

Computer Resource Optimization for Differentiated Customer Services

Kaiqi Xiong

Department of Computer Science
North Carolina State University
Raleigh, NC 27965-7534, USA
xiong@unity.ncsu.edu

Harry Perros

Department of Computer Science
North Carolina State University
Raleigh, NC 27965-7534, USA
hp@unity.ncsu.edu

Abstract

In enterprise computing, customer requests often need to be distinguished, with different request characteristics and customer's different service requirements. In this paper, we consider a set of computer resources used by a service provider to host enterprise applications for differentiated customer services subject to a service level agreement. We present an approach for resource optimization in such an environment that minimizes the total cost of computer resources used by a service provider for such an application while satisfying the QoS metric that the response time for executing differentiated service requests is statistically bounded. That is, each $\gamma^{(r)}$ % of the time the response time is less than a pre-defined value for class r customers. This QoS metric is more realistic than the mean response time typically used in the literature. Numerical results show the applicability of the approach and validate its accuracy.

1 Introduction

With the number of e-Business applications dramatically increasing, computer resource management will play an important part in enterprise computing. In this paper, we consider a collection of computer resources used by a service provider to host enterprise applications for multiple class business customers. An enterprise application running in such a computing environment is associated with a service level agreement (SLA). That is, the service provider is required to execute service requests from a customer within negotiated QoS requirements for a given price. Figure 1 depicts a scenario for such an environment. A customer represents a business that generates service requests at a given rate to be processed by the service provider's resource stations according to QoS requirements and for a given fee. As shown in Figure 1 a service request is transmitted to the service provider over a network provider. After it is processed at the various resource stations of the service provider the

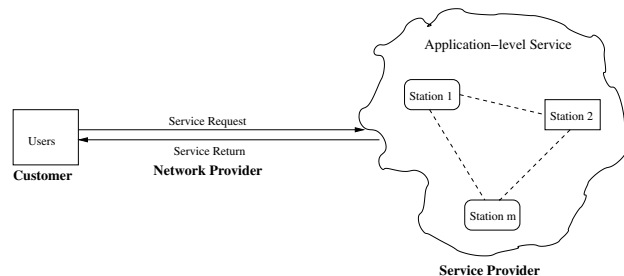


Figure 1. Execution of Service Requests

final result is sent back to the customer. For presentation purposes, we assume that each resource station has only one type of server associated with cost c_j . If they have multiple types of servers, we can decompose each resource station into several individual stations so that each one only contains one type of server with the same cost.

Let N_j be the number of servers at station j ($j = 1, 2, \dots, m$). Thus, the resource allocation is quantified by solving for n_j ($n_j = 1, 2, \dots, N_j$) in the following optimization problem:

$$I = \min_{n_1, \dots, n_m} (n_1 c_1 + \dots + n_m c_m) \quad (1)$$

subject to SLA constraints. Performance and price are the two most important components for a variety of SLAs for business applications, based on customer's different service requirements. Hence, in this paper, we consider the minimization of the overall cost of the service provider's computing resources allocated to the business customer so that $\gamma^{(r)}$ % of the time the response time, i.e., the time to execute a service request, is less than a pre-defined value for class r customers. Typically, in the literature the response time is taken into account through its mean. However, this may not be a meaningful quality of service as far as the customer is concerned, who may be more interested in a statistical bound of the response time.

In the real world customer requests often need to be distinguished, with different request characteristics and cus-

customer's different service requirements. Imposing a priority structure with preemption-resume is one way to implement a service for satisfying multiple class customer requests. A priority discipline does not depend on the state of a queue at the arrival of a customer, but is determined by a classification of arriving customers according to some criterion which is independent of the state of the queue. Suppose arriving customers to a single queue are classified into R different classes, where type r_1 customers ($r_1 = 1, 2, \dots, R-1$) have always priority for service over those of type r_2 ($r_2 > r_1$; $r_1, r_2 = 1, 2, \dots, R$), while customers of the same type are served in order of arrival (or FIFO). Then, it is said that the single queue operates according to a priority discipline with R classes.

In this paper, we consider a *preemptive-resume* priority discipline, that is, the service of a class r_2 customer can be interrupted if a higher-priority customer of class r_1 ($r_2 > r_1$) arrives during its service. The interrupted customer resumes its service from where it stopped after the higher-priority customer, and any other customer with priority higher than r_2 that may arrive during its service, complete their service. There are many other situations of practical interest-in the fields of modern computer systems, communication networks, computer server maintenance, and computer security checking, for example-in which the order of servicing is determined by preemptive-resume.

In this paper, we present an approach for the resource optimization that minimizes the total cost of computer resources required while preserving a given percentile of the response time for priority-class customers. We calculate the number of servers in each resource station that minimize a cost function that reflects operational costs. We first analyze an open tandem queueing network, and then we extend our work to an open tandem queueing network with feedback. We note that the proposed approach can be also applied to queueing networks consisting of nodes arbitrarily linked.

For notational simplicity, we only consider two priority customers in this paper. High-priority class customers are indexed 1 and low-priority class customers 2. The obtained results in this paper can be easily extended to the case of multiple priority customers by using an approach of class aggregation. It will be discussed in another paper due to the page limit.

The rest of the paper is organized as follows. Related work is briefly reviewed in section 2. In section 3 we define the SLA performance metric considered in this paper and formulate the resource optimization problem for differential customer services. In section 4, we first give the probability distribution of the response time distribution for a single priority queue with preemptive-resume. Then, we present two typical real-life models and propose an approach for solving the optimization problem for two priority-class cus-

tomers. In section 5, numerical simulations demonstrate the applicability and validity of the proposed approach. Finally, the conclusions are given in section 6.

2 Related Work

Resource optimization problems subject to performance metrics such as response time, bandwidth, and link utilization have been extensively studied. It was studied for multiple packet networks in [3], different classes of flows at network nodes in [5], IP networks in [11], wireless networks in [1], and grid computing in [12]. The calculation of the response time often becomes critical in solving the resource optimization problem. Martin and Nilsson [11] measured the average response time of a service request. In [14], Osogami and Wierman analyzed an $M/GI/k$ system with two priority classes and a general phase-type distribution and evaluated the optimal number of servers based on overall *mean* response time. A framework for service management in grid computing was defined in [12], but they did not provide a method for calculating the probability distribution of the response time.

In order to compute a percentile of the response time one has to first find the probability distribution function (pdf) of the response time. This is not an easy task in a complex computing environment involving many computing nodes. The calculation of the pdf of the response time is relatively simple for overtake-free paths in Jackson and Gordon-Newell networks. Walrand and Varaiya [18] showed that in any open Jackson network, the response times of a customer at the various nodes of overtake-free path are all mutually independent. Daduna [7] further proved that the same result is valid for overtake-free paths in Gordon-Newell networks. The pdf of the response time was derived for a closed queueing network in [13], and the passing time distribution was calculated for large Markov chains in [10].

3 The SLA Performance Metric and The Resource Optimization Problem

A SLA is a contract between a customer and a service provider that defines all aspects of the service that is to be provided. In this paper the SLA consists of service performance and a fee. We consider the percentile of the response time as the performance metric. This is the time it takes for a service request to be executed on the service provider's multiple resource stations.

Assume that $f_T(t)$ is the probability distribution function of a response time T of a certain priority class. For example, in the case of two priority classes, for the high-priority class $T = T^{(1)}$ and for the low-priority class $T = T^{(2)}$. $T_D^{(r)}$ is

a desired target response time for priority class r ($r = 1, 2$) that a customer requests and agrees upon with its service provider based on a fee paid by the customer. The SLA performance metric used in this paper can be expressed as follow:

$$\int_0^{T_D^{(r)}} f_{T^{(r)}}(t) dt \geq \gamma^{(r)}\%, \quad (r = 1, 2) \quad (2)$$

That is, $\gamma^{(r)}\%$ of the time a customer will receive its service in less than $T_D^{(r)}$ ($r = 1, 2$).

As an example let us consider an $M/M/1$ queue with an arrival rate $\lambda^{(r)}$ and a service rate $\mu^{(r)}$ ($r = 1, 2$). The service discipline is preemptive-resume. We want to describe the SLA performance metric (2) for the high-priority class. As discussed in section 1, in this case, the steady-state probability of the system as far the high-priority class is concerned is $p_0 = 1 - \rho^{(1)}$, and $p_k = (1 - \rho^{(1)})(\rho^{(1)})^k$, $k > 0$, where $\rho^{(1)} = \frac{\lambda^{(1)}}{\mu^{(1)}}$ (see [15]). The response time $T^{(1)}$ is exponentially distributed with the parameter $\mu^{(1)}(1 - \rho^{(1)})$, i.e., the probability distribution of the high-priority response time is given by

$$f_{T^{(1)}}(t) = \mu^{(1)}(1 - \rho^{(1)})e^{-\mu^{(1)}(1 - \rho^{(1)})t} \quad (3)$$

Using the definition given in (2), we have that

$$\int_0^{T_D^{(1)}} f_{T^{(1)}}(t) dt = 1 - e^{-\mu^{(1)}(1 - \rho^{(1)})T_D} \geq \gamma^{(1)}\% \quad (4)$$

or $\mu^{(1)} \geq \frac{-\ln(1 - \gamma^{(1)}\%)}{T_D^{(1)}} + \lambda^{(1)}$. This means that in order to guarantee a higher SLA service level, $\mu^{(1)}$ increases when $T_D^{(1)}$ decreases. Similarly, for any given arrival rate $\lambda^{(1)}$ and service rate $\mu^{(1)}$, we can use (4) to find the percentile of $\gamma^{(1)}$.

Then, the computer resource optimization problem can be formulated as the following optimization problem.

Resource Optimization Problem:

Find integers n_j ($1 \leq n_j \leq N_j$; $j = 1, 2, \dots, m$) in the n -dimensional integer optimization problem (1) under the constraints of percentile response times for differential customer services as expressed by (2), and the constraint: $I \leq C_D$, where C_D is a fee negotiated and agreed upon between a customer and the service provider.

4 Approaches for Resource Optimization

In this section, we propose an approach to solving the resource optimization problem for two typical service models that depict the path that service requests have to follow through the service provider's resource stations. Before presenting the approach, we need to derive the Laplace-Stieltjes transforms (LST) of the response time distributions for priority-class customers.

4.1 The LSTs of Response Time Distributions for Two Priority Customers

Let us recall that high-priority class customers are indexed 1 and low-priority class customers 2. In this section, we assume that the arrival processes of the two classes are independent of each other. Let $B^{(r)}(t)$ be the service time cumulative distribution of class r with mean $1/v^{(r)}$ and second moment $1/v_2^{(r)}$ ($r = 1, 2$). The total service time distribution $B(t)$ is given by $B(t) = \frac{\lambda^{(1)}}{\lambda^{(1)} + \lambda^{(2)}}B^{(1)}(t) + \frac{\lambda^{(2)}}{\lambda^{(1)} + \lambda^{(2)}}B^{(2)}(t)$, and the arrival rate into the queue is $\lambda = \lambda^{(1)} + \lambda^{(2)}$. It follows that the total utilization $\rho = \rho^{(1)} + \rho^{(2)}$, where $\rho^{(r)} = \frac{\lambda^{(r)}}{v^{(r)}}$, equals to the occupation time given by $\lambda \int_0^\infty t dB(t) = \frac{\lambda^{(1)}}{v^{(1)}} + \frac{\lambda^{(2)}}{v^{(2)}} = \rho^{(1)} + \rho^{(2)}$. Assume that the stability condition of the queueing system holds, i.e., $\rho = \rho^{(1)} + \rho^{(2)} < 1$.

The LST of the service time distribution of class r is

$$g^{(r)}(s) = \int_0^\infty e^{-st} dB^{(r)}(t), \quad r = 1, 2 \quad (5)$$

and the LST of the residual service time distribution for class r , ($r=1, 2$) is:

$$g_e^{(r)}(s) = \int_0^\infty e^{-st} (1 - B^{(r)}(t)) v^{(r)} dt = \frac{(1 - g^{(r)}(s)) v^{(r)}}{s} \quad (6)$$

The busy period time is the time between an interruption moment at which the server becomes busy due to an arriving customer and the first moment at which the server becomes available again. The LST of the busy time distribution of the high priority class, denoted by $\delta^{(1)}(s)$, is the smallest root of the *Kendall functional equation* (Cohen [6])

$$\delta^{(1)}(s) = g^{(1)}(s + \lambda^{(1)}(1 - \delta^{(1)}(s))) \quad (7)$$

where $g^{(1)}$ is defined by (5).

Let c be the completion time of a customer, that is, the time elapsed from the moment when the service of the customer begins to the moment where the customer is completely served. As derived in (3.51) of [6], the LST, $L_c(s)$, of the complete time c for both preemptive-resume and non-preemptive resume, is:

$$L_{c(s)} = g^{(2)}(z(s)) \quad (8)$$

where

$$z(s) = s + \lambda^{(1)}(1 - \delta^{(1)}(s)) \quad (9)$$

For the preemptive-resume discipline, class 2 does not exist as far as class 1 is concerned. Thus in this case the waiting time distribution of the high-priority class is the same as in the FIFO queue without priorities. The probability distribution of the high-priority class was given in (3). Hence, in the following discussion of this section we

only need to find the LSTs of the waiting time distribution of the low-priority class.

Denote $W^{(2)}$ the cumulative distribution of the steady-state waiting time of the low-priority customers. According to (3.63) and (3.67) in [6], the LST of the low-priority waiting time $W^{(2)}(t)$ is given by

$$L_{W^{(2)}}(s) = \frac{1 - \rho}{1 - \rho f(s)}, \quad s > 0 \quad (10)$$

where

$$f(s) = \frac{\rho^{(1)}}{\rho} h_0^{(1)}(s) + \frac{\rho^{(2)}}{\rho} g_e^{(2)}(z(s)) \quad (11)$$

$$h_0^{(1)}(s) = \frac{1 - \delta^{(1)}(s)}{\frac{1}{v^{(1)}} s + \rho^{(1)}(1 - \delta^{(1)}(s))} \quad (12)$$

Note that $g_e^{(2)}(z(s))$ and $\delta^{(1)}(s)$ are determined by (5) and (7) respectively.

Therefore, the low-priority response time denoted by $T^{(2)}$ equals to the sum of the low-priority waiting time and the low-priority completion time whose LST is given by (8). The low-priority response time distribution is a convolution of the distributions of the low-priority waiting time and the low-priority completion time. Thus the LST of the low-priority response time distribution denoted by $L_{T^{(2)}}(s)$ is: $L_{T^{(2)}}(s) = L_{W^{(2)}}(s) \times L_{c(s)}$. That is,

$$L_{T^{(2)}}(s) = \frac{(1 - \rho) \times g^{(2)}(z(s))}{1 - \rho f(s)}, \quad s > 0 \quad (13)$$

The distributions of the low-priority response time can be obtained by numerically inverting the LST of (13) respectively. This procedure can usually be done via the software package provided in [9].

For presentation purposes, we only consider an $M/M/1$ queue in this paper since the distributions of the low-priority response time given by (13) can be specified and simplified, as discussed in the following subsection.

4.1.1 The LST of The Low-priority Response Time Distribution In An $M/M/1$ Queue

In an $M/M/1$ priority queue, assume that the service rates are $\mu^{(r)}$ ($r = 1, 2$). This means that $v^{(r)} = \mu^{(r)}$. In this case, from (5) and (6) we have that for $r = 1, 2$,

$$g^{(r)}(s) = \frac{\mu^{(r)}}{s + \mu^{(r)}} \quad (14)$$

$$g_e^{(r)}(s) = \frac{\mu^{(r)}}{s + \mu^{(r)}} \quad (15)$$

Therefore, from (7) and (14) we derive the relation:

$$\delta^{(1)} = \frac{1 - \delta^{(1)}}{\frac{1}{v^{(1)}} s + \rho^{(1)}(1 - \delta^{(1)})} \quad (16)$$

and by solving for $\delta^{(1)}$ we obtain

$$\delta^{(1)} = \frac{\eta - \sqrt{\eta^2 - 4\lambda^{(1)}\mu^{(1)}}}{2\lambda^{(1)}} \quad (17)$$

where η is determined by

$$\eta = s + \lambda^{(1)} + \mu^{(1)} \quad (18)$$

Furthermore, it follows from (11), (12) and (16) that $h_0^{(1)}(s) = \delta^{(1)}$, and

$$f(s) = \frac{\rho^{(1)}}{\rho} \frac{\eta - \sqrt{\eta^2 - 4\lambda^{(1)}\mu^{(1)}}}{2\lambda^{(1)}} + \frac{\rho^{(2)}}{\rho} \frac{\mu^{(2)}}{z(s) + \mu^{(2)}} \quad (19)$$

where $z(s)$ is given by (9).

Moreover, from (7), (9) and (14) we have that

$$\delta^{(1)} = \frac{\mu^{(1)}}{z(s) + \mu^{(1)}}, \quad \text{or} \quad z(s) = \mu^{(1)}[(\delta^{(1)})^{-1} - 1] \quad (20)$$

Thus due to (20), the LST of the low-priority waiting time distribution given in (10) reduces to

$$\begin{aligned} L_{W^{(2)}}(s) &= \frac{1 - \rho}{1 - \rho^{(1)}\delta^{(1)} - \rho^{(2)}\frac{\mu^{(2)}}{z(s) + \mu^{(2)}}} \\ &= \frac{(1 - \rho)\{\mu^{(1)}[(\delta^{(1)})^{-1} - 1] + \mu^{(2)}\}}{(\mu^{(1)} - \mu^{(2)})\rho^{(1)}\delta^{(1)} + \mu^{(1)}(\delta^{(1)})^{-1} - \xi} \end{aligned} \quad (21)$$

where ξ is given by

$$\begin{aligned} \xi &= \mu^{(1)}(\rho^{(1)} + \rho^{(2)}) + (\mu^{(1)} - \mu^{(2)})(1 - \rho^{(2)}) \\ &= \mu^{(1)}\rho + (\mu^{(1)} - \mu^{(2)})(1 - \rho^{(2)}) \end{aligned} \quad (22)$$

Hence, from (17) and (21) we have the LST of the low-priority waiting distribution for an $M/M/1$ queue

$$L_{W^{(2)}}(s) = \frac{(1 - \rho)(\frac{\eta}{2} + \frac{1}{2}\sqrt{\eta^2 - 4\lambda^{(1)}\mu^{(1)}} - (\mu^{(1)} - \mu^{(2)}))}{(1 - \frac{\mu^{(2)}}{2\mu^{(1)}})\eta - \xi + \frac{\mu^{(2)}}{2\mu^{(1)}}\sqrt{\eta^2 - 4\lambda^{(1)}\mu^{(1)}}} \quad (23)$$

Consequently, replacing (23) in (13) gives the LST of the low-priority response time distribution

$$\begin{aligned} L_{T^{(2)}}(s) &= \mu^{(2)}(1 - \rho)\left[\frac{\eta}{2} + \frac{1}{2}\sqrt{\eta^2 - 4\lambda^{(1)}\mu^{(1)}}\right. \\ &\quad \left. - (\mu^{(1)} - \mu^{(2)})\right]\{(z(s) + \mu^{(2)}) \times [(1 - \frac{\mu^{(2)}}{2\mu^{(1)}})\eta \\ &\quad - \xi + \frac{\mu^{(2)}}{2\mu^{(1)}}\sqrt{\eta^2 - 4\lambda^{(1)}\mu^{(1)}}]\}^{-1}, \quad s > 0 \end{aligned} \quad (24)$$

where $z(s)$, η and ξ are given by (9), (18) and (22) respectively.

Additionally, when the service rates of the two priority classes are the same, i.e., $\mu^{(1)} = \mu^{(2)}$, (19) reduces to the following expression:

$$\begin{aligned} f(s) &= \frac{\rho^{(1)}}{\rho} \delta^{(1)} + \frac{\rho^{(2)}}{\rho} \frac{\mu^{(2)}}{z(s) + \mu^{(2)}} \\ &= \frac{\rho^{(1)}}{\rho} \delta^{(1)} + \frac{\rho^{(2)}}{\rho} \frac{\mu^{(1)}}{z(s) + \mu^{(1)}} = \delta^{(1)} \end{aligned}$$

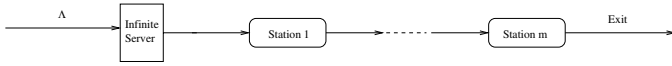


Figure 2. A Tandem-station Service Model

due to (20). Hence, equation (10) becomes $L_{W^{(2)}}(s) = \frac{1-\rho}{1-\rho\delta^{(1)}}$. Thus, the LST of the low-priority response time distribution is

$$L_{T^{(2)}}(s) = \frac{(1-\rho)\delta^{(1)}}{1-\rho\delta^{(1)}} \quad (25)$$

based on (13), (20) and $\mu^{(1)} = \mu^{(2)}$. It should point out that when the utilization ρ is fixed, the LST of low-priority response time distribution does not depend on the arrival rate $\lambda^{(2)}$ of the low-priority class.

4.2 Algorithms for The Resource Optimization Problem

In this subsection, we study two queueing network models that depict the path that service requests have to follow through the service provider's resource stations. These two models are shown in Figures 2 and 3. We refer to these two queueing models as service models since they depict the resources used to provide a service to a customer.

The first service model consists of a single infinite server and m stations numbered sequentially from 1 to m as shown in Figure 2. Each station j is modeled as a priority queue served by n_j identical servers, each providing a service at the rate $\mu_j^{(r)}$, where $r = 1, 2$. Let $\Lambda^{(r)}$ be the external arrival rate to the infinite server, and let $\lambda^{(r)}$ and $\lambda_j^{(r)}$ be the effective arrival rates to the infinite server and station j , $j = 1, 2, \dots, m$. We assume that all service times are exponentially distributed and the external arrival to the infinite server occurs in a Poisson fashion.

The infinite server represents the total propagation delay from the user to the service provider and back, and also from station 1 to m . Each station carries out a particular function. For instance, it could be a database server, a file server, a web server, a group of CPUs and local disks, etc. We consider two priority classes of customers in this paper. In the following discussion each station is modeled as a single $M/M/1$ priority queue with arrival rate $\lambda_j^{(r)}$ and service rate $\psi^{(r)}(n_j)\mu_j^{(r)}$, where $\psi^{(r)}(n_j)$ is a known function of n_j ($r = 1, 2$), and depends on the configuration of servers and the type of customers at each station. It is non-decreasing and can be inverted, i.e., $(\psi^{(r)})^{-1}$ exists ($r = 1, 2$). For instance, suppose that a station represents a group of CPUs. Then, $\psi^{(r)}(n)$ can be seen as a CPU scaling factor for the number of CPUs from 1 to n . According to [4], $\psi^{(r)}(n) = (\xi^{(r)})^{\log n}$, where $\xi^{(r)}$ is a basic scaling

factor from 1 CPU to 2, and it ranges from 1 to 2 ($r = 1, 2$). So, $(\psi^{(r)})^{-1}(n) = (\xi^{(r)})^{-\log n}$. Additionally, each station that is modeled as a *single M/M/1* priority queue with service rate $\psi^{(r)}(n_j)\mu_j^{(r)}$ can only serve either high-priority or low priority customers at one time. Hence, $\psi^{(1)}(n_j)\mu_j^{(1)}$ is considered as the same as $\psi^{(2)}(n_j)\mu_j^{(2)}$.

Since the queueing network is overtake-free, the waiting time of a customer at a station is independent of its waiting times at other stations (see [7] and [18]). Let $X^{(r)}$ be the service time at the infinite server and $X_j^{(r)}$ be the time elapsed from the moment a customer arriving at station j to the moment it departs from the station ($r = 1, 2$). Then, the total response time is $T^{(r)} = X^{(r)} + X_1^{(r)} + X_2^{(r)} + \dots + X_m^{(r)}$, and hence the LST (Laplace-Stieltjes transform) of the response time T is

$$L_{T^{(r)}}(s) = L_{X^{(r)}}(s)L_{X_1^{(r)}}(s)\cdots L_{X_m^{(r)}}(s) \quad (26)$$

where $L_{X^{(r)}}(s)$ is the LST of the service time $X^{(r)}$ given by

$$L_{X^{(r)}}(s) = \frac{\lambda^{(r)}}{s + \lambda^{(r)}} \quad (27)$$

and $L_{X_j^{(r)}}(s)$ is the LST of the response time $X_j^{(r)}$ at the j -th station, where $r = 1, 2$.

Due to the preemptive-resume priority, the high-priority response time is that same as in the single class FIFO $M/M/1$ whose probability distribution can be expressed as (3) in section 3. Thus $L_{X_j^{(1)}}(s)$ is determined by

$$L_{X_j^{(1)}}(s) = \frac{\psi^{(1)}(n_j)\mu_j(1-\rho_j^{(1)})}{s + \psi^{(1)}(n_j)\mu_j(1-\rho_j^{(1)})}, \quad (j = 1, 2, \dots, m) \quad (28)$$

and $L_{X_j^{(2)}}(s)$ is the LST of the low-priority response time given by

$$L_{X_j^{(2)}}(s) = \frac{(1-\rho_j)\delta_j^{(1)}}{1-\rho_j\delta_j^{(1)}}, \quad (j = 1, 2, \dots, m) \quad (29)$$

due to (25), where $\rho_j^{(r)} = \frac{\lambda_j^{(r)}}{\psi^{(r)}(n_j)\mu_j}$, $\rho_j = \frac{\lambda_j^{(1)} + \lambda_j^{(2)}}{\psi^{(r)}(n_j)\mu_j}$, $\delta_j^{(1)}$

is given by $\delta_j^{(1)} = \frac{\eta_j - \sqrt{\eta_j^2 - 4\psi^{(1)}(n_j)\lambda_j^{(1)}\mu_j^{(1)}}}{2\psi^{(1)}(n_j)\lambda_j^{(1)}}$, and $\eta_j = s + \lambda_j^{(1)} + \psi^{(1)}(n_j)\mu_j^{(1)}$ for $j = 1, 2, \dots, m$.

From (26), (27), (28) and (29) we have that

$$L_{T^{(1)}}(s) = \frac{\lambda^{(1)}}{s + \lambda^{(1)}} \prod_{j=1}^m \frac{\psi^{(1)}(n_j)\mu_j(1-\rho_j^{(1)})}{s + \psi^{(1)}(n_j)\mu_j(1-\rho_j^{(1)})}$$

and

$$L_{T^{(2)}}(s) = \frac{\lambda^{(2)}}{s + \lambda^{(2)}} \prod_{j=1}^m \frac{(1-\rho_j)\delta_j^{(1)}}{1-\rho_j\delta_j^{(1)}}$$

We observe that $f_{T^{(r)}}(t)$ and $F_{T^{(r)}}(t)$ ($r = 1, 2$) are usually nonlinear functions of t and n_j . Hence, the resource optimization problem is an n -dimensional linear optimization problem subject to nonlinear constraints. In general, it is not easy to solve this problem. However, the complexity of the problem can be significantly reduced by requiring that the service rates of the queues in the service model in Figure 2 are all equal. That is, we find the optimum value of n_1, \dots, n_m such that $\psi^{(r)}(n_1)\mu_1^{(r)} = \dots = \psi^{(r)}(n_m)\mu_m^{(r)}$ ($r = 1, 2$), called *balanced utilization*. (We note that in production lines, it is commonly assumed that the service stations are balanced.)

From the traffic equations: $\lambda^{(r)} = \lambda_j^{(r)} = \Lambda^{(r)}$ ($j = 1, 2, \dots, m$), we have that the utilization of each station $\rho_j^{(r)} = \frac{\lambda_j^{(r)}}{\psi^{(r)}(n_j)\mu_j^{(r)}} = \frac{\Lambda^{(r)}}{\psi^{(r)}(n_j)\mu_j^{(r)}}$. Thus we have that for the high-priority queue, $\hat{a}_i = \psi^{(1)}(n_i)\mu_i^{(1)}(1 - \rho_i^{(1)}) = \psi^{(1)}(n_j)\mu_j(1 - \rho_j^{(1)}) = \hat{a}_j \triangleq \hat{a}$ that implies $n_j = (\psi^{(1)})^{-1}(\frac{\hat{a} + \lambda_j^{(1)}}{\mu_j^{(1)}})$ ($i, j = 1, 2, \dots, m$). Hence, from (26) we have $f_{T^{(1)}}(t) = L^{-1}\{\frac{\lambda^{(1)}}{s + \lambda^{(1)}} \cdot \frac{\hat{a}^m}{(s + \hat{a})^m}\}$, and subsequently obtain that

$$F_{T^{(1)}}(t) = L^{-1}\left\{\frac{\lambda^{(1)}}{s(s + \lambda^{(1)})} \cdot \frac{\hat{a}^m}{(s + \hat{a})^m}\right\} \quad (30)$$

Moreover, for the low-priority queue, we have that $\rho_{j_1} = \rho_{j_2} \triangleq \hat{\rho}$, and $\delta_{j_1}^{(1)} = \delta_{j_2}^{(1)} \triangleq \hat{\delta}^{(1)}$, for $j_1, j_2 = 1, 2, \dots, m$, due to $\psi^{(2)}(n_1)\mu_1^{(2)} = \dots = \psi^{(2)}(n_m)\mu_m^{(2)} \triangleq \hat{b}$. Thus $L_{T^{(2)}}(s)$ reduces to

$$L_{T^{(2)}}(s) = \frac{\lambda^{(2)}}{s + \lambda^{(2)}} \frac{(1 - \hat{\rho})^m (\hat{\delta}^{(1)})^m}{(1 - \hat{\rho} \hat{\delta}^{(1)})^m} \quad (31)$$

which is a function of only one variable \hat{b} , since $\hat{\rho}$ and $\hat{\delta}^{(1)}$ are considered as functions of only one variable \hat{b} . Consequently, $\sum_{j=1}^m n_j c_j$ reduces to a function of variable \hat{a} due to $n_j^{(1)} = \lceil (\psi^{(1)})^{-1}(\frac{\hat{a} + \lambda_j^{(1)}}{\mu_j^{(1)}}) \rceil$ for a high-priority customer and $n_j^{(2)} = \lceil (\psi^{(2)})^{-1}(\frac{\hat{b}}{\mu_j^{(2)}}) \rceil$ for a low-priority customer. Thus the resource optimization problem can be decomposed into the two one-dimensional resource optimization problems for both high-priority and low-priority customers respectively. We have the following algorithm for the resource optimization problem.

Algorithm 1:

1. The minimization problem for high-priority customers: Find \hat{a} in the one dimensional optimization problem:

$$\hat{a}^{min} \leftarrow \arg \min_{\hat{a}} F_{T^{(1)}}(t)|_{t=T_D^{(1)}}$$

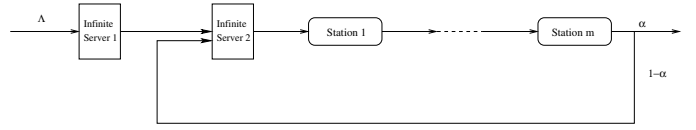


Figure 3. A Service Model With Feedback

subject to the constraint $F_{T^{(1)}}(t)|_{t=T_D^{(1)}} \geq \gamma^{(1)}\%$ at $\hat{a} = \hat{a}^{min}$, where $F_{T^{(1)}}(t)$ is given by (30).

2. The minimization problem for low-priority customers: Find \hat{b} in the one-dimensional optimization problem:

$$\hat{b}^{min} \leftarrow \arg \min_{\hat{b}} F_{T^{(2)}}(t)|_{t=T_D^{(2)}}$$

subject to the constraint $F_{T^{(2)}}(t)|_{t=T_D^{(2)}} \geq \gamma^{(2)}\%$ at $\hat{b} = \hat{b}^{min}$, where $F_{T^{(2)}}(t)$ is given by (31).

3. Compute integers $n_j^{(1)}$ and $n_j^{(2)}$ by using $n_j^{(1)} = \lceil (\psi^{(1)})^{-1}(\frac{\hat{a}^{min}}{\mu_j^{(1)}(1 - \rho_j^{(1)})}) \rceil$ and $n_j^{(2)} = \lceil (\psi^{(2)})^{-1}(\frac{\hat{b}^{min}}{\mu_j^{(2)}}) \rceil$. Then, calculate $n_j = \max\{n_j^{(1)}, n_j^{(2)}\}$ ($j = 1, 2, \dots, m$).
4. Check if $1 \leq n_j \leq N_j$ ($j = 1, 2, \dots, m$) and $I \leq C^D$ are satisfied. If yes, the obtained n_j is the number of servers required at each station. Otherwise, print “the problem cannot be solved.”

Note that when the balanced utilization as above is satisfied, the suboptimal solution obtained by Algorithm 1 is optimal for the resource optimization problem. This statement is also true for Algorithm 2 that will be given later.

In order to allow for a more complex execution path of a service request, we extended the above model to a service model with feedback, as shown in Figure 3. Infinite server 1 represents the total propagation delay within a network provider and infinite server 2 represents the propagation delay within the service provider, i.e. among stations 1 to m . In this figure, a customer upon completion of its service at the m -th station, exits the system with probability α , or returns to the beginning of the system with probability $1 - \alpha$. We note that the model shown in Figure 3 can be easily extended to a network of queues arbitrarily connected. We reuse the notation in the first model shown in Figure 2: $\Lambda^{(r)}$ as the external arrival rate, $\lambda_d^{(r)}$, $\lambda^{(r)}$ and $\lambda_j^{(r)}$ as the effective arrival rates to the second infinite server and station j , and $\mu_j^{(r)}$ as the service rate at station j , where $j = 1, 2, \dots, m$ and $r = 1, 2$.

The main difficulty of this resource optimization problem is to find $f_{T^{(r)}}(t)$, the probability distribution function of $T^{(r)}$ ($r = 1, 2$). We obtain this probability distribution

function assuming that the waiting time of a customer at a station is independent of its waiting time at other stations, and each visit at the same station j is independent of the others. (We note that this assumption of independence does not hold in queueing networks with feedback. However, as will be discussed in the validation section the solution obtained has a good accuracy.) We first have the traffic equations: $\lambda_d^{(r)} = \Lambda^{(r)}$, $\lambda^{(r)} = \Lambda^{(r)} + (1 - \alpha)\lambda_m^{(r)}$ and $\lambda_j^{(r)} = \lambda^{(r)}$ that implies $\lambda_j^{(r)} = \lambda^{(r)} = \frac{\Lambda^{(r)}}{\alpha}$. and the utilization of each station is $\rho_j^{(1)} = \frac{\lambda_j^{(r)}}{\psi^{(r)}(n_j)\mu_j} = \frac{\Lambda^{(r)}}{\alpha\mu_j\psi^{(r)}(n_j)}$ ($j = 1, 2, \dots, m$).

Then, the high-priority and low-priority response times of the k -th pass at the infinite station and the j -th station are considered as the sum of $m + 2$ random variables $T^{(r)}(k) = D^{(r)} + X^{(r)} + X_1^{(r)} + \dots + X_m^{(r)}$, where we assume that the high-priority (or low-priority) waiting time of a customer at a station is independent of its high-priority (or low-priority) waiting times in other visits to the same station. $D^{(r)}$ and $X^{(r)}$ are the service times of class r at the first and second infinite servers respectively, and $X_j^{(r)}$ is the time elapsed from the moment a class r customer arrives at station j to the moment it departs from it. Then, the total response time is $T^{(r)} = \sum_{k=0}^{\infty} p(k)T^{(r)}(k)$, where $p(k)$ is the steady state probability that a request will circulate k times at the infinite station and the j -th station through the computing system. $p(k)$ is determined by $p(k) = \alpha(1 - \alpha)^{k-1}$. Thus the LST of the response time $T^{(r)}$ is $L_{T^{(r)}}(s) = L_D(s) \sum_{k=0}^{\infty} p(k)L_X^k(s)L_{X_1^{(r)}}^k(s) \dots L_{X_m^{(r)}}^k(s)$, which can be re-written as follows:

$$L_{T^{(r)}}(s) = \frac{\alpha L_{D^{(r)}}(s)L_X(s) \prod_{j=1}^m L_{X_j^{(r)}}(s)}{1 - (1 - \alpha) L_X(s) \prod_{j=1}^m L_{X_j^{(r)}}(s)} \quad (32)$$

where $L_{D^{(r)}}(s)$ is the LST of the service time $D^{(r)}$ given by $L_{D^{(r)}}(s) = \frac{\Lambda^{(r)}}{s + \Lambda^{(r)}}$, and replacing $L_X(s)$ and $L_{X_j^{(r)}}(s)$ ($j = 1, 2, \dots, m$) by (27), (28) and (29) in (32), we have that

$$L_{T^{(1)}}(s) = \frac{(\Lambda^{(1)})^2 \prod_{j=1}^m \hat{a}_j}{(s + \Lambda^{(1)})[(s + \lambda^{(1)}) \prod_{j=1}^m (s + \hat{a}_j) - (1 - \alpha)\lambda^{(1)} \prod_{j=1}^m \hat{a}_j]} \quad (33)$$

where $\hat{a}_j = \psi^{(1)}(n_j)\mu_j^{(1)}(1 - \rho_j^{(1)})$, and

$$L_{T^{(2)}}(s) = (\Lambda^{(2)})^2 \prod_{j=1}^m [(1 - \rho_j)\delta_j^{(1)}](s + \Lambda^{(2)})^{-1} \times \{(s + \lambda^{(2)}) \prod_{j=1}^m (1 - \rho_j \delta_j^{(1)}) - (1 - \alpha)\lambda^{(2)} \prod_{j=1}^m [(1 - \rho_j)\delta_j^{(1)}]\}^{-1} \quad (34)$$

To find the response time distribution $f_{T^{(r)}}(t)$, we are required to invert the LST given by (33) using partial fraction decomposition of a rational function. However, the partial fraction decomposition of the rational function requires searching for roots of a high-order polynomial. It is

usually not an easy task when the order of the polynomial is more than 5. Instead, in this paper the LST given by (33) is inverted numerically, as (34).

Similarly, we want to find n_1, \dots, n_m such that the best utilization of these stations is achieved, which implies that each station has the same maximal service capacity. That is, $\hat{a}_i = \hat{a}_j = \hat{a}$, $\hat{\rho}_i = \hat{\rho}_j = \hat{\rho}$ and $\hat{\delta}_i^{(1)} = \hat{\delta}_j^{(1)} = \hat{\delta}^{(1)}$ ($i, j = 1, \dots, m$). Then, from equations (33) and (34), and $F_{T^{(r)}}(t) = L^{-1}\{L_{T^{(r)}}(s)/s\}$ ($r = 1, 2$) we have

$$F_{T^{(1)}}(t) = L^{-1}\left\{\frac{(\Lambda^{(1)})^2 \hat{a}^m}{s(s + \Lambda^{(1)})[(s + \lambda^{(1)})(s + \hat{a})^m - (1 - \alpha)\lambda^{(1)}\hat{a}^m]}\right\} \quad (35)$$

and

$$F_{T^{(2)}}(t) = L^{-1}\{(\Lambda^{(2)})^2 [(1 - \hat{\rho})\hat{\delta}^{(1)}]^m s^{-1} (s + \Lambda^{(2)})^{-1} \times \{(s + \lambda^{(2)})(1 - \hat{\rho}\hat{\delta}^{(1)})^m - (1 - \alpha)\lambda^{(2)} [(1 - \hat{\rho})\hat{\delta}^{(1)}]^m\}^{-1}\} \quad (36)$$

Thus we have the following algorithm for the resource optimization problem in the model shown in Figure 3.

Algorithm 2:

Steps 1-4 are the same as Steps 1-4 in Algorithm 1 except $F_{T^{(2)}}(t)$ and $F_{T^{(2)}}(t)$ given by (35) and (36) respectively.

Note that if we cannot get a solution for the resource optimization problem using Algorithm 1 (or 2), then the service provider cannot execute the service request for the service model 1 (or 2) due to at least one of the following reasons: (i) the service provider has an insufficient resource (i.e., N_j is too small), (ii) a pre-specific fee is too low (i.e., $I > C_D$), and (iii) a network connection is either too slow or has a problem so that (2) cannot be satisfied. Using these information, we may detect and debug either a network problem or a service provider's capacity problem, or the SLA needs to be re-negotiated.

5 Numerical Validations

In this section we demonstrate the accuracy and applicability of our proposed approximation method.

Two types of errors are introduced in our proposed approximation method. The first, hereafter referred to as Class I error, comes from numerically inverting the Laplace transform. The other, hereafter referred to as Class II error, is due to the assumptions that the waiting time of a customer at each station is independent of the waiting times at the other stations, and it is also independent of its waiting times in other visits to the same station.

The relative error % is used to measure the accuracy of the approximate results compared to model simulation results, and it is defined as follows

$$\text{Relative error \%} = \frac{\text{Approximate Result} - \text{Simulation Result}}{\text{Simulation Result}} \times 100$$

As is seen, our proposed approximation method heavily depends on the computation of the inverse Laplace transform

$$f(t) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} e^{ts} \hat{f}(s) ds$$

where $\hat{f}(s)$ is the image function of the inverse Laplace function $f(t)$ defined by $\hat{f}(t) = \int_0^\infty e^{-st} f(t) dt$, where $f(t)$ is called the original function of $\hat{f}(s)$ and it is a real or complex-valued function defined on the positive part \mathcal{R}_+ .

The numerical inversion of the Laplace transform has been widely studied and several efficient methods have been proposed in the past a few decades (see Graf [9]). However, as is described in [9], the numerical computation of an inverse Laplace transform is an ill-posed problem. The inverse Laplace transform is determined by the singularities of the image function $\hat{f}(s)$. This means that the behavior of the image function near the singularities determines the inverse Laplace transform. Hence, we need to consider the singularities of the image function in our numerical validations. Additionally, whereas some numerical methods work for certain image functions well, they may provide poor results for other image functions. No single method works for any image functions. Thus, in our numerical validations we employed several different numerical methods for a given image function. If two or more methods can reach about the same results, then we are confident that the derived numerical inverse Laplace transform is correct. These numerical methods include the inversion methods using Laguerre functions and Fourier functions (see Graf [9]), Gaussian quadrature formulas (see Piessens [16]), and the method by Gaver [8] and Stehfest [17].

We study the accuracy of our proposed approximation method using several examples below.

Contrary to the distribution of the high-priority response time whereby it is the single class FIFO $M/M/1$ case, as given in section 3, the distribution of the low-priority response time in an $M/M/1$ queue does not have a closed-form solution. The distribution whose Laplace transform is given by (25) have to be evaluated numerically by using one of the numerical methods for inverting a Laplace transform (e.g., [9]). Hence, we first verify the accuracy of the approximate method for the single queue given by (25). Let $\mu^{(1)} = \mu^{(2)} = \frac{1000}{9} = 111.11$ and $\lambda^{(r)}$ is varied ($r = 1, 2$).

We simulated the queueing network using Arena (see [2]) and the analytical method was implemented in Mathematica using the package of the inverse Laplace transform in Graf [9]. The simulation results are considered as “exact” since the simulation model is an exact representation of the queueing network under study.

Tables 1 and 2 show the simulated and approximate cumulative distributions of the low-priority response times for the two different cases: (i) $\lambda^{(1)} = \lambda^{(2)} = 50$; (ii)

Table 1. The Cumulative Distribution of The Low-priority Response Time for $\lambda^{(1)} = \lambda^{(2)} = 50$

Response Time	Simul	Approx	R-Err %
0.1	0.5183	0.4821	-6.9844
0.3	0.8691	0.8318	-4.2918
0.5	0.9691	0.9447	-2.5178
0.7	0.9911	0.9818	-0.9384
0.9	0.9997	0.9940	-0.5702
1.1	1.0000	0.9980	-0.2000
1.3	1.0000	0.9994	-0.0600
1.5	1.0000	0.9998	-0.0200
1.7	1.0000	0.9999	-0.0100
1.9	1.0000	1.0000	0.0000

Table 2. The Cumulative Distribution of The Low-priority Response Time for $\lambda^{(1)} = 75$ and $\lambda^{(2)} = 25$

Response Time	Simul	Approx	R-Err %
0.1	0.4175	0.3837	-8.0958
0.3	0.7115	0.6783	-4.6662
0.5	0.8617	0.8225	-4.5491
0.7	0.9338	0.9003	-3.5875
0.9	0.9669	0.9435	-2.4201
1.1	0.9835	0.9679	-1.5862
1.3	0.9922	0.9817	-1.0583
1.5	0.9970	0.9895	-0.7523
1.7	0.9982	0.9940	-0.4208
1.9	0.9996	0.9966	-0.3001
2.1	1.0000	0.9980	-0.2000
2.3	1.0000	0.9999	-0.0100
2.5	1.0000	1.0000	0.0000

$\lambda^{(1)} = 75$ and $\lambda^{(2)} = 25$. In these two tables, the column labeled “Simul” gives the simulation result, the column labeled “Approx” gives the approximate result, and the column labeled “R-Err %” gives their relative errors. The same abbreviations are also used in other tables in the rest of this section. It appears that using the package of the inverse Laplace transform in Graf [9] we can get a good accuracy for the numerical inversion of the low-priority response time distribution given by (25) respectively.

Then, we verify the accuracy of our approach for the first service model shown in Figure 2. Let $m = 8$, $\lambda^{(1)} = 100$, $\lambda^{(1)} = 50$, $\mu_1^{(1)} = 48$, $\mu_2^{(1)} = 18$, $\mu_3^{(1)} = 85$, $\mu_4^{(1)} = 32$, $\mu_5^{(1)} = 49$, $\mu_6^{(1)} = 24$, $\mu_7^{(1)} = 28$, $\mu_8^{(1)} = 38$, $\mu_1^{(2)} = 42$, $\mu_2^{(2)} = 15$, $\mu_3^{(2)} = 60$, $\mu_4^{(2)} = 25$, $\mu_5^{(2)} = 41$, $\mu_6^{(2)} = 18$, $\mu_7^{(2)} = 26$, and $\mu_8^{(2)} = 35$. We also choose $N_j = 100$, $c_j = 1$, $\psi^{(1)}(n_j) = 1.5^{\log n_j}$, $\psi^{(2)}(n_j) = 1.55^{\log n_j}$ and $C_D = 300$ ($j = 1, \dots, 8$).

Tables 3 and 4 show the simulated and approximate cumulative distributions of the high-priority and low-priority

Table 3. The Cumulative Distribution of The High-priority Response Time

Response Time	Simul	Approx	R-Err %
0.02	0.0002	0.0002	0.0000
0.04	0.0214	0.0214	0.0000
0.06	0.1542	0.1528	-0.9079
0.08	0.4085	0.4075	-0.2448
0.10	0.6691	0.6672	-0.2840
0.12	0.8459	0.8450	-0.1064
0.14	0.9385	0.9379	-0.0639
0.16	0.9781	0.9780	-0.0102
0.18	0.9931	0.9929	-0.0201
0.20	0.9980	0.9979	-0.0100
0.22	0.9995	0.9994	-0.0100
0.24	0.9998	0.9998	0.0000
0.26	0.9999	1.0000	0.0100
0.28	1.0000	1.0000	0.0000

Table 4. The Cumulative Distribution of The Low-priority Response Time

Response Time	Simul	Approx	R-Err %
0.2	0.1767	0.1473	-16.6384
0.3	0.4538	0.4396	-3.1291
0.4	0.6982	0.7082	1.4323
0.5	0.8541	0.8719	2.0841
0.6	0.9389	0.9503	1.2142
0.7	0.9774	0.9824	0.5116
0.8	0.9925	0.9942	0.1713
0.9	0.9978	0.9982	0.0401
1	0.9993	0.9995	0.0200
1.1	0.9998	0.9999	0.0100
1.2	1.0000	1.0000	0.0000

response times respectively. It appears that the results obtained by Algorithm 1 are very accurate. It is shown in Table 5 that the optimal number of servers is required for 97.5% of the high-priority response time to be less than $T_D^{(1)} = 0.16$ and for 97.5% of the low-priority response time to be less than $T_D^{(2)} = 0.7$. Moreover, the optimal number of servers is required for satisfying both the high-priority and the low-priority response times is shown in Table 5. The exact optimal number of servers, obtained by exhaustive search using the simulation model, and assuming that each station has balanced utilization, is consistent with the ones shown in Table 5. So, $I = 214 < C_D$. We point out that the relative errors shown in Tables 3 and 4 are only due to the Class I error since the Class II error is not present for this model.

Let us now consider an example of the service model 2 shown in Figure 3. We choose $m = 8$, $\Lambda^{(1)} = 55$, $\Lambda^{(2)} = 42$, $\mu_1^{(1)} = 12$, $\mu_2^{(1)} = 46$, $\mu_3^{(1)} = 95$, $\mu_4^{(1)} = 25$, $\mu_5^{(1)} = 35$, $\mu_6^{(1)} = 20$, $\mu_7^{(1)} = 10$, and $\mu_8^{(1)} = 98$, $\mu_1^{(2)} = 15$, $\mu_2^{(2)} = 42$, $\mu_3^{(2)} = 90$, $\mu_4^{(2)} = 18$, $\mu_5^{(2)} = 28$, $\mu_6^{(2)} = 15$, $\mu_7^{(2)} = 5$,

Table 5. The Optimal Number of Servers

Station	1	2	3	4	5	6	7	8
High-priority Customer	12	62	5	23	12	38	29	18
Low-priority Customer	12	61	7	27	13	46	26	16
All the Customers	12	62	7	27	13	46	29	18

and $\mu_8^{(2)} = 82$, and $\alpha = 0.67$. Let us also select $N_j = 250$, $c_j = 1$, $\psi^{(1)}(n_j) = 1.5^{\log n_j}$, $\psi^{(2)}(n_j) = 1.55^{\log n_j}$, and $C_D = 800$ ($j = 1, \dots, 8$). Thus it follows from equation: $\lambda^{(r)} = \frac{\Lambda^{(r)}}{\alpha}$ ($r = 1, 2$) that $\lambda^{(1)} = 82.09$ and $\lambda^{(2)} = 62.69$.

We obtained the cumulative distribution of the response time by solving (35) and (36) using the package of the inverse Laplace transforms in Graf [9]. Table 6 shows the number of servers in the eight stations necessary to ensure the 95% SLA guarantee for $T_D^{(1)} = 0.3$ and $T_D^{(2)} = 1.0$ respectively, and the number of servers in the eight stations necessary to ensure all the customers. We also sim-

Table 6. The Optimal Number of Servers

Station	1	2	3	4	5	6	7	8
High-priority Customer	123	13	4	35	20	52	168	4
Low-priority Customer	61	12	4	46	23	61	342	5
All the Customers	123	13	4	46	23	61	342	5

ulated the tandem queueing network and validated using the brute-force approach that these numbers of servers obtained by our approximate method are in fact optimal, provided that each station has balanced utilization. The optimal number of servers is given in Table 6. It derives that $I = 617 < C_D$, i.e., Step 4 in Algorithm 2 is met. Tables 7 and 8 gives the cumulative distributions of the high-priority and low-priority response times obtained using the approximate method and the simulation method, and their relative error %. The relative error comes from both Classes I and II error. We note that our approximate method has a very good accuracy when percentiles are high. Extensive numerical results (not reported here due to lack of space) point to the fact that the independence assumption has little impact on the accuracy of the results when the number of nodes is large. A contributing factor is that typically we are interested in values of the cumulative distribution of the response time that correspond to very high percentiles for which the approximate results seem to have a very good accuracy.

6 Conclusions

We proposed an approach for resource optimization in a service provider's computing environment, whereby we minimize the total cost of computer resources allocated to a priority-class customer so that satisfies a given percentile of

Table 7. The Cumulative Distribution of The High-priority Response Time

Response Time	Simul	Approx	R-Err %
0.1	0.3879	0.3850	-0.7476
0.2	0.8343	0.8332	-0.1318
0.3	0.9528	0.9547	0.1994
0.4	0.9867	0.9877	0.1013
0.5	0.9961	0.9966	0.0502
0.6	0.9989	0.9991	0.0200
0.7	0.9997	0.9998	0.0100
0.8	0.9999	0.9999	0.0000
0.9	1.0000	1.0000	0.0000

the response time for each class customer. We have formulated the resource optimization problem as an optimization subject to SLA constraints for a service model with or without feedback. We have derived the LSTs of a customer's high-priority and low-priority response times. We further developed an efficient and accurate numerical solution for inverting the LSTs of a high-priority and low-priority customer's response time numerically. In the resource optimization problem we are typically interested in values of the cumulative distribution of the response time that correspond to very high percentiles. Validation testes showed that our approach has a very good accuracy in this case.

7 Acknowledgments

The authors would like to thank Professor Urs Graf for his help with the software for the numerical inversion of a Laplace transform, presented in his book [9], and Professor Tayfur Altioek for his help with Arena. Special thanks also to anonymous reviewers for their useful comments.

References

[1] I. Aib, N. Agoulmine, and G. Pujolle. The generalized service level agreement model and its application to the SLA driven management of wireless environments. In *International Arab Conference on Information Technology*, ACIT, December 2004.

[2] T. Altioek and B. Melamed. *Simulation Modeling and Analysis with Arena*. Cyber Research, Inc. and Enterprise Technology Solutions, Inc., 2001.

[3] E. Bouillet, D. Mitra, and K. Ramakrishnan. The structure and management of service level agreements in networks. *IEEE Journal on Selected Areas in Communications*, 20(4):691–699, 2002.

[4] J. Chang. Processor performance, Update 1. In <http://www.sql-server-performance.com>, 2005.

[5] C. Chassot, F. Garcia, G. Auriol, A. Lozes, E. Lochin, and P. Anelli. Performance analysis for an IP differentiated services network. In *Proceedings of IEEE International Conference on Communication (ICC'02)*, pages 976–980, 2002.

Table 8. The Cumulative Distribution of The Low-priority Response Time

Response Time	Simul	Approx	R-Err %
0.2	0.1679	0.1467	-12.6266
0.4	0.6144	0.6116	-0.4557
0.6	0.8219	0.8207	-0.1460
0.8	0.9126	0.9177	0.5588
1.0	0.9567	0.9622	0.5749
1.2	0.9784	0.9826	0.4293
1.4	0.9889	0.9920	0.3135
1.6	0.9943	0.9963	0.2011
1.8	0.9971	0.9978	0.0702
2.0	0.9985	0.9992	0.0701
2.2	0.9993	0.9996	0.0300
2.4	0.9996	0.9998	0.0200
2.6	0.9998	0.9999	0.0100
2.8	0.9999	1.0000	0.0100
3.0	0.9999	1.0000	0.0100
3.2	1.0000	1.0000	0.0000

[6] J. Cohen. *The Single Server Queue*. North-Holland, Amsterdam, New York, Oxford, 1982.

[7] H. Daduna. Burke's theorem on passage times in Gordon-Newell networks. *Adv. Appl. Prob.*, 16, 1984.

[8] D. Gaver. Observing stochastic processes, and approximate transform inversion. *Operation Research*, 14(3), 1966.

[9] U. Graf. *Applied Laplace Transforms and z-Transforms for Scientists and Engineers*. Birkhauser Verlag, Basel-Boston-Berlin, 2004.

[10] P. Harrison and W. Knottenbelt. Passage time distributions in large Markov chains. In *ACM SIGMETRICS*, 2002.

[11] J. Martin and A. Nilsson. On service level agreements for IP networks. In *IEEE INFOCOM '02*, June 2002.

[12] D. Menasce and E. Casalicchio. A framework for resource allocation in grid computing. In *Proceedings of the 12th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'04)*, pages 259–267. IEEE, October 2004.

[13] J. Muppala, K. Trivedi, V. Mainkar, and V. Kulkarni. Numerical computation of response time distributions using stochastic reward nets. *Annals of Oper. Res.*, 48, 1994.

[14] T. Osogami, A. Wierman, M. Harchol-Balter, and A. Scheller-Wolf. How many servers are best in a dual-priority FCFS system? In *CMU Technical Report: CMU-CS-03-201, November, 2003*. Carnegie Mellon University, November 2003.

[15] H. Perros. *Queueing Network with Blocking, Exact and Approximate Solutions*. Oxford University Press, 1994.

[16] R. Piessens. Gaussian quadrature formulas for the numerical integration of Bromwich's integral and the inversion of the Laplace transform. *Journal of Engineering Mathematics*, 5(1), 1971.

[17] H. Stehfest. Algorithm 386, numerical inversion of Laplace transforms. *Communications of the ACM*, 13(1), January 1970.

[18] J. Walrand and P. Varaiya. Sojourn times and the overtaking condition in Jacksonian networks. *Adv. Appl. Prob.*, 12, 1980.