MA-402
Queueing Models for Performance Analysis
Assignment # 1

Japsimran Singh
04010218

Department of Electronics and Communications Engineering
IIT Guwahati

A Queueing Model for Call Blending in Call Centers

I. INTRODUCTION

In this paper, the implementation of the queueing model with its typical application in call centers is discussed .The queueing system is divided into two types of jobs .The first type of jobs have a constraint on the performance, i.e., the average waiting time has to be below a certain level. Next to this time-constrained type there is a second type of jobs, available in an infinite quantity,for which the objective is to serve as many as possible. The arrivals of the first job type are determined by a Poisson process and the service times of both job types are independent exponentially distributed.Both job types are served by a common pool of s servers under a non-preemptive discipline. The question that we will answer in this note is how to schedule these s servers to maximize the throughput of type 2 jobs while satisfying the waiting time constraint on the type 1 jobs.

Modern call centers deal with a mixture of incoming and outgoing calls.Of course, there are constraints on the waiting time of incoming calls.The traditional solution is to assign call center employees (often called agents") to either incoming or outgoing calls. However, the call rate fluctuates over the day and in order to handle the calls during peak
periods usually a substantial number of agents needs to be assigned to incoming calls. Consequently, the productivity of the agents, i.e., the fraction of time that agents are busy, is low during other periods. On the other hand, assigning fewer agents to incoming calls increases the productivity, but leads to longer waiting times. Hence, there is a need to balance productivity and waiting times. The solution is call blending,dynamically assigning agents either to incoming or outgoing traffic.

II. MODEL DESCRIPTION

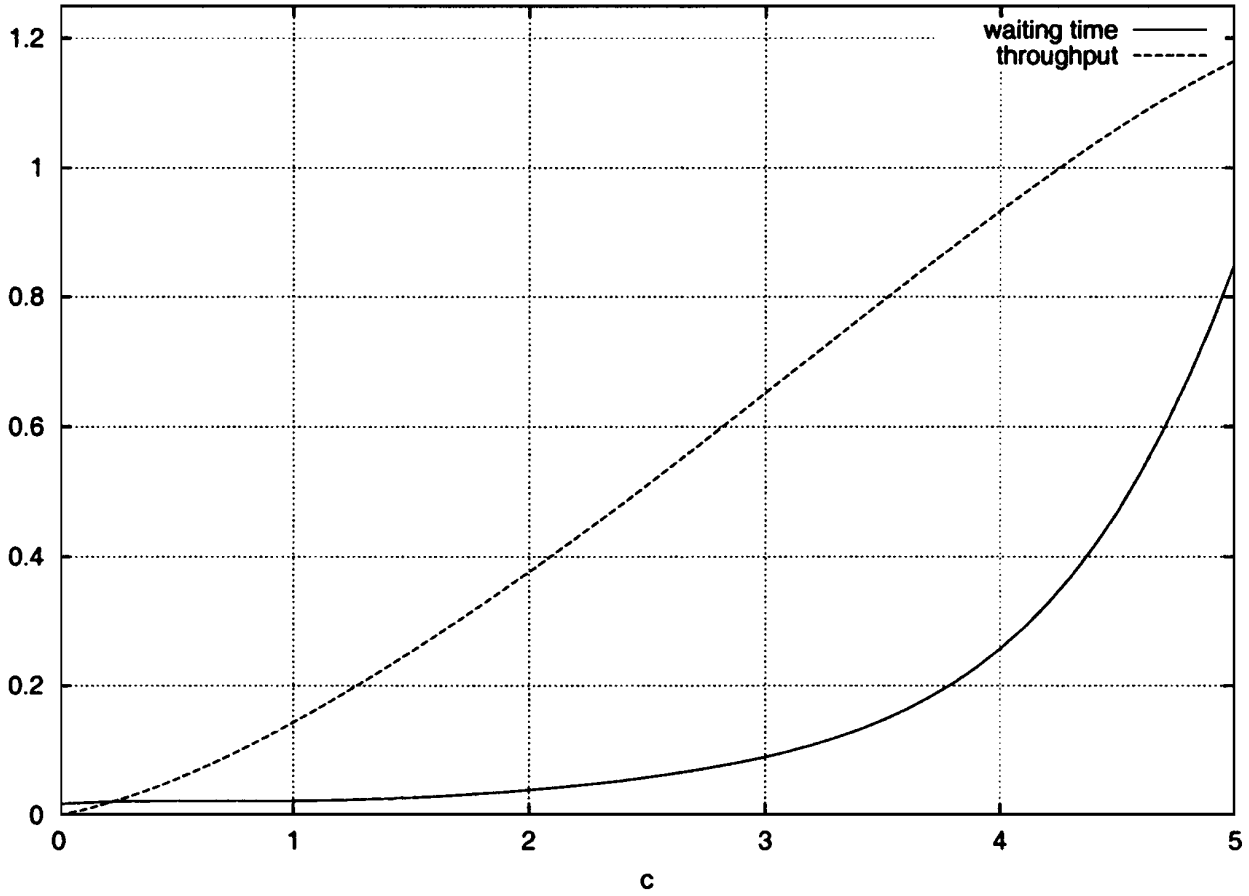
There are two types of traffic, type 1 and type 2, having independent exponentially distributed service requirements with rates $\mu 1$ and $\mu 2$. Type 1 jobs arrive according to a Poisson process with rate λ, and there is an infinite waiting queue for jobs that cannot be served yet. There is an infinite supply of type 2 jobs. There are a total of s identical servers. The long-term average waiting time of the type 1 jobs should be below a constant .Waiting excludes the service time; if the response time is to be considered, then the average service time, $1/\mu 1$, should be added to the average waiting time. The objective for type 2 jobs is to maximize its throughput, i.e., to serve on average per unit of time as many type 2 jobs as possible,of course at the same time obeying the constraint on the type 1 waiting time. The following control actions are possible. The moment a server finishes service, or, more generally at any moment that a server is idle, it can take one of the following three actions: start serving a type 1 job (if one or more are waiting in the queue for service), start serving a type 2 job, or remain idle.

Note that in our model preemption of jobs in service is not allowed here. When preemption is allowed the problem is trivial. The optimal policy will assign all servers to type 2 jobs when no

type 1 jobs are present in the system. When a type 1 job arrives then it is clearly optimal to interrupt the service of a type 2 job and to serve the type 1 job. Hence,the waiting time constraint is satisfied and the type 2 throughput is equal to $\mu 2$ (s-$\lambda/\mu 1$).Note that any work-conserving policy that satisfies the waiting time constraint is optimal and achieves the same throughput. In practice the jobs that can be preempted in call centers are e-mail messages. Therefore, it is beneficial for call centers to encourage customers to send their requests by e-mail.This finishes the description of our model. In the next two sections,we will deal with the case $\mu 1 = \mu 2$ and $\mu 1 \neq \mu 2$, respectively. A first question that has to be answered is whether it is at all possible to find a policy that satisfies the waiting time constraint of the type 1 traffic.

III. EQUAL SERVICE REQUIREMENTS

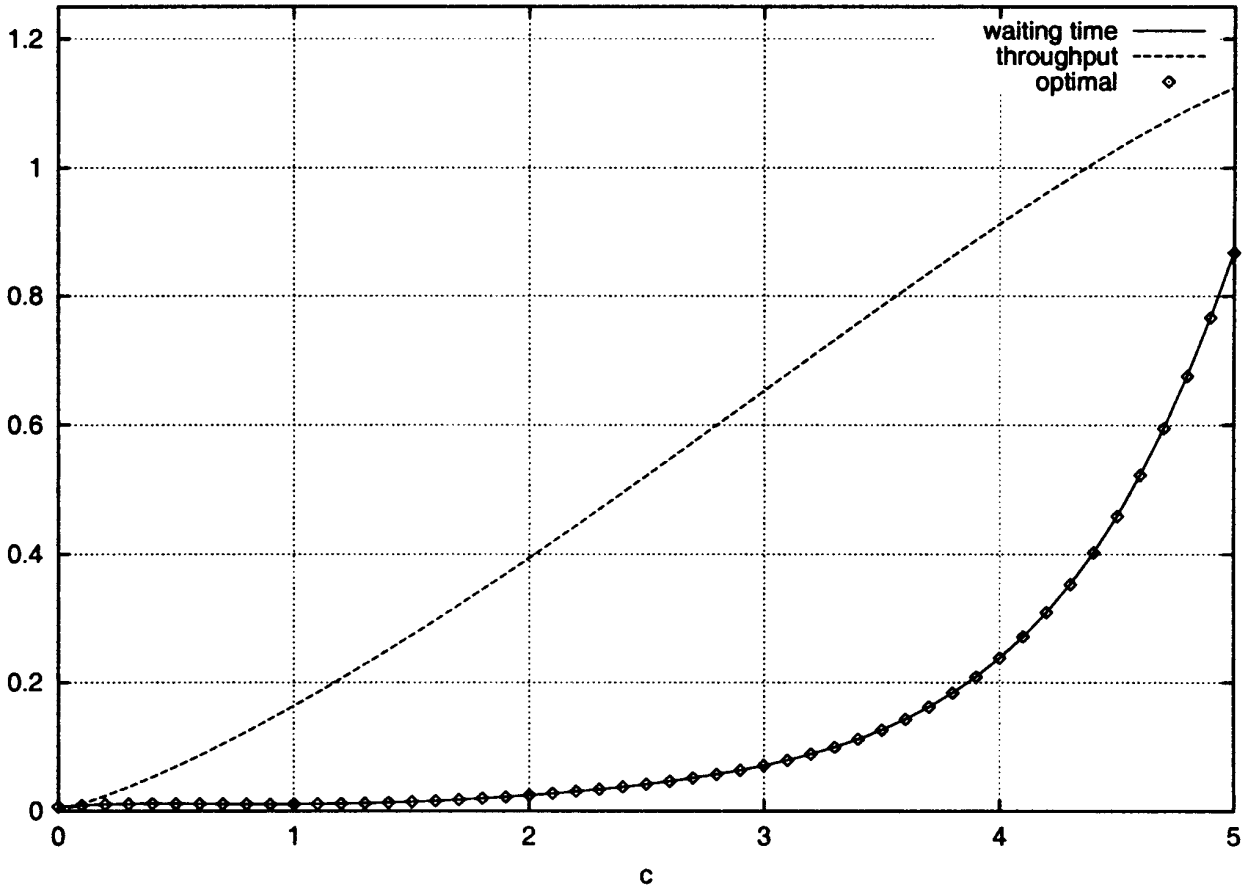
Let $\mu=\mu 1=\mu 2$,Consider the event that a server becomes idle,and that there are one or more type 1 jobs waiting. Then the controller has to choose between scheduling a type 1 or a type 2 job (or idling,but this is evidently suboptimal). Giving priority to a type 2 job and delaying type 1 jobs obviously leads to higher waiting times. Delaying the processing of a type 2 job does not change the performance for this class, as we are interested in the long-term throughput. This intuitive argument implies that, when a server becomes idle and a type 1 job is waiting, it is optimal to assign this type 1 job to the server.



$\mu=\mu 1=\mu 2=1/3$, s=5, $\lambda=1/2$

IV. UNEQUAL SERVICE REQUIREMENTS

When $\mu 1 \neq \mu 2$, the analysis is more complicated. In this case we have to differentiate between type 1 customers and type 2 customers.The optimal policy will depend on these different classes and can be very complicated. From a practical viewpoint, these policies can also be difficult to implement in call center software. Therefore, we prefer to study simpler policies and we restrict ourselves to the class of threshold policies. Numerical experiments indicate that this theorem also holds in case of unequal service requirements.The restriction to the class of threshold policies forces the policies to be simple and appealing. Moreover, we will show by numerical computation that threshold policies are a good approximation to the optimal policy.



$\mu 1=4/10, \mu 2=3/10$, s=5, $\lambda=1/2$

Again, we see that the average waiting time increases slowly, while the throughput increases nearly linearly. Since the threshold policy does not need to be the optimal policy, it is interesting to numerically compute the performance of the optimal policy. This can be done by using the dynamic programming operator for a fixed value of the Lagrange parameter Υ yielding a policy π. The waiting time of type 1 jobs is obtained by iterating the dynamic programming operator following policy π without the reward rate for scheduling type 2 jobs. Similarly, the

throughput of type 2 jobs is obtained by iterating the dynamic program-ming operator following policy without the cost rates for waiting.

For the fixed value of the Lagrange parameter Υ,we thus obtain the optimal waiting time with the corresponding throughput.The graph of the optimal policy is generated by computing the waiting times with the corresponding throughput for various values of the Lagrange parameter Υ. Matching the throughput with the throughput computed by the threshold policy yields a value of c.The minimal waiting time that can be achieved for that level c is the waiting time computed by the optimal policy. This waiting time is denoted by a dot in the graph.It is interesting to note that the optimal policy yields a performance very close to the approximative threshold policy. Extensive experiments show that for other parameter values the same result is obtained.The experiments indicate that the optimal policy behaves nearly as a threshold policy, but minor differences occur when x + y is close to the threshold value.