

Boundary Problem in a System with global FCFS and Presorting

Willem Mélange, Joris Walraevens, Dieter Claeys, Bart Steyaert and Herwig Bruneel

*Department of Telecommunicatie en Informatieverwerking, Ghent University,
Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium*

Abstract. In this paper we consider a continuous-time queueing system with two different types (1 and 2) of customers with two dedicated servers (also named 1 and 2). This means server 1 (2) can only serve customers of type 1 (2). The goal of this paper is to study the boundary conditions for a system with global FCFS and presorting service discipline, i.e., all arriving customers are accommodated in one single FCFS queue, regardless of their types, with an exception of the first N customers. For the first N customers the FCFS rule holds only within the types, i.e. customers of different types can overtake each other in order to be served. The motivation for our work comes from traffic and is to be able to give advice about the optimal length of filter lanes, i.e. lanes reserved for vehicles making a specific turn at a junction. This paper is a first step in this process.

Keywords: queueing; continuous time; Markov chain; global FCFS; Presorting; boundary

PACS: 02.50.Ga; 02.50.Ey

INTRODUCTION

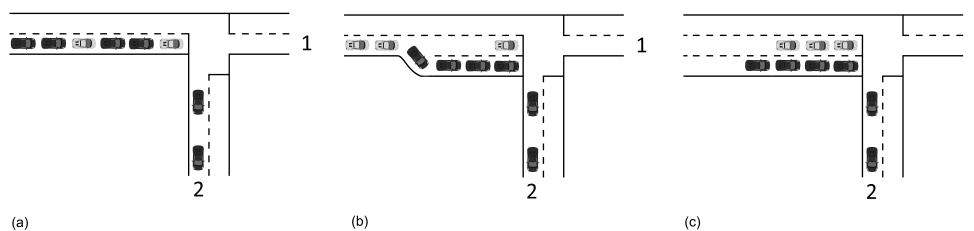


FIGURE 1. Light grey vehicles with destination 1 and dark grey vehicles with destination 2 approaching a traffic junction

The motivation for this work is an every day problem in traffic. Traffic jams might occur for multiple reasons. One reason are traffic junctions. Consider for instance the following situation: vehicles approach a junction with two possible destinations (1 and 2) as seen in Fig. 1. In traffic context, it is often not physically feasible to provide two separate lanes for each possible destination (as seen in Fig. 1c). If it would be, vehicles for both directions can be kept apart completely. The other extreme occurs much more frequently, is that there is one lane on the main road (Fig. 1a). In the case where there is only one lane on the main road, it is possible that vehicles on that road heading for destination 1 may be hindered or even blocked by vehicles heading for destination 2, even when the subroad leading to destination 1 is free, simply because cars that go to 2 are in front of them. In other words, there is a first-come-first-serve (FCFS) order on the main road regardless what destination they have. In the rest of this paper we will call this service discipline global FCFS (gFCFS). In queueing theory terms, a service discipline where there are 2 types of customers that are accommodated in a single queue and who are served in a FCFS manner regardless of their type. When we look at the case in Fig. 1a, there is even a global first-in-first-out (gFIFO) order on the main road. At any given time, at most one server will be working. A possible way to minimize the impact of this blocking phenomenon is the use of filter lanes, i.e. lanes reserved for vehicles making a specific turn at a junction (as seen in Fig. 1b). It is clear that in this case we cannot longer talk about gFCFS or gFIFO as service discipline. In the rest of the paper we will call this new service discipline, which can be seen as sort of relaxation of the gFCFS service discipline, gFCFS with presorting (P-gFCFS). Again in queueing theory terms, a service discipline where there are 2 types of customers that are accommodated in a single queue and who are served in a FCFS manner regardless of their type with an exception for the first N customers. Thus the customers can only be served if they are one of the first N customers in the system

and there are no customers of the same type in front of them. Fig. 1b is an example of such a system with a P-gFCFS service discipline. In this specific case, N is equal to 4. If the vehicle is in the first 4 vehicles on the main road and there is no vehicle in front of him with the same direction, he will be able to drive without delay caused by other vehicles to his destination. However if there are 4 vehicles with another destination in front of him, even when there are no vehicles in front of him with the same destination, the vehicle is not able to drive without delay caused by other vehicles to his destination (as seen in Fig. 1b).

PROBLEM FORMULATION

We consider a continuous-time queueing model with infinite waiting room. There are two servers, where server 1 is working at rate μ_1 and server 2 at rate μ_2 (exponential service times). There are two types (classes) of customers. Each of the two servers is dedicated to a given class of customers. In this case, server 1 always serves customers of type 1 and server 2 always serves customers of type 2. The customers are served as follows: if both types are present in the first N customers in the system, the first customer of each type can be served by its server. The customers not in the first N customers are served in a global FCFS way, i.e., they are blocked not only by customers of their own type in front of them, but also by customers of different type. We will call this service discipline global FCFS with presorting (P-gFCFS). For example, if the first $i - 1$ customers are of type 1 and the i -th customer is of type 2, then this customer can be served by server 2 if $i \leq N$. However, if the first N customers are of type 1 and the $N + 1$ -th customer is of type 2, then this customer cannot be served by server 2 even if the server is idle.

The customers enter the system according to a Poisson arrival process with mean arrival rate λ . With probability σ , the customer is of type 1 and with probability $1 - \sigma$ the customer is of type 2.

Note that we are only interested in the number of customers of type 1 and 2 in the first N customers of the system. Thus the exact queueing order of the types of the first N customers is of no importance. These observations lead to a Quasi-birth-death process with $N + 1$ phases (m), and the levels are represented by the number of customers in the system. The phase m represents the number of customers of the first N in the system that are of type 2 (and thus $N - m$ of type 1). The number of states in the boundary conditions is thus dependent on the parameter N . For example, when $N = 10$, we have 55 number of states in the boundary conditions. But when $N = 100$, we have already 5050 number of states. An efficient method for solving the boundary conditions imposes itself.

SOLUTION

The transition rate matrix S of a QBD (see also [1]) is given by

$$S = \begin{bmatrix} B_{0,0} & B_{0,1} & 0 & & \\ B_{1,0} & A_1 & A_2 & & \\ 0 & A_0 & A_1 & A_2 & \\ & & \ddots & \ddots & \ddots \end{bmatrix}, \quad (1)$$

and to solve the boundary conditions, the following equation is used

$$[\pi_0 \quad \pi_1] Q = [\pi_0 \quad \pi_1] \begin{bmatrix} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} B_{0,0}^* & B_{0,1} \\ \\ \\ B_{1,0}^* & A_1 + RA_0 \end{bmatrix} \end{bmatrix} = [1 \quad 0 \cdots 0], \quad (2)$$

where R is the rate matrix, π_0 are the stationary probabilities to be in the boundary states and π_1 are the stationary probabilities to be in the first states of the repeating portion of the QBD. Notice that the first column is replaced by the

TABLE 1. Comparison different methods in matlab in msec

N*	block inv [†]	inv	linsolve
10	3	2	4
25	16	7	20
50	187	215	668
75	1038	2052	6528
100	3686	10634	34189

* with $\lambda = 0.8, \mu_1 = 1.2, \mu_2 = 0.9$ and $\sigma = 0.2$

[†] method described in this paper

with $i = 1, 2, \dots, N$ where $d_{0,0}^L = Q_{0,0}$ and an upward sweep,

$$c_{i+1}^R = -Q_{i,i+1}(d_{i+1,i+1}^R)^{-1}, \quad (6)$$

$$d_{i,i}^R = Q_{i,i} + c_{i+1}^R Q_{i+1,i}, \quad (7)$$

with $i = N-1, N-2, \dots, 0$ where $d_{N,N}^R = Q_{N,N}$. After these two sweeps, the required elements of $G = Q^{-1}$ can be computed in a number of largely independent steps,

$$G_{i,i} = (-Q_{i,i} + d_{i,i}^L + d_{i,i}^R)^{-1} \quad (8)$$

where $i = 0, 1, \dots, N$ and

$$G_{i,j} = \begin{cases} G_{i,i} c_{i+1}^R c_{i+2}^R \cdots c_j^R, & \text{for } i < j, \\ G_{i,i} c_{i-1}^L c_{i-2}^L \cdots c_j^L, & \text{for } i > j. \end{cases} \quad (9)$$

CONCLUSION

By adding transitions (without influencing the stationary probabilities), we made it possible to use a block tridiagonal algorithm as presented in previous Section and saving computational time for solving the boundary probabilities. In Table 1 we compared the block tridiagonal algorithm in MATLAB with the methods `inv()` and `linsolve()` of MATLAB to solve the boundary probabilities. Here we note that the higher the parameter N , the better the block tridiagonal algorithm performs compared to the other methods. Notice also that in [2] the algorithm gets optimized for parallel computation and even better times could be achieved.

ACKNOWLEDGMENTS

This research has been funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office.

REFERENCES

1. G. Latouche and V. Ramaswami, *Introduction to Matrix Geometric Methods in Stochastic Modeling*, 1999.
2. S. Skelboe, *The Scheduling of a Parallel Tiled Matrix Inversion Algorithm*, 2009.