Queueing Theory without Probabilities

A. Udaya Shankar

June 5, 1995

1 Introduction

Queueing theory is all about the performance of systems where customers compete for nonshareable resources. A queueing system has servers, representing resources, and a stream of customers with service requirements. If several customers want service at the same server, all but one of the customers wait according to some queueing discipline.

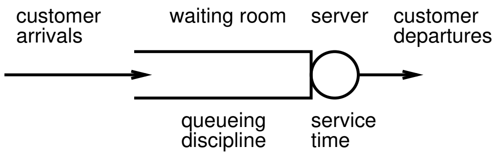
Operating systems are examples of queueing systems; here, customers correspond to processes, io-requests, etc., and resources correspond to cpu, memory, disks, etc. Hence, queueing theory allows us to examine the effect of varying customers, resources, and scheduling disciplines on OS performance, without having to resort to costly emulation or implementation.

Road traffic is another source of queueing systems; here, the customers are cars, trucks, etc., and the servers are traffic lights, construction sites, intersections, etc. Indeed, road traffic, because of our intimate familiarity with it, provides a convenient "reality" check for queueing results.

The following introduces some simple results of queueing theory while assuming no background in probability theory and stochastics. *Note that "queue", when used as a noun, refers to the waiting room plus server, and "queue", when used as verb, does not imply FCFS (first-come-first-served) discipline.*

2 The Reason for Queues

The simplest queueing system is the **single-server queue**. It consists of a single server and a waiting room. Customers arrive, each with a service requirement. An arriving customer either starts getting service or waits according to some queueing discipline.



Consider a road with a construction site bottleneck. Suppose a car takes 3 s to go through the site. This can be modeled as a single-server queue, where the server is the site and the waiting room is the road where cars line up in case of congestion. Cars wait in FCFS order. The waiting time for a car is the time from its arrival until entry to the construction site. The response time for a car is the time from its arrival until exit from the construction site. The throughput is the average number of cars departing per second.

2.1 Light load

Suppose cars arrive separated by 4 s, starting at time 0. Let $N(t)$ denote the number of cars in the system at time t. $N(t)$ evolves as follows:

```
  N(t)
3 _|
2 _|      C1                 C2                  C3
1 _|______________      ______________      ______________       __
0 _|______________|____|______________|____|______________|_____|____ time
   0    1    2    3    4    5    6    7    8    9    10   11   12

   C1             C1   C2             C2   C3             C3    C4
   arr            dep  arr            dep  arr           dep    arr
```

Since 1 car departs every 4 s, the throughput is 1/4 cars/s. The waiting time for each car is 0 s. The response time for each car is 2 s. The average number of cars in the queue is 2/3, because there is one car for 2 s out of every 3 s interval. The average number of waiting cars in the queue is 0: a very lightly loaded system.

2.2 Heavy load

How can we get a queue to build up? Suppose cars arrive separated by 2 s. We get the following:

```
  N(t)
  _|
4 _|                                        C3    C4   C4        C5
3 _|           C1   C2     C2       C3     _____      _________|___
2 _|   C1     ____      _________|________|     |____|
1 _|_________|    |____|
0 _|_________________________________________________________________ time
   0    1    2    3    4    5    6    7    8    9    10   11   12

   C1        C2   C1   C3      C2 C4       C5   C3   C6       C4 C7
   arr       arr  dep  arr    dep arr      arr  dep  arr     dep arr
```

Note that $N(t)$ repeats every 6 s (the least common multiple of 2 and 3), except that each time it moves up by 1. The queue, and hence the delays, increase without bound. The system is said to be **unstable**. The waiting time is 0 s for car 1, 1 s for car 2, 2 s for car 3, and, in general, k s for car k.

Since 1 car departs every 3 s, the throughput is 1/3 cars/s. This is less than the arrival rate, which is 1/2 cars/s.

2.3 Bunched load

Unstable queues are also unrealistic, because in real-life queues grow *and* shrink. Cars do queue up at construction sites, but we know from experience that over, say, an hour, these queue sizes fluctuate about some average, rather than steadily increasing over time. The reason for this is *variations* in customer arrival times and service requirements. Real-life cars travel in clusters and take differing times to go through a bottleneck.

Suppose we bunch up cars so that car 1 arrives at time 0, car 2 at time 2, and the pattern repeats every 8 s (i.e. cars 3 and 4 at times 8 and 10, cars 5 and 6 at times 12 and 14, etc.).

We get the following:

```
  N(t)
  _|
3 _|        C1                               C3
2 _|   C1   ___     C2                  C3   ___     C4
1 _|_______|   |___________         ________|   |___________
0 _|_______________________|_______|________________________|________ time
   0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15

   C1      C2  C1          C2      C3      C4  C3          C4
   arr     arr dep         dep     arr     arr dep         dep
```

The waiting time is 0 s for odd cars and 1 s for even cars. The response time is 3 s for odd cars and 4 s for even cars. The average number of cars in the system is 7/8, since every 8 s interval has 1 car for 5 s and 2 cars for 1 s. The average number of waiting cars is 1/8.

Since 2 cars depart every 8 s, the throughput is 2/8 cars/s, which is the same as in the first case.

3 Performance Measures

We now define various performance measures for a queueing system. We have already seen some of these, e.g. throughput, response time, average number of customers. Some measures, such as throughput, are defined per unit time (e.g. per second). Some measures, such as response time, are defined per customer. Some measures, referred to as *instantaneous* measures, are defined over a finite time interval or a particular customer. Some measures, referred to as *steady-state* measures, are defined over all time or over all customers (perhaps of some class).

3.1 Instantaneous measures

Let customer i be the ith customer to arrive at the queue, where $i = 1, 2, 3, \cdots$. The stream of customers can be unending or it can be finite. For customer i, let

A_i denote its arrival time to the system.

S_i denote its service requirement.

D_i denote its departure time from system. (function of A_i's, S_i's, and queueing discipline)

To define an unending stream of customers, we usually define a finite stream of customers and have that repeat with some period.

Let

- $N(t)$ denote the number of customers in the system at time t (both waiting and in service).
- $N_W(t)$ denote the number of waiting customers in the system at time t.
- $Y(t)$ denote the unfinished work in the system at time t; i.e. the remaining service time of the customer being served (if any) plus the sum of the service times of the waiting customers (if any).

For any customer i:

- R_i denotes the response time (departure time minus arrival time): $R_i = D_i - A_i$

- W_i denotes the waiting time: $W_i = R_i - S_i$

We next define some "per-time" measures for an arbitrary time interval $[t_1, t_2]$, where $t_1 < t_2$:

- Arrival rate over $[t_1, t_2] = \dfrac{\text{number of arrivals in } [t_1, t_2]}{t_2 - t_1}$

- Offered load (or arriving work) over $[t_1, t_2] = \dfrac{S_{i_1} + S_{i_2} + \cdots + S_{i_n}}{t_2 - t_1}$
  where $i_1, i_2, \cdots, i_n$ are the customers arriving in $[t_1, t_2]$.

- Throughput (or departure rate) over $[t_1, t_2] = \dfrac{\text{number of departures in } [t_1, t_2]}{t_2 - t_1}$

- Average number of customers in system for time interval $[t_1, t_2] = \dfrac{\int_{t_1}^{t_2} N(t)\,dt}{t_2 - t_1}$
  This is the area under $N(t)$ in the time interval $[t_1, t_2]$.

- Utilization of a server over $[t_1, t_2] = \dfrac{\text{fraction of } [t_1, t_2] \text{ where the server is busy}}{t_2 - t_1}$

3.2 Steady-state measures

We have seen that a queueing system with an unending stream of customers can be stable or unstable. An unstable system always has waiting customers and its queues keep getting bigger. A stable system cycles through busy periods and idle periods, and its queues grow and shrink but they don't blow up.

For a stable queueing system, we are interested in "steady-state" performance measures, i.e. over the time interval $[0, \infty)$, or over all customers, or over all customers of some class (e.g. the odd cars in the above example). Steady-state measures are not usually interesting for systems with a finite stream of customers, because the measures are usually zero.

We have the following:

- Average service time $S = \lim_{n\to\infty} \dfrac{\sum_{i=1}^{n} S_i}{n}$

- Average response time $R = \lim_{n\to\infty} \dfrac{\sum_{i=1}^{n} R_i}{n}$ (assumes stable system)

- Average waiting time $W = \lim_{n\to\infty} \dfrac{\sum_{i=1}^{n} W_i}{n}$ (assumes stable system)

- Average number of customers in system $N = \lim_{t\to\infty} \dfrac{\int_0^t N(t)\,dt}{t}$ (assumes stable system)

- Arrival rate $\lambda = \lim_{t\to\infty} \dfrac{\text{number of arrivals in } [0, t]}{t}$

- Offered load $\rho \;=\; \lim_{t \to \infty} \dfrac{\text{offered load for } [0,\ t]}{t}$

- Throughput $X \;=\; \lim_{t \to \infty} \dfrac{\text{number of departures in } [0\ ,\ t]}{t}$

- Utilization $U \;=\; \lim_{t \to \infty} \dfrac{\text{Utilization for } [0\ ,\ t]}{t}$

Observations:

- *Offered load* $\rho \;=\; \lambda S$
- *A single-server system is unstable if* $\rho > 1$
- *For stable system:* $X \;=\; \lambda$ *and* $U \;=\; \rho$
- *For an unstable system:* $X \;=\; \dfrac{1}{S}$ *and* $U \;=\; 1$

4 Effect of Queueing Disciplines

We examine the effect of four queueing disciplines: FCFS, SJF (shortest-job-first), SJFP (shortest-job-first-preemptive), and RR (round-robin). Consider a single-server queue with customers as follows (all times are in seconds):

customer	A_i	S_i
1	1.0	3.0
2	2.0	2.0
3	2.5	1.0

repeated every 10 seconds (e.g. customers 4, 5, 6 arrive at times 11.0, 12.0, 12.5, with service requirements 3.0, 2.0, 1.0)

4.1 First-come-first-served (FCFS)

The dynamics of the queue can be seen by plotting $N(t)$ versus time. Also indicated are the ids of customers arriving, departing, and getting service.

```
   N(t)
   _|
 4 _|                            C1
 3 _|                  C1    ____________    C2
 2 _|          C1    ______|            |_________   C3
 1 _|       _______|                             |________
 0 _|______|_____________________________________________|________ time
    0      1       2      2.5          4         6        7       (seconds)

          C1      C2      C3           C1        C2       C3
          arr     arr     arr          dep       dep      dep
```

The system becomes empty at time 7. Thus the system is stable and the pattern repeats at times 10, 20, etc. The customer departure, response, and wait times are as follows:

customer	A_i	S_i	D_i	R_i	W_i
1	1.0	3.0	4.0	3.0	0.0
2	2.0	2.0	6.0	4.0	2.0
3	2.5	1.0	7.0	4.5	3.5

$$\text{Average response time } R = \frac{3.0+4.0+4.5}{3} = \frac{11.5}{3}$$

$$\text{Average wait time } W = \frac{0.0+2.0+3.5}{3} = \frac{5.5}{3}$$

Throughput $X = \dfrac{3}{10}$. (The instantaneous throughput over [0, 10] is $\frac{3}{10}$, over [0, 3] it is $\frac{0}{3} = 0$, and over [0, 5] it is $\frac{1}{5}$.)

Utilization $U = \dfrac{6}{10}$. (The instantaneous utilization over [0, 3] is $\frac{2}{3}$, over [0, 5] it is $\frac{4}{5}$, and over [0, 10] it is $\frac{6}{10}$.)

Average number of customers in the system $N = \dfrac{11.5}{10}$. (The instantaneous measure over [0, 10] is $\frac{(7-1)+(6-2)+(4-2.5)}{10} = \frac{11.5}{10}$; note that it's easier to calculate the area by partitioning it in horizontal rectangles of unit height than vertical rectangles of unit width. Over [0, 3] the measure is $\frac{(3-1)+(3-2)+(3-2.5)}{3} = \frac{3.5}{3}$, and over [0, 5] it is $\frac{(5-1)+(5-2)+(4-2.5)}{5} = \frac{8.5}{5}$.)

4.2 Shortest-job first (SJF)

```
   N(t)
   _|
 4 _|                           C1
 3 _|                 C1    ___________        C3
 2 _|         C1    ______|            |______      C2
 1 _|       _______|                          |____________
 0 _|______|______________________________________________|________ time
    0      1       2      2.5          4      5           7        (seconds)

          C1      C2     C3            C1     C3          C2
          arr     arr    arr           dep    dep         dep
```

The system becomes empty at time 7, as with FCFS. Thus it is stable and the pattern repeats every 10 s. The (steady-state) throughput $X = \frac{3}{10}$ and utilization $U = \frac{6}{10}$, as with FCFS. The customer departure, response, and wait times are:

customer	A_i	S_i	D_i	R_i	W_i
1	1.0	3.0	4.0	3.0	0.0
2	2.0	2.0	7.0	5.0	3.0
3	2.5	1.0	5.0	2.5	1.5

$$R = \frac{3.0+5.0+2.5}{3} = \frac{10.5}{3} \qquad \text{(smaller than with FCFS)}$$

$$W = \frac{0.0+3.0+1.5}{3} = \frac{4.5}{3} \qquad \text{'' ''}$$

$$N = \frac{(7-1)+(5-2)+(4-2.5)}{10} = \frac{10.5}{10} \qquad \text{'' ''}$$

4.3 Shortest-job-first-preemptive (SJFP)

```
   N(t)
   _|
 4 _|                          C3
 3 _|                    C1   ______     C1
 2 _|           C1    ______|       |___________   C2
 1 _|        _______|                           |____________
 0 _|______|__________________________________________|________ time
    0      1       2    2.5    3.5         5            7       (seconds)

           C1      C2    C3     C3          C1          C2
           arr     arr   arr    dep         dep         dep
```

When customer 2 arrives, customer 1 has remaining service time of 2 s. Since customer 2's service time is also 2 s, it does not preempt customer 1. When customer 3 arrives, customer 1 has remaining service time of 1.5 s. Since customer 2's service time of 1 s is less, it preempts customer 1. When customer 3 departs, customer 1 takes priority over customer 2 since customer 1 has lower remaining service time.

As with FCFS and SJF, the system becomes empty at time 7, the throughput $X = \frac{3}{10}$, and utilization $U = \frac{6}{10}$. The customer departure, response, and wait times are:

customer	A_i	S_i	D_i	R_i	W_i
1	1.0	3.0	5.0	4.0	1.0
2	2.0	2.0	7.0	5.0	3.0
3	2.5	1.0	3.5	1.0	0.0

$$R = \frac{4.0+5.0+1.0}{3} = \frac{10.0}{3} \qquad \text{(smaller than with SJF)}$$

$$W = \frac{1.0+3.0+0.0}{3} = \frac{4.0}{3} \qquad \text{” ”}$$

$$N = \frac{(7-1)+(5-2)+(3.5-2.5)}{10} = \frac{10.0}{10} \qquad \text{” ”}$$

4.4 Round-robin (RR) with quantum size 1 s

```
   N(t)
   _|
 4 _|                   C2     C1       C3
 3 _|              C2   ____|______|________   C2
 2 _|       C1    ____|                     |______  C1
 1 _|     ______|                                   |______
 0 _|____|_____________________________________________|________ time
    0    1      2    2.5    3      4       5       6       7      (seconds)

         C1     C2   C3                    C3      C2      C1
         arr    arr  arr                   dep     dep     dep

Just after time:      1.0    2.0        3.0    4.0    5.0    6.0    7.0

Order of customers    C1-3   C2-2       C1-2   C3-1   C2-1   C1-1
and their remaining          C1-2       C3-1   C2-1   C1-1
service                                 C2-1   C1-1
```

As with the previous disciplines, the system becomes empty at time 7, the throughput $X = \frac{3}{10}$, and utilization $U = \frac{6}{10}$.

The customer departure, response, and wait times are:

customer	A_i	S_i	D_i	R_i	W_i
1	1.0	3.0	7.0	6.0	3.0
2	2.0	2.0	6.0	4.0	2.0
3	2.5	1.0	5.0	2.5	1.5

$$R = \frac{6.0+4.0+2.5}{3} = \frac{12.5}{3} \qquad \text{(larger than with previous disciplines)}$$

$$W = \frac{3.0+2.0+1.5}{3} = \frac{6.5}{3} \qquad \text{" "}$$

$$N = \frac{(7-1)+(6-2)+(5-2.5)}{10} = \frac{12.0}{10} \qquad \text{" "}$$

4.5 Relationship between N and R

The disciplines achieve the same $U(=\frac{6}{10})$ and $X(=\frac{3}{10})$ but different R and N:

	FCFS	SJF	SJFP	RR
R	$\frac{11.5}{3}$	$\frac{10.5}{3}$	$\frac{10.0}{3}$	$\frac{12.5}{3}$
N	$\frac{11.5}{10}$	$\frac{10.5}{10}$	$\frac{10.0}{10}$	$\frac{12.5}{10}$

However, the ratio of R and N is constant. In fact, $N = R \times X$ holds for each discipline. More generally, we have the following:

Little's Law: *For any queueing system in steady-state* $N = R \times X$

Little's Law applies to single-server queues as well as networks of queues. It applies to queueing systems with one or more classes of customers. It applies to any stable subsystem of a queueing system. For example, if a queueing system has several classes of customers and X_i, R_i, and N_i are, respectively, the throughput, response time, and average number of customers of the ith class, then $N_i = R_i \times X_i$. This is true even if the overall system is unstable, e.g. even if N_j blows up for some class $j \neq i$.

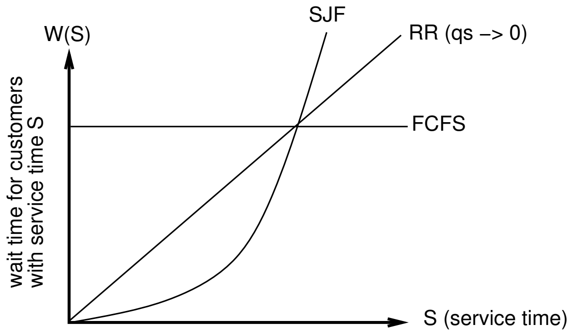
For another example, consider the waiting room of a single-server queue. Applying Little's Law to this subsystem, we get $N_W = W \times X$, where N_W is the number of waiting customers in the queue, W is the average waiting time in the queue, and X is the throughput of the queue (since every customer that leaves the waiting room also eventually leaves the server).

4.6 Relationship between R and service time

The different disciplines distinguish customers based on their service times:

service time	W_{FCFS}	W_{SJF}	W_{SJFP}	W_{RR}
1.0	3.5	1.5	0.0	1.5
2.0	2.0	3.0	3.0	2.0
3.0	0.0	0.0	1.0	3.0

In general, SJFP favors customers with small service times at the expense of customers with large service times. SJF also does this, but less severely. RR also does this, but even less severely and linearly (for small quantum sizes). Disciplines such as FCFS and LIFO do not discriminate on service times. This property is illustrated by plotting $W(S)$, the average waiting time for customers with service time S, versus S:



More generally, a queueing discipline can discriminate customers into different classes, either implicitly (based usually on customer service times) or explicitly (based on preassigned priorities).

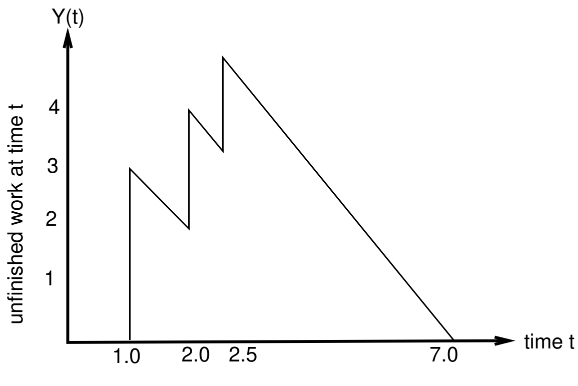
4.7 Relationship between idle and busy periods

Note that the system emptied at the same time for each of the above queueing disciplines. This too is can be generalized for single-server queues.

Observe that the server cycles between *idle periods* and *busy periods*. A busy period starts when a customer arrives to an empty queue. An idle period starts when a customer departs leaving an empty queue. A discipline is said to be *work-conserving* if the server is not idle when there is a customer waiting.

Work-conserving Law: *The sequence of idle and busy periods, and hence the utilization, is independent of queueing discipline, provided the discipline is work-conserving.*

This is obvious when we observe that the evolution of $Y(t)$, the unfinished work in the queue at time t, is independent of the queueing discipline. When customer i arrives, $Y(t)$ jumps up by S_i. Whenever $Y(t) > 0$, it decreases with slope -1. $Y(t) = 0$ is an empty system.



5 M/M/1 queues

We have seen that queueing arises because of variations in customer arrival times or service requirements. Different kinds of variations gives rise to different values of performance measures. Ideally we would like to average over arrival and service sequences that reflect the variability occurring in practice. But this is difficult to do because we often do not know the variations occuring in practice, and even when we do we often cannot analyze the resulting performance expressions.

The good news is that there are analytically simple probabilistic ways to define arrival and service sequences so that the resulting performance measures accurately describe many real-life situations. The simplest of these ways is the so-called M/M/1 assumption: that interarrival times and service times are independent and exponentially distributed with means $1/\lambda$ and S, respectively.

[For insight, we can compare the sequence of interarrival times to the sequence of numbers resulting by repeatedly throwing a fair dice. Each number is uniformly distributed over the values 1, 2, 3, 4, 5, 6 (and hence has a mean of 3.5). The numbers are independent of each other, i.e. a number's value does not tell us anything about the other numbers. In the case of interarrival times, each interarrival time is a real number that is exponentially distributed with mean $1/\lambda$, and it is independent of all other interarrival times.]

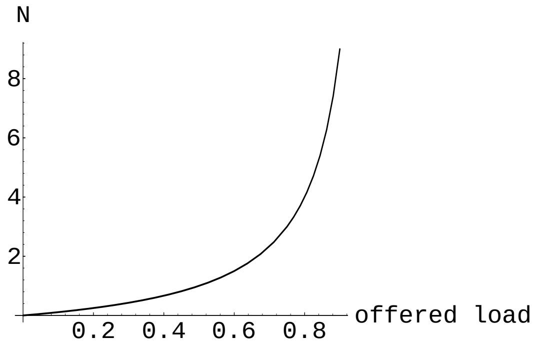
In "M/M/1", the first "M" indicates memoryless arrivals, the second "M" indicates memoryless service times, and the 1 indicates the number of servers. It turns out that if one tries to do something without *any* memory of the past, then the time taken to do that something is exponentially distributed. This is true, for example, of the time taken by elementary particles to decay.

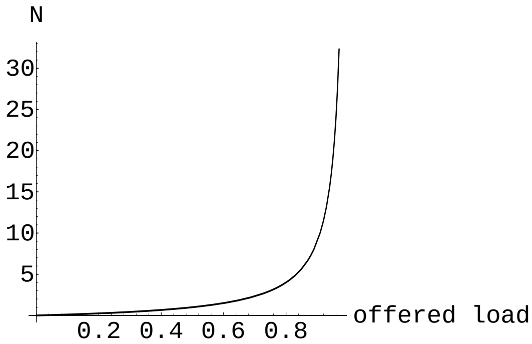
For a stable M/M/1 queue

$$N = \frac{\rho}{1-\rho}$$

Recall that $\rho > 1$ results in instability. The above equation indicate that $\rho = 1$ is also unstable. This happens when arrival and service times are defined probabilistically. Although this may seem unintuitive (and does not happen if arrival and service times are defined deterministically), it is in fact what happens in real-life.

N grows at an increasing rate as ρ approaches 1 from below. The following two plots indicate the rapidity of this growth:





Because $\rho = \lambda S$ and $N = R\lambda$ (Little's law), we have for a stable M/M/1

$$R = \frac{S}{1-\rho}$$

and, from $W = R - S$,

$$W = \frac{S\rho}{1-\rho}$$

6 A Priority Queue

Consider a single-server M/M/1 queue that gets customers with average service requirement S and arrival rate λ. Assuming $\lambda S < 1$, we have

$$N = \frac{\lambda S}{1 - \lambda S}$$

Now suppose that the stream of customers is divided into two classes, G and H, using some distinction (which is of no concern to us), e.g. H may be customers that are tall, or customers with service times above some threshold. Let λ_G and λ_H be the arrival rates for G and H customers respectively. Let S_G and S_H be the average service times for G and H customers respectively. Clearly

$$\begin{aligned} \lambda_G + \lambda_H &= \lambda \\ \lambda_G S_G + \lambda_H S_H &= \lambda S \end{aligned}$$

Suppose G customers get preemptive priority over H customers. Then H customers do not exist as far as G customers are concerned. Thus G customers perceive the server as dedicated to them. Assuming M/M/1 for the G customers, we get the average number of G customers in the system to be (the subscripts identify the customer class to which the measure applies):

$$\begin{aligned} N_G &= \frac{\lambda_G S_G}{1 - \lambda_G S_G} \\ R_G = \frac{N_G}{\lambda_G} &= \frac{S_G}{1 - \lambda_G S_G} \end{aligned}$$

H customers get whatever is left of the server, which is $1 - \lambda_G S_G$, since the work brought by G customers is $\lambda_G S_G$. Thus it takes S'_H seconds to give an H customer S_H seconds of service, where

$$S'_H = \frac{S_H}{1 - \lambda_G S_G}$$

If we assume that M/M/1 holds for the H customers also, we get

$$\begin{aligned} N_H = \frac{\lambda_H S'_H}{1 - \lambda_H S'_H} &= \frac{\lambda_H S_H}{1 - \lambda_G S_G - \lambda_H S_H} \\ R_H &= \frac{S_H}{1 - \lambda_G S_G - \lambda_H S_H} \end{aligned}$$

Note that the H subsystem is stable if $\lambda_H S'_H < 1$, which simplifies to $\lambda S < 1$. Thus both the H and G subsystems are stable given the stability of the original system, which is to be expected since the priority queueing discipline is work-conserving.

Unfortunately, the above expressions are not correct because the H customers do not satisfy M/M/1. The correct expression is [Kleinrock, vol 2, ch 3, page 125, eq 3.39]

$$R_H = \frac{S_H + \lambda_G S_G (S_G - S_H)}{(1 - \lambda_G S_G)(1 - \lambda_G S_G - \lambda_H S_H)}$$

What is the relationship between the performance measures of the original queue and the priority queue for each class of customers? Do the following hold?

$$\begin{aligned} N_G + N_H &> N \\ R_G &< R \\ R_H &> R \end{aligned}$$