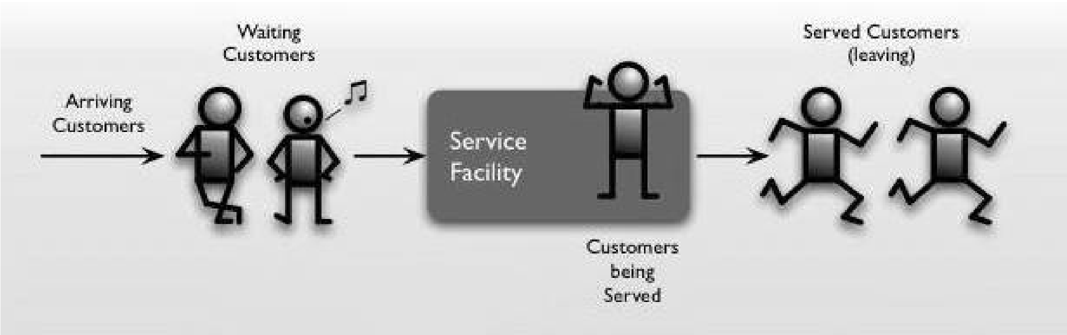
Performance Analysis of a Customer Service Center with Chat Sessions

Georgios Galvas

MSc Business Analytics

August 2015



supervisor: René Bekker

VU University Amsterdam
Faculty of Sciences
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands

Preface

The Master's in "Business Analytics" is a multidisciplinary programme, aimed at improving business processes by applying a combination of methods based on mathematics, computer science and business management. Theories from Business Analytics are typically applied to areas such as supply chain planning, data mining, call center management, revenue management and risk management.

As part of the BA programme, students are required to produce a 'thesis'. This is an account of a research project undertaken by the student to a specific problem statement. The input for this research may involve the use of computer-generated data, although it can also be drawn from the existing literature.

I want to thank my supervisor René Bekker, assistant professor at VU University, for guiding me throughout this research. He always made time for me, gave constructive criticism and helpful suggestions. That helped me a lot throughout the research. I would also like to thank Alex Roubos for providing me with the call center data I needed in order to complete my research.

Georgios Galvas
August 2015

Contents

1 Introduction

A *queueing system* is a system in which customers, orders or jobs of any kind arrive so that they get serviced or processed by a number of servers or machines. The primary goal of *queueing theory*, the mathematical study of these systems, is to predict queue lengths and waiting time. The results are often used in the decision making process of how to allocate resources that provide these services in a way that balances the waiting time with the cost of the servers.

Our main interest is to analyse the performance of a queueing system in which customers arrive and request service by a number of servers. More specifically, we deal with a center that serves incoming contact requests from customers. For convenience, we will call these systems *Customer Service Centers* or shortly *CSC's*. In a CSC, the machines or servers are substituted by the employees of this center and from now on we will refer to them as *agents*.

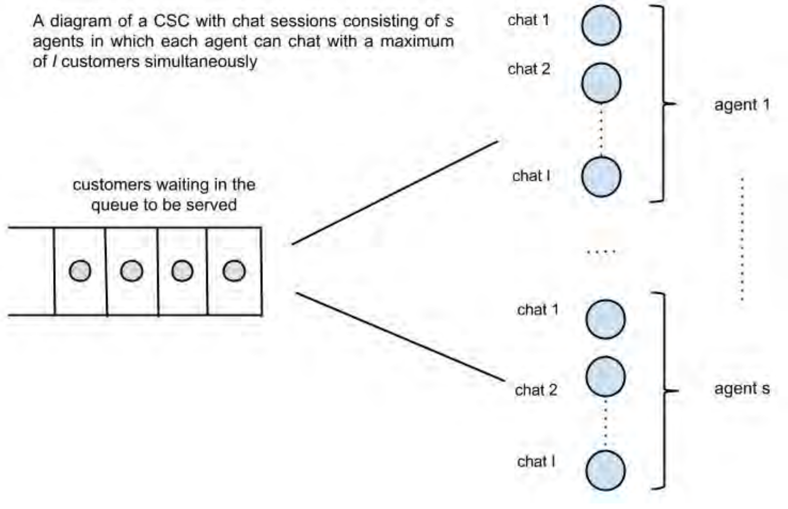
1.1 Customer service centers with chat sessions

Many studies have been carried out on Customer Service Centers in which arriving customers enter the system and require to be served by a number of agents. In these studies, each agent could serve only one client at a time. After the completion of the service and departure of the customer, the agent proceeds immediately with the service of the next customer waiting in the queue, if there is any.

In the present study we incorporate an extra feature into the CSC: the ability of simultaneous service of more than one customer by a single agent (see figure 1). These centers offer service in a different way than that of a telephone or email service: clients enter the website of the company and communicate in real time with the CSC's agents through instant messaging. We refer to these systems in which agents provide their services by chatting simultaneously with more than one clients as *Customer Service Centers with chat sessions* or simply *CSC with chat sessions*.

Chat sessions are gaining popularity in contact centers, especially when it comes to computer and software companies: solutions to customers' problems can be handled easier with chats than telephone servicing and the procedure is more interactive than that of email servicing. Consequently, the efficiency of the chatting service is higher. Furthermore, customers while waiting for a response from the agent can perform other computer-based tasks.

Figure 1:



A diagram of a CSC with chat sessions consisting of *s* agents in which each agent can chat with a maximum of *l* customers simultaneously

1.2 Research question

The main goal of the paper is to answer the following question:

How can we develop a suitable model to determine service levels for chat sessions in a customer service center?

When a customer that requires service finds more than one agent with a free slot, he should be assigned to one of them. The rule under which the customer is assigned to an agent will from now on be referred to as the *routing policy.* Considering this, another interesting question is:

In a contact center with chat sessions that consists of multiple agents what is the difference in the performance under different routing policies?

1.3 Overview

We begin in section 2 with a general analysis of the CSC with chat sessions as a Markov chain, together with some notation and assumptions. The analysis of a CSC with chat sessions with only one agent is presented in section 3. Then, in section 4, the case of two agents is considered and analysed under two different perspectives regarding the routing policy. We create the models for these cases and prove the distribution of the number of customers present in the center together with the distribution of the waiting time, the probability of waiting and the probability of finding at least one idle agent. Next, in section 5, by using real data from a call center with chat sessions that operates in Brazil we compute the parameters needed for the computations of the above measures. Finally, several comparisons are made regarding the performance of this center depending on the different routing policies we choose. These are presented in section 6.

2 The chat model

First, before proceeding with the analysis of the model, we introduce some notation together with some definitions that will be useful throughout the rest of the paper:

2.1 Notation and assumptions

- By saying that customers arrive in the CSC or they enter the system we actually mean that they request to be served through the chatting application of the center.

- It is assumed that there are no impatient customers. We do not consider the case where customers leave the system before they get service.

- As soon as a customer is served and departs, the first customer in the queue takes his place in the chat. Agents do not have breaks after a departure and they immediately proceed to the service of the new customer, with the exception when there are no remaining customers waiting in the queue.

- The number of agents in the center is denoted by s.

- Every agent can chat simultaneously with at most I number of customers and this number is the same for all the agents of the center. We call this number *maximum chat load.*

- We use the term *'level i'* to refer to the activity (or task) of helping i customers at the same time, and an agent is said to be at level i at a certain time if that agent is chatting with i customers.

- An arriving customer finding all agents being at the same level i (with $i < I$) is always assigned to the agent with the smallest index.

- The *service rate* depends on the level i. We denote by μ_i the service rate considered only when the agent is chatting with i customers.

- Customers arrive according to a Poisson process with rate λ. Hence, interarrival times are exponentially distributed.

- When the number of customers in the chat is fixed an equal to i then the service time is exponentially distributed. Note that when a customers enters or leaves the chat the service rate changes.

- The *occupation rate*, which we denote with ρ is

  $$\rho = \frac{\lambda}{s \cdot \mu_I}$$

  For the system to be stable we demand that the rate in which customers arrive is less than the total service rate when there are I chats. Otherwise, the queue is growing to infinity and the system explodes. Thus, we require that

  $$\rho < 1$$

Considering that the purpose of this paper is to make a performance analysis of a CSC with chat sessions and not to identify what the optimal maximum chat load should be under several circumstances, we choose I to be constant. Specifically, based also on the data we have, we choose $I = 3$ for the rest of the paper. Consequently, the parameters of interest are μ_1, μ_2 and μ_3, i.e. the service rate when an agent is chatting with 1, 2 or 3 customers respectively, and thus the occupation rate is

$$\rho = \frac{\lambda}{s \cdot \mu_3}$$

2.2 Routing policies

We are going to analyse the CSC with chat sessions under two different perspectives depending on the way that customers are assigned to the agents. There are two different angles under which we can view this system, according to the routing policies:

1. *'Maximum load'*
   The chat is routed to the agent with the most chats (as long as the number of chats does not exceed I) in order to keep a number of agents available for other tasks.

2. *'Balanced load'*
   The chat is routed to the agent with the least number of chats. This way, the aim is to balance the load between all the s agents.

As a rule, for the case of the Maximum load routing policy, when there are more than one agents having the same maximum number of customers in their chat (less than I) we arbitrarily choose the arriving customer to be routed to the agent with the least index between them. On the same line of reasoning, for the Balanced load case, when there are more than one agents having the same least number of chats the next customer to arrive will be signed to the agent with the least index among them.

2.3 Defining a Markov chain

As a first step we need to find a stochastic process that describes our system. There are different descriptions and the most appropriate may depend on the circumstances. These circumstances can be the number of agents staffed in the contact center, the maximum number of customers each agent can chat with or the routing policies we choose.

One way of describing the state of the system is by a $(s+1)-$dimensional random variable, with the first dimension counting the number of customers waiting in the queue and the remaining s dimensions keeping track of the number of customers chatting with each agent. Then the state of the system at time t is described by the following random variable:

$$X(t) = (L_q(t), a_1(t), a_2(t), ..., a_s(t))$$

where $a_i(t)$ is the number of chats of agent i at time t, with $i \in \{1, 2, ..., s\}$ and $L_q(t)$ is the number of customers waiting in the queue at time t.

An alternative description is by counting the number of customers that are present in the queue and the number of agents that are chatting with $0, 1, 2, ..., I$ customers. The state description will then be given by the random variable below:

$$X(t) = (L_q(t), c_0(t), c_1(t), c_2(t), ..., c_I(t))$$

where $c_j(t)$ is the number of agents chatting with j customers at time t, with $j \in \{0, 1, ..., I\}$.

In cases where the number of agents is smaller than the maximum chat load or equal, i.e. $s \leq I$, the first is a more suitable description. Unfortunately, as the number of agents grows, so does the dimensions of the state description, which leads to computational problems. On the other hand, the second is a more convenient way when aiming to compare the performance of systems with constant I and for different values of s since it more easily scales with the number of agents.

As agreed, for the rest of the paper the maximum chat load of each agent is $I = 3$ customers, and it is easy to see that in the first case, for $s = 2$ agents we have a $3-$dimensional state description. Moreover, by counting the number of agents with $0, 1, 2$ and 3 customers in their chat session and those waiting in the queue, the second description yields a 5 dimensional state space regardless the number of agents.

3 Chat sessions with 1 agent

We start by considering a system with only one agent. Even though it is not realistic for a CSC to have only one agent, the analysis of this case will be instructive when building the model for the cases of more agents, i.e. $s \geq 2$. In the system with one agent, customers arrive according to a Poisson process with rate λ and they either wait in the queue if the agent is busy chatting with other customers, or they join the chat if there are less than 3 slots available. The service rates obviously depend on the number of customers that are present in the chat.

As a state description for the system at time t we choose the number of customers in the system, i.e. all the customers waiting in the queue and those chatting with the agent. From figure 2 we can see that the system with one agent has only two kind of state transitions, an increase or decrease by one (a "birth" or a "death" respectively) and so we can identify that this is a *"birth-death" Markov process*. The analysis of a system like this is quite standard and is presented here for illustration reasons before the case of $s = 2$ in which the analysis is similar to some extend.

3.1 Number of customers in the center

We denote with $p_n(t)$, $n = 0, 1, 2, ...$, the probability that there are n customers in the system at time t. Since we are interested in the limiting behaviour of the system, we focus on p_n, where:

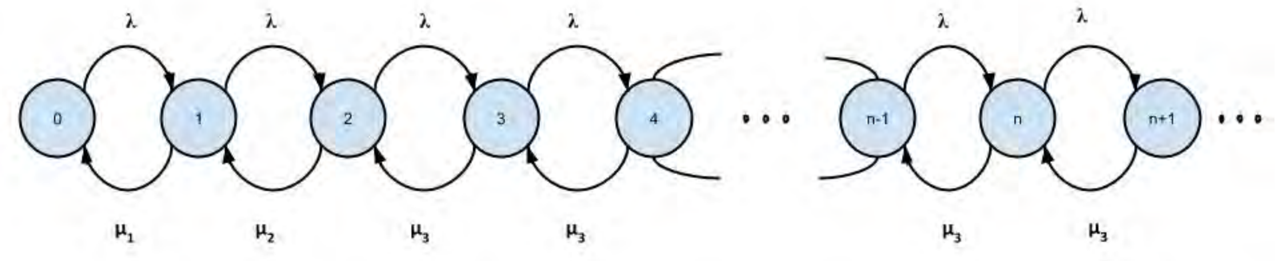
$$p_n = \lim_{t \to +\infty} p_n(t)$$

In order to find the distribution of the number of customers in the system, we first need to derive the equilibrium equations. We do that by looking at the diagram in figure 2. Considering the *Global balance principle* which states that the flow into any set of states is equal to the flow out of that set and by applying it here to the set $\{0, 1, 2, ..., n-1\}$ we get the following balance equations:

$$\lambda \cdot p_0 = \mu_1 \cdot p_1$$
$$\lambda \cdot p_1 = \mu_2 \cdot p_2$$
$$\lambda \cdot p_2 = \mu_3 \cdot p_3$$

Notice that these three equations refer to the system when there are no customers waiting in the

Figure 2: State transition diagram for the case of 1 agent



queue. From the above, by expressing all probabilities in terms of p_0 we get the following:

$$p_1 = \frac{\lambda}{\mu_1} \cdot p_0$$

$$p_2 = \frac{\lambda}{\mu_2} \cdot p_1 = \frac{\lambda}{\mu_2} \cdot \frac{\lambda}{\mu_1} \cdot p_0 \qquad \Rightarrow \qquad p_2 = \frac{\lambda^2}{\mu_1 \mu_2} \cdot p_0$$

$$p_3 = \frac{\lambda}{\mu_3} \cdot p_2 = \frac{\lambda}{\mu_3} \cdot \frac{\lambda^2}{\mu_1 \mu_2} \cdot p_0 \qquad \Rightarrow \qquad p_3 = \frac{\lambda^3}{\mu_1 \mu_2 \mu_3} \cdot p_0$$

Gathering the above in one expression we have:

$$p_n = \frac{\lambda^n}{\prod_{j=1}^{n} \mu_j} \cdot p_0 \qquad or \qquad p_n = \lambda^n \cdot \prod_{j=1}^{n} \mu_j^{-1} \cdot p_0, \qquad for\ \ n = 0, 1, 2, 3.$$

As seen from the state transition diagram, when there are more than 3 customers in the center the system behaves like an M/M/1 queue, in which the rate that a transition from n to $n+1$ occurs is λ (arrival) and the rate that a transition from $n+1$ to n occurs is μ_3 (departure). Hence,

$$\lambda \cdot p_{n-1} = \mu_3 \cdot p_n \qquad \Rightarrow \qquad p_n = \frac{\lambda}{\mu_3} \cdot p_{n-1} \quad , \qquad for\ \ n > 3$$

or

$$p_n = \left(\frac{\lambda}{\mu_3}\right)^{n-3} \cdot p_3, \qquad for\ \ n \geq 3$$

This can also be written as:

$$p_{3+n} = \left(\frac{\lambda}{\mu_3}\right)^{n} \cdot p_3, \qquad for\ \ n = 0, 1, 2, ...$$

We have already found p_3 and by substituting it in the above equation we get:

$$p_{3+n} = \left(\frac{\lambda}{\mu_3}\right)^{n} \cdot \frac{\lambda^3}{\mu_1 \mu_2 \mu_3} \cdot p_0, \qquad for\ \ n = 0, 1, 2, ...$$

Now we have expressed all probabilities in terms of p_0 and we are left with finding p_0. From the normalization property we have:

$$
\begin{aligned}
1 &= \sum_{n=0}^{+\infty} p_n \\
&= (p_0 + p_1 + p_2) + \sum_{n=0}^{+\infty} p_{3+n} \\
&= p_0 \cdot \left(1 + \frac{\lambda}{\mu_1} + \frac{\lambda^2}{\mu_1\mu_2}\right) + \sum_{n=0}^{+\infty} \left(\frac{\lambda}{\mu_3}\right)^n \cdot p_3 \\
&= p_0 \cdot \left(\frac{\mu_1\mu_2 + \mu_2\lambda + \lambda^2}{\mu_1\mu_2}\right) + \frac{\lambda^3}{\mu_1\mu_2\mu_3} \cdot p_0 \cdot \sum_{n=0}^{+\infty} \left(\frac{\lambda}{\mu_3}\right)^n \\
&= p_0 \cdot \left(\frac{\mu_1\mu_2 + \mu_2\lambda + \lambda^2}{\mu_1\mu_2} + \frac{\lambda^3\mu_3}{\mu_1\mu_2\mu_3 \cdot (\mu_3 - \lambda)}\right) \\
&= p_0 \cdot \left[\frac{(\mu_1\mu_2 + \mu_2\lambda + \lambda^2) \cdot (\mu_3 - \lambda) + \lambda^3}{\mu_1\mu_2 \cdot (\mu_3 - \lambda)}\right]
\end{aligned}
$$

This yields:

$$
p_0 = \frac{\mu_1\mu_2 \cdot (\mu_3 - \lambda)}{\mu_1\mu_2 \cdot (\mu_3 - \lambda) + \lambda\mu_2 \cdot (\mu_3 - \lambda) + \lambda^2\mu_3}
$$

or

$$
p_0 = \frac{\mu_1 \cdot \mu_2 \cdot (\mu_3 - \lambda)}{\mu_2 \cdot (\mu_1 + \lambda) \cdot (\mu_3 - \lambda) + \lambda^2 \cdot \mu_3} \tag{1}
$$

Combining the above, the distribution of the number of customers present in the system is given by

$$
p_n = \begin{cases} \lambda^n \cdot \prod_{j=1}^{n} \mu_j^{-1} \cdot p_0, & for\ n = 1, 2, 3 \\ \\ \left(\frac{\lambda}{\mu_3}\right)^{n-3} \cdot \frac{\lambda^3}{\mu_1\mu_2\mu_3} \cdot p_0, & for\ n > 3 \end{cases}
$$

where p_0 was defined in equation (1).

3.2 Probability of waiting

Next, we determine the distribution of the waiting time. To do so, we first derive the probability that an arriving customer has to wait before he enters the chat. We refer to this probability as Π_W. We also introduce two random variables: we denote by L the random variable that describes the number of customers in the center and by L^a the random variable that indicates the number of customers in the center that are seen by an arriving customer. Then Π_W is equal to the probability that a customer upon arrival finds the agent busy, i.e., he finds three or more customers in the center when he arrives. This can also be written as $P(L^a \geq 3)$.
Having assumed Poisson arrivals the PASTA property is valid: the fraction of customers finding on arrival n customers in the system is equal to the fraction of time there are actually n customers in the system. Consequently, from the above we have

$$\Pi_W = P(L^a \geq 3) = P(L \geq 3) = \sum_{n=3}^{+\infty} p_n$$

The distribution of p_n was derived in the previous section. Thus, we have

$$\Pi_W = \sum_{n=3}^{+\infty} \left(\frac{\lambda}{\mu_3}\right)^{n-3} \cdot p_3 = p_3 \cdot \sum_{n=0}^{+\infty} \left(\frac{\lambda}{\mu_3}\right)^{n} = p_3 \cdot \frac{1}{1-\frac{\lambda}{\mu_3}} \quad \Rightarrow \quad \Pi_W = p_3 \cdot \left(\frac{\mu_3}{\mu_3 - \lambda}\right) \tag{2}$$

Remember that we already found both p_3 and p_0. This gives us

$$\Pi_W = \frac{\lambda^3}{\mu_1\mu_2\mu_3} \cdot p_0 \cdot \frac{\mu_3}{\mu_3 - \lambda} = \frac{\lambda^3}{\mu_1\mu_2 \cdot (\mu_3 - \lambda)} \cdot \frac{\mu_1\mu_2 \cdot (\mu_3 - \lambda)}{\mu_2 \cdot (\mu_1 + \lambda) \cdot (\mu_3 - \lambda) + \lambda^2 \cdot \mu_3}$$

Therefore, the probability that an arriving customer has to wait before he enters the chat is equal to

$$\Pi_W = \frac{\lambda^3}{\mu_2 \cdot (\mu_1 + \lambda) \cdot (\mu_3 - \lambda) + \lambda^2 \cdot \mu_3} \tag{3}$$

The probability Π_W is also referred to as the *delay probability.*

3.3 Queue length

We start by introducing the random variable L_q^a which denotes the number of customers waiting in the queue seen by an arriving customer. Again, due to PASTA, the fraction of customers finding on arrival n customers waiting in the queue is equal to the fraction of time that there are actually n customers in the queue:

$$P(L_q^a = n) = P(L_q = n)$$

It is important to find the distribution of the number of customers waiting in the queue before we find the distribution of the waiting time. Notice that the probability there are n customers waiting in the queue is equal to the probability that there are 3 customers being served by the agent, plus the n waiting. This is equal to being $n+3$ customers in the system:

$$P(L_q = n) = p_{3+n} = p_3 \cdot \left(\frac{\lambda}{\mu_3}\right)^n = p_3 \cdot \left(\frac{\mu_3}{\mu_3 - \lambda}\right) \cdot \left(\frac{\mu_3 - \lambda}{\mu_3}\right) \cdot \left(\frac{\lambda}{\mu_3}\right)^n = \left(\frac{\mu_3 - \lambda}{\mu_3}\right) \cdot \Pi_W \cdot \left(\frac{\lambda}{\mu_3}\right)^n$$

which can also be written more elegantly as

$$P(L_q = n) = \left(1 - \frac{\lambda}{\mu_3}\right) \cdot \Pi_W \cdot \left(\frac{\lambda}{\mu_3}\right)^n \tag{4}$$

We can now find the mean number of customers waiting in the queue:

$$E[L_q] = \sum_{n=0}^{+\infty} n \cdot P(L_q = n) = \sum_{n=0}^{+\infty} n \cdot \left(1 - \frac{\lambda}{\mu_3}\right) \cdot \Pi_W \cdot \left(\frac{\lambda}{\mu_3}\right)^n = \left(1 - \frac{\lambda}{\mu_3}\right) \cdot \Pi_W \cdot \frac{\frac{\lambda}{\mu_3}}{\left(1 - \frac{\lambda}{\mu_3}\right)^2}$$

$$\Rightarrow \qquad E[L_q] = \Pi_W \cdot \left(\frac{\lambda}{\mu_3 - \lambda}\right) \tag{5}$$

3.4 Waiting time

Finally, we focus on the waiting time. By applying *Little's law* in the queue, we get

$$E[L_q] = \lambda \cdot E[W]$$

from which we can easily find the mean waiting time:

$$E[W] = \Pi_W \cdot \left(\frac{1}{\mu_3 - \lambda}\right) \tag{6}$$

Next, we determine the distribution of the waiting time. We derive it using the Laplace-Stieltjes transform. Remember that the Laplace-Stieltjes transform of a random variable X that is exponentially distributed with parameters μ is:

$$\tilde{X}(s) = \frac{\mu}{\mu + s}$$

We introduce a new random variable D_k, which will be the k^{th} interdeparture time when the agent is working with 3 chats. Then D_k, $k = 1, 2, ...$, are independent and exponentially distributed random variables with mean $\frac{1}{\mu_3}$.

For the waiting time we have by definition that:

$$\tilde{W}(s) = \sum_{n=0}^{2} E[e^{-sW}|L_q = n] \cdot P(L_q = n) + \sum_{n=3}^{+\infty} E[e^{-sW}|L_q = n] \cdot P(L_q = n)$$

When $n \leq 2$, the waiting time is zero and so we have:

$$E[e^{-sW}|L_q = n] = E[e^{-s \cdot 0}|L_q = n] = 1$$

Expanding the Laplace-Stieltjes transform we have:

$$\tilde{W}(s) = \sum_{n=0}^{2} 1 \cdot p_n + \sum_{n=3}^{+\infty} E[e^{-sW}|L_q = n] \cdot P(L_q = n)$$

The waiting time of an arriving customer given that he finds n customers waiting in the queue is equal to the sum of the n interdeparture times, i.e. $D_1 + D_2 + ... + D_n$. But he also has to wait for the remaining time of the customer that is in the chat, whose interdeparture time is exponentially distributed and has the memoryless property. Thus he has to wait for all the $n+1$ interdeparture times and hence

$$W = D_1 + D_2 + ... + D_{n+1}$$

The Laplace-Stieltjes transform then takes the form

$$\begin{aligned}
\tilde{W}(s) &= (1-\Pi_W) + \sum_{n=0}^{+\infty} E[e^{-s\cdot(D_1+D_2+...+D_{n+1})}] \cdot p_{3+n} \\
&= (1-\Pi_W) + \sum_{n=0}^{+\infty} \left(\frac{\mu_3}{\mu_3+s}\right)^{n+1} \cdot \left(\frac{\lambda}{\mu_3}\right)^{n} \cdot p_3 \\
&= (1-\Pi_W) + p_3 \cdot \left(\frac{\mu_3}{\mu_3+s}\right) \cdot \sum_{n=0}^{+\infty} \left(\frac{\mu_3}{\mu_3+s} \cdot \frac{\lambda}{\mu_3}\right)^{n} \\
&= (1-\Pi_W) + p_3 \cdot \left(\frac{\mu_3}{\mu_3+s}\right) \cdot \left(\frac{\mu_3+s}{\mu_3+s-\lambda}\right) \\
&= (1-\Pi_W) + p_3 \cdot \mu_3 \cdot \frac{1}{\mu_3-\lambda+s} \\
&= (1-\Pi_W) + p_3 \cdot \left(\frac{\mu_3}{\mu_3-\lambda}\right) \cdot \frac{\mu_3-\lambda}{\mu_3-\lambda+s} \\
&= (1-\Pi_W)\cdot 1 + \Pi_W \cdot \frac{\mu_3 \cdot \left(1-\frac{\lambda}{\mu_3}\right)}{\mu_3 \cdot \left(1-\frac{\lambda}{\mu_3}\right)+s}
\end{aligned}$$

Then $\tilde{W}(s)$ can be expressed in terms of ρ:

$$\tilde{W}(s) = (1-\Pi_W)\cdot 1 + \Pi_W \cdot \frac{\mu_3 \cdot (1-\rho)}{\mu_3 \cdot (1-\rho)+s}$$

Taking into account the fact that $\tilde{W}(0) = 1$ and the above, we have the following:

$$W \sim \begin{cases} 0, & \textit{with probability} \quad 1-\Pi_W \\ \\ Exp(\mu_3(1-\rho)), & \textit{with probability} \quad \Pi_W \end{cases}$$

So we have proved that the distribution of the waiting time, given that you have to wait, is exponential with parameter $\mu_3 \cdot (1-\rho)$. Hence:

$$P(W > t) = \Pi_W \cdot e^{-\mu_3\cdot(1-\rho)\cdot t}, \qquad t \geq 0 \tag{7}$$

4 Chat sessions with 2 agents

Now we consider the case in which there are two agents serving the arriving customers. In order to find the equilibrium distribution of the number of customers in the center, we first derive the global balance equations. The routing policy is relevant for the performance of the system, since it defines the way we build the state transition diagram and hence the form of the global balance equations. We start with the ”Maximum Load” and then continue with ”Balanced Load” routing policy.

4.1 Number of customers in the system

Since $s \leq I$ ($s = 2$ and $I = 3$) we choose the first state description. Focusing only in the limiting behaviour, the state description for both routing policies will be given by the following random variable:

$$X = \{L_q, a_1, a_2\}$$

The probability $p_{(L_q,a_1,a_2)}$ can be interpreted as the fraction of time that there are L_q customers waiting to be served and that agent 1 and agent 2 are chatting with a_1 and a_2 customers, respectively.

4.1.1 Maximum load routing policy

Due to the assumption made, an arriving customer finding both agents at the same level i (with $i < 3$) is assigned to agent 1 and consequently, for both routing policies, the following transitions are prohibited:

$$\begin{array}{ccc} (0,0,0) & \longrightarrow & (0,0,1) \\ (0,1,1) & \longrightarrow & (0,1,2) \\ (0,2,2) & \longrightarrow & (0,2,3) \end{array}$$

We first consider the case in which the customers are assigned to the agent that has currently the maximum number of chats. Therefore, as we can also see from the state transition diagram in figure 3, not all transitions are possible. Below are the prohibited transitions due to the maximum load policy:

$$\begin{array}{ccc} (0,0,1) & \longrightarrow & (0,1,1) \\ (0,1,0) & \longrightarrow & (0,1,1) \\ (0,0,2) & \longrightarrow & (0,1,2) \\ (0,1,2) & \longrightarrow & (0,2,2) \\ (0,2,1) & \longrightarrow & (0,2,2) \\ (0,2,0) & \longrightarrow & (0,2,1) \end{array}$$

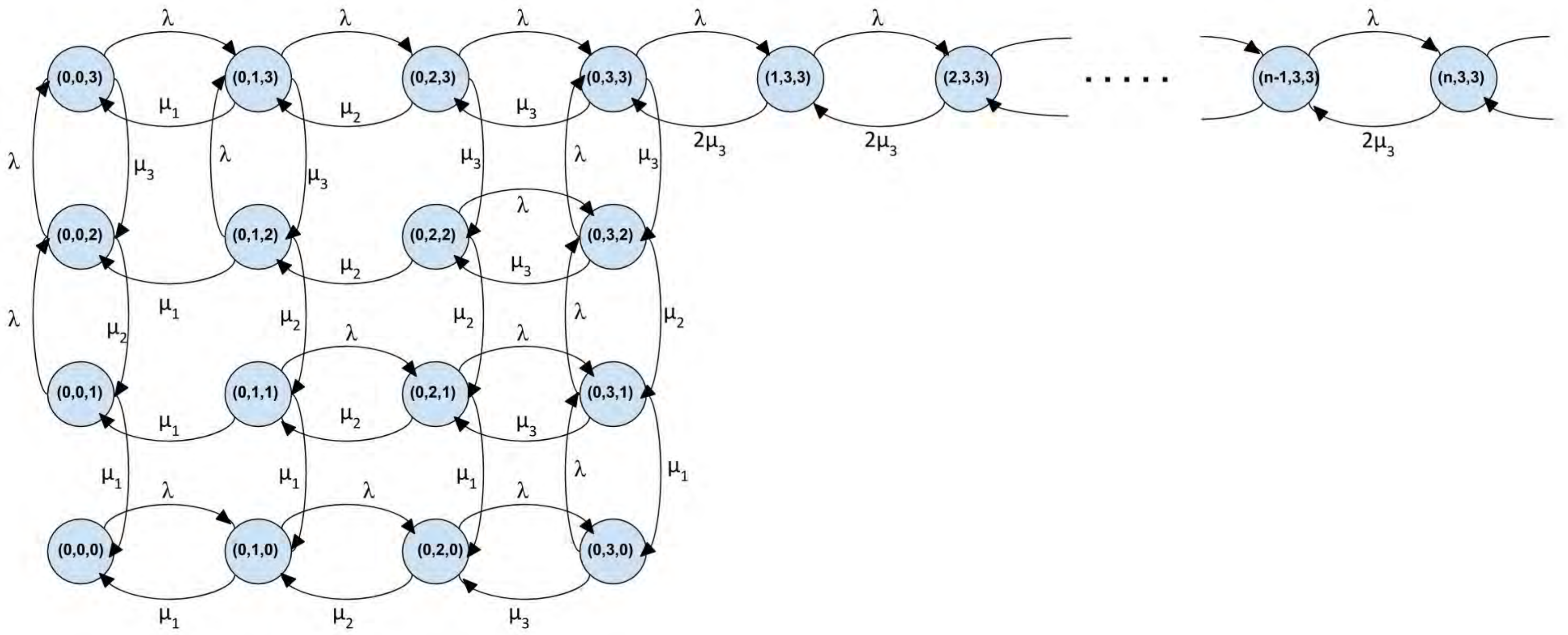


Figure 3: State transition diagram for two agents when applying the "Maximum load" routing policy

We first find the balance equations in the boundary part, i.e. the probabilities when there is no customer waiting in the queue ($L_q = 0$). By applying the *Global Balance principle* to a single state, we equate the flow out with the flow into every state and get the following system of equations:

$$
\begin{aligned}
\lambda \cdot p_{(0,0,0)} &= \mu_1 \cdot p_{(0,1,0)} + \mu_1 \cdot p_{(0,0,1)} \\
(\lambda + \mu_1) \cdot p_{(0,1,0)} &= \lambda \cdot p_{(0,0,0)} + \mu_1 \cdot p_{(0,1,1)} + \mu_2 \cdot p_{(0,2,0)} \\
(\lambda + \mu_2) \cdot p_{(0,2,0)} &= \lambda \cdot p_{(0,1,0)} + \mu_1 \cdot p_{(0,2,1)} + \mu_3 \cdot p_{(0,3,0)} \\
(\lambda + \mu_3) \cdot p_{(0,3,0)} &= \lambda \cdot p_{(0,2,0)} + \mu_1 \cdot p_{(0,3,1)} \\
(\lambda + \mu_1) \cdot p_{(0,0,1)} &= \mu_1 \cdot p_{(0,1,1)} + \mu_2 \cdot p_{(0,0,2)} \\
(\lambda + 2\mu_1) \cdot p_{(0,1,1)} &= \mu_2 \cdot p_{(0,2,1)} + \mu_2 \cdot p_{(0,1,2)} \\
(\lambda + \mu_1 + \mu_2) \cdot p_{(0,2,1)} &= \lambda \cdot p_{(0,1,1)} + \mu_2 \cdot p_{(0,2,2)} + \mu_3 \cdot p_{(0,3,1)} \\
(\lambda + \mu_1 + \mu_3) \cdot p_{(0,3,1)} &= \lambda \cdot p_{(0,3,0)} + \lambda \cdot p_{(0,2,1)} + \mu_2 \cdot p_{(0,3,2)} \\
(\lambda + \mu_2) \cdot p_{(0,0,2)} &= \lambda \cdot p_{(0,0,1)} + \mu_1 \cdot p_{(0,1,2)} + \mu_3 \cdot p_{(0,0,3)} \\
(\lambda + \mu_1 + \mu_2) \cdot p_{(0,1,2)} &= \mu_2 \cdot p_{(0,2,2)} + \mu_3 \cdot p_{(0,1,3)} \\
(\lambda + 2\mu_2) \cdot p_{(0,2,2)} &= \mu_3 \cdot p_{(0,3,2)} + \mu_3 \cdot p_{(0,2,3)} \\
(\lambda + \mu_2 + \mu_3) \cdot p_{(0,3,2)} &= \lambda \cdot p_{(0,3,1)} + \lambda \cdot p_{(0,2,2)} + \mu_3 \cdot p_{(0,3,3)} \\
(\lambda + \mu_3) \cdot p_{(0,0,3)} &= \lambda \cdot p_{(0,0,2)} + \mu_1 \cdot p_{(0,1,3)} \\
(\lambda + \mu_1 + \mu_3) \cdot p_{(0,1,3)} &= \lambda \cdot p_{(0,1,2)} + \lambda \cdot p_{(0,0,3)} + \mu_2 \cdot p_{(0,2,3)} \\
(\lambda + \mu_2 + \mu_3) \cdot p_{(0,2,3)} &= \lambda \cdot p_{(0,1,3)} + \mu_3 \cdot p_{(0,3,3)} \\
(\lambda + 2\mu_3) \cdot p_{(0,3,3)} &= \lambda \cdot p_{(0,3,2)} + \lambda \cdot p_{(0,2,3)} + 2\mu_3 \cdot p_{(1,3,3)}
\end{aligned}
$$

For the states in which there are customers waiting in the queue ($L_q > 0$) we equate the flow in with the flow out of a set of states of the form $\{\{0, 1, 2, ..., n-1\}, 3, 3\}$ and we get the following balance equations:

$$\lambda \cdot p_{(n-1,3,3)} = 2\mu_3 \cdot p_{(n,3,3)}, \qquad n \geq 1$$

Note that in our case with 2 agents and $I = 3$ chats we get 16 boundary equations. In general, for a CSC with chat sessions with s agents and maximum chat load equal to I we get $(I+1)^s$ boundary equations.

4.1.2 Balanced load routing policy

Similar to section 4.1.1 the process cannot make all boundary transitions. The workload that arrives must be equally divided among the two agents. Hence, after an arrival the customer that entered is always routed to the agent with the least chats. In addition to the three prohibited transitions in case of equal number of customers, the following transitions are impossible:

$$
\begin{aligned}
(0{,}0{,}1) &\longrightarrow (0{,}0{,}2)\\
(0{,}1{,}0) &\longrightarrow (0{,}2{,}0)\\
(0{,}0{,}2) &\longrightarrow (0{,}0{,}3)\\
(0{,}1{,}2) &\longrightarrow (0{,}1{,}3)\\
(0{,}2{,}0) &\longrightarrow (0{,}3{,}0)\\
(0{,}2{,}1) &\longrightarrow (0{,}3{,}1)
\end{aligned}
$$

Below are the *global balance equations* which can be found from the state transition diagram of figure 4 by equating the flow out with the flow into every single state:

$$
\begin{gathered}
\lambda \cdot p_{(0,0,0)} = \mu_1 \cdot p_{(0,0,1)} + \mu_1 \cdot p_{(0,1,0)}\\
(\lambda + \mu_1) \cdot p_{(0,1,0)} = \lambda \cdot p_{(0,0,0)} + \mu_1 \cdot p_{(0,1,1)} + \mu_2 \cdot p_{(0,2,0)}\\
(\lambda + \mu_2) \cdot p_{(0,2,0)} = \mu_1 \cdot p_{(0,2,1)} + \mu_3 \cdot p_{(0,3,0)}\\
(\lambda + \mu_3) \cdot p_{(0,3,0)} = \mu_1 \cdot p_{(0,3,1)}\\
(\lambda + \mu_1) \cdot p_{(0,0,1)} = \mu_1 \cdot p_{(0,1,1)} + \mu_2 \cdot p_{(0,0,2)}\\
(\lambda + 2\mu_1) \cdot p_{(0,1,1)} = \lambda \cdot p_{(0,1,0)} + \lambda \cdot p_{(0,0,1)} + \mu_2 \cdot p_{(0,1,2)} + \mu_2 \cdot p_{(0,2,1)}\\
(\lambda + \mu_1 + \mu_2) \cdot p_{(0,2,1)} = \lambda \cdot p_{(0,1,1)} + \lambda \cdot p_{(0,2,0)} + \mu_2 \cdot p_{(0,2,2)} + \mu_3 \cdot p_{(0,3,1)}\\
(\lambda + \mu_1 + \mu_3) \cdot p_{(0,3,1)} = \lambda \cdot p_{(0,3,0)} + \mu_2 \cdot p_{(0,3,2)}\\
(\lambda + \mu_2) \cdot p_{(0,0,2)} = \mu_1 \cdot p_{(0,1,2)} + \mu_3 \cdot p_{(0,0,3)}\\
(\lambda + \mu_1 + \mu_2) \cdot p_{(0,1,2)} = \lambda \cdot p_{(0,0,2)} + \mu_2 \cdot p_{(0,2,2)} + \mu_3 \cdot p_{(0,1,3)}\\
(\lambda + 2\mu_2) \cdot p_{(0,2,2)} = \lambda \cdot p_{(0,2,1)} + \lambda \cdot p_{(0,1,2)} + \mu_3 \cdot p_{(0,3,2)} + \mu_3 \cdot p_{(0,2,3)}\\
(\lambda + \mu_2 + \mu_3) \cdot p_{(0,3,2)} = \lambda \cdot p_{(0,2,2)} + \lambda \cdot p_{(0,3,1)} + \mu_3 \cdot p_{(0,3,3)}\\
(\lambda + \mu_3) \cdot p_{(0,0,3)} = \mu_1 \cdot p_{(0,1,3)}\\
(\lambda + \mu_1 + \mu_3) \cdot p_{(0,1,3)} = \lambda \cdot p_{(0,0,3)} + \mu_2 \cdot p_{(0,2,3)}\\
(\lambda + \mu_2 + \mu_3) \cdot p_{(0,2,3)} = \lambda \cdot p_{(0,1,3)} + \mu_3 \cdot p_{(0,3,3)}\\
(\lambda + 2\mu_3) \cdot p_{(0,3,3)} = \lambda \cdot p_{(0,2,3)} + \lambda \cdot p_{(0,3,2)} + 2\mu_3 \cdot p_{(1,3,3)}
\end{gathered}
$$

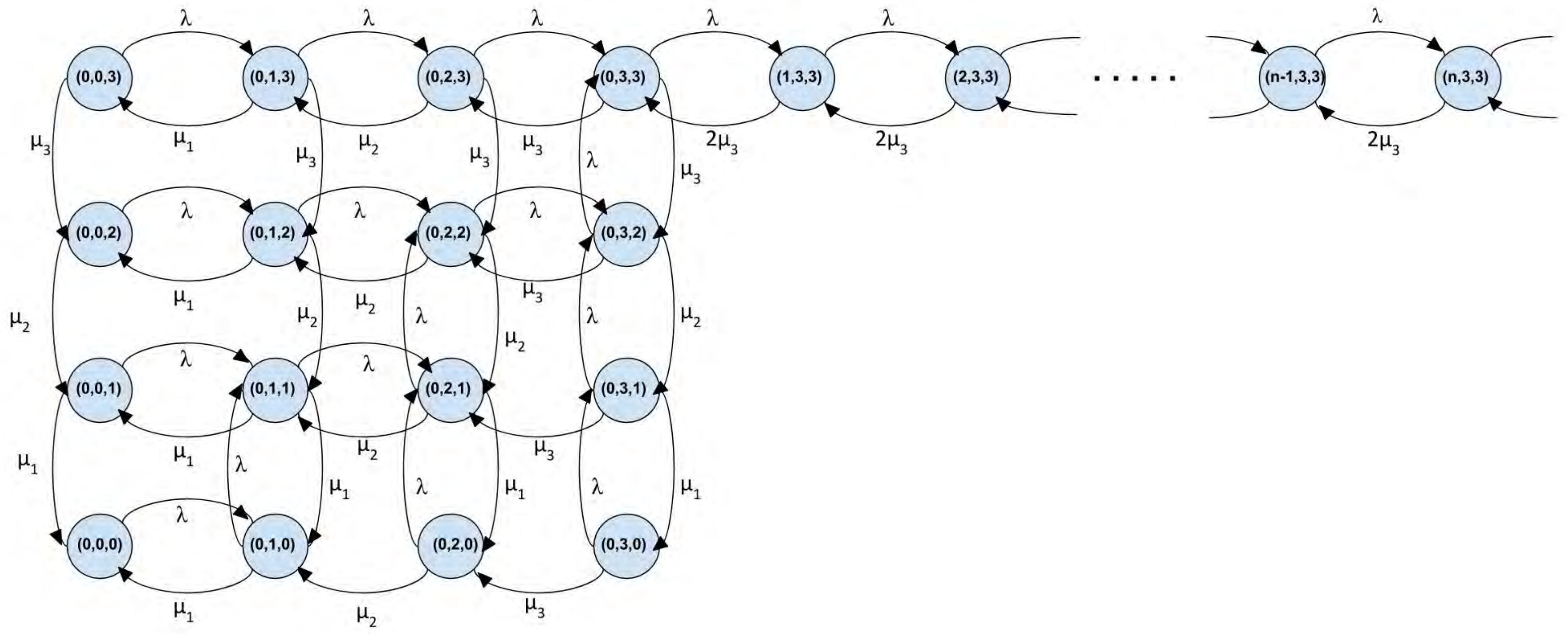


Figure 4: State transition diagram for two agents when applying the "Balanced load" routing policy

For the states when there are customers waiting in the queue, the system behaves like an M/M/2 system with service rate equal to μ_3. Thus, we have

$$\lambda \cdot p_{(n-1,3,3)} = 2\mu_3 \cdot p_{(n,3,3)}, \qquad n \geq 1$$

Notice that the above equation is the same for both different routing policies. Also, from the diagrams we can see that the behaviour of the system for these states is the same. The above equation can also be written as

$$p_{(n,3,3)} = \frac{\lambda}{2\mu_3} \cdot p_{(n-1,3,3)} = \left(\frac{\lambda}{2\mu_3}\right)^2 \cdot p_{(n-2,3,3)} = \dots \quad \Rightarrow \quad p_{(n,3,3)} = \left(\frac{\lambda}{2\mu_3}\right)^n \cdot p_{(0,3,3)} \tag{8}$$

From the normalization property we get the following:

$$\sum_{i=0}^{3}\sum_{j=0}^{3} p_{(0,i,j)} + \sum_{n=1}^{+\infty} p_{(n,3,3)} = 1 \tag{9}$$

The second term of equation (9) can be computed from formula (8):

$$\sum_{n=1}^{+\infty} p_{(n,3,3)} = \sum_{n=1}^{+\infty} \left(\frac{\lambda}{2\mu_3}\right)^n \cdot p_{(0,3,3)} = \frac{\lambda}{2\mu_3} \cdot p_{(0,3,3)} \cdot \sum_{n=0}^{+\infty} \left(\frac{\lambda}{2\mu_3}\right)^n = \frac{\lambda}{2\mu_3 - \lambda} \cdot p_{(0,3,3)}$$

The normalization property in (9) can then be rewritten as

$$p_{(0,0,0)} + p_{(0,0,1)} + p_{(0,1,0)} + p_{(0,1,1)} + \dots + p_{(0,2,3)} + p_{(0,3,2)} + p_{(0,3,3)} + \frac{\lambda}{2\mu_3 - \lambda} \cdot p_{(0,3,3)} = 1$$

or

$$p_{(0,0,0)} + p_{(0,0,1)} + p_{(0,1,0)} + p_{(0,1,1)} + \dots + p_{(0,2,3)} + p_{(0,3,2)} + \left(\frac{2\mu_3}{2\mu_3 - \lambda}\right) \cdot p_{(0,3,3)} = 1 \tag{10}$$

To find all these probabilities, we solve the system of the 16 equations we created from the global balance equations presented on pages 17 and 18. In both cases though, we ignore the last equation and we use the normalization property instead, as it is in equation (10). A more analytical description on how to solve these systems with MatLab is given in the appendix.

4.2 Probability of waiting

As we explained before, the probability that an arriving customer has to wait upon arrival is equal to the probability that an arriving customer finds both agents fully occupied. According to PASTA this is equal to the fraction of time that both agents are fully occupied which in this case is when there are 6 or more customers in the center. The probability that a customer has to wait in a CSC with 2 agents is:

$$P_W = P(L^a \geq 6) = P(L \geq 6) \quad \Rightarrow \quad P_W = \sum_{n=0}^{+\infty} p_{(n,3,3)} \tag{11}$$

From equations (8) and (11), we derive the following:

$$P_W = \sum_{n=0}^{+\infty} \left(\frac{\lambda}{2\mu_3}\right)^n \cdot p_{(0,3,3)} = \left(\frac{1}{1-\frac{\lambda}{2\mu_3}}\right) \cdot p_{(0,3,3)} \quad \Rightarrow \quad P_W = \frac{2\mu_3}{2\mu_3 - \lambda} \cdot p_{(0,3,3)} \tag{12}$$

This is the *delay probability* for the case of $s = 2$, which is the counterpart of Π_W of the one agent case.

4.3 Queue length

The distribution of the number of customers waiting in the queue is:

$$P(L_q = n) = p_{(n,3,3)} = \left(\frac{\lambda}{2\mu_3}\right)^n \cdot p_{(0,3,3)} = \left(\frac{\lambda}{2\mu_3}\right)^n \cdot \left(\frac{2\mu_3 - \lambda}{2\mu_3}\right) \cdot \left(\frac{2\mu_3}{2\mu_3 - \lambda}\right) \cdot p_{(0,3,3)}$$

From formula (12) and by using the fact that $\rho = \frac{\lambda}{2\mu_3}$, we can write this as:

$$P(L_q = n) = (1 - \rho) \cdot P_W \cdot \rho^n \tag{13}$$

With the proof being similar to the case where $s = 1$, the mean number of customers waiting in the queue is:

$$E[L_q] = \sum_{n=0}^{+\infty} n \cdot P(L_q = n) = \sum_{n=0}^{+\infty} n \cdot (1 - \rho) \cdot P_W \cdot \rho^n$$

$$\Rightarrow \quad E[L_q] = P_W \cdot \left(\frac{\rho}{1-\rho}\right) \tag{14}$$

Hence, the main difference with the case where $s = 1$ lies on the delay probabilities Π_W and P_W.

4.4 Waiting time

From Little's law the mean waiting time is

$$E[W] = \frac{1}{\lambda} \cdot E[L_q] = \frac{1}{\lambda} \cdot P_W \cdot \left(\frac{\lambda}{2\mu_3 - \lambda} \right) \Rightarrow$$

$$\Rightarrow \qquad E[W] = P_W \cdot \frac{1}{2\mu_3(1-\rho)} \tag{15}$$

The proof of the distribution of the waiting time given that an arriving customer has to wait is similar to the case of one agent. As a result we have

$$P(W > t) = P_W \cdot e^{-2\mu_3 \cdot (1-\rho) \cdot t}, \qquad t \geq 0 \tag{16}$$

where $\rho = \frac{\lambda}{2\mu_3} < 1$.

4.5 Finding an idle agent

Finally we compute the probability that an arriving customer finds at least one idle agent (i.e. all his chat slots are empty):

$$P(idle) = P(agent_1 = idle) + P(agent_2 = idle) + P(both.agents = idle)$$

The probability of finding both agents idle is of course equal to $p_{(0,0,0)}$. For our case, where $I = 3$, the above is equal to:

$$P(idle) = \sum_{i=1}^{3} p_{(0,0,i)} + \sum_{j=1}^{3} p_{(0,j,0)} + p_{(0,0,0)} \tag{17}$$

5 Computing the service rates

In this section we describe the steps we followed to compute the service rates from the data. The data is taken from a Brazilian customer service center with chat sessions. Our purpose is to compute the service rate when there are one, two or three customers in the chat. In mathematical terms, we need to compute μ_1, μ_2 and μ_3.

5.1 Data description and changes

From the data we know the exact time and date that customers entered the service center, their waiting time and their sojourn time (total time they spend in the system waiting and being served). By adding the arriving time to the waiting time we find the time that each customer entered the chat and the adding of the arriving time and the sojourn time yields the time that a customer has completed his service and is ready for departure.

We separate the data per agent and conduct the analysis for each of them individually. Knowing the arriving and departure time of every customer, we create a new attribute in the data which stores whether the agent has 1, 2 or 3 chats at each time. An arrival or departure occurring when the agent is chatting with other customers increases or decreases respectively the number of chats of the agent by one. By adding all these time intervals for each of the three cases, we create another category for the total time that an agent spent in chatting with 1, 2 or 3 customers.

Finally, for every agent, we create another attribute counting the total number of service completions for each of the 3 cases in which the agent was chatting with 1,2 or 3 customers. It is important to know before computing the service parameters.

5.2 Parameter results

The service rate for level i by definition is:

$$\mu_i = \frac{number\ of\ service\ completions\ when\ there\ are\ i\ customers\ in\ the\ chat}{total\ time\ spent\ in\ chatting\ with\ i\ customers}$$

for $i = 1, 2$ and 3.

In table 1 we present the parameters we found for the 15 most "busy" agents. We also compute a weight for each agent according to the number of customers he served. From the table we can see that for almost all agents, $\mu_1 < \mu_2 < \mu_3$. This is reasonable since it takes more time for an agent to complete the services when more customers are present in the chat. On the other hand we see that $\mu_2 < 2\mu_1$ (and $\mu_3 < 3\mu_1$) which means that an agent is servicing 2 (or 3) customers faster when chatting with both(or the three) of them simultaneously than if he was servicing each one of them separately.

Agent	μ_1	μ_2	μ_3	**weight**
Adriana Pereira	0.0987	0.1346	0.1677	0.043
Ana Lucia	0.0719	0.1282	0.1377	0.038
Vanessa Silva	0.1180	0.1835	0.1659	0.059
Ednaldo Faria	0.0750	0.1249	0.1703	0.076
Camila Gabriele	0.0987	0.1716	0.1833	0.071
Sabrina Patricio	0.1174	0.1196	0.1706	0.058
Ellen Martins	0.0889	0.1692	0.2149	0.063
Aline Colella	0.1159	0.1436	0.1963	0.079
Fernanda Gomes	0.1228	0.1644	0.2146	0.071
Camila Murano	0.0790	0.0998	0.1524	0.056
Jackeline Alencar	0.0787	0.1075	0.147	0.074
Flavia Freitas	0.189	0.1499	0.2203	0.079
Joao Pedro	0.0916	0.1369	0.1857	0.085
Juliana dos Santos	0.1031	0.1384	0.1661	0.076
Debora Andrade	0.0957	0.1532	0.1459	0.063

Table 1: The service rates found for several different agents

By finding the weighted average using the service rates from each of these agents, we compute the parameters we will use in our numerical computations in the next section:

$$\mu_1 = 0.1047$$
$$\mu_2 = 0.1422$$
$$\mu_3 = 0.1783$$

As time unit we considered one minute. Thus μ_i is the number of customers served per minute when there are i customers in the chat.

6 Numerical analysis

As we already mentioned in the introduction, we will compare the performance of the CSC with chat sessions under the two different routing policies. Using the parameters μ_1, μ_2 and μ_3 that we found in the previous section, we compute numerically with MatLab the distribution of the number of customers in the system, the probability of waiting, the mean waiting time and the probability that an arriving customers finds at least one idle agent. We derive results only for the case of two agents.

As arrival rate we arbitrarily choose a value with the condition that the occupation rate is lower than 1, so that the queue does not grow to infinity. Consequently, the restriction for λ is

$$\rho < 1 \quad \Leftrightarrow \quad \frac{\lambda}{2\mu_3} < 1 \quad \Leftrightarrow \quad \lambda < 2\mu_3 \quad \Leftrightarrow \quad \lambda < 0.3567$$

6.1 Mean waiting time

For different values of λ we computed the mean waiting time of a customer for the two different routing policies. The differences are very small, especially for big values of λ and therefore we present in figure 5 the values of the mean waiting time only for $\lambda < 0.2$ so that the differences in the graph are observable easier. As time unit we considered to be one minute:

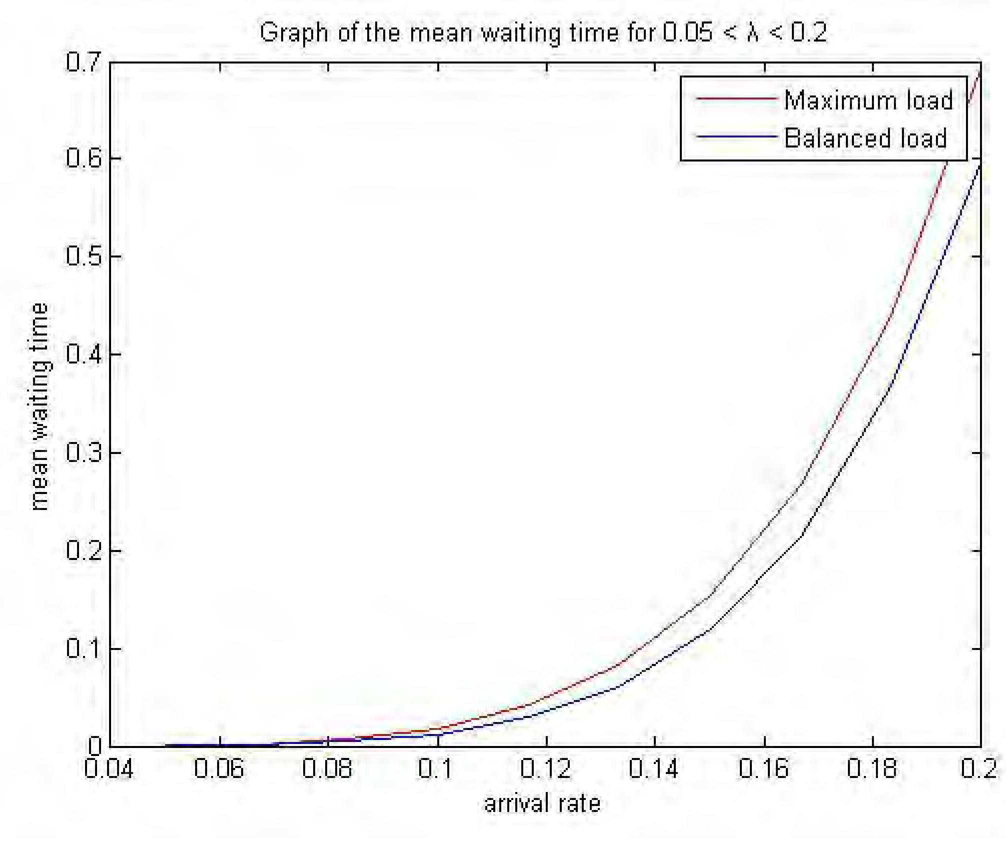


Figure 5:

From the graph we can see that in the Maximum load routing policy customers have to wait longer before they enter the chat compared to the case of the Balanced load policy. This can be explained as follows: in the Maximum load policy, customers are routed to the agent with the most chats, keeping the second agent free of work. But, as we see from the results of the parameters, serving two customers at the same time with rate μ_2 takes more time on average than two different agents serving one customer each, due to the fact that $\mu_2 < 2\mu_1$. Similarly, when there are three arriving customers it is faster to distribute them among both agents, in which one agent will serve two customers and the other agent one customer, than route all the three of them to only one agent, since $\mu_3 < \mu_1 + \mu_2$.

6.2 Probability of waiting

The probability that an arriving customer has to wait upon arrival, for several different service rate values is given in the graph of figure 6:

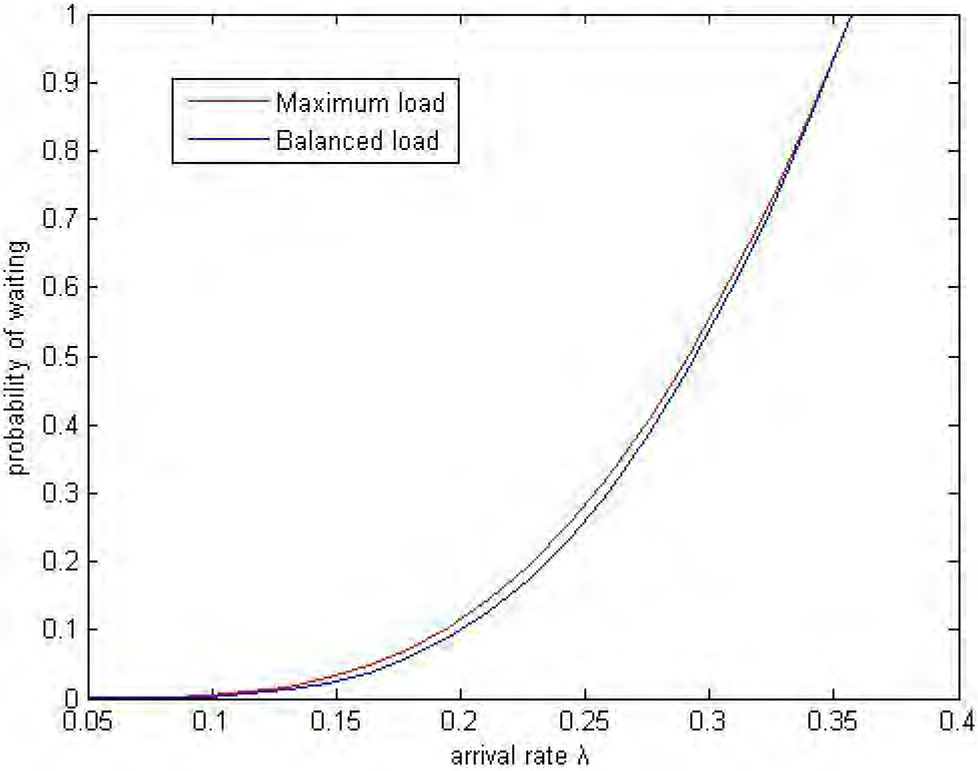


Figure 6: Graph of the probability of waiting time for several values of λ

As was expected, in a CSC which routes customers equally to the agents, arriving customers have lower probability to wait before they enter the chat compared to a CSC that always routes the customers to the agent with the maximum number of chats. For $\lambda < 0.15$ and for $\lambda > 0.3$

the probability of waiting is very close for the two cases. The biggest differences in the delay probabilities are for $0.15 < \lambda < 0.3$, i.e. when the workload is between 40% to 80% but also these can still be considered relatively small.

We now present the table for the probability that a customers has to wait less than t minutes, for some values of t. In the table, we assumed that the occupation rate is 80% such that $\lambda = 0.2853$.

	Maximum load	**Balanced load**
$P(W < 20\ sec)$	0.5499	0.5693
$P(W < 1\ min)$	0.5709	0.5893
$P(W < 2\ min)$	0.6005	0.6176

In figure 7 below we present the graph for the case of $t = 1$, for all possible λ's. As we can see the differences are quite small:

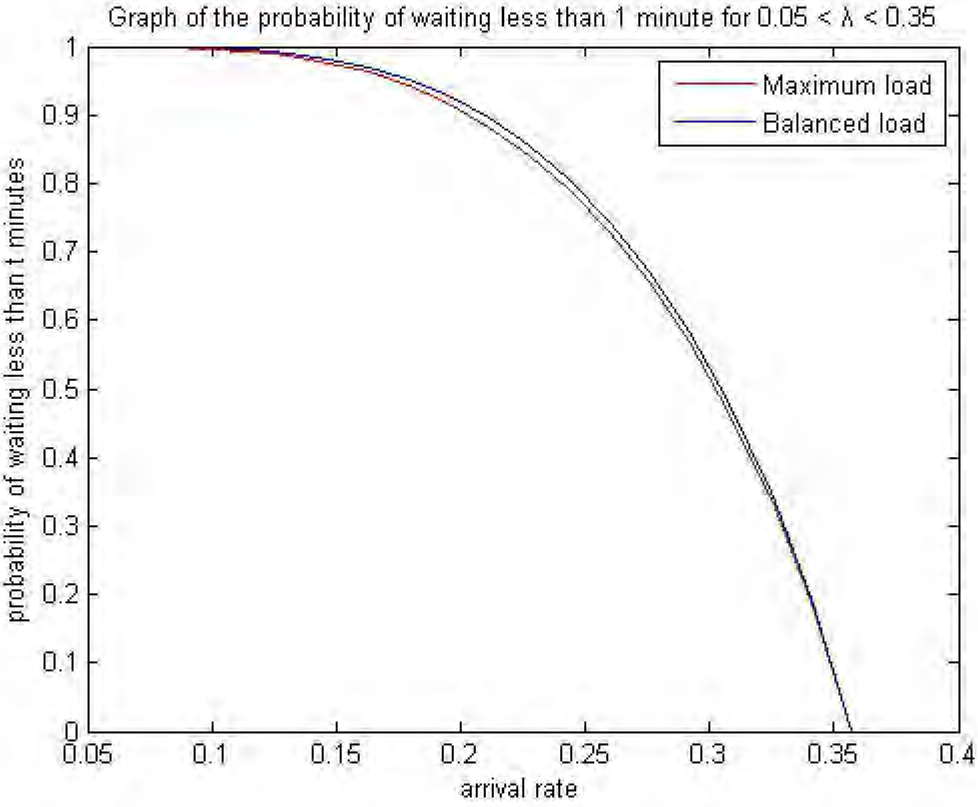


Figure 7: Probability of waiting less than 1 minute for several values of λ

6.3 Idle agent

Finally, for the arrival rate $\lambda = 0.2853$, we find the probability of an idle agent. In the case of a CSC with the Maximum load routing policy, 21.7% of the time an arriving customer will find at least one agent idle while in the case of the Balanced load routing policy this will happen 13.4% of the time.

	Maximum load	**Balanced load**
$P(at\ least\ 1\ agent\ idle)$	0.2170	0.1341

Observe that, while for all the performance measures the Balanced load outmatches the Maximum load policy, this does not happen when it comes to the probability of finding at least one idle agent. As we can see from figure 8, the difference is considerable and regardless of the arrival rate, an arriving customers is more likely to find at least one idle agent in the Maximum load policy. This was expected - the purpose of this policy is to keep many agents as possible free for other tasks.

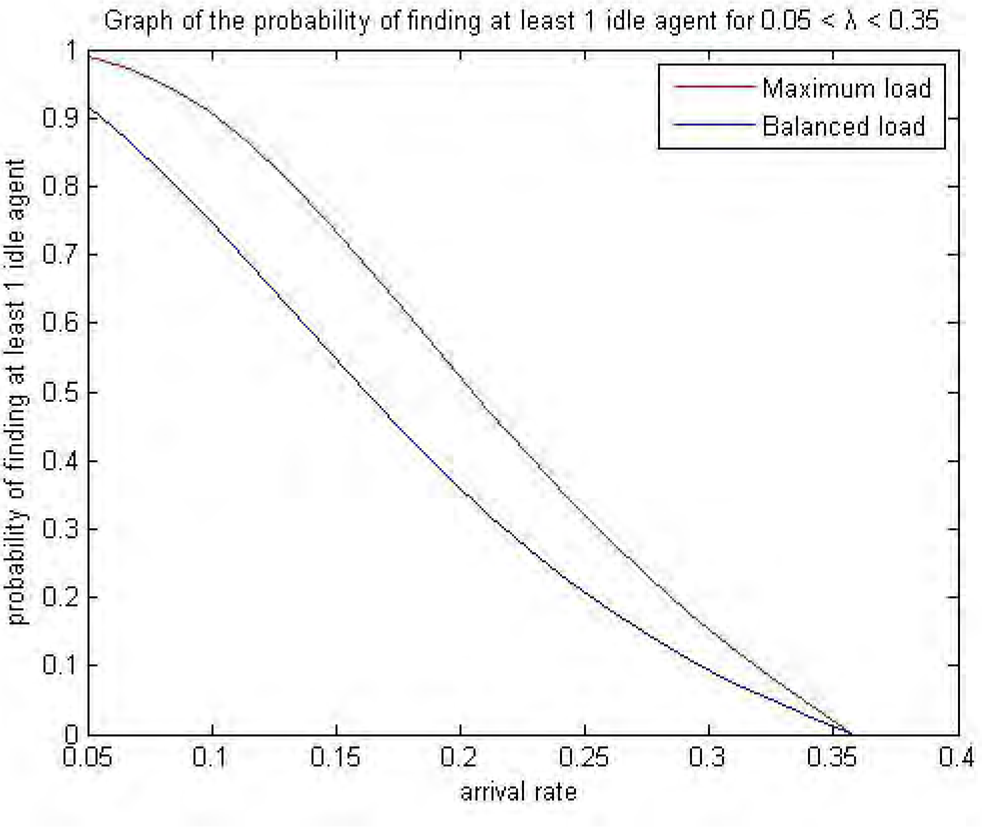


Figure 8: Graph of the probability of finding at least one idle agent for several values of λ

6.4 Similar performances

Finally we would like to present here the similarity between a CSC with chat sessions and the M/M/c queueing system with c servers. The performance of a CSC with chat sessions and service rates $\mu_2 = 2\mu_1$ and $\mu_3 = 3\mu_1$ should perform on the same way with an M/M/6 system in which the service rate is $\mu = \mu_1 = \frac{\mu_3}{3}$. We will compare the performance of these two systems for the following parameters:

$$\lambda = 0.3 \qquad and \qquad \mu = \frac{0.17836}{3} = 0.05945$$

From the results below, we can verify that this is true:

	Maximum L.	**Balanced L.**	**M/M/6**
E[W]	10.6519	10.6519	10.65

For a further explanation look for the "Verification of the system" in the appendix.

7 Conclusion

Customer service centers with chat sessions have proved that they deserve a high preference in lots of companies and therefore this justifies the importance to study the performance of these systems. In this paper, we attempted to model and analyse a system like this for the cases of 1 and 2 agents with the assumption that each of them can serve a maximum of 3 customers simultaneously. For the case of two agents, two different routing policies were considered: the Maximum load routing policy in which an arriving customer is assigned to the agent with the maximum number of chats and the Balanced load routing policy in which the arriving customer is signed to the agent with the least number of chats.

Several performance measures were found such as the mean waiting time and the probability of waiting and have been compared between the two policies. For these measures, the Balanced load policy proved to be more preferable than the Maximum load routing policy. On the other hand, an arriving customer is more likely to find an idle agent when customers are routed according to the Balanced load policy.

Although the present research was as thorough as possible, there are still many interesting questions to be answered. How does the chat system behave when the number of agents increases? Is there a big improvement in the performance measures? What is the optimal maximum chat load for a specific number of agents or what is the optimal number of agents for a specific maximum chat load such that some performance target is met? How can we model a system in which service times are generally distributed? What changes in the modelling or performance of the system if abandons of customers are possible or agents take breaks for a random amount of time?

A Solving the system of equations in MatLab

We give a brief description on how we solved the global balance equations in MatLab. For both routing policies we solve a system of 16 boundary equations with 16 unknowns. The unknowns are the probabilities $p_{(0,i,j)}$ for $i, j = 0, 1, 2, 3$. The probabilities $p_{(n,3,3)}$ with $n > 0$ can be expressed in terms of $p_{(0,3,3)}$ as we have shown in formula (8). We set the variables of the systems as follows:

$$\begin{array}{llll}
x_1 = p_{(0,0,0)} & x_5 = p_{(0,0,1)} & x_9 = p_{(0,0,2)} & x_{13} = p_{(0,0,3)} \\
x_2 = p_{(0,1,0)} & x_6 = p_{(0,1,1)} & x_{10} = p_{(0,1,2)} & x_{14} = p_{(0,1,3)} \\
x_3 = p_{(0,2,0)} & x_7 = p_{(0,2,1)} & x_{11} = p_{(0,2,2)} & x_{15} = p_{(0,2,3)} \\
x_4 = p_{(0,3,0)} & x_8 = p_{(0,3,1)} & x_{12} = p_{(0,3,2)} & x_{16} = p_{(0,3,3)}
\end{array}$$

By considering the above, the normalization property as it is in formula (10) presented in section 4.2 takes the following form:

$$x_1 + x_2 + x_3 + ... + x_{13} + x_{14} + x_{15} + \left(\frac{2\mu_3}{2\mu_3 - \lambda}\right) \cdot x_{16} = 1 \tag{18}$$

which will be the last equation of the system as we mentioned before.

For the *maximum load* routing policy, the system of the global balance equations is written as:

$$\begin{gathered}
\lambda \cdot x_1 - \mu_1 \cdot x_2 - \mu_1 \cdot x_5 = 0 \\
(\lambda + \mu_1) \cdot x_2 - \lambda \cdot x_1 - \mu_1 \cdot x_6 - \mu_2 \cdot x_3 = 0 \\
(\lambda + \mu_2) \cdot x_3 - \lambda \cdot x_2 - \mu_1 \cdot x_7 - \mu_3 \cdot x_4 = 0 \\
(\lambda + \mu_3) \cdot x_4 - \lambda \cdot x_3 - \mu_1 \cdot x_8 = 0 \\
(\lambda + \mu_1) \cdot x_5 - \mu_1 \cdot x_6 - \mu_2 \cdot x_9 = 0 \\
(\lambda + 2\mu_1) \cdot x_6 - \mu_2 \cdot x_7 - \mu_2 \cdot x_{10} = 0 \\
(\lambda + \mu_1 + \mu_2) \cdot x_7 - \lambda \cdot x_6 - \mu_2 \cdot x_{11} - \mu_3 \cdot x_8 = 0 \\
(\lambda + \mu_1 + \mu_3) \cdot x_8 - \lambda \cdot x_4 - \lambda \cdot x_7 - \mu_2 \cdot x_{12} = 0 \\
(\lambda + \mu_2) \cdot x_9 - \lambda \cdot x_5 - \mu_1 \cdot x_{10} - \mu_3 \cdot x_{13} = 0 \\
(\lambda + \mu_1 + \mu_2) \cdot x_{10} - \mu_2 \cdot x_{11} - \mu_3 \cdot x_{14} = 0 \\
(\lambda + 2\mu_2) \cdot x_{11} - \mu_3 \cdot x_{12} - \mu_3 \cdot x_{15} = 0 \\
(\lambda + \mu_2 + \mu_3) \cdot x_{12} - \lambda \cdot x_8 - \lambda \cdot x_{11} - \mu_3 \cdot x_{16} = 0 \\
(\lambda + \mu_3) \cdot x_{13} - \lambda \cdot x_9 - \mu_1 \cdot x_{14} = 0 \\
(\lambda + \mu_1 + \mu_3) \cdot x_{14} - \lambda \cdot x_{10} - \lambda \cdot x_{13} - \mu_2 \cdot x_{15} = 0 \\
(\lambda + \mu_2 + \mu_3) \cdot x_{15} - \lambda \cdot x_{14} - \mu_3 \cdot x_{16} = 0 \\
x_1 + x_2 + x_3 + ... + x_{15} + \left(\frac{2\mu_3}{2\mu_3 - \lambda}\right) \cdot x_{16} = 1
\end{gathered}$$

The system of equations for the *balanced load* routing policy is:

$$\lambda \cdot x_1 - \mu_1 \cdot x_5 - \mu_1 \cdot x_2 = 0$$
$$(\lambda + \mu_1) \cdot x_2 - \lambda \cdot x_1 - \mu_1 \cdot x_6 - \mu_2 \cdot x_3 = 0$$
$$(\lambda + \mu_2) \cdot x_3 - \mu_1 \cdot x_7 - \mu_3 \cdot x_4 = 0$$
$$(\lambda + \mu_3) \cdot x_4 - \mu_1 \cdot x_8 = 0$$
$$(\lambda + \mu_1) \cdot x_5 - \mu_1 \cdot x_6 - \mu_2 \cdot x_9 = 0$$
$$(\lambda + 2\mu_1) \cdot x_6 - \lambda \cdot x_2 - \lambda \cdot x_5 - \mu_2 \cdot x_{10} - \mu_2 \cdot x_7 = 0$$
$$(\lambda + \mu_1 + \mu_2) \cdot x_7 - \lambda \cdot x_6 - \lambda \cdot x_3 - \mu_2 \cdot x_{11} - \mu_3 \cdot x_8 = 0$$
$$(\lambda + \mu_1 + \mu_3) \cdot x_8 - \lambda \cdot x_4 - \mu_2 \cdot x_{12} = 0$$
$$(\lambda + \mu_2) \cdot x_9 - \mu_1 \cdot x_{10} - \mu_3 \cdot x_{13} = 0$$
$$(\lambda + \mu_1 + \mu_2) \cdot x_{10} - \lambda \cdot x_9 - \mu_2 \cdot x_{11} - \mu_3 \cdot x_{14} = 0$$
$$(\lambda + 2\mu_2) \cdot x_{11} - \lambda \cdot x_7 - \lambda \cdot x_{10} - \mu_3 \cdot x_{12} - \mu_3 \cdot x_{15} = 0$$
$$(\lambda + \mu_2 + \mu_3) \cdot x_{12} - \lambda \cdot x_{11} - \lambda \cdot x_8 - \mu_3 \cdot x_{16} = 0$$
$$(\lambda + \mu_3) \cdot x_{13} - \mu_1 \cdot x_{14} = 0$$
$$(\lambda + \mu_1 + \mu_3) \cdot x_{14} - \lambda \cdot x_{13} - \mu_2 \cdot x_{15} = 0$$
$$(\lambda + \mu_2 + \mu_3) \cdot x_{15} - \lambda \cdot x_{14} - \mu_3 \cdot x_{16} = 0$$
$$x_1 + x_2 + x_3 + ... + x_{15} + \left(\frac{2\mu_3}{2\mu_3 - \lambda} \right) \cdot x_{16} = 1$$

A system like this can be expressed in matrix form. By creating a 16×16 matrix with the coefficients of the x_i, lets say A, and a 16×1 matrix containing the constants, lets say B, the above system can be written as:

$$A \cdot X = B$$

As an indication, the first row of matrix A in the case of the Balanced load routing policy will look like

$$A_1 = [\lambda, -\mu_1, 0, 0, -\mu_1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

and B will be as follows:

$$B = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]^T$$

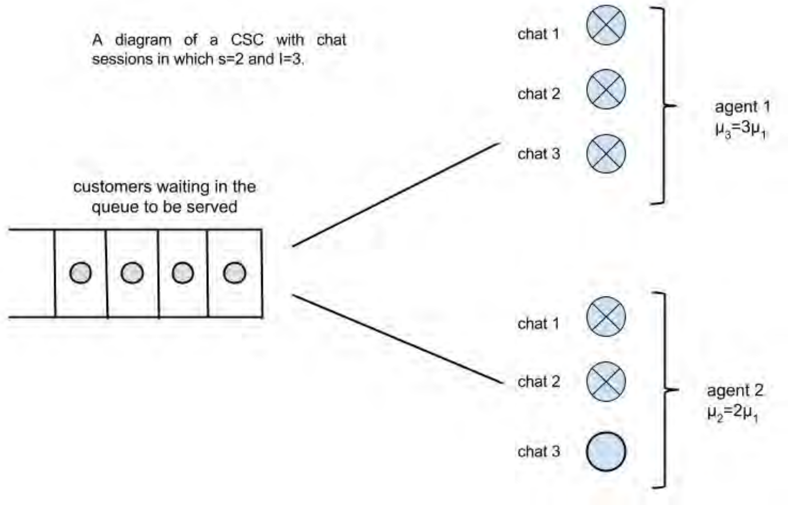
Then the unknowns $x_1, x_2, ..., x_{16}$ are the elements of the matrix X. A is a square and invertible matrix, and thus solving in terms of X yields

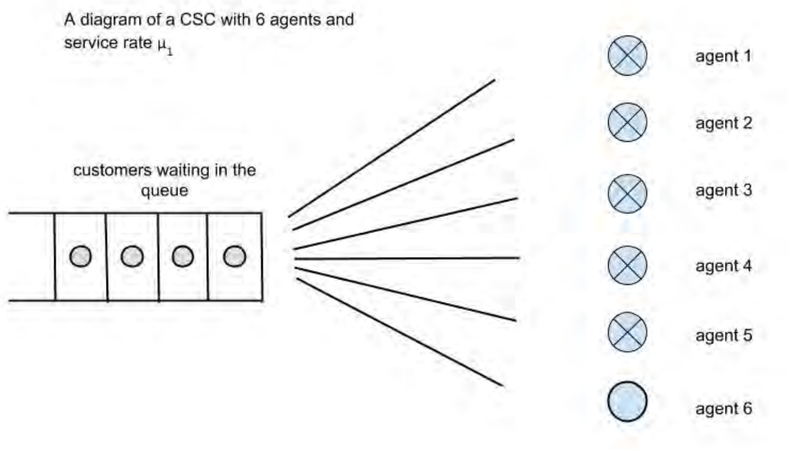
$$X = A^{-1} \cdot B$$

where A^{-1} is the inverse matrix of A. We use the command **linsolve(A,B)** in MatLab for this computation.

B Verification of the system

The chat system with 2 agents where each agent can serve at most 3 chats at the same time, with service rates μ_1, μ_2 and μ_3, behaves as an M/M/6 queueing system in which $\mu_2 = 2\mu_1$ and $\mu_3 = 3\mu_1$. For example, when in the CSC with chat sessions and 2 agents there are 5 customers, the total service rate is $\mu_2 + \mu_3 = 2\mu_1 + 3\mu_1 = 5\mu_1$, the same total service rate we have in the M/M/6 for these 5 customers (see figures below).


A diagram of a CSC with chat sessions in which s=2 and l=3.
customers waiting in the queue to be served
chat 1
chat 2
chat 3
agent 1
$\mu_3=3\mu_1$
chat 1
chat 2
chat 3
agent 2
$\mu_2=2\mu_1$


A diagram of a CSC with 6 agents and service rate μ_1
customers waiting in the queue
agent 1
agent 2
agent 3
agent 4
agent 5
agent 6

Thus we can verify our results for the mean waiting time and waiting time distribution with an Erlang-C calculator that can easily be found on the web. One link is:

http://www.math.vu.nl/ koole/ccmath/ErlangC/