Academic Journal of Applied Mathematical Sciences
ISSN(e): 2415-2188, ISSN(p): 2415-5225
Vol. 4, Issue. 6, pp: 49-52, 2018
URL: http://arpgweb.com/?ic=journal&journal=17&info=aims

Academic Research Publishing Group

Original Research Open Access

Queues with Server Utilization of One

Robert Aidoo
Department of Mathematics and Statistics, Faculty of Science, University of Windsor, Ontario, Canada

Myron Hlynka*
Department of Mathematics and Statistics, Faculty of Science, University of Windsor, Ontario, Canada

Abstract

Generally, server utilization must be less than 1 for a queueing system to be stable. One exception that maintains stability with server utilization equal to 1, is the D/D/1 case with identical interarrival times and service times. In this paper, we present several other models which are stable with server utilization equal to 1.

Keywords: Queues; Server utilization; Lindley recursion.

1. Introduction

The server utilization, or traffic intensity, for a G/G/1 queue is defined to be the ratio of the mean service time to the mean interarrival time, and is denoted by ρ. The server utilization represents the long run proportion of time that the server is busy [1, 2]. The M/G/1 and G/M/1 systems are null recurrent for ρ=1 [3]. Thus, the system will return to its current state with probability 1, but the expected time to return is infinite.

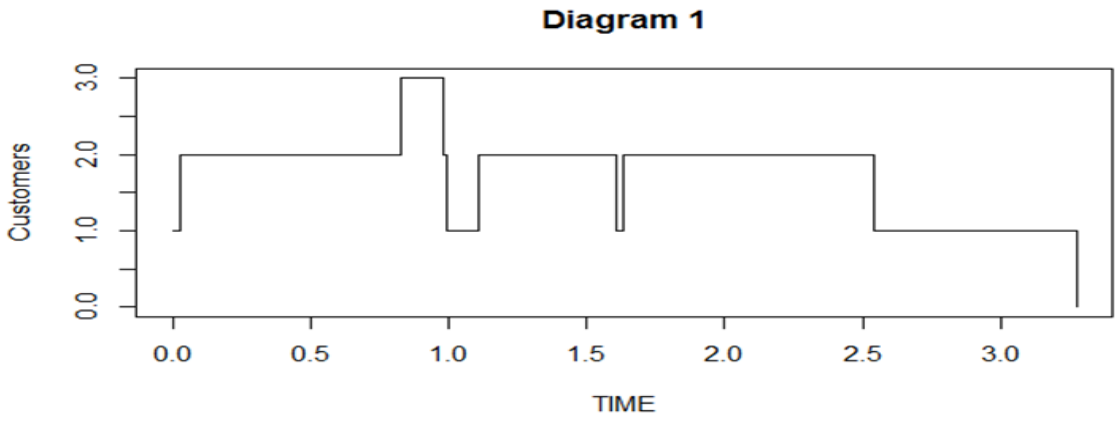
According to Kleinrock [1], "For the system G/G/1 to be stable, it must be that $0 \leq \rho < 1$. Occasionally we permit the case ρ=1 within the range of stability, in particular for the system D/D/1." In this paper, we present some modifications of the D/D/1 system with ρ=1, which maintain stability.

To begin, we present R code for plotting the number of customers in the system vs time. We assume the first arrival occurs at time 0. To illustrate, we generate 4 more random interarrival times. Then we generate 5 random service times. We obtain the actual arrival times as the cumulative sum of the interarrival times. The first customer has no wait. We use the Lindley recursion formula [see Brill [4]] to determine the other waits. We obtain the actual completion times by summing the arrival times plus the waiting times plus the service times. We merge and sort the arrival times and the completion times, tagging the arrival times with +1 and tagging the service completions with -1. The cumulative sum of the tags gives the number of customers in the system at each point in time. Here is the code.

```
#IA= interarrival times, always start at 0    A= actual arrival times
# S= service times,      W=wait times using Lindley recursion
#END= end (completion) times for each customer
#TYPE: +1 for arrival(up), -1 for service completion(down)
#TIME=merge and sort all arrival times and service times
#NCUST= number of customers in the system
IA=c(0,runif(4));   A=cumsum(IA);   S=runif(5);  L=length(IA)
W=c(0); for (i in 2:L){W=c(W,max(0,W[i-1]+S[i-1]-IA[i]))} #Lindley recursion
END=A+W+S
TYPE=c(rep(1,L),rep(-1,L));   UD=TYPE[order(c(A,END))]; Customers=cumsum(UD)
TIME=sort(c(A,END))
plot(TIME, Customers, "s", xlim=range(0:7))
```

*Corresponding Author

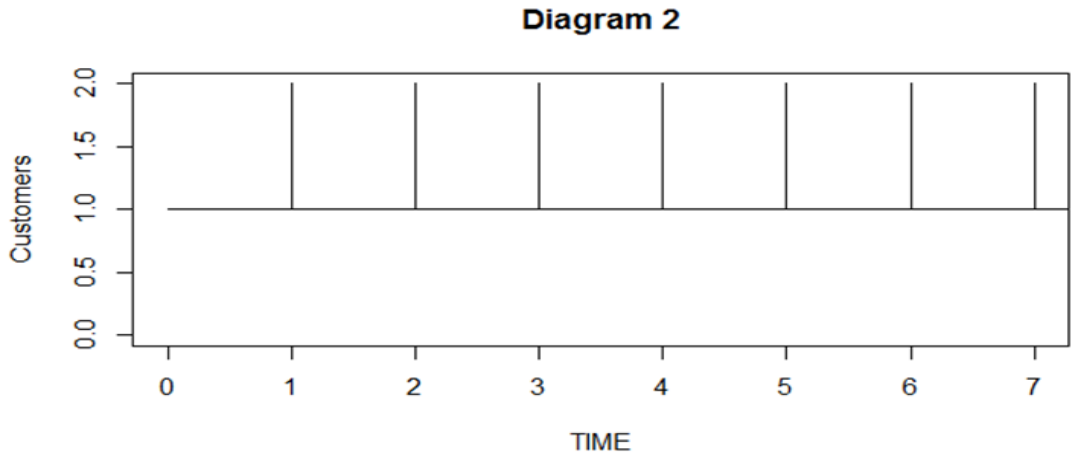
Typical output is Diagram 1 below



1.1. Standard D/D/1 Model With ρ=1

Assume that we have a D/D/1 model with the interarrival times and the service times exactly 1.

In the previous code, replace interarrival times IA and service times S by IA=c(0,1,1,1,1,1,1,1,1,…) and S=c(1,1,1,1,1,1,1,1,1,…). The result is Diagram 2.



The spikes occur since there is an arrival and service completion at exactly the same time. There is always exactly 1 customer in the system, so E(L)=1, E=0.

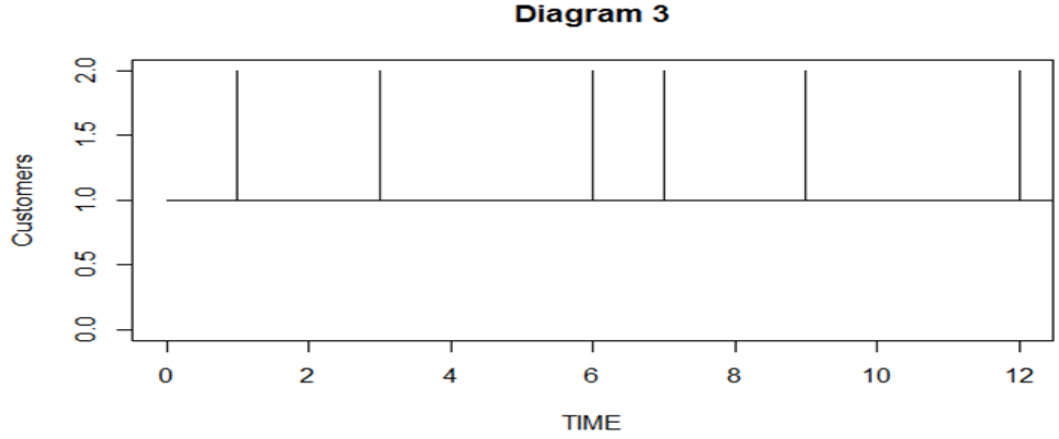
Section 2 will introduce two new models with ρ=1. Section 3 will draw some conclusions and make further comments.

2. New Models

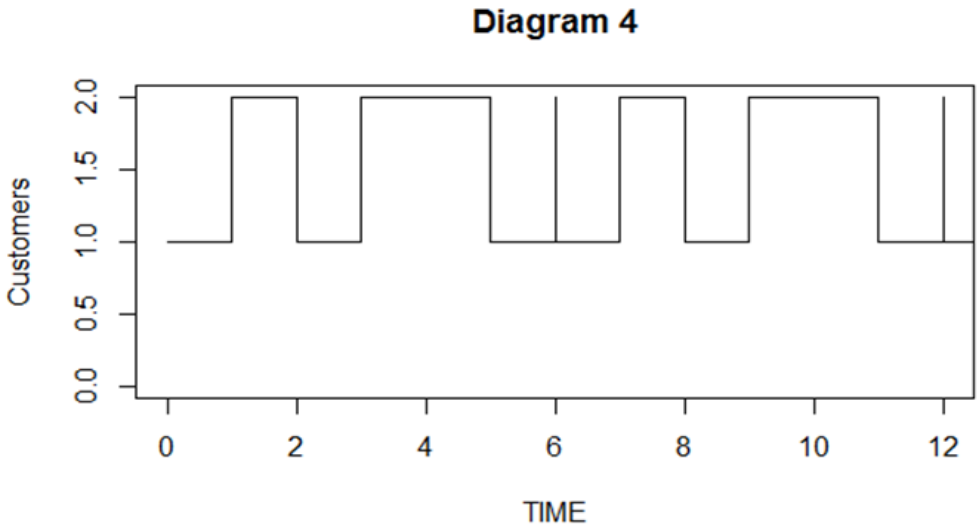
2.1. New Model 1 (Cyclical Interarrival and Service Times, ρ=1)

We apply the same R code to the following queueing model.

The interarrival times and service times are cyclical. So, we no longer have a proper G/G/1 system as both the independence and identically distributed properties are removed. As an example of cyclical interarrival and service times, we use interarrivals times IA=c(0,1,2,3,1,2,3,1,2,3,…) and service times S=c(1,2,3,1,2,3,1,2,3,…). This results in Diagram 3.



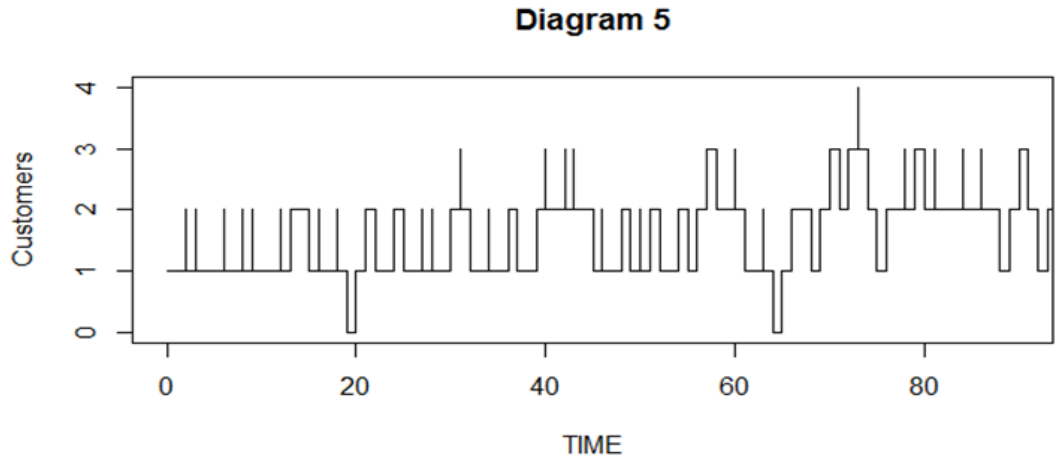
In this case, we again have E(L)=1, E=0. The diagram gives a nice illustration that if systems are scheduled, with customers and servers arriving and finishing perfectly on schedule, there should be no waiting. We next repeat this example but change the service time to S=c(2,3,1,2,3,1,2,3,1,…). The result is Diagram 4.



This time there are some differences. The cycle ends at time 6 and repeats thereafter. Now E(L)=1.5, E=1. This illustrates that bad scheduling will result in unnecessary waits. This model lacks independence of interarrival times and independence of service times and the identically distributed property of the standard G/G/1 system.

2.2. New Model 2 (Randomized Cyclical Interarrival and Service Times, ρ=1)

Here we introduce a novel type of cycling model with interarrival times in randomized groups of 1,2,3, and service times in randomized groups of 1,2,3. By this, we mean that the we use all interarrival times 1,2,3 as a group but that any random permutation of these is allowable, and similarly for the service times. As a result, each interarrival time is equally likely to be 1,2,3 so we really do have identically distributed interarrival and service times. However, we still do not have independence. A typical interarrival vector might be IA=c(0, 2,1,3, 2,1,3, 1,3,2 ,2,1,3 ,3,1,2 ,1,3,2 ,3,1,2, 1,3,2, 2,1,3,…) and a typical service vector is S=c(2,1,3, 2,1,3, 3,1,2, 1,2,3, 2,1,3, 1,2,3, 3,2,1, 2,1,3, 1,2,3, …). The result is Diagram 5.



In this case there is no clear cycle, although there actually are some built-in restrictions. The first 4 arrivals came at times 0,2,3,6. The next 3 arrivals must arrive by time 6+1+2+3=12. The next three must arrive by time 12+1+2+3=18. The zero customer level (which appears in Diagram 5) cannot reappear indefinitely. If it did, there would be an imbalance between interarrival and service times. In fact, the maximum time spent at level zero is 2 time units (for this specific randomized cycle). Also, the maximum number of customers is 4 and this can only appear instantaneously.

3. Conclusions

In this paper, we have presented 2 new queueing models which have utilization equal to 1, yet are stable systems, and always finite. One of these has discarded the standard queueing assumptions of independent and identically distributed interarrival times and service times. The other model has discarded only the independence condition. In the first cyclic model, we observed that a good choice of scheduling can reduce customer waiting time to zero.

The models have some interest in their own right, not just as interesting examples when ρ=1. In reality, we might use 3 different servers in sequence in a manner which always has exactly one server working, but the servers have different expected times for service (such as 1,2,3). In the randomized cyclical system, the expected system time is not equal to the mean of the three server times, and the expected number of customers in the system is not equal to 1.

It is possible to make the system look very close to an M/M/1 system, with ρ=1. For example, we could generate 100 independent exponential interarrival times. Then cycle the 100 times in a permuted order as the service times. Repeat as often as desired. This type of system would appear to a viewer to be an M/M/1 system, with ρ=1. However, it would fail in the independence assumption yet this failure would be very hard to detect.

References

[1] Kleinrock, L., 1975. "Queueing systems." *Wiley Interscience*, vol. 1.

[2] Gross, D., Shortle, J. F., Thompson, J. M., and Harris, C. M., 2008. *Fundamentals of queueing theory*. 4th ed. Wiley.

[3] Bhat, U. N., 2015. "An introduction to queueing theory: Modeling and analysis in applications." *Birkhäuser*.

[4] Brill, P. H., 2018. *Level crossing methods in stochastic models*. 2nd ed. Springer.