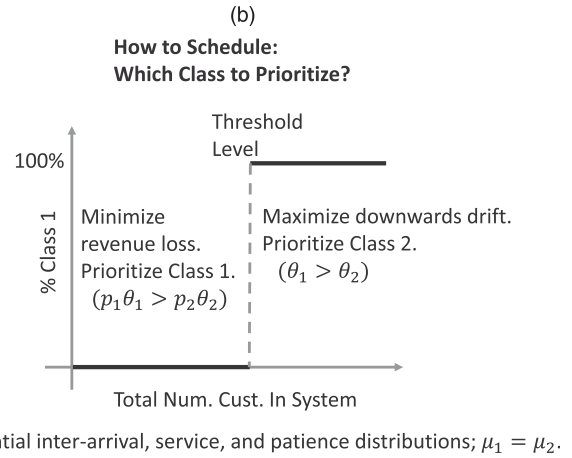
where $R_j(T,\pi)$ tracks the cumulative number of reneging customers in $[0,T]$ from class $j \in \{1,2\}$ under scheduling policy π.

Assume the classes are labeled so that $p_1\theta_1 > p_2\theta_2$. The solution to the approximating fluid control problem in equation (23) in Puha and Ward (2019) motivates using a static priority scheduling rule that gives priority to class 1 customers and only serves class 2 customers when no class 1 customers are waiting. When customer interarrival, service, and patience times are all exponentially distributed, Atar et al. (2010), proposition 4.1 and theorem 6.1, establishes that the aforementioned static priority scheduling rule is asymptotically optimal under many-server fluid scaling and overload conditions. However, under the additional assumption that $\mu_1 = \mu_2$, the solution to the approximating diffusion control problem in equation (16) in Kim et al. (2018), proposition 1, only motivates using the aforementioned static priority scheduling rule when $\theta_1 \le \theta_2$ and, otherwise, motivates using the state-dependent threshold rule illustrated in Figure 1(b) (see also Harrison and Zeevi (2004)). The intuition is that, when $\theta_1 \le \theta_2$, class 2 has both a lower instantaneous abandonment cost ($p_1\theta_1 > p_2\theta_2$) and a higher reneging rate ($\theta_1 \le \theta_2$), meaning there is a fortunate alignment between cost and downward drift from reneging. Otherwise ($\theta_1 > \theta_2$), prioritizing class 2 over class 1 gains the additional reneging rate $\theta_1 - \theta_2$, which trims congestion (thereby reducing future reneging), presenting a trade-off between deciding priorities based on drift and deciding priorities greedily based on instantaneous cost.

Figure 1. The Many-Server Queueing Model



Two classes and flexible servers.



A DCP-based scheduling policy.

The Open Question

We would like to bound the performance of the state-independent static priority scheduling policy in comparison with the optimal scheduling policy. When customer interarrival, service, and patience times are all exponentially distributed, the scheduling control problem can be written as a Markov decision problem, in which case we also want to bound the performance of threshold scheduling policies. In this setting, we are hopeful the methodology in Braverman et al. (2019) could be helpful. (We note that a similar question holds in the single-server setting, which could be an easier place to begin.) The larger (and harder) question is to be able to say something more generally when distributions are not exponential.

Acknowledgement

The author thanks Jan A. Van Mieghem (Northwestern University) for suggesting Figure 1(b) to illustrate the intuition behind the scheduling policy.

References

Atar R, Giat C, Shimkin N (2010) The $c\mu/\theta$ rule for many-server queues with abandonment. *Oper. Res.* 58(5):1427–1439.

Braverman A, Gurvich I, Huang J (2019) On the Taylor expansion of value functions. *Oper. Res.* Forthcoming.

Harrison JM, Zeevi A (2004) Dynamic scheduling of a multiclass queue in the Halfin-Whitt heavy traffic regime. *Oper. Res.* 52(2):243–257.

Kim J, Randhawa RS, Ward AR (2018) Dynamic scheduling in a many-server multi-class system: The role of customer impatience in large systems. *Manufacturing Service Oper. Management* 20(2):285–301.

Puha A, Ward AR (2019) Scheduling an overloaded multiclass many-server queue with impatient customers. *TutORials Oper. Res.* (INFORMS, Catonsville, MD), 189–217.

Downloaded from informs.org by [216.73.217.74] on 15 July 2026, at 04:23 . For personal use only, all rights reserved.