ABSTRACT

Title of Dissertation: PERFORMANCE ANALYSIS OF A MULTI-CLASS, PREEMPTIVE PRIORITY CALL CENTER WITH TIME-VARYING ARRIVALS

Ahmad Ridley, Doctor of Philosophy, 2004

Dissertation directed by: Professor Michael Fu
Applied Mathematics and Scientific Computation Program

We model a call center as a an $M_t/M/n$, preemptive-resume priority queue with time-varying arrival rates and two priority classes of customers. The low priority customers have a dynamic priority where they become high priority if their waiting time exceeds a given service-level time. The performance of the call center is estimated by the mean number in the system and mean virtual waiting time for both classes of customers. We discuss some analytical methods of measuring the performance of call center models, such as Laplace transforms. We also propose a more-robust fluid approximations method to model a call center.

The accuracy of the performance measures from the fluid approximation method depend on an asymptotic scheme developed by Halfin and Whitt. Here,

the offered load and number of servers are scaled by the same factor, which maintains a constant system utilization. The fluid approximations provide estimates for the mean number in system and mean virtual waiting time. The approximations are solutions of a system of nonlinear differential equations.

We analyze the accuracy of the fluid approximations through a comparison with a discrete-event simulation of a call center. We show that for a large enough scale factor, the estimates of the performance measures derived from the fluid approximations method are relatively close to those from the discrete-event simulation. Finally, we demonstrate that these approximations remain relatively close to the simulation estimates as the system state varies between under-loaded and over-loaded status.

PERFORMANCE ANALYSIS OF A MULTI-CLASS, PREEMPTIVE PRIORITY CALL CENTER WITH TIME-VARYING ARRIVALS

by

Ahmad Ridley

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2004

Advisory Committee:

Professor Michael Fu, Chairman/Advisor
Professor William A. Massey, Co-Chairman/Co-Advisor
Professor Jeffrey Herrmann
Professor John Osborn
Professor Eric Slud

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Call Center Basics

1.1 Introduction

Call centers have become the primary channel of customer interactions, sales, and service for many businesses. Traditional call center performance modelling is based on simple Markovian queueing models, developed to analyze telephone traffic across the Public Switched Telephone Network (PSTN). Closed-form solutions for most of these queueing models are only available for steady-state behavior. Thus, these solutions are not applicable to practical call centers because of the time-varying, or transient, behavior of the arrival call process. In addition, these traditional models become problematic as call centers progress from handling only voice calls to handling multiple types of calls, such as voice, e-mail, faxes, and Web chat sessions. In other words, they do not accurately analyze the performance of modern, multimedia call centers.

To better measure the performance of multimedia call centers over time, we develop mathematical fluid approximations instead of using simple Markovian queueing models. We model a multimedia call center as a preemptive-resume priority queue with time-varying arrival rates and two priority classes of cus-

tomers. The high priority customer class consists of regular telephone, or voice, calls, while the low priority customer class contains e-mail calls. The low priority calls have a dynamic priority where they are upgraded to high priority status based on their service level. Usually, this service level is defined as the probability that the waiting-time in queue is less than a given time duration, although sometimes it is defined as the probability that the mean waiting-time is less than a given duration.

The call center performance measured by our fluid approximation is the mean number of calls in the system and the mean virtual waiting time for each customer class. Our preemptive-resume, time-varying model cannot be easily solved with traditional Markovian queueing techniques. The fluid approximations are computed using an asymptotic scheme where the ratio of the offered load to the number of servers remains constant. The mean number in system for both customer classes is a solution to a system of differential equations. We investigate the effectiveness of the fluid approximations through a comparison with the stochastic, discrete-event simulation method and measure the difference between the mean number in system computed using both methods. We also discuss our results and describe our future efforts for computing the mean virtual delay for both customer classes.

1.2 Overview

Traditionally, customers contacted a call center by talking to a customer service representative (CSR), or agent, over the telephone. Now, customers can contact an agent over the Internet, either by e-mail or chat session. Many companies use call centers, such as banks, financial institutions, information technology (IT)

help desks, and government agencies. The growth of call centers has been substantial over the last two decades. According to industry estimates, there were $69,500$ call centers in the United States. That number is expected to grow to approximately $78,000$ by the end of 2003 [15]. The industry is expected to have an annual growth rate of twenty (20) percent over the next few years [19]. These numbers represent explosive growth over the numbers from the late $1970s$ [39]. Also, 4.5 million people worked in North American call centers in 1995, and over 10 million will have worked in call centers by 2004 [39]. Currently, 70 percent of all business transactions are done over the telephone. The managers of these call centers attempt to provide their customers with efficient and convenient service. However, their job is much more difficult today, because there are far more products and services being sold and supported than a few years ago. Thus, the managers struggle to deliver different service levels to different types of customers with different needs and issues.

1.2.1 Technical Components

A traditional call center has several main components, namely, an automatic call distributor (ACD), an interactive voice response unit (IVR), desktop computers, and telephones. The ACD is a telephone switch located at a customer's premises and provides methods for the distribution of customer calls [8]. There are a finite number of trunks (i.e., telephone lines) connecting the ACD to the PSTN. However, a large ACD switch can connect approximately $30,000$ lines physically to the PTSN, and process roughly $250,000$ calls per hour [5]. As customer calls arrive, the ACD receives and routes them either to the IVR where customer transactions are handled automatically, or to an idle CSR, who provides the

necessary service. If no CSR is available, the calls are placed in a queue (i.e., on hold). The CSR responds to the calls routed to them using their telephone and desktop computer. For example, if the agent is answering a telephone call, that agent can access the customer information database through the desktop computer. The heart of a traditional call center is this dynamic routing of a new or pending call by the ACD to the most appropriate and available CSR. This call routing or assignment process must take into consideration such factors as the call priority, call arrival time, and CSR skills and availability. It requires a dynamic, real-time management of all CSR skill levels and availability, the call/caller identity and status, and customer information databases. Therefore, the flow of an arriving call through a call center can be complex.

Many managers of established, or traditional, call centers enhance their existing infrastructure by enabling Web integration, instead of implementing all-Internet call centers, where all customer interaction occurs over the Internet [9]. Thus, a call center owner typically provides bandwidth access to the Internet and installs an Internet call manager application. Also, the owner typically adds software to existing ACD systems, CTI applications, and agent stations. Finally, a voice over Internet (VOIP) gateway device is connected to the ACD to allow the call center to handle incoming voice calls over the Internet.

We provide a diagram of a Web-enabled call center in Figure 1.1.

1.2.2 Management of a Call Center

A business manager must determine how to improve the performance of a call center to meet an ever-increasing demand. This job involves determining the capacity of the telephone trunk lines that connect the call centers to the customers,

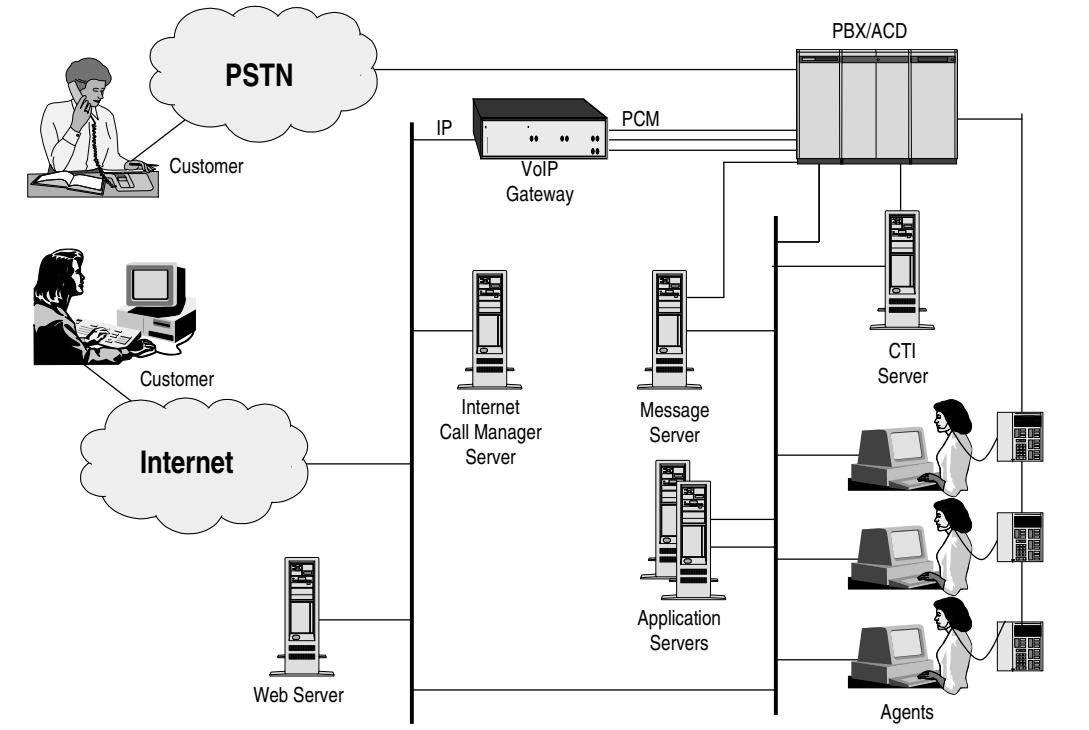

Figure 1.1: Web-Enabled Call Center

and assigning an appropriate number of agents to the call center. Both trunk capacity and agent staffing contribute greatly to the cost of the call center. The manager wants to minimize this cost while controlling the desired blocking probability (i.e., the probability a customer receives a busy signal) and improving the customer response times. Thus, the proper sizing of the call center becomes a non-trivial task and is critical to the success of the business operation.

Multimedia customer service capability is also critical to the success of the today's call center business operation. Business managers must now account for different types of interactions between customers and call agents (i.e., CSRs) besides standard telephone calls. These different types of interactions are mainly Web-enabled customer services. The rapid growth of e-commerce, which provides detailed and timely customer information, is spurring the development of

Web-enabled call centers. The Web-based technology that has made this development possible includes instant messaging, e-mail, faxes, and click-to-call links, which are Web-site buttons that generate agent callbacks over the Public Switched Telephone Network (PSTN) [39]. Although the traditional telephone PBX/ACD switches and the PSTN are still the mainstays of most of today's call center operations, there is a dramatic industry shift in call centers towards including Internet Protocol (IP) networking in support of multimedia customer communications. Thus, Web-enabled call centers will not only handle calls from the telephone network, but also traffic from the Internet. For example, customers can access a business website through information retrieval, business transaction data entry, or an e-mail exchange and simultaneously converse with a CSR over a voice telephone conversation. Eventually, traditional call centers will evolve into purely Web-based multimedia call centers, where all customer interactions will occur strictly over the Internet (i.e., no calls will use the PSTN) [8].

The advancement in call center technologies provides more benefits, but also more challenges. For example, current technologies provide managers greater flexibility in routing and queueing calls by prioritizing certain types of incoming calls and allowing customers to access call agents with different skill sets. The manager's job of scheduling agents and satisfying multiple customer service levels therefore becomes more complex.

1.2.3 Operation of a Call Center

Multimedia, or Web-enabled, call centers operate somewhat differently than traditional call centers. Here, an agent can handle all call types (voice, e-mail, or fax, for instance), two call types, or only one call type. Thus, agents can have

multiple skills or only one skill. When different types of calls arrive at the call center, they wait for service, or queue, at different places. For example, voice calls made over the Internet or the telephone network queue at the ACD, while customer e-mails queue at the e-mail server. Usually, telephone, or voice, calls have the highest priority in the call center. If an agent has a choice between responding to a voice call and e-mail, or voice call and fax, then the agent will answer the voice call first. E-mails have the next highest priority, and faxes have the lowest priority. E-mails arrive and queue at the e-mail server. When there is no telephone call in the call center, any e-mail, arriving or in queue, will be serviced by the next appropriate agent. However, faxes can arrive at the fax server over the Internet, or at the ACD over the telephone network. The faxes at the ACD are directed to a fax machine. Thus, faxes can queue at the fax server or the fax machine. When there is no telephone call or e-mail in the call center, any fax in the system will be handled by the next appropriate agent.

Since telephone calls have the highest priority, these calls are allowed to interrupt any other call type receiving service from an agent. For example, if an agent is responding to an e-mail and the telephone rings on his/her desk, then the agent will stop working on the e-mail and answer the telephone call. Once the agent has finished with the voice call and no other voice call arrives, then he/she will finish responding to the e-mail. E-mails will be allowed to interrupt faxes in a similar manner.

Besides this priority service discipline, the voice calls have another important characteristic. The voice calls will wait in queue for only a certain period of time before abandoning the system, i.e., customers calling over the telephone will get impatient and leave the system. Some customers will call again (i.e., retry for

service) after some additional time. Thus, voice calls have some probability of abandoning the system while in queue.

The key to operating these multimedia call centers effectively is computer telephony integration (CTI). Computer telephony integration is a broad technology aimed at improving telephone call handling activities by using intelligent computer information systems [14]. CTI technology is used to selectively route voice calls to automated, self-service application processes (such as the IVR system) or to call agents. This technology provides a business with an opportunity to improve the efficiency of its customer-relationships. CTI functions allow dynamic information about incoming and outbound calls to be linked in real-time with business applications and database information. For example, CTI-based strategies already assist traditional ACD technology with the accurate reporting of expected waiting times to a telephone caller in queue and effective switching of callers in queue to the IVR [14]. Also, using CTI, an agent can automatically access, almost instantaneously, a customer's file in the companys database, instead of searching for a paper file in a central archive. For example, suppose a customer calls from a telephone help-desk for technical support. The customer can usually be automatically identified by the ACD, using ANI (Automatic Number Identification). The information from the customer's file, which may be relevant to this specific request, is then displayed on the agents computer screen. This information may also provide the agent with tips on supporting the customer's request. After identifying the customers need, the agent could almost respond instantaneously with an automatic e-mail or fax that resolves the customers problem [44]. Thus, with the assistance of CTI technology, an agent can possibly respond much more efficiently to a customer's request.

Now, with the convergence of voice and data technologies, CTI-based strategies have become even more important in the queueing of both Web-based "calls" and telephone calls at the ACD in multimedia call centers. For example, CTI and ANI are used to route different types of calls (such as phone calls, e-mails, and faxes) to appropriately skilled agents. Therefore, CTI enables faster and more effective responses for all call types, reduces CSR call handling time, and minimizes call handling errors, each of which is an important task.

Therefore, the operation and management of call centers have become more complex. As customers interact in more ways with agents than just the telephone, call handling tasks have become more difficult to control. As the number of call centers continues to rise, businesses must determine efficient methods to improve system performance.

1.3 Research Contributions

Fluid approximations have been used by many researchers to model queueing systems. Newell [55] developed fluid and diffusion approximations to estimate queue lengths and the mean waiting-time for customers in non-stationary queues. Also, Halachmi and Franta [26] used fluid and diffusion heuristic approaches to compute the mean waiting-time of customers. Recently, Mandelbaum et al. [51] derived fluid approximations to estimate the queue length and virtual waiting-time for time-varying queues with abandonment and retrials under an asymptotic scheme. However, their model assumes only a single class of customers and a first-come-first-serve (FCFS) discipline. Although they expanded their model to handle customer priorities, they only approximate the queue length, or number in system, process. Additionally, their customer priorities are static, or constant

over time, for the low priority customers.

We make contributions to the previous call center research, mentioned above, in several ways. First, we develop an extension of the fluid model studied by Mandelbaum et al. Unlike their model, our model incorporates two different customer classes with a preemptive-resume priority service discipline. In our model, the low priority customers have dynamic priorities. Thus, at some point in time, we allow these customers to be upgraded to the high priority class. Second, although our model computes the same fluid approximations as those determined by Mandelbaum et al., we compute these approximations for two separate priority classes of customers. Third, we develop a low priority algorithm to analyze the flow of low priority customers through our call center model. With our algorithm, we determine the fluid approximations for the mean number in system and mean virtual waiting time for low priority customers. Finally, we give further evidence of the usefulness of these fluid approximations for modelling call centers. By comparing the approximations with performance estimates from a discrete-event simulation model, we show that our fluid approximations are accurate estimates of the system performance measures. Also, our model provides much more scalable approximations than those from the discrete-event simulation of a call center. Specifically, the complexity of our fluid model does not increase as the size (i.e., number of agents/staff) of the call center substantially increases, whereas the computational burden, in terms of the number of events tracked and run-time, of a discrete-event simulation will increase proportionally.

Chapter 2

Literature Review

2.1 Overview of Call Centers

Call centers, or their modern-day equivalent, contact centers, are the preferred and prevalent way for many companies to communicate with their customers. The percentage of U. S. workers who are employed by call centers is approximately three (3) percent, or roughly 1.55 million agents. A call center workspace usually consists of a large room of agents stationed in cubicles, with a computer and telephone in each cubicle. In some of the largest, best-practice call centers, agents handle thousands of calls per hour, customers rarely abandon while waiting for service, and about half of the calls are answered immediately [19]. Call centers that operate at such high levels of agent utilization and customer service levels rely on sound scientific principles for management and design. In fact, many call centers use some level of mathematical analysis, from classical Erlang approximations to a wide-range of heuristic algorithms, to model their operations.

Call center managers have increasingly relied on scientific research on call centers to effectively design their operations [19]. This research includes analysis of

call forecasting, optimal staffing levels, infrastructure planning (i.e., number and type of ACDs and circuits), and workforce management. For example, Pinedo et al. [58] gives the basics of call center management in the financial and other industries. Anupindi and Smythe [3] describe computer and equipment technology that will enable future call centers, and Duxbury et al. [17] examine standard techniques used in agent-customer interactions and their possibly evolution. Also, Brigandi et al. [12] use a discrete-event simulation model to design and evaluate a network of call centers. Finally, Gans, Koole, and Mandelbaum [19] provide a comprehensive overview of the research areas related to call centers.

We provide a summary of some of the queueing theory-related research used to analyze the performance of call centers. We discuss research related to applying queueing models to call centers, the types of distributions used for call center data, fluid and diffusion models of call centers, computational methods for the waiting time distributions and their inversion, and staffing levels for call centers.

2.1.1 Queueing Models for Call Centers

Simple call centers are a natural application for queueing models based on their operational structure. In a queueing model of a call center, the customers are calls, servers (i.e., resources) are telephone agents or communication equipment, and the queues consist of callers that await service from a system resource. A Markovian queueing model is represented symbolically as $M/M/N/L$. The first M identifies the arrival process as a stationary Poisson process, where the inter-arrival times of customers, or calls, are exponentially distributed with a mean constant call rate. (Note that M_t identifies a non-stationary Poisson process, where the arrival call rates vary over time.) If M were replaced by GI, then the

inter-arrival times would have a general (i.e. any) distribution with independent observations. The second M identifies the service times of the calls as exponentially distributed random variables. If this M were replaced with a G, then the service times would have a general distribution. The N represents the number of servers, or call agents, at the queue. Finally, the L represents the number of spaces available in the system, i.e., the total number of servers and queue spaces. In call center terminology, this value L is known as the total number of trunk lines available to calls.

The simplest and most-widely used call center model is the $M/M/n$ queue, also known as the Erlang C queue [19]. For most applications, however, Erlang C oversimplifies the real-world problem. For example, it assumes that no customers are blocked from the system (i.e., no busy signals) and that customers do not abandonment or retry for service. But the modern call center is often a much more complicated queueing network. Brandt et al. [11] discusses why call centers that allow customers to access an IVR, prior to joining an agents queue, should be modelled as two queues in tandem. In many systems, the customer's time spent at the IVR can be negligible compared to their time spent with an agent, in which case the two queue model can be simplified to one. Garnett and Mandelbaum [20] and Bhulai and Koole [20] use models incorporating multiple groups of agents with varying skill levels and exhibit the increase in the complexity of their models. In additiona, the Erlang C model becomes insufficient when geographically dispersed groups of agents over multiple interconnected call centers are used as discussed in Kogan et al. [43]. Erlang C does not provide good performance estimates for the time-varying arrival and service rate models employed by Mandelbaum et al. [53], or for the multiple class of customer models

discussed in the research of Aksin and Harker [1] and Armony and Maglaras [4].

In both the Erlang B and Erlang C models, the arrival of the calls to the call center are modelled as a stationary Poisson process. The Poisson process is a process from a broader class of stochastic processes known as counting processes, which count the cumulative number of random events that have occurred up to some point in time. A counting process, $N(t)$, has the following properties:

1. $N = \{N(t) : t \geq 0\}$ and takes values in $S = \{0, 1, 2, \ldots\}$.

2. $N(0) = 0$; if $s < t$ then $N(s) \leq N(t)$

A counting process is a *Poisson process* with rate λ if [25]:

1. The process has independent increments, meaning that the numbers of events in any pair of disjoint time intervals are statistically independent.

2. The process has stationary increments, meaning that the distribution of the number of events in any time interval depends only on the length of the time interval and not on when the interval occurred.
3. $$P\left(N(t+h) = n + m \mid N(t) = n\right) = \begin{cases} \lambda h + o(h) & \text{if} \quad m = 1; \\ o(h) & \text{if} \quad m > 1; \\ 1 - \lambda h + o(h) & \text{if} \quad m = 0; \end{cases}$$

where h is small and $o(h)$ is a summation of terms of order h^2 and above such that $\lim_{h \to 0} \frac{o(h)}{h} = 0$. Therefore, the Poisson process can be summarized as follows:

- The probability that a customer arrives at any time does not depend on when other customers arrived.

- The probability that a customer arrives within a small interval of time starting at any time does not depend on the current time.
- Customers arrive one at a time.

Finally, the Poisson process has the following properties:

1. The number of events $N(t)$ in any time interval of length t has a Poisson distribution with mean λt, i.e., $P(N(t) = x) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$, $x = 0, 1, 2, \ldots$.
2. The inter-arrival times are independent exponential random variables with mean $\frac{1}{\lambda}$.

The Erlang-B model can be represented as an $M/M/n/n$ queue. Again, $\rho = \frac{\lambda}{\mu \cdot n}$, where the quantity λ/μ is defined as the offered load of the traffic. Whenever n calls are present in the system, a call may be blocked from entering the call center. This blocking probability, β_n, is an important performance measure and is given by the following steady-state formula:

$$\beta_n = P(\text{all } n \text{ servers are busy}) = \frac{\frac{(\frac{\lambda}{\mu})^n}{n!}}{\sum_{k=0}^{n} \frac{(\frac{\lambda}{\mu})^k}{k!}}. \tag{2.1}$$

The above formula is also referred to as the Erlang B, or Erlang Loss formula.

The Erlang-C model can be represented as the $M/M/n/$ queue. There is no probability of blocking incoming calls since there is infinite waiting space. In this model, the probability of waiting in queue (i.e. probability of call delay), or $P(D > 0)$, is important to measure and is given by the following steady-state formula:

$$P(D > 0) = P(\text{at least } n \text{ calls in system}) = \frac{(\frac{(n\rho)^n}{n!})(\frac{1}{1-\rho})}{[\sum_{k=0}^{n-1} \frac{(n\rho)^k}{k!} + (\frac{(n\rho)^n}{n!})(\frac{1}{1-\rho})]}, \tag{2.2}$$

where D is the delay of a customer call. Also, the mean delay, $E[D]$, is given by [41]:

$$E[D] = \frac{(P(D > 0)(e^{-(n-\rho)\mu t}))}{\mu(n-\rho)}. \tag{2.3}$$

Now, the steady-state waiting time distribution is well-known for the $M/M/1$ and $M/M/n$ queues. In both cases, their Laplace transforms are inverted to obtain the following steady-state formulas:

$$P(W \leq x) = W(x) = 1 - \rho e^{-\mu(1-\rho)x},\ \text{x} >= 0,\ \text{for}\ M/M/1\ \text{and}, \tag{2.4}$$

$$P(W \leq x \mid W > 0) = P(W(x)|W > 0) = 1 - e^{-(n\mu-\lambda)x},\ x \geq 0,\ \text{for}\ M/M/n. \tag{2.5}$$

The above formula for the $M/M/n$ Markovian model is used in practice to approximate the number of call agents required to satisfy customer performance at given service levels. Similarly, the formula for the $M/M/n/n$ Markovian model is used to estimate the mean waiting time in queue experienced by customers. Although these models provide valuable insight into the real system, they are often based on the following, limiting assumptions:

1. Every call is of the same type;
2. Every call agent can handle calls equally fast;
3. The inter-arrival rates are always stationary (i.e., they never vary with time); thus, the system can enter steady-state as ρ approaches 1;
4. Calls are queued on a first-come-first-serve basis.

Unfortunately, under these assumptions, the Markovian approximations can sometimes differ significantly from the real-world call center performance measures.

Although queueing theory can be used to model call centers, the existing theory on call center management has a few issues in its applications to real-world problems [19]. First, the majority of research on queueing theory either are not developed for practical problems, or do not provide enough of a practical solution to real-world problems. Second, researchers often do not validate their models by applying them to real-world instances of their problem. Finally, researchers have trouble developing accurate real-world models, because unpredictable human factors, such as abandonments and retrials, need to be incorporated. Accurate empirical data is often difficult to collect for such factors.

2.1.2 Abandonment, Retrials, and Blocking Models

However, there has been some research performed to model the human behavior of customers and agents. Zohar et al. [54] present empirical data and propose dynamic learning models to measure customer abandonment decisions. Also, Kort [45] develops customer opinion and behavior models to assess abandonment, retrials, and complaint behavior. Palm [57] developed the first models for human factors in telephone services in the 1940s [19]. He studied the behavior of people as they made telephone calls. He observed that callers abandoned their call while waiting for a dial tone, while dialing the telephone number, or while waiting for the connection to be completed across the network. Palm and Kort ultimately showed that the time that callers wait for a dial tone can be modelled with the Weibull distribution. Baccelli and Hebuterne [7] showed that the distribution of

the waiting time until a call is completed across the network can be modelled as an Erlang phase-type distribution with three (3) phases.

For call centers, the most common analytical models for performance analysis are the $M/M/n/n$, or Erlang B, and the $M/M/n$, Erlang C queues. Each one has its limitations though. The Erlang B does not allow customers to wait in queue if all servers are busy. Thus, too many customers may receive busy signals, and be blocked from entering the system. Some call center managers provision a large number of telephone lines to reduce the number of blocked customers. However, in queueing models with infinite capacity, such as the Erlang C, customers tend to experience long delays, especially when the number of customers in queue becomes large. These long delays can also increase customer abandonment.

There are some research models that attempt to compensate for the limitations of the Erlang B and C models. Baccelli and Hebuterne [7] show that the $M/M/n/B + G$ queue, where B represents the overall number of lines and $(B \geq n)$, and G is a general distribution for the customer abandonment, is a good model for balancing blocking and delay requirements. Finally, Riordan [62] and Garnett et al. [21] provide mathematical details for an analytically tractable model is the $M/M/n/B + M$, where patience is assumed to be exponentially distributed [19].

2.1.3 Call Center Data

Existing performance models are based on data collected by the ACD, or telephone switch located on the customer premises. The ACD routes calls to agents and captures each calls arrival time, waiting time in the tele-queue, and service duration. Managers use the ACD data to create reports consisting of total

counts and averages over 30 minute periods, and weekly periods for example [19]. However, call centers do not always have sufficient historical data to develop forecasts. Furthermore, certain factors, such as weather conditions, cannot be predicted. However, Jongbloed and Koole [35] offer a possible solution. They develop a method to derive intervals for arrival rates rather than point estimates. Gordon and Fowler [22] also offer a solution to this problem.

Call Arrivals

The arrival process of calls to a call center is a random process, where customers decide to call independently of each other. There is a small probability that each customer will call during a short period of time, i.e., a 1 minute interval of time. Also, there is a potentially large number of statistically identical customers of the call center. An arrival process with these properties can be modelled as a Poisson process. If more customers are likely to call at one time as opposed to another, the arrival process would have the properties of a time-inhomogeneous Poisson process. Call center modelers often assume that arrival rates are constant over individual periods of time, such as 30 minute intervals. Thus, the true arrival rate function can be often approximated by a piecewise constant function. Therefore, standard steady-state analysis and, more importantly, well-known analytical queueing formulas for estimates of system performance can be used during each time interval. However, these performance estimates will only be accurate if steady state is achieved relatively fast during these intervals [28]. Finally, the Poisson assumption on the arrival process fails when customers experience frequent busy-signals, i.e., calls are blocked from entering the call center, or retrials occur often, i.e., customer satisfaction is low.

Service Duration

In most queueing theory models of call centers, the service time distribution is assumed to be exponentially distributed. We make this assumption in our call center model. This exponential assumption leads to the application of models that are analytically tractable, with well-known formulas for performance measures.

There exist models that show the exponential assumption for the service times is reasonable. For example, Kort [45] validates that exponential service time distributions are acceptable. Harris et al. [29], who analyze IRS call centers, uses exponentially distributed service times in their model of the large IRS call center for the United States federal government. However, other types of distributions have been used for the service time. For example, Mandelbaum et al. [52] discuss a good fit of the lognormal family to the service times for an banking call center model.

Often, there is a practical need for non-standard service time distributions. First, various aspects of the call center have associated service times, such as the IVR and agents work after a call is completed. Currently, not much is known about the IVR service distributions, although the time a customer spends at the IVR is usually negligible. Also, the call handling time, which the sum of the call's service time and any "after-call" work performed by the agent after a call has been completed, is an important parameter to managers. Harris et al. also show that the after-call work time can be ignored, if it is less than 5 percent of the total call handling time. Thus, the call handling time can be made equivalent to the service time in such cases. Second, management decisions could dramatically affect service duration. For example, agents can artificially inflate

the number of calls served during a day by hanging up on customers before service is satisfactorily completed to meet a incentive programs. Thus, customers delay would be small, but customer service levels would suffer. Next, for call centers with a complex set of services, agents with specialized skills can be grouped together to increase response times. Whitt [70] discusses how such a partition of agent skills can lead to efficient models. Finally, the human behavior of agents can affect service times, or work rates, during different times of the day, week, or month [65].

2.2 Performance Models

Queueing models are used to analyze the performance of a call center. By computing performance measures, such as actual customer service levels and agent utilization, researchers can determine the affect of maintaining target service levels on the efficiency of a call center's operations. Typically, these measures are estimated using functions of the incoming traffic, or calls, and available resources, such as agents and telephone lines.

2.2.1 Single Customer Class, Single-Skill Agents

The simplest and most used performance model is the stationary $M/M/n$ queue. It describes a single-customer class call center with n single-skill agents. The calls arrive randomly as a Poisson process to the queue. The time-period is assumed to be short-enough such that calls arrive at a constant rate. The staffing level and service rates are also assumed constant. The model assumes out busy signals, abandonment, retrials and time-varying conditions. The fluid and diffusion

approximations of Mandelbaum et al. [51] incorporates all of these conditions, except for busy signals. Since they assume an infinite queue capacity, they do not account for the blocking of some arriving calls. These approximations are relatively new and have not been developed much for serious applications [19].

For call center models, the useful approximations typically occur in heavy-traffic, which is usually defined by the offered load converging to 1. In the $M/G/n$ queue, Kleinrock [41] provides the Kingman's classical result for the waiting time being approximately exponential, for a small to moderate number of agents n. However, Halfin and Whitt [27] show that, for large n, the waiting times do not necessarily converge asymptotically to an exponential distribution in the $M/M/n$ queue. Thus, the number of servers, or agent staffing level, representing the largest cost in call center, can greatly influence customer waiting times.

2.2.2 Time-Varying Arrival Rates

More realistic models incorporate time-varying arrival rates, which makes performance analysis more complex. Thus, the arrival process is modelled as an inhomogeneous Poisson process. To measure performance in this setting, Green and Kolesar [24] propose the pointwise stationary approximation. Here, the weighted sums of interval performance measures are taken, using the individual arrival rate for each interval. An alternative way to measure performance is to use the average arrival rate as the input for a model. Green and Kolesar [23] [24] show that this can give extremely bad results, even if the staffing levels are constant.

Sudden significant changes in the arrival rate, and hence offered load, cause stationary methods to be less effective. Borst, Mandelbaum and Reiman [10] study the asymptotic behavior of the minimal required staffing as the load tends

to ∞. Overloading could occur from an external event, such as advertising a telephone number on TV, or opening the call center in the middle of the day [19]. Fluid and diffusion models, as studied by Mandelbaum et al. [50], account for such abrupt changes in the offered load. These results are extended in Mandelbaum, Massey, Reiman, Rider, and Stolyar [51]. Unfortunately, Altman, Jimenez, and Koole [2] argue that these fluid approximations do not work as well in under loaded situations [19]. A numerical way to include non-stationary behavior in the modelling of staffing levels is described in Fu, Marcus and Wang [18]. Finally, Jennings et al. [34] developed heuristic staffing guidelines, in opposition to the pointwise stationary approximation, that give rise to a time-varying square-root staffing principle.

2.2.3 Fluid and Diffusion Approximations

Numerical Integration of ODEs

For their time-varying arrival rate model, Mandelbaum et al. [51] used Euler's method to compute the fluid and diffusion approximations for the mean number in system, mean virtual waiting time, variance of the number in system and virtual waiting time, and their corresponding distributions. Their results compared favorably to results from a simulation of their stochastic service system. The formula for Euler's method is:

$$y_{n+1} = y_n + hf(x_n, y_n) + O(h^2), \tag{2.6}$$

where y_n is the approximate solution of the true solution $y(x)$ at x_n, h is the length of the subinterval $[x_n, x_{n+1})$ step-size, and f is the right-hand side of the differential equation. At each step, the order of the error for the method is $O(h^2)$.

$O(h)$ is a summation of terms of order h and above such that $\lim_{h\to 0} \frac{O(h)}{h} = C$ for some constant term C. However, this method is not as accurate as more sophisticated methods and not always stable. If a more accurate and stable method is needed, then the Runge-Kutta method can be implemented. The classical fourth-order Runge-Kutta formula is the most often used form of Runge-Kutta. Its formula is (see Stoer and Bulirsch [64]):

$$\begin{aligned} k1 &= hf(x_n, y_n), \\ k2 &= hf(x_n + \frac{h}{2}, y_n + \frac{k1}{2}), \\ k3 &= hf(x_n + \frac{h}{2}, y_n + \frac{k2}{2}), \\ k4 &= hf(x_n + h; y_n + k3); \\ y_{n+1} &= y_n + k1 + k2 + k3 + k4 + O(h^5), \end{aligned} \tag{2.7}$$

where the method requires four evaluations of the right-hand side of the differential equation.

Fluid Models

A more realistic arrival process for a call center is a non-stationary Poisson process for which the arrival rate varies over time. More specifically, the counting process, $N(t)$, is a non-stationary Poisson process if [25]:

1. The process has independent increments.
2. $$P\left(N(t+h) = n + m \mid N(t=n) = \begin{cases} \lambda_t h + o(h) & \text{if} \quad m = 1; \\ o(h) & \text{if} \quad m > 1; \\ 1 - \lambda_t h + o(h) & \text{if} \quad m = 0. \end{cases}\right.$$

where λ_t = the arrival rate at time t. The definition is identical to the stationary Poisson process defined in Section 2.1.1, except that the arrival rate, λ_t is now a function of time. The non-stationary Poisson process does not have the property that the inter-arrival times are exponential random variables. However, Hall states that it does have several properties in common with the stationary Poisson process. [28] Some properties are:

1. The number of arrivals over the interval $[a, b]$ is Poisson with mean $E[N(b) - N(a)] = \int_a^b \lambda_t dt = \Lambda(b) - \Lambda(a)$, where $\Lambda(t)$ is the expected number of arrivals between 0 and t.

2. If $N(t)$ is the number of events in $[0, \tau]$, then the unordered event times are defined by $N(t)$ independent random variables with probability distribution $P(T \leq t) = \frac{\Lambda(t)}{\Lambda(\tau)}$, where T is the random variable for the event time.

The last property states that the event times can have any distribution as defined by $\Lambda(t)$. Note that for a stationary Poisson process, this property means that the event times have a conditionally uniform probability distribution on $[0, \tau]$, given $N(t)$.

Non-stationary Poisson processes have two types of variation: random variation and predictable variation [28]. The *predictable* is associated with the function $\Lambda(t)$, which gives the expected number of arrivals as a function of time. The *random* variation is reflected in the exact arrival times of customers. A sample path of the function, $N(t)$, of the exact number of arrivals, $A(t)$, is susceptible to random variation. Thus, $\Lambda(t)$ and $N(t)$ will have somewhat different values over time. Because the number of arrivals in any time interval has a Poisson distribution, the mean, $\Lambda(t)$, must equal its variance. Thus, the coefficient of variation

in A(t), which is the ratio of its standard deviation to its mean, is the following:

$$\mathsf{C}[A(t)] = \frac{\sqrt{variance}}{mean} = \frac{\sqrt{\Lambda(t)}}{\Lambda(t)} = \frac{1}{\sqrt{\Lambda(t)}} \tag{2.8}$$

As shown in Equation (2.8), the larger the value of $\Lambda(t)$, the smaller the random variations between the precise number of arrivals, $A(t)$, and the expected number of arrivals, $\Lambda(t)$.

For busy queueing systems, sometimes these random variations are minor compared to the predictable variations. For example, a busy highway toll plaza might have an average of $8,000$ arrivals per hour. Over a one (1) hour period, there will be $8,000$ customers expected to arrive at the plaza. If the coefficient of variation CV equals $1/\sqrt{8000} = 0.011$, then, since the CV is small, A(t) is assumed to be known with certainty and equal to $\Lambda(t)$, in which case, a non-stationary Poisson arrival pattern can be approximated by a ***deterministic*** model.

Deterministic queueing models are usually classified as ***fluid approximations***. Although customers are discrete, not continuous, quantities, a large number of customers can be approximated by a continuous variable and thus modelled as a fluid [28]. A helpful method of visualizing a fluid queueing model is by imaging water filling and draining from a tub. A faucet fills the tub with water, and a drain empties the water from the tub. As water fills the tub, the tub becomes a queue, and the water becomes the customers entering and leaving the queue. The arrival rate is the rate at which the water flows out of the faucet into the tub. Also, the service rate is the speed at which the water drains from the tub. If the water enters the tub faster than it exits, then its level will rise, equivalent to a queue forming when customers arrive faster than they are served. Finally, if the water is drained faster than it enters, then its level will decrease, until all

the water has left the tub.

The validity of the deterministic approximation depends on the variability of the service and inter-arrival times. For the $M_t/M/1$ queue, random queues will form when $\rho^*(t) < 1$, where $\forall s \varepsilon [0,t)$, $\frac{\rho^*(t) = sup\{\int_s^t \Lambda(r)dr}{\mu \cdot (t-s)\}}$. However, the fluid approximation predicts that queues only form when $\rho^*(t) > 1$. An accurate fluid approximation should account for these random queues.

A queueing system with a non-stationary arrival process, i.e., time-varying arrival rates, will never enter into steady-state. In other words, the probability distribution of performance measures, such as the number of customers in the system, will not converge to a steady-state distribution, where the probability becomes independent of any initial conditions, or transient effects. However, steady-state equations can be used to *approximate* the behavior of the system, particularly if the:

- arrival rate changes slowly, and
- the system operates below capacity, i.e. $\rho^*(t) < 1$

When the conditions above are satisfied, the behavior of a non-stationary queuing system can be modelled with steady-state equations during periods of constant arrival rates, and the system is said to be in ***quasi-steady state***.

Now, as $\rho^*(t)$ increases from a number much smaller than one (1) to a number much greater than one (1), estimating the expected queue length becomes more difficult. For the following values of $\rho^*(t)$, we explain the difficulties (see Hall [28]):

1. $\rho^*(t) \ll 1$: The quasi-steady state model is valid, and provides a good queue length estimate;

2. $\rho^*(t) \leq 1$, (1-$\rho^*(t)$) small: The queue lengths are difficult to predict. The quasi-steady state model is not valid. The deterministic, or fluid, approximation is not valid either because it predicts a queue length of zero (random queues are only predicted for stage 3 as noted above);

3. $\rho^*(t) > 1$: The *growth* of the expected queue length is accurately predicted by the deterministic approximation. Note that the quasi-steady state model not applicable here.

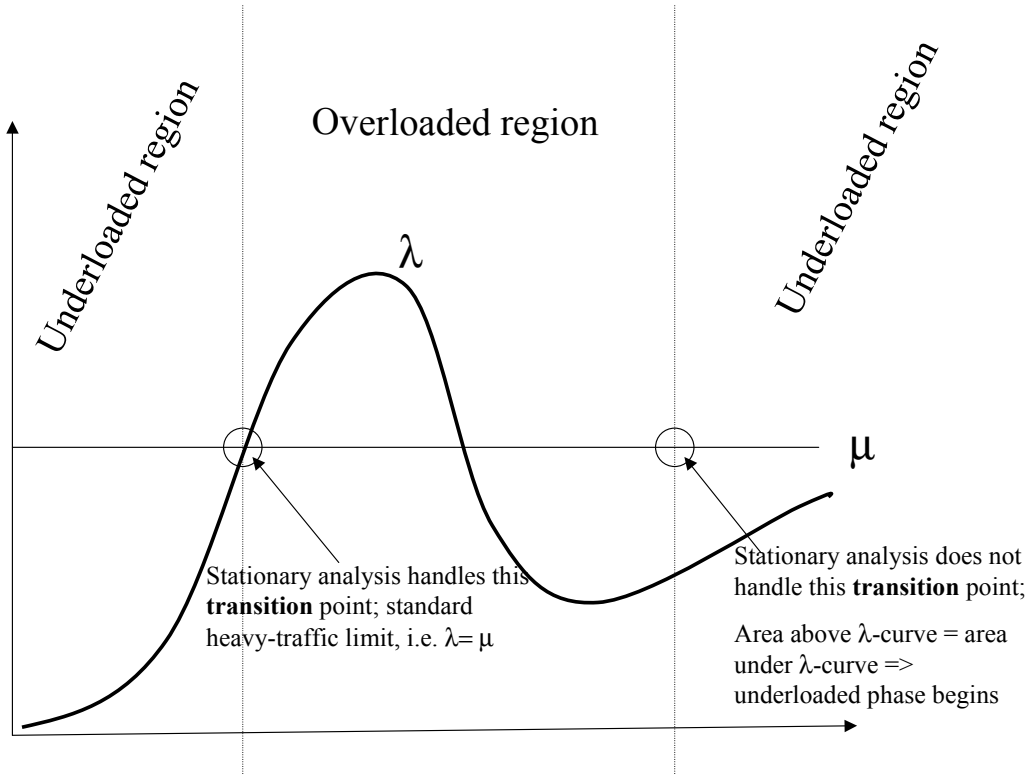

Figure 2.1: Queue Length Phases for Time-Varying Systems

In Figure 2.1, we give a graphical view of the queue length phases.

Finally, in the second stage, there are two ways to estimate the queue length. The first uses a ***diffusion model***, discussed in the next section, and the second uses ***simulation***, discussed in the next chapter.

Diffusion Models

Diffusion models are used in physics to represent the molecular diffusion of fluids, but are also useful in the analysis of the stochastic behavior of non-stationary queueing systems. Diffusion models provide both relatively simple and robust results when an exact analysis of these systems is extremely difficult. As discussed in the previous section, deterministic fluid models can be used to approximate queue behavior. Stochastic diffusion models can also be used. The rate at which a fluid diffuses across a boundary is similar to the transition rate across a boundary line between two states in a transition rate diagram [28].

There are two types of diffusion models. One is the ***diffusion equation***, which is a differential equation first developed for molecular diffusion. Newell [56] examines the derivation of the diffusion equation, and its role in developing non-stationary queueing results. The other is the ***diffusion process*** which is a stochastic process where the time between events are independent, normal random variables. A special case of the diffusion process is Brownian motion. As applied to queueing theory, the fundamental assumption of the diffusion equation is the following (see Hall [28]):

- The arrival and departure processes behave like diffusion processes, and
- The arrival and departure processes are mutually independent, whenever the queue size is positive.

Thus, stochastic processes, including Poisson processes, can be approximated with a diffusion process.

Single Customer Class

Mandelbaum et al. [49], [51] derive fluid and diffusion approximations for the number in system and virtual waiting time for the single customer class, first-come first-serve (FCFS), $M_t/M/n$ queue. Their model incorporates abandonments, retrials, and time-varying arrival rates. The concepts and methods presented by these researchers form the basis for our fluid and diffusion model. Note that in [51], Mandelbaum et al. developed the method for the single customer class, FCFS, $M_t/M/n$ queue with abandonments and retrials. Ultimately, this method will be extended to the two customer class, preemptive-resume priority, $M_t/M/n$ queue with abandonments, which is the call center model of interest. The limit theorem results will yield fluid and diffusion approximations to the virtual waiting-time distribution for both high and low priority customers.

Sample Path Construction

To motivate the sample path construction of the single customer class, FCFS, $M_t/M/n$ queue with abandonments, we give a brief description of this queue without abandonments. Note that for a single-server queue, FCFS is the same service discipline as FIFO, or first-in-first-out. Thus, the first customer arrival will depart the system before the second customer arrival. However, for multi-server queues, FCFS is not always the same as FIFO. In other words, the second customer arrival might depart the system before the first one.

The $M_t/M_t/$n mean number in system process $Q \equiv \{Q(t) \mid t \geq 0\}$ is a continuous-time Markov chain with time-varying instantaneous transition rates. Each customer has a first-come, first-serve service discipline within the class. The arrival process is a time-inhomogeneous Poisson process with rate function

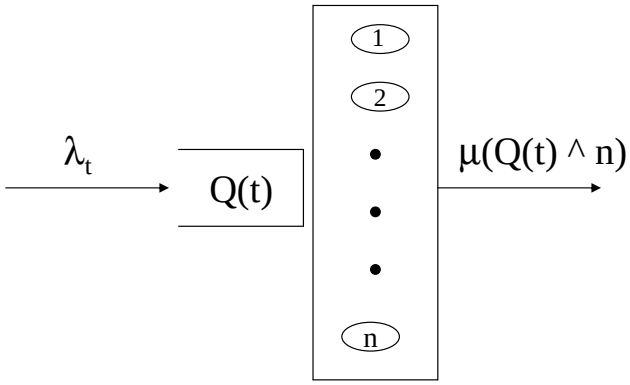

Figure 2.2: The Single-Customer Class $M_t/M/n$ queue

$\{\lambda_i(t) \mid i = 1, 2; t \geq 0\}$, where each $\lambda_i(t)$ is assumed to be locally integrable. The queue has a fixed number of servers, n, where each server has an independent, exponentially distributed service time with rate μ.

We provide a single customer class queue diagram in Figure 2.2. Note that $x \wedge y$ represents the minimum between x and y.

Because there is only one type of customer, the sample path construction reduces to the one-dimensional case. The standard approach to constructing the sample path distribution for this queueing process is to state that its transition probabilities, i.e.,

$$p_{i,j}(t) = \mathsf{P}(Q(t) = j \mid Q(0) = i), \tag{2.9}$$

for all non-negative integers i and j, are the unique solutions to the forward equations:

$$\begin{aligned} \frac{d}{dt}p_{i,0}(t) &= \mu \cdot p_{i,1}(t) - \lambda_t \cdot p_{i,0}(t), \text{ if } j = 0; & (2.10) \\ \frac{d}{dt}p_{i,j}(t) &= \lambda_t \cdot p_{i,j-1}(t) + \mu \cdot \min(j+1, n) \cdot p_{i,j+1}(t) \\ &\quad -(\lambda_t + \mu \min(j, n))p_{i,j}(t), \text{ if } j \geq 1 & (2.11) \end{aligned}$$

where $p_{i,j}(0) = 1 \Leftrightarrow i = j$ and $p_{i,j}(0) = 0$ otherwise. (For more details, see Wolff,

[73].)

The $M_t/M/n$ queueing process is the canonical example for a special family of continuous-time Markov chains (CTMCs) called Markovian service networks [51]. Markovian service networks are discussed in detail by Mandelbaum et al. in [49]. This family can be defined precisely by an alternative method to the computation of the forward equations. Instead, an implicit definition of the transition probabilities can be used to construct the random sample paths directly [51]. The sample paths for the queueing process are the unique set of solutions to the functional equation:

$$Q(t) = Q(0) + \Pi^1\Big(\int_0^t \lambda_s ds\Big) - \Pi^2\Big(\int_0^t \mu \cdot (Q(s) \wedge n) ds\Big), \tag{2.12}$$

where $Q(t)$ is the number of customers in the system (waiting in queue and at the server). Also, $\{\Pi^j(t) \mid t \geq 0, j = 1, 2\}$ are independent, standard (mean rate 1) Poisson processes, and λ_t is an integrable function of time t. Note that $\forall$ real x and y , $x \wedge y \equiv \min(x,y)$.

Similarly, the random sample paths of the number-in-system process, $Q(t)$, for the above $M_t/M/n$ queue, with customers abandoning at a rate of β, are uniquely determined by the following equation:

$$\begin{aligned} Q(t) &= Q(0) + \Pi^1\Big(\int_0^t \lambda_s ds\Big) - \Pi^2\Big(\int_0^t \mu \cdot (Q(s) \wedge n) ds\Big) \\ &\quad -\Pi^3\Big(\int_0^t \beta \cdot (Q(s) - n)^+ ds\Big), \end{aligned} \tag{2.13}$$

where $\{\Pi^j(t) \mid t \geq 0, j = 1, 2, 3\}$ are independent, standard Poisson processes, λ_t is an integrable function of time t, and $x^+ \equiv \max(x, 0)$ [49]. Using the theory of strong approximations for Poisson processes, the above sample path construction can be employed to do an asymptotic sample path analysis. The asymptotic analysis can then be used to obtain the fluid and diffusion limit theorems.

Asymptotic Mean Number in System Results

The asymptotic regime that will be implemented consists of scaling up the number of servers to counter a similar scaling up of the arrival rate of customers. Halfin and Whitt [27] describe several asymptotic regimes, including the one used by Mandelbaum et al. [51]. Specifically, only two parameters will be scaled by an index η, not including the initial conditions for the mean number in system process, $Q(0)$, which will vary with η as $Q^{\eta}(0) = \eta Q^{(0)}(0) + \sqrt{\eta} Q^{(1)}(0) + o(\sqrt{\eta})$ for constants $Q^{(0)}(0)$ and $Q^{(1)}(0)$. These constants will be formally defined shortly. The first scaled parameter is the external arrival rate, λ_t, which is the intensity of the Poisson arrival process. It will be scaled as $\lambda_t^{\eta} = \eta \cdot \lambda_t$. The last scaled parameter is the number of servers, n, which will be scaled as $n^{\eta} = \eta \cdot n$ [51]. Actually, $Q^{\eta}(0)$ and n^{η} should be integer-valued, so their expressions should be denoted as the greatest integer less than or equal to their scaled values. Thus, the scaled number in system process $Q^{\eta}(t)$ is uniquely determined by the relation:

$$\begin{aligned} Q^{\eta}(t) &= Q^{\eta}(0) + \Pi^1\Big(\int_0^t \eta\lambda_s ds\Big) - \Pi^2\Big(\int_0^t \mu \cdot (Q^{\eta}(s) \wedge \eta n) ds\Big) \\ &\quad -\Pi^3\Big(\int_0^t \beta \cdot (Q^{\eta}(s) - \eta n)^+ ds\Big). \end{aligned} \tag{2.14}$$

The results and theorems presented here have been adapted from those stated by Mandelbaum, Massey, and Reiman [51] and Mandelbaum et al. [49]. Their results are based on a similar model that incorporated customer retrials. Thus, customers are allowed to abandon the system completely, or abandon the system, enter a secondary queue, and re-enter the system after a random period of time. The fundamental concepts and theorems used here are discussed in detail in [49] and proven in [51]. Now, for the model of interest, the limit theorem for the functional strong law of large numbers can be stated. The initial conditions for

the mean number in system process satisfy the following asymptotic assumption:

$$\lim_{\eta\to\infty}\frac{1}{\eta}Q^{\eta}(0) = Q^{(0)}(0) \text{ a.s.}, \tag{2.15}$$

where $Q^{(0)}(0)$ is a constant. Thus, the functional strong law of large numbers (FSSLN) limit theorem is:

Theorem 2.1

$$\lim_{\eta\to\infty}\frac{1}{\eta}Q^{\eta} = Q^{(0)}, \ \textit{a.s.}, \tag{2.16}$$

where the convergence is uniform on compact sets of t. Moreover, $Q^{(0)} = \{Q^{(0)}(t) \mid t \geq 0\}$ *is uniquely determined by* $(Q^{(0)}(0)$ *and the differential equation:*

$$\frac{d}{dt}Q^{(0)}(t) = \lambda_t - \mu\cdot(Q^{(0)}(t)\wedge n) - \beta\cdot(Q^{(0)}(t)-n)^{+} \tag{2.17}$$

This theorem states rigorously that $Q^{\eta} \approx \eta Q^{(0)}$ for large η, independent of the Poisson process assumption on $Q(t)$, where $Q^{(0)}$ is called the *fluid approximation* for Q^{η}. The proof of the theorem is given in [49].

However, this fluid approximation can be refined using the functional central limit theorem. In this case, the initial conditions satisfy the following assumption:

$$\lim_{\eta\to\infty}\sqrt{\eta}\left(\frac{1}{\eta}Q^{\eta}(0) - Q^{(0)}\right)\stackrel{d}{=}Q^{(1)}(0), \tag{2.18}$$

where $Q^{(1)}(0)$ is a constant. Before the theorem is stated, some notation must be defined. First, $\lim_{\eta\to\infty} X_n \stackrel{d}{=} Y$ denotes that $\{X_n \mid n \geq 0\}$ converges in distribution to Y. Second, $X \stackrel{d}{=} Y$ implies that the random variables X and Y have the same distribution. Now, the functional central limit theorem is:

Theorem 2.2

$$\lim_{\eta\to\infty}\sqrt{\eta}\left(\frac{1}{\eta}Q^{\eta} - Q^{(0)}\right)\stackrel{d}{=}Q^{(1)}, \tag{2.19}$$

where $Q^{(1)} = \left\{Q^{(1)}(t) \mid t \geq 0\right\}$ *is a diffusion process. This is a convergence in distribution of the stochastic processes in an appropriate functional space [49].*

Moreover, if the set of time points $\left\{t \geq 0 \mid Q^{(0)}(t) = n\right\}$ *has measure zero for the multi-server queue with abandonment model, then* $\left\{Q^{(1)}(t) \mid t \geq 0\right\}$ *is a Gaussian process. The mean for* $Q^{(1)}$ *then solves the differential equation:*

$$\frac{d}{dt}\mathsf{E}\left[Q^{(1)}(t)\right] = -\left(\mu \cdot 1_{\{Q^{(0)}(t)\leq n\}} + \beta \cdot 1_{\{Q^{(0)}(t)\geq n\}}\right)\mathsf{E}\left[Q^{(1)}(t)\right] \tag{2.20}$$

Finally, the variance for $Q^{(1)}$ *solves the differential equation:*

$$\begin{aligned}\frac{d}{dt}\mathsf{Var}\left[Q^{(1)}(t)\right] &= 2\left(\beta \cdot 1_{\{Q^{(0)}(t)\geq n\}} + \mu \cdot 1_{\{Q^{(0)}(t)\leq n\}}\right)\mathsf{Var}\left[Q^{(1)}(t)\right] + \\ &\quad \lambda_t + \beta \cdot \left(Q^{(0)}(t) - n\right)^+ + \mu \cdot \left(Q^{(0)}(t) \wedge n\right).\end{aligned} \tag{2.21}$$

This theorem states rigorously that $Q^\eta \approx \eta Q^{(0)} + \sqrt{\eta}Q^{(1)}$ for large η, where $Q^{(1)}$ is called the *diffusion approximation* for Q^η.

Time-varying queues alternate between three phases. For a given time t, these phases are:

- Underloaded $\Rightarrow Q^{(0)}(t) < n$,
- Critically-loaded $\Rightarrow Q^{(0)}(t) = n$, and
- Overloaded $\Rightarrow Q^{(0)}(t) > n$.

Note that the fluid model for the $M_t/M/n$ queue must be allowed to alternate between the under-loading and overloading phases to guarantee the results of Theorem (2.2). However, if the model "lingers" too long in the critically-loaded phase, then the approximations will be affected by this lingering behavior. Thus, the model should only remain in the critically-loaded phase for values of t, where the set $\left\{t \mid Q^{(0)}(t) = n\right\}$ has measure zero. [51]

Asymptotic Virtual Waiting-Time Results

Once the mean number in system process approximations are determined, the asymptotic results for the virtual waiting-time can be computed. As in the mean number in system process section, the results and theorems presented here have been adapted and summarized from those stated by Mandelbaum, Massey, and Reiman [51] and Mandelbaum et al. [49]. To compute the waiting-time of a virtual customer arriving to the system at a fixed time $\tau_i \geq 0$, $i = 1, 2, \cdots$, an additional assumption is required. After this time τ_i, the original model will be modified as follows:

- There are no new exogenous arrivals into the system after time τ_i.
- In particular, the servers only process any remaining customers in the system at time τ_i.

Theorem 2.1 and Theorem 2.2 still apply to the modified model; however, certain terms in their equations, corresponding to the arrivals after time τ_i, will become zero [51].

The asymptotic results also require some new notation. Denote the arrival and departure processes for the system by:

$$A^\eta = \{A^\eta \mid t \geq 0\}, \text{ and } \Delta^\eta = \{\Delta^\eta \mid t \geq 0\} \tag{2.22}$$

respectively. By convention, let the arrival process include the customers in the system at time 0. So, $A^\eta(0) = \hat{Q}^\eta(0)$, $\Delta^\eta(0) = 0$, and $A^\eta(t) - \Delta^\eta(t) = \hat{Q}^\eta(t), t \geq 0$, where $\hat{Q}^\eta(t)$ is the mean number in system process for the modified queue.

The previous assumptions and notations lead to the following *fluid* limit result:

Theorem 2.3 *As a joint process,*

$$\lim_{\eta\to\infty}\frac{1}{\eta}\left(\hat{Q}^{\eta},A^{\eta},\Delta^{\eta}\right)=\left(\hat{Q}^{(0)},A^{(0)},\Delta^{(0)}\right)\ \ a.s. \tag{2.23}$$

and this convergence is uniform on compact sets of t. The fluid limit $\hat{Q}^{(0)}(t)$ *satisfies equation 7.1 for* $t<\tau$*. For* $t\geq\tau$*, the following properties hold:*

1. *The future evolution of* $\hat{Q}^{(0)}(t)$ *is determined by the differential equation:*

$$\frac{d}{dt}\hat{Q}^{(0)}(t)=-\mu\cdot(\hat{Q}^{(0)}(t)\wedge n)-\beta\cdot(\hat{Q}^{(0)}(t)-n)^{+}. \tag{2.24}$$

2. *There are no future arrivals, so that* $A^{(0)}(t)=A^{(0)}(\tau)$*.*

3. *The deterministic process* $\Delta^{(0)}$ *is a continuously differentiable non-decreasing function in* $[0,\infty]$ *.*

Also, the additional assumption leads to the following *diffusion* limit result:

Theorem 2.4

$$\lim_{\eta\to\infty}\sqrt{\eta}\left(\frac{1}{\eta}\hat{Q}^{\eta}-\hat{Q}^{(0)},\frac{1}{\eta}A^{\eta}-A^{(0)},\frac{1}{\eta}\Delta^{\eta}-\Delta^{(0)}\right)\overset{d}{=}\left(\hat{Q}^{(1)},A^{(1)},\Delta^{(1)}\right). \tag{2.25}$$

Moreover, if the set of time points $\left\{t\geq 0\mid\hat{Q}^{(0)}(t)=n\right\}$ *has measure zero for the multi-server queue with abandonment model, then* $\left\{\hat{Q}^{(1)}(t)\mid t\geq 0\right\}$ *is a Gaussian process. For* $t\geq\tau$*,* $\mathsf{Var}\left[\hat{Q}^{(1)}(t)\right]$ *solves the differential equation:*

$$\begin{aligned}\frac{d}{dt}\mathsf{Var}\left[\hat{Q}^{(1)}(t)\right] &= -2\left(\beta\cdot 1_{\{\hat{Q}^{(0)}(t)\geq n\}}+\mu\cdot 1_{\{\hat{Q}^{(0)}(t)\leq n\}}\right)\mathsf{Var}\left[\hat{Q}(1)(t)\right]+\\ &\quad\beta\cdot\left(\hat{Q}^{(0)}(t)-n\right)^{+}+\mu\cdot\left(\hat{Q}^{(0)}(t)\wedge n\right).\end{aligned} \tag{2.26}$$

It follows from the above theorem and definitions that

$$\hat{Q}^{(1)}(t)=A^{(1)}(t)-\Delta^{(1)}(t). \tag{2.27}$$

Before the asymptotic result for the virtual waiting time distribution can be stated, a few more definitions and assumptions must be given. The *potential service initiation* process D^η for the server is defined as:

$$D^\eta(t) = \Delta^\eta(t) + \eta n,\ t \geq 0. \tag{2.28}$$

Recall that $A^\eta(t) - \Delta^\eta(t) = \hat{Q}^\eta(t), t \geq 0$. So, if $\hat{Q}^\eta(t) < \eta n$, then $A^\eta(t) < D^\eta(t)$. Thus, by Theorem 4.3,

$$\lim_{\eta\to\infty} \frac{1}{\eta} D^\eta(\cdot) = D^{(0)}(\cdot) \text{ a.s.}, \tag{2.29}$$

where the convergence is uniform on compact sets of t and $D^{(0)}(t) = \Delta^{(0)}(t) + n, t \geq 0$. Note that $D^{(0)}(t)$ is continuously differentiable because $\Delta^{(0)}(t)$ is continuously differentiable as the fluid limit of the departure process. Thus, the derivative of $D^{(0)}(t)$ is denoted by $d^{(0)}(t)$. The following assumption for $D^{(0)}(t)$ is important, but not too restrictive for the virtual waiting time result [51]:

$$\lim_{t\to\infty} D^{(0)}(t) > A^{(0)}(\tau), \tag{2.30}$$

where $D^{0)}(t)$ is continuously differentiable with *strictly positive* derivative. Note that, based on previous definitions, $A^{(0)}(\cdot)$ and $A^{(0)}(\tau)$ are constant on the interval $[\tau, \infty)$. Also, it is convenient to assume that all processes are defined on the interval $[-T, \infty)$ where $T = n/d^{(0)}(0)$ instead of $[0, \infty)$. This interval extension assumes that there are no arrivals or departures within the interval $[-T, 0)$.

Now, Theorem 4.3 and Theorem 4.4 can be written in terms of D as follows:

$$\lim_{\eta\to\infty} \frac{1}{\eta} \left(Q^\eta, A^\eta, D^\eta\right) = \left(\hat{Q}^{(0)}, A^{(0)}, D^{(0)}\right) \tag{2.31}$$

and,

$$\lim_{\eta\to\infty} \sqrt{\eta} \left(\frac{1}{\eta}\hat{Q}^\eta - \hat{Q}^{(0)}, \frac{1}{\eta}A^\eta - A^{(0)}, \frac{1}{\eta}D^\eta - D^{(0)}\right) \stackrel{d}{=} \left(\hat{Q}^{(1)}, A^{(1)}, D^{(1)}\right), \tag{2.32}$$

where $D^{(1)} = \Delta^{(1)}$ and $t \geq -T$.

Note that $A^{(0)}, D^{(0)}, A^{(1)}, D^{(1)}$ are continuous and $D^{(0)}(-T) = D^{(1)}(-T) = 0$ [51]. Let the *first attainment* process, $\{S^{(\eta)}(t)\}$, be defined for all $t \geq -T$ as:

$$S^{\eta}(t) = \inf\{s \geq -T : D^{(\eta)}(s) > A^{\eta}(t)\}, \tag{2.33}$$

and,

$$S^{(0)}(t) = \inf\{s \geq -T : D^{(0)}(s) > A^{(0)}(t)\}. \tag{2.34}$$

Similarly, define the *attainment* waiting time process as:

$$W^{\eta}(t) = S^{\eta}(t) - t, \tag{2.35}$$

and,

)

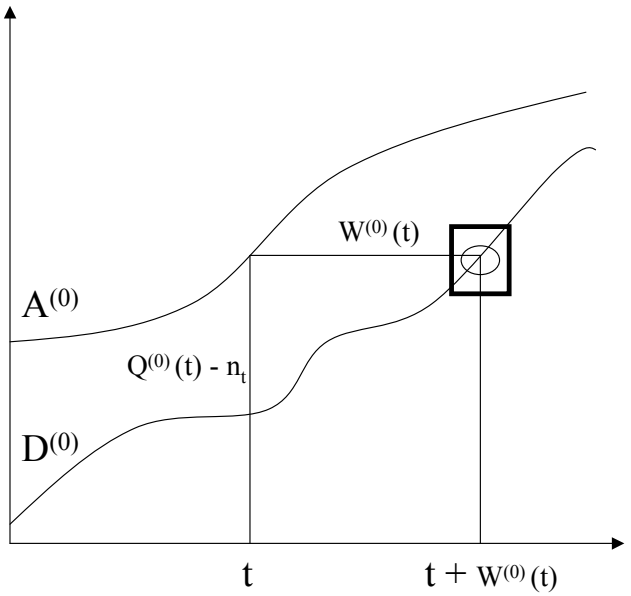

Figure 2.3: Fluid Approximation of Waiting Time

In Figure 2.3, we provide a graphical explanation of the fluid approximation, $W^{(0)}(t)$, to the attainment waiting time.

The conventions and assumption defined above allow the previous processes to be well-defined and finite with probability 1 for sufficiently large η.

Now, define the *virtual* waiting time at τ_i, $\hat{W}^{\eta}(\tau)$, as the time a customer arriving to the queueing service node at time τ_i would have to wait until its service starts, assuming that customer *does not* the queue [51]. Thus, the virtual waiting time, $\hat{W}^{\eta}(\tau)$, and the attainment waiting time, $W^{\eta}(t)$, are related as:

$$\hat{W}^{\eta}(\tau) = W^{\eta}(\tau)^{+}. \tag{2.37}$$

So, if $\hat{Q}^{\eta}(\tau) < \eta n$, then $W^{\eta}(\tau)$ (and $W^{(0)}(\tau)$) will be negative. Therefore, by definition, $\hat{W}^{\eta}(\tau) = 0$. If $W^{\eta}(\tau)$ is non-negative, then $\hat{W}^{\eta}(\tau)$ will have the same value as $W^{\eta}(\tau)$.

The next theorem follows directly from Equation (4.28), Equation (4.29), and the theorem in Puhalskii [60]. Those results yield the following convergence theorem:

Theorem 2.5

$$\lim_{\eta\to\infty} \frac{1}{\eta}\left(\hat{Q}^{\eta}, A^{\eta}, D^{\eta}, W^{\eta}\right) = \left(\hat{Q}^{(0)}, A^{(0)}, D^{(0)}, W^{(0)}\right), \ \textit{a.s.}, \tag{2.38}$$

and,

$$\lim_{\eta\to\infty} \sqrt{\eta}\left(\frac{1}{\eta}\hat{Q}^{\eta} - \hat{Q}^{(0)}, \frac{1}{\eta}A^{\eta} - A^{(0)}, \frac{1}{\eta}D^{\eta} - D^{(0)}, W^{\eta} - W^{(0)}\right) \stackrel{d}{=} \left(\hat{Q}^{(1)}, A^{(1)}, D^{(1)}, W^{(1)}\right), \tag{2.39}$$

where

$$W^{(1)}(t) = \frac{A^{(1)}(t) - D^{(1)}(S^{(0)}(t))}{d^{(0)}(S^{(0)}(t))} \ \textit{and}\ S^{(0)}(t) = \textit{inf}\{s \geq -T : D^{(0)}(s) > A^{(0)}(t)\}. \tag{2.40}$$

In Figure 2.4, we show a magnified view of the small boxed area in Figure 2.3. The figure gives a graphical definition of the attainment waiting time, which was

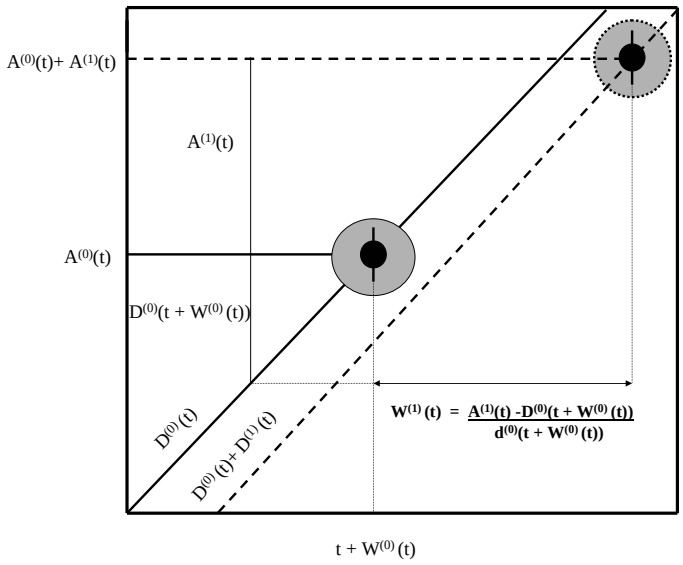

Figure 2.4: Diffusion Approximation of Waiting Time

defined above as the first time after time t that the number of departures equals the number of arrivals. Here, the fluid approximations, $A^{(0)}(t)$ and $D^{(0)}(t)$, to the arrival and queue departure processes are improved by the addition of their corresponding diffusion terms, $A^{(1)}(t)$ and $D^{(1)}(t)$. Thus, the fluid approximation, $W^{(0)}(t)$, to the attainment waiting time must also be adjusted by its diffusion term, $W^{(1)}(t)$. Remember that the virtual waiting time and attainment waiting time are related by the Equation (2.37).

Since the processes $A^{(1)}(t)$, $D^{(1)}(t)$, $\hat{Q}^{(1)}(t)$, $W^{(1)}(t)$ are continuous with probability one, their finite dimensional distributions converge [51]. In particular, consider the non-trivial case $S^{(0)}(\tau) \geq \tau$, which is equivalent to $\hat{Q}^{(0)}(\tau) \geq n$. Moreover, assume that the set of points $\left\{t \mid \hat{Q}^{(0)}(t) = n\right\}$ has measure zero on $[0, \tau]$. Then:

$$\lim_{\eta \to \infty} W^{\eta}(\tau) = W^{(0)}(\tau) \text{ a.s.}$$

and

$$\lim_{\eta\to\infty}\sqrt{\eta}(W^{\eta}(\tau)-W^{(0)}(\tau))\stackrel{d}{=}W^{(1)}(\tau)=\frac{\hat{Q}^{(1)}(S^{(0)}(\tau))}{d^{(0)}(S^{(0)}(\tau))}$$

where $\hat{Q}^{(1)}(S^{(0)}(\tau))$ is a Gaussian process with a mean and variance computed as follows. First, solving equation 4.21 for $\hat{Q}^{(0)}(\cdot)$ in the interval $[\tau,\infty]$ yields:

$$\frac{d}{dt}\hat{Q}^{(0)}(t)=-\beta\hat{Q}^{(0)}(t)+(\beta-\mu)n,\ t\geq\tau.$$

Now, by definition,

$$S^{(0)}(\tau)=\min\left\{t\geq\tau\mid\hat{Q}^{(0)}(t)=n\right\}.$$

Second, the mean, $\mathsf{E}\left[\hat{Q}^{(1)}(S^{(0)}(\tau))\right]$, and variance, $\mathsf{Var}\left[\hat{Q}^{(1)}(S^{(0)}(\tau))\right]$, are computed as the solutions to the following equations:

$$\frac{d}{dt}\mathsf{E}[\hat{Q}^{(1)}(t)]=-\beta\cdot\mathsf{E}[\hat{Q}^{(1)}(t)],\ t\geq\tau. \tag{2.41}$$

and

$$\frac{d}{dt}\mathsf{Var}[\hat{Q}^{(1)}(t)]=-2\beta\mathsf{Var}[\hat{Q}^{(1)}(t)]+\beta(\hat{Q}^{(0)}(t)-n)+\mu n,\ t\geq\tau. \tag{2.42}$$

Observe that since zero is a solution to equation 2.41, the mean can be assumed to be zero. Finally, noting that $d^{(0)}\left(S^{(0)}(\tau)\right)=n\mu$ when $S^{(0)}(\tau)\geq\tau$ gives:

$$\mathsf{Var}\left[W^{(1)}(\tau)\right]=\frac{\mathsf{Var}\left[\hat{Q}^{(1)}(S^{(0)}(\tau))\right]}{(n\mu)^2} \tag{2.43}$$

2.3 Simulation of Queueing Models

A simulation algorithm consists of techniques for using computers to model the operations of real-world processes or facilities, often called systems. In order to analyze a system, certain assumptions are made about how it works. These

assumptions are used to develop mathematical or logical relationships that constitute a model. Sometimes, these relationships are relatively simple. In such cases, mathematical methods, like algebra, calculus, or probability theory, might be used to derive exact, or analytical, solutions. However, most real-world systems are too complex for any model to provide analytical solutions. Thus, a computer simulation must be used to evaluate a model numerically. For example, a simulation of a queueing model for a call center must be used to incorporate many of the real-world characteristics of a call center, such as abandonments, retrials, and time-varying rates, in the performance analysis of the system.

Simulation is one of the most widely used operations research techniques. Lane, Monsour, and Harpell [46] found in their longitudinal study that simulation was consistently ranked as one of the three most important operations research techniques [47]. However, there are some disadvantages to the method [47]. First, the models used to study large-scale systems can be very complex. Also, the task of writing computer programs to execute these models can be difficult, despite the development of excellent simulation software packages. Second, a large amount of computer time is sometimes required to evaluate a model. However, the problem is made less severe by the creation of faster and cheaper computers.

Law and Kelton formally define a system modelled by a simulation as a collection of entities, such as people or equipment, that interact together toward some logical end [47]. In practice, a system is defined based on the objectives of a particular analysis. For example, imagine that the goal of an analysis is to determine the number of checkout lines needed to provide adequate service in a supermarket. Here, the system can be defined as the number of open checkout lines and the number of customers waiting in line or being served. Additionally,

the state of the system is defined to be that collection of variables necessary to describe the system at a particular time, relative to the goals of the analysis [47]. In a supermarket analysis, examples of possible state variables are the number of busy checkout clerks, the number of customers in the store, and the arrival time of each customer in the store.

We categorize systems that are simulated into two types: discrete-event and continuous. A *discrete-event* system consists of state variables which change instantaneously at distinct points in time [47]. A supermarket is an example of a discrete-event system since state variables, i.e. the number of customers in the store, change only when a customer arrives or completes service and departs. However, a *continuous* system consists of state variables which change continuously with respect to time. A train moving along a railroad line is an example of a continuous system, since state variables such as position and velocity can change continuously with respect to time. Note that the discrete-event or continuous classification is not dependent on whether time itself is discrete or continuous, but on the possible values of the state variables.

Discrete-event and continuous simulations can be defined analogously to the way discrete and continuous systems were defined above. Note that a discrete model is not always used to model a discrete system, and vice versa [47]. Rather, the specific objectives of the analysis determines the type of simulation model. For example, a model of traffic flow on a highway can be discrete-event, if the movement of individual cars is important, or continuous, if the cars can be viewed "in the aggregate." Discrete-event simulation deals with modelling a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time [47]. Mathematically speaking, the system

can change at only a countable number of points in time. Events, such as an arrival into or departure out of the system, occur at these points in time. Such events are defined as instantaneous occurrences that may change the state of the system. Since we are interested in the movement of individual customers (i.e, calls), we use discrete-event simulation to model our call center, which is defined in Chapter 3.

2.3.1 Waiting-Time Computational Methods

A manager usually requires that eighty (80%) of the calls are answered within twenty (20) seconds [8]. Thus, the probability that customers wait less than twenty (20) seconds should be at least eighty (80) percent. Mathematically, this is represented, in general, as $P(W \leq t) \geq \alpha$, where t could be twenty seconds, and α could be eighty percent. This quantity is the waiting time distribution and is used to assess the customer service levels received in a call center. Sometimes, this probability is written in terms of the"tail" distribution as $P(W \geq t) \leq 1-\alpha$. In other words, the probability that customers wait more than 20 seconds should be less than 20 percent.

There exists a lot of research on customer waiting-times for different types of queueing systems. Although early researchers focused on computing the mean waiting-time, recent ones have derived methods for approximating the waiting-time distribution. For the $M/G/1$, FCFS queue, the Laplace transform of the waiting-time distribution is determined from the Pollacek-Khinchin $(P-K)$ transform equation for the number of customers in the system. This equation was first published by Khinchin [38] in 1932 and studied by Pollaczek [59] in 1930. Kleinrock [41] and Wolff [73], among others, provide derivations of the

waiting-time distribution Laplace transform using the $P - K$ equations.

Customer waiting-times have been analyzed for the $M/G/n$, FCFS queue as well. Kingman [40] derives bounds on the mean waiting-time. Newell [55] and Halachmi and Franta [26] use diffusion approaches to compute the mean waiting-time. Unlike the previous researchers, Hokstad [31] computes approximate results for the distribution of the number of customers in the system and the distribution of the waiting-time. Whitt [71] uses the Laplace transform of the waiting-time distribution to predict queueing delays of customers before they enter service. In this case, delay information based on the distribution is seen as more insightful than delay information based on the mean. Finally, Kleinrock also derives waiting-time distribution transform for the $G/M/n$, FCFS queue. He shows that the distribution has an asymptotic exponential form. Although his transform is computed for a conditional waiting-time distribution, given that a customer must queue, the unconditional waiting-time distribution can also be computed.

The waiting-time distribution has also been studied under the priority queue discipline. Jaiswal [33] derives the results for the Laplace transform of the waiting-time distribution for the $M/G/n$, non-preemptive priority queue. He first computed the Laplace transform for the busy period containing customers of higher priority than the given customer. Then, the waiting-time distribution transform can be written as a function of the busy period transform. Kleinrock, Wolff, and Takagi [67] also describe similar results to Jaiswal's for the $M/G/n$, non-preemptive priority queue. Takagi inverts the waiting-time distribution for the $M/G/1$ queue. However, his resulting formula involves evaluating an infinite series. Thus, obtaining useful values for the waiting-time distribution in queueing applications can be rather complex.

One of the earliest results for the waiting-time distribution for a priority queue was published by Cobham [13] and Kesten and Runnenburg [37]. Shortly after, Takacs [66] expanded on their techniques and provided a simple method to determine the waiting-time distribution for a customer in any class p. For the stationary $M/G/1$, preemptive and non-preemptive priority, queues, he computed the waiting-distribution as a function of the moments of the service time distribution. Another early result was published by Davis [16]. He gives an explicit formula for the waiting-time distribution for an arbitrary customer in the $M/M/n$, non-preemptive priority queue. He not only derived the Laplace transform, but also inverted this transform using contour integration. However, his subsequent waiting-time distribution formula was somewhat complex, like Takagi's. Thus, computing values of the waiting-time distribution is as easy as other methods, when applied to real-world problems, such as call centers. Williams [72] expands on Hokstad's earlier work on the $M/G/n$, FCFS queue. He computes the waiting-time distribution transform for high and low priority customers in a two-customer, non-preemptive priority setting. Also, for the $M/M/n$, non-preemptive priority queue, Kella and Yechiali [36] derives a waiting time distribution transform result similar to Davis's. They use a less elaborate method that uses the probabilistic equivalence of the waiting times in the $M/G/1$ queue with server vacations to those in their queue setting.

Finally, there are other types of priority queues that have been studied. Wolff derives the waiting-time distribution transform for high and low priority customers in the $M/G/1$, preemptive-resume priority queue. He first computed an ordinary and exceptional-first-service busy period duration transform for his work. Also, Kleinrock [42] derives only the mean waiting-time for $M/G/1$, dy-

namic priority queue. He uses a dynamic priority service discipline that uses a time-dependent priority structure where priorities increase or decrease linearly over time. Finally, Jackson [32] gives further results for other dynamic priority queues. His results were some of the earliest on dynamic priority queues.

2.3.2 Waiting-Time Distribution

The Laplace transform of the waiting-time distribution has been derived for several different types of queues. As discussed previously, this transform depends on the transform of the duration of the busy period. The waiting-time transform will be stated for the general $M/G/n$ queue under the first-come-first-serve (FCFS), non-preemptive priority, and preemptive-resume priority queue discipline. Since our call center model assumes that service time distribution, $\tilde{G}(s)$, is the exponential distribution, the transforms derived can be simplified by substituting the following for $\tilde{G}(s)$:

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

The transform results will be stated first for the single-server case, and then expanded to the multi-server case.

2.3.3 FCFS Queueing Models

For the $M/G/1$, first-come-first-serve, single-customer class case, the waiting-time transform follows directly from the well-known Pollaczek-Khinchin(P-K) transform equation for the number of customers in the system. The $P-K$ transform was first proven by Pollaczek [59] and Khinchin [38]. Next, we give a summary of Kleinrock's [41] derivation of the waiting-time $P-K$ transform equation.

First, let q_n be the number of customers left in the system after the departure of the nth customer. Note that z-transform is defined as $Q(z) = \sum_{k=0}^{\infty} P[q_n = k]z^k = E\left[z^{q_n}\right]$, $0 \leq z \leq 1$. Then, $P-K$ transform for q_n is:

$$Q(z) = \tilde{G}(\lambda - \lambda z)\frac{(1-\rho)(1-z)}{\tilde{G}(\lambda - \lambda z) - z}. \tag{2.44}$$

Now, let v_n be the number of arrivals during the service time, x_n, of the nth customer. The z-transform of v_n is:

$$V(z) = \tilde{G}(\lambda - \lambda z).$$

Also, define s_n as the total time spent in the system by the nth customer, with service time distribution $S(s)$ and z-transform $\tilde{S}(s)$. Note that $s_n = w_n + x_n$ where w_n is the waiting-time of the nth customer. Then Kleinrock [41] shows that v_n, $V(z)$, and $\tilde{G}(z)$ are each equivalent to q_n, $Q(z)$, and $\tilde{S}(s)$. Thus, $Q(z)$ satisfies the same equation as $V(z)$, namely $Q(z) = \tilde{S}(\lambda - \lambda z)$.

Next, by substituting the value for $Q(z)$ from equation 2.44, we have:

$$\tilde{S}(\lambda - \lambda z) = \tilde{G}(\lambda - \lambda z)\frac{(1-\rho)(1-z)}{\tilde{G}(\lambda - \lambda z) - z}. \tag{2.45}$$

By making the change of variable $s = \lambda - \lambda z \Rightarrow z = 1 - \frac{s}{\lambda}$, the above equation becomes:

$$\tilde{S}(s) = \tilde{G}(s)\frac{s(1-\rho)}{s - \lambda + \lambda\tilde{G}(s)}, \tag{2.46}$$

which is the explicit expression for the P-K equation for the Laplace transform of the time-in-system distribution.

Finally, assuming the system is ergodic, we note that s_n, x_n, and w_n have limiting values $\tilde{s}$, $\tilde{x}$, $\tilde{w}$ with probability 1 as the system reaches equilibrium. Since a customer's service time is independent from its waiting time, we also

have that $\tilde{s}$ is the sum of two independent random variables $\tilde{x}$, and $\tilde{w}$. Thus, the Laplace transform of $\tilde{s}$ is:

$$\tilde{S}(s) = \tilde{W}(s)\tilde{G}(s). \tag{2.47}$$

Thus, by substituting the Laplace transform value of $\tilde{S}(s)$ from 2.45, the waiting-time distribution Laplace transform is given by:

$$\tilde{W}(s) = \frac{s(1-\rho)}{s - \lambda + \lambda\tilde{G}(s)}, \tag{2.48}$$

where $\rho = \frac{\lambda}{\mu}$, and $\tilde{G}(s)$ is the service-time distribution Laplace transform. Therefore, we have the $M/G/1$, single customer class, FCFS P-K formula for the waiting time Laplace transform.

2.3.4 Priority Queueing Models

Under complex settings, such as time-varying rates and priority service disciplines, the usual Markovian models might become intractable. In other words, for these models, no closed-form may exist for the mean waiting time or waiting-time distribution. In those cases, the Laplace transform is too complex to invert analytically. However, values for the waiting-time distribution can still be obtained through numerical inversion, as discussed in Chapter 2. Now, for these models, the calls will have different levels of priorities. Although the waiting-time distribution is known under the first-come-first-serve queue discipline, those results are not valid in the priority discipline. Thus, more complex priority service models must be considered. Some analysis of the models under a priority service discipline has been done. However, the expected system performance, such as queue length and waiting-time distributions, have not been determined in closed-form for models beyond the $M/M/1$ queue. For some models, the Laplace

transforms for the queue length and waiting time distributions are known. For example, for the $M/G/n$ non-preemptive priority queue, the Laplace transform for the waiting-time distribution is the following:

$$W_k^*(s) = (1-\pi) + \pi \frac{n\mu(1-\sigma_k)(1-G^*(s))}{s - \lambda_k + \lambda_k G^*(s)}, \tag{2.49}$$

where $W_k^*(s)$ is the waiting-time Laplace transform of the kth class and π is probability all servers are busy. However, similar results for the $M/G/n$, preemptive priority queue are not well-known. Although the distributions can be obtained from inverting the transform, this process is often very complex. Also, these distributions usually can not be written in closed-form, or, if they can, are not practical for deriving probabilities.

Preemptive-Resume Priority Discipline

The transform for multiple-customer class, priority queue scenario is not as well-known as the FCFS, single-class transform. However, some results do exist for the single-server queue. Takacs [66] proved some of the early stationary results for the Laplace transforms of the $M/G/1$, preemptive-resume priority, multi-class queue. He derived the transform, distribution, and moments for the waiting-time of a customer of priority p. Wolff's results are similar to those of Takacs. Wolff [73] derives the transforms for the $M/G/1$, two-class, preemptive-resume priority queue using the exceptional-first-service busy period concept discussed in his book. The high priority, class-1 waiting-time distribution transform is:

$$\tilde{W}_1(s) = \frac{s(1-\rho_1)}{s - \lambda_1 + \lambda_1 \tilde{G}_1(s)}, \tag{2.50}$$

where $\rho_1 = \frac{\lambda_1}{\mu_1}$, λ_1 and μ_1 are the inter-arrival and service rates for class-1 customers, and $\tilde{G}_1(s)$ is the service-time distribution Laplace transform for class-1

customers.

Now, the low priority, class-2 Laplace transform is the product of two transforms, $\tilde{B}_{D_{2f}}(s)$ and $\tilde{B}_{\mathcal{C}}(s)$, where D_{2f} is the delay of a class-2 customer prior to entering service for the first time, and $\mathcal{C}$ is the completion time of a class-2 customer from the time service begins until service is completed. D_{2f} and $\mathcal{C}$ are exceptional first-service busy periods initiated by the arrival of class-1, i.e., high priority, customers. Thus, the class-2 waiting-time distribution transform is:

$$\tilde{W}_2(s) = \tilde{B}_{D_{2f}}(s)\tilde{B}_{\mathcal{C}}(s), \tag{2.51}$$

where

$$\tilde{B}_{D_{2f}}(s) = \frac{\lambda_1\mu_1}{\lambda(\mu_1 + s + \lambda_1 + \lambda_1\tilde{B}(s))} + \frac{\lambda_2\mu_2}{\lambda(\mu_2 + s + \lambda_1 + \lambda_1\tilde{B}(s))},$$

$$\text{and } \tilde{B}_{\mathcal{C}}(s) = \frac{\mu_2}{\mu_2 + s + \lambda_1 + \lambda_1\tilde{B}(s)}.$$

Again, $\tilde{B}(s)$ is the Laplace transform for the distribution of an ordinary busy period duration.

Non-Preemptive Priority Discipline

There are results for the non-preemptive priority, multi-customer class transform as well. Kella and Yechiali [36] computed the waiting time distribution transform of a class-p customer for a $M/G/n$, non-preemptive priority, multi-class queue. They use the probabilistic equivalence between the $M/G/1$ queue with multiple server vacations and the $M/M/n$ queue. As in the preemptive-resume case, Takacs [66] proved some of the early stationary results for the Laplace transforms of the $M/G/n$, non-preemptive priority, multi-class queue. Again, he derived the transform, distribution, and moments for the waiting-time of a customer of

priority p. For the $M/G/m$, non-preemptive priority, multi-class queue, Williams [72] derived the waiting-time distributions for two classes of customers. His results are similar to those of Takacs and Kella and Yechiali. He showed that the high priority, class-1 waiting-time distribution Laplace transform is:

$$\tilde{W}_1(s) = 1 - \Pi + \frac{\lambda_1(1 - \tilde{G}(\frac{s}{n}))(1-\rho_1)\Pi}{\rho_1(s - \lambda_1(1 - \tilde{G}(\frac{s}{n})))}, \tag{2.52}$$

where $\rho_1 = \frac{\lambda_1}{n\mu_1}$ and:

$$\Pi = \left\{1 + (1-\rho)\sum_{k=0}^{n-1}(m\rho)^{k+1-m}(n-1)!/(k!\rho)\right\}.$$

The class-2 transform is somewhat different, but similar in form. It is given by:

$$\tilde{W}_2(s) = 1 - \Pi + \frac{\lambda(1 - \tilde{B}_1(s))(1-\rho)\Pi}{\rho(s - \lambda_2(1 - \tilde{B}_1(s)))}, \tag{2.53}$$

where $\rho = \rho_1 + \rho_2$, $\lambda = \lambda_1 + \lambda_2$, and $\tilde{B}_1(s)$ is the distribution of an ordinary busy period duration initiated by class-1 customers.

2.3.5 Staffing Models

For a single-skill call center, the problem of determining the work hours of each agent involves first determining the shifts, and then assigning the agents to shifts. There are different approaches for determining the shifts. A heuristic approach is advocated in Henderson and Berry [30], while Segal [63] uses a linear programming one. Other aspects, such as shift break placements, are also studied in detail by Aykin [6]. An overview of the area, though not necessary applied to call centers, is discussed by Thompson [68]. However, in practice there are many additional constraints to the above problem. A few of these problems are union regulations on labor hours and unpredictable availability of agents because

of weather or emergencies. These constraints make it necessary to use other optimization techniques [19].

For a multiple-skill call center, the problem is even more complicated. Many different agent combinations might be possible for fulfilling service requirements and can potentially solve the staffing model, even if the shift durations are fixed [19]. Thus, the integration of the performance and staffing model is necessary. However, there is not a lot of research on so-called "skill-based routing" queueing models.

2.4 Conclusions

The basic structure of call centers allow them to be easily modelled as queueing systems. However, call centers can become very complex depending on their applications to different types of businesses. Thus, the call center research is a challenging and diverse area for the applications of queueing models. The importance of call centers has exploded over the last two decades in helping businesses meet their increasing demand. Thus, managers require more complex and robust call center models from researchers. The development of these complex models will benefit customers and help businesses provide efficient service to their various classes of customers through different forms of interaction.

Chapter 3

Call Center Modelling

The basic structure of the call center can be described as a finite capacity, multi-server system. Customer calls arrive at the call center at varying rates on a finite number of trunks. These calls are terminated at the ACD/PBX switch and are routed to a group of call agents. In a multimedia call center, these calls can be voice, email, fax, or (eventually) video.

3.1 Problem Setting

We study a complex call center system for which simple Markovian queueing models do not apply. Our goal is to develop alternative methods to estimate the transient performance for our system, rather than approximating them with steady-state $M/M/N/L$ queueing systems. Specifically, we develop a fluid model and a separate simulation model to approximate the mean number in system and mean virtual waiting time for both high and low priority customers at different points in time. We also measure the variance of the virtual waiting time for both classes of customers. Our call center is a help desk with two-customer classes and a preemptive-resume priority queue discipline. The high priority customer class

consists of voice calls, while the low priority customer class consists of e-mails. Here, we assume that the trunk capacity L, i.e, the number of telephone lines for the high priority customers, is large enough to prevent any call blocking. Also, we assumed that the service level for the high priority class is high enough that no calls abandon the system. Note that in a general call center environment these assumptions are not always valid.

In our model, the customers are served from two distinct virtual queues. The customers from the lowest priority class, i.e., e-mail, will abandon the low priority queue and enter the higher priority queue based on a specified service level parameter. In this regard, the low priority calls will have dynamic priorities, i.e., be upgraded. Our goal is to show that the fluid approximations of the call center performance are close to the actual performance, as measured by a discrete-event simulation model.

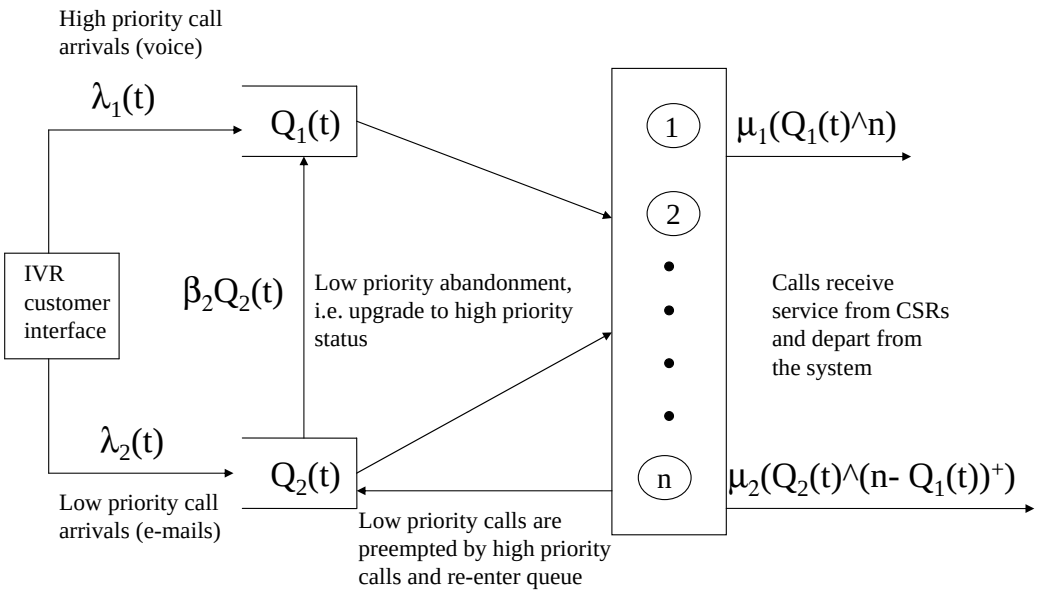

Figure 3.1: Two Class, Preemptive-Resume Model with Low Priority Abandonments

A diagram of the model is shown in Figure 3.1. The variables listed in Figure 3.1 are defined as follows:

- $\lambda_i(t)$ represents the arrival rate for class i customers into queue i, $(i = 1, 2)$;
- $Q_i(t)$ represents the number of class i customers in the system;
- β_2 is the abandonment rate of low priority customers out of the low priority queue;
- n is the number of servers, or CSRs, in the system, which remains constant over time.

Two important system performance measures in modern call centers are the mean waiting time and waiting-time distribution for customers. Call center managers must verify the customer's quality of service, which can be measured by the mean waiting time being less than some target delay, and/or the probability that a customer's waiting time is less than some target delay. Although there is no true industry standard, most managers aim to have at least 80 percent of their customers waiting less than 30 seconds for service. It is becoming more common in modern call centers to inform customers of delay predictions. However, providing only point-estimate predictions of delay, such as mean waiting time, does not provide as much value as predictions of the distribution of the waiting time. Therefore, the waiting-time distribution provides a better measure of the call center's quality of service, or service level, for the customer and the manager than the mean waiting time.

3.2 Research Methodology

We will use two methods to determine the mean number in system, mean virtual waiting-time, and the variance of the virtual waiting times performance measures

for our call center. First, we will develop fluid approximations for the performance measures. Also, we recommend an optimal staffing level to ensure that the mean virtual waiting time for both customer classes simultaneously satisfy their given service levels. In other words, given target delays of 30 seconds for high priority customers and 8 hours for low priority customers, we compute the number of agents required such that the mean virtual waiting-time for both classes is less than their targets. These approximations are determined by using asymptotic limit results based on scaling the inter-arrival rate and number of servers upwards to infinity. Note that these models do not require the restrictive assumptions of the standard Erlang models, such as stationary inter-arrival rates. Second, we construct a discrete-event simulation of the real call center system. Here, the mean number in system and the mean virtual waiting-time are computed for each customer class.

3.2.1 Priority Models with Voice and E-mail Calls

In this two class model, e-mail messages will have a lower priority than voice calls. There are two types of e-mail messages, or calls. The first type is the standard, or unconstrained, ones many people use to communicate with friends or co-workers. Typically, these messages are not limited by length or content. The second type of messages are ones where a customer completes a standard form or application on-line, and submits it to the call center. These constrained e-mail messages have a higher priority than unconstrained e-mail. Online registration forms are one example of this type of e-mail traffic.

The e-mail "abandonment", or upgrade, process is the same for both types of e-mail messages. When an e-mail has been in queue close to its abandonment

time, it leaves the low priority queue. However, it does not leave the system. Instead, it becomes a high priority call and is placed at the end of the voice call queue. We set the abandonment time equal to its service level time, s, which is used to set the maximum time a e-mail will wait to begin service in the low priority queue. In practice, the service level, s, for e-mail messages usually varies from 2 to 8 to 24 hours, depending on the size and type of call center. For example, a call center manager can strive to have 80 percent of the average waiting times for e-mails to be less than or equal to 2 hours. Also, each e-mail message will be modelled as having the same abandon time, equal to the service level, s. In other words, no email message can abandon its queue before one that arrived at an earlier time.

3.2.2 Priority Models with Voice and Fax Calls

If voice and fax calls are the classes in the call center, then the fax calls will have the lower priority. Here, the "abandonment" process for the fax calls can be modelled in two ways. The first way is the same as the e-mail abandonment process. Thus, fax calls leave their queue after waiting close to their abandonment time, which is equivalent to the target service level time, s.

The second way is much different from the e-mail process. The "abandonment" time for each fax is the exactly the same. These calls are allowed to queue. When the abandonment time is reached, all of the faxes in the queue leave together, as opposed to each one leave leaving separately as in the first way discussed above. They become high priority calls, and are placed at the end of the voice call queue. In other words, the fax calls still in the queue are upgraded in batches after a fixed period of time. For example, if they have a service level

of twenty-four (24) hours, then after each 24 hour period, all the faxes still in the low priority queue move to the high priority voice queue at the same instant. Afterwards, the fax calls will receive service in the high priority queue at the same rate as a voice call. Any new fax calls that arrive during the next 24 hour period either receive service immediately or wait for service (if necessary) at the low priority queue.

3.3 Our Call Center Model

Call center queueing models may yield complex solutions for traditional Markovian queueing systems. For example, for the stationary $M/G/n$, preemptive priority queue, solutions for the performance measures of the waiting time distribution are not well-known. Also, these distributions usually can not be written in closed-form, or, if they can, are not practical for deriving distribution values. Our call center queueing model is the $M_t/M/n$ model with a preemptive-resume priority service discipline. Since this model also yields complex solutions using traditional solution methods, such as Markovian models, a fluid approximation method will be used to compute estimates of the performance measures. Our fluid method will provide accurate approximations to the queue length, mean virtual waiting time, and the mean and variance of the virtual waiting times, as compared with our simulation estimates of the actual call center performance. The arrival process of the calls will be a Poisson process with time-varying inter-arrival rates. In practice, the number of servers, or agents, varies over time, resembling the shift scheduling of agents over the course of a day, week, or month in a call center. However, we assume that the number of servers remains constant over time. The service time distribution will be the exponential distribution, with the

service times of each call type having the same service rate. Finally, we assume that the number of trunk lines is infinite in our call center. Thus, there is no limit on the queue size in our model.

In our model, we use two classes of customers, voice and e-mail. The voice calls will always have the higher priority. We assume that the service level is high enough such that the number of high priority calls that abandon is insignificant. Thus, we do not allow the high priority customers to abandon the system. The lower priority of calls are e-mails. In practice, customers currently access most call centers through the use of e-mails over 50% of the time, compared with other forms of Web access such as Web chat (27%), click-to-talk (11.5%), and voice-over-the-Internet (2%). We choose e-mails as our low priority call type. The preemptive-resume priority discipline will be implemented, where higher priority voice calls will interrupt lower priority calls in service. However, if the lower priority calls are not meeting their service level requirements, then they will be allowed to upgrade their class and enter into the higher priority class. In other words, the priority of the lower class of calls will be dynamic, i.e., change over time. The abandonment process is the same as described above for e-mails. The abandonment rate is β_2 where $\beta_2 = 1/s$. Now, the overall abandonment rate out of the low priority queue will $\beta_2 \cdot (Q_2(t) - (n\Lambda Q_1(t))^+)^+$, where $(Q_2(t) - (n \cdot \Lambda Q_1(t))^+)^+$ is the number of e-mails waiting in the low priority queue. A simple representation of our call center model is shown in Figure 3.2.

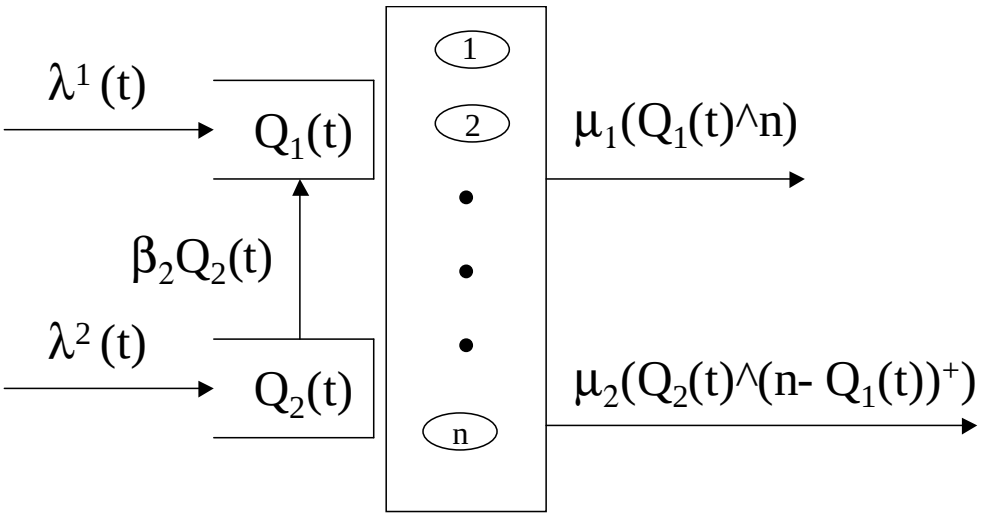

Figure 3.2: Multi-Class, Preemptive Priority Call Center with Dynamic Priorities

Theoretically, there is also a limit on the number of calls that an ACD can handle simultaneously. However, as discussed in Chapter 1, that limit has steadily increased as the ACD technology has advanced over the years. Thus, for our model, there will be no practical limit on the queue length.

Chapter 4

Fluid and Diffusion Approximations

Service systems models, such as call center models, belong to the class of stochastic service network models. These network models form a special family of non-stationary Markov processes where parameters such as inter-arrival and service rates are time-dependent. More importantly, these models have functional strong laws of large numbers and functional central limit theorem results for the number of customers in the system and the waiting time in queue [49]. The results are developed using an asymptotic limiting process, where the number of servers are scaled up in response to a scaling up of the arrival rates. The individual service and abandonment rates are not scaled. The resulting limit theorems are diffusion, and not heavy-traffic, limit results.

These limit theorems lead to a tractable set of network fluid and diffusion approximations in the form of a system of ordinary differential equations (ODEs). By numerically solving these differential equations, values for the distribution of performance measures such as the waiting time in queue can be computed. More importantly, we can approximate solutions of otherwise analytically intractable models. Therefore, an alternative, robust, methodology can be developed and applied to the performance analysis of service systems, such as call centers.

4.1 Multiple Customer Class

We compute the fluid approximations for the mean number in the system and mean virtual waiting time for the ***two customer class***, preemptive-resume priority, $M_t/M/n$ queue. Like Mandelbaum et al. [49], we use the Euler method to compute the fluid approximations for our model. If the approximations computed using Euler's method are vastly different from the simulation estimates, we will then use the Runge-Kutta method to improve the accuracy of our approximations. Since the high priority customers can preempt the lower priority ones, these customers will essentially receive service as if no other type of customer is present in the system. Thus, the high priority customer class results will be almost the same as the results for the single customer class. The only difference from the single customer class case is the dynamic priority process for the low priority customers, where these customers can abandon their queue and enter the high priority queue as a high priority customer. Each low priority customer, or e-mail, will have a exponential abandonment rate β_2 where $\beta_2 = 1/s$, and s is the service level time. The overall abandonment rate out of the low priority queue will $\beta_2 \cdot (Q_2(t) - (n\Lambda Q_1(t))^+)^+$, where $(Q_2(t) - (n \cdot \Lambda Q_1(t))^+)^+$ is the number of e-mails waiting in the low priority queue. This abandonment process adds the above abandonment rate term to the differential equations describing the mean number in system and waiting-time process for the high priority customers (as shown will be shown below). Mandelbaum et al. [49] developed fluid and diffusion approximations for the multiple customer class, preemptive-priority, $M_t/M/n$ queue. However, they only studied the mean number in system distributions for each customer class and not the waiting-time distributions.

Now, the $M_t/M/$n mean number in system process $Q \equiv \{Q(t) \mid t \geq 0\}$,

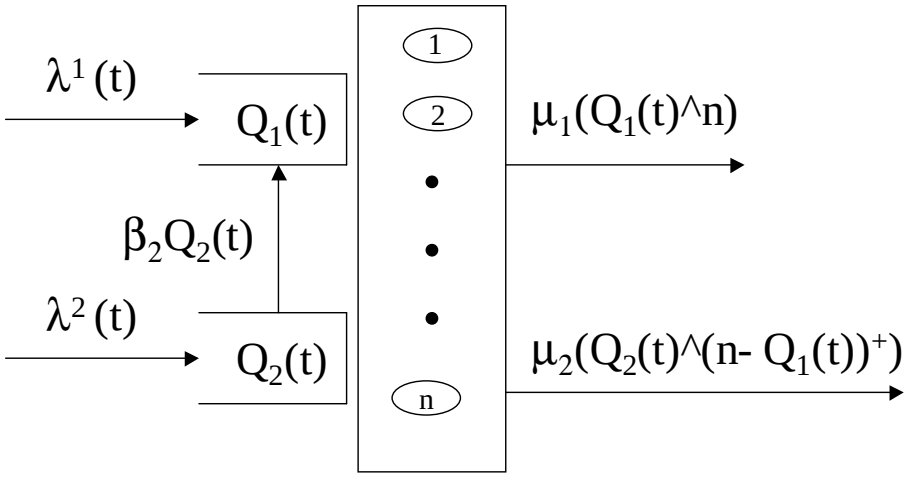

Figure 4.1: The Two-Customer Class $M_t/M/n$ Queue with Abandonment

previously defined for the single customer class case, must be defined for two customer classes.

The diagram of the new queueing model is shown in Figure 4.1.

4.1.1 Asymptotic Mean Number in System Results

Again, the results and theorems presented in this section are adapted from those stated by Mandelbaum, Massey, and Reiman [51] and Mandelbaum, Massey, Reiman, et al. [49]. However, customers are now grouped into two classes: high priority and low priority. High priority customers are labelled as class-1 customers while low priority customers are labelled as class-2 customers. Thus, all of the random variables of the stochastic processes discussed earlier are now random vectors. In other words, the random variables such as $Q(t)$ and $W(t)$ are now defined as:

$$\mathbf{Q}(t) = \{Q_1(t), Q_2(t)\}\,, \text{ and } \mathbf{W}(t) = \{W_1(t), W_2(t)\}\,. \tag{4.1}$$

for all t. Here, $Q_1(t)$, $W_1(t)$ and $Q_2(t)$, $W_2(t)$ are the corresponding quantities for class-1 and class-2 customers, respectively.

Now, the limit theorem for the functional strong law of large numbers can be restated for the new model. The initial conditions for the mean number in system process satisfy the following asymptotic assumption:

$$\lim_{\eta\to\infty}\frac{1}{\eta}\mathbf{Q}^{\eta}(0) = \mathbf{Q}^{(0)}(0) \text{ a.s.}, \tag{4.2}$$

where $\mathbf{Q}^{(0)}(0) = \left\{Q_1^{(0)}(0), Q_2^{(0)}(0)\right\}$ is constant. Note that $\mathbf{Q}^{\eta}$ was defined in Chapter 2, Section 2.2.3 as the number in system process scaled by a factor of η. Thus, the functional strong law of large numbers theorem for the new model is:

Theorem 4.1

$$\lim_{\eta\to\infty}\frac{1}{\eta}\mathbf{Q}^{\eta} = \mathbf{Q}^{(0)}, \ \textit{a.s.}, \tag{4.3}$$

where the convergence is uniform on compact sets of t. Moreover, $\mathbf{Q}^{(0)} = \{\mathbf{Q}^{(0)}(t) \mid t \geq 0\} = \left\{Q_1^{(0)}(t), Q_2^{(0)}(t) \mid t \geq 0\right\}$ *is uniquely determined by* $\mathbf{Q}^{(0)}(0)$ *and the differential equation:*

$$\begin{aligned}
\frac{d}{dt}Q_1^{(0)}(t) &= \lambda_1(t) - \mu_1(Q_1^{(0)}(t) \wedge n) - \beta[Q_2^{(0)}(t) - (n - Q_1^{(0)}(t))^{+}]^{+}; \quad (4.4)\\
\frac{d}{dt}Q_2^{(0)}(t) &= \lambda_2(t) - \mu_2[Q_2^{(0)}(t) \wedge (n - Q_1^{(0)}(t))^{+}]^{+}\\
&\quad -\beta[Q_2^{(0)}(t) - (n - Q_1^{(0)}(t))^{+}]^{+} \qquad (4.5)
\end{aligned}$$

where $[Q_2^{(0)}(t) - (n - Q_1^{(0)}(t))^{+}]^{+}$ *represents the number of customers in the low priority queue.*

This theorem states rigorously that $\mathbf{Q}^{\eta} \approx \eta\mathbf{Q}^{(0)}$ for large η, where $\mathbf{Q}^{(0)}$ is called the *fluid approximation* for $\mathbf{Q}^{\eta}$. The proof of the theorem is given in [49].

As discussed for the single customer class case, the fluid approximation in Theorem 2.1 can be refined using the functional central limit theorem. Here, the

initial conditions satisfy the following assumption:

$$\lim_{\eta\to\infty}\sqrt{\eta}\left(\frac{1}{\eta}\mathbf{Q}^{\eta}(0)-\mathbf{Q}^{(0)}(0)\right)\stackrel{d}{=}\mathbf{Q}^{(1)}(0), \tag{4.6}$$

where $\mathbf{Q}^{(1)}$ is a constant. Now, for the new model, the functional central limit theorem is:

Theorem 4.2

$$\lim_{\eta\to\infty}\sqrt{\eta}\left(\frac{1}{\eta}\mathbf{Q}^{\eta}-\mathbf{Q}^{(0)}\right)\stackrel{d}{=}\mathbf{Q}^{(1)}, \tag{4.7}$$

where $\mathbf{Q}^{(1)}=\left\{\mathbf{Q}^{(1)}(t)\mid t\geq 0\right\}$ *is a diffusion process. This is a convergence in distribution of the stochastic processes in an appropriate functional space [49].*

Moreover, if the set of time points $\left\{t\geq 0\mid \mathbf{Q}^{(0)}(t)=n\right\}$ *has measure zero for the multi-server queue with abandonment model, then* $\left\{\mathbf{Q}^{(1)}(t)\mid t\geq 0\right\}$ *is a Gaussian process. The mean for* $\mathbf{Q}^{(1)}$ *then solves the following differential equations for:*

$$\begin{aligned}
\frac{d}{dt}\mathsf{E}\left[Q_1^{(1)}(t)\right] &= (\beta 1_{\{Q_1^{(0)}(t)\leq n\}}-\mu_1 1_{\{}Q_1^{(0)}(t)\leq n\})\mathsf{E}[Q_1^{(1)}(t)] \\
&= +\beta Q_2^{(1)}(t)\mathsf{E}[Q_2^{(1)}(t)]; \qquad (4.8)\\
\frac{d}{dt}\mathsf{E}\left[Q_2^{(1)}(t)\right] &= (-\beta 1_{\{Q_1^{(0)}(t)\leq n\}}+\mu_2 1_{\{Q_2^{(0)}(t)\leq (n-Q_1^{(0)}(t))^+\}})\mathsf{E}[Q_1^{(1)}(t)] \\
&\quad -(\beta Q_2^{(0)}(t)+\mu_2 1_{\{Q_2^{(0)}(t)\leq (n-Q_1^{(0)}(t))^+\}})\mathsf{E}[Q_2^{(1)}(t)]. \qquad (4.9)
\end{aligned}$$

Similarly, the variance for $\mathbf{Q}^{(1)}$ *solves the following set of differential equation:*

$$\begin{aligned}
\frac{d}{dt}\mathsf{Var}\left[Q_1^{(1)}(t)\right] &= 2(\beta 1_{\{Q_1^{(0)}(t)\leq n\}}-\mu_1 1_{\{Q_1^{(0)}(t)\leq n\}})\mathsf{Var}[Q_1^{(1)}(t)]; \qquad (4.10)\\
&\quad +\beta Q_2^{(1)}(t)\mathsf{Cov}[Q_1^{(1)}(t),Q_2^{(1)}(t)]+\lambda_1(t) \\
&\quad +\mu(Q_1^{(0)}(t)\wedge n)+\beta[Q_2^{(0)}(t)-(n-Q_1^{(0)}(t))^+]^+. \qquad (4.11)
\end{aligned}$$

$$\frac{d}{dt}\mathsf{Var}\left[Q_2^{(1)}(t)\right] = -2\beta 1_{\{Q_1^{(0)}(t)\leq n\}}\mathsf{Cov}[Q_1^{(1)}(t),Q_2^{(1)}(t)]$$

$$
\begin{aligned}
&+2\mu_2 1_{\{Q_2^{(0)}(t)\le (n-Q_1^{(0)}(t))^+\}}\mathsf{Cov}[Q_1^{(1)}(t),Q_2^{(1)}(t)]\\
&+(-\beta Q_2^{(0)}(t)-\mu_2 1_{\{Q_2^{(0)}(t)\le (n-Q_1^{(0)}(t))^+\}})\mathsf{Var}[Q_2^{(1)}(t)]\\
&+\lambda_2(t)+\mu[Q_2^{(0)}(t)\wedge (n-Q_1^{(0)}(t))^+]^+\\
&+\beta[Q_2^{(0)}(t)-(n-Q_1^{(0)}(t))^+]^+. \qquad (4.12)
\end{aligned}
$$

Finally, the covariance for $\mathbf{Q}^{(1)}$ *solves the following differential equation:*

$$
\begin{aligned}
\frac{d}{dt}\mathsf{Cov}\left[Q_1^{(1)}(t),Q_2^{(1)}(t)\right] &= -\beta 1_{\{Q_1^{(0)}(t)\le n\}}\mathsf{Var}[Q_1^{(1)}(t)]\\
&\quad +\mu_2 1_{\{Q_2^{(0)}(t)\le (n-Q_1^{(0)}(t))^+\}}\mathsf{Var}[Q_1^{(1)}(t)]\\
&\quad +\beta Q_2^{(1)}(t)\mathsf{Var}[Q_2^{(1)}(t)]\\
&\quad +\beta 1_{\{Q_1^{(0)}(t)\le n\}}\mathsf{Cov}[Q_1^{(1)}(t),Q_2^{(1)}(t)]\\
&\quad -\mu_1 1_{\{Q_1^{(0)}(t)\le n\}}\mathsf{Cov}[Q_1^{(1)}(t),Q_2^{(1)}(t)]\\
&\quad -\beta Q_2^{(0)}(t)\mathsf{Cov}[Q_1^{(1)}(t),Q_2^{(1)}(t)]\\
&\quad -\mu_2 1_{\{Q_2^{(0)}(t)\le (n-Q_1^{(0)}(t))^+\}}\mathsf{Cov}[Q_1^{(1)}(t),Q_2^{(1)}(t)]\\
&\quad -\beta(Q_2^{(0)}(t)-(n-Q_1^{(0)}(t))^+). \qquad (4.13)
\end{aligned}
$$

Theorem 4.2 states rigorously that $\mathbf{Q}^\eta \approx \eta\mathbf{Q}^{(0)}+\sqrt{\eta}\mathbf{Q}^{(1)}$ for large η, where $\mathbf{Q}^{(1)}$ is called the *diffusion approximation* for $\mathbf{Q}^\eta$. We used Theorem 5.2 in Mandelbaum et al. [51] applied to our model to derive the Equation (4.23), Equation (4.12), and Equation (4.13). Specifically, from Theorem 5.2, we have:

$$
\frac{d}{dt}\mathsf{Var}\left[Q_1^{(1)}(t)\right]=2a_{11}\mathsf{Var}\left[Q_1^{(1)}(t)\right]+2a_{21}\mathsf{Cov}\left[Q_1^{(1)}(t),Q_2^{(1)}(t)\right]+b_{11} \qquad (4.14)
$$

$$
\frac{d}{dt}\mathsf{Var}\left[Q_2^{(1)}(t)\right]=2a_{22}\mathsf{Var}\left[Q_2^{(1)}(t)\right]+2a_{12}\mathsf{Cov}\left[Q_1^{(1)}(t),Q_2^{(1)}(t)\right]+b_{22} \qquad (4.15)
$$

$$
\begin{aligned}
\frac{d}{dt}\mathsf{Cov}\left[Q_1^{(1)}(t),Q_2^{(1)}(t)\right] &= a_{12}\mathsf{Var}\left[Q_1^{(1)}(t)\right]+a_{21}\mathsf{Var}\left[Q_2^{(1)}(t)\right]\\
&\quad +(a_{11}+a_{22})\mathsf{Cov}\left[Q_1^{(1)}(t),Q_2^{(1)}(t)\right]+b_{12} \qquad (4.16)
\end{aligned}
$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{4.17}$$

is the Jacobian of the right-hand side of Equation(7.1) for the high and low priority customers. Additionally,

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \tag{4.18}$$

is the tensor product of the right-hand side of Equation (7.1) for the high and low priority customers. Therefore, by computing the Jacobian matrix and tensor product matrix, we determine the coefficients of our variance and covariance differential equations.

4.1.2 Asymptotic Virtual Waiting-Time Results

The asymptotic results for the mean virtual waiting-time depend on the mean number in system process approximations. As in the mean number in system process section, the results and theorems presented here have been adapted and summarized from those stated by Mandelbaum, Massey, and Reiman [51] and Mandelbaum et al. [49]. To compute the waiting-time of virtual customer arriving to the system at a fixed time $\tau \geq 0$, an additional assumption is required. The original model will be modified as follows:

- There are no new exogenous arrivals into the system after time τ_i.
- In particular, the servers only process any remaining customers in the system at time τ_i.

Virtual Waiting Time Methodology

Finally, in the fluid and diffusion approximations, time, τ_i, is the time when the arrival rate for the two class of customers is set to 0. This is analogous to turning off the source of the fluid, and analyzing the fluid level as it drains out of its container. The sequence of τ_i's are evenly-spaced over the time interval $[0, T]$, separated from each other by subintervals of width, h. Therefore, the number of subintervals, nos, is defined as $nos = \frac{T}{h}$, and the sequence $\{\tau_i\}$ has index i where $i = 1, \ldots nos$. Thus, after time τ_i, the waiting-time in queue for a high priority customer is the time until the high priority queue empties, i.e., the length is reduced to 0. The low priority customer's waiting-time is similar, but more complex, which will be discussed later in this chapter. Therefore, the virtual waiting-time and the waiting-time distribution can be computed for a customer arriving to the system at time τ_i.

High Priority Customers

We use Theorem 4.1 and 4.2 to derive the fluid and diffusion approximations for the virtual waiting time of the high priority customers. After time τ_1, we show that certain terms in their equations, corresponding to the external arrivals to the system, will become zero [51]. The asymptotic results for the virtual waiting time require some new notation. We denote the arrival and departure processes for the system by:

$$A^\eta = \{A^\eta \mid t \leq 0\}, \text{ and } \Delta^\eta = \{\Delta^\eta \mid t \leq 0\} \tag{4.19}$$

respectively. By convention, let the arrival process include the customers in the system at time 0. So, $A^\eta(0) = \hat{Q}_1^\eta(0)$, $\Delta^\eta(0) = 0$, and $A^\eta(t) - \Delta^\eta(t) = \hat{Q}_1^\eta(t), t \geq 0$, where $\hat{Q}_1^\eta(t)$ is the mean number in system process for the modified queue.

The previous assumptions and notations lead to the following *fluid* limit result:

Theorem 4.3 *As a joint process,*

$$\lim_{\eta\to\infty}\frac{1}{\eta}\left(\hat{Q}_1^{\eta},A^{\eta},\Delta^{\eta}\right)=\left(\hat{Q}_1^{(0)},A^{(0)},\Delta^{(0)}\right),\ a.s. \tag{4.20}$$

and this convergence is uniform on compact sets of t. The fluid limit $\hat{Q}_1^{(0)}(t)$ *satisfies equation 7.1 for* $t<\tau$*. For* $t\geq\tau$*, the following properties hold:*

1. *The future evolution of* $\hat{Q}_1^{(0)}(t)$ *is determined by the differential equation:*

$$\frac{d}{dt}\hat{Q}_1^{(0)}(t)=-\mu\cdot(\hat{Q}_1^{(0)}(t)\wedge n)-\beta\cdot(\hat{Q}_1^{(0)}(t)-n)^{+}. \tag{4.21}$$

2. *There are no future arrivals, so that* $A^{(0)}(t)=A^{(0)}(\tau)$*.*

3. *The deterministic process* $\Delta^{(0)}$ *is a continuously differential non-decreasing function in* $[0,\infty]$ *.*

Also, the additional assumption leads to the following *diffusion* limit result:

Theorem 4.4

$$\lim_{\eta\to\infty}\sqrt{\eta}\left(\frac{1}{\eta}\hat{Q}_1^{\eta}-\hat{Q}_1^{(0)},\frac{1}{\eta}A^{\eta}-A^{(0)},\frac{1}{\eta}\Delta^{\eta}-\Delta^{(0)}\right)\stackrel{d}{=}\left(\hat{Q}_1^{(1)},A^{(1)},\Delta^{(1)}\right). \tag{4.22}$$

Moreover, if the set of time points $\left\{t\geq 0\mid \hat{Q}_1^{(0)}(t)=n\right\}$ *has measure zero for the multi-server queue with abandonment model, then* $\left\{\hat{Q}_1^{(1)}(t)\mid t\geq 0\right\}$ *is a Gaussian process. For* $t\geq\tau$*,* $\mathsf{Var}\left[\hat{Q}^{(1)}(t)\right]$ *solves the differential equation:*

$$\begin{aligned}\frac{d}{dt}\mathsf{Var}\left[Q_1^{(1)}(t)\right] &= 2(\beta 1_{\{\hat{Q}_1^{(0)}(t)\leq n\}}-\mu_1 1_{\{\hat{Q}_1^{(0)}(t)\leq n\}})\mathsf{Var}[\hat{Q}_1^{(1)}(t)]\\ &\quad+\beta\hat{Q}_2^{(1)}(t)\mathsf{Cov}[\hat{Q}_1^{(1)}(t),Q_2^{(1)}(t)]\\ &\quad+\mu(\hat{Q}_1^{(0)}(t)\wedge n)+\beta[\hat{Q}_2^{(0)}(t)-(n-\hat{Q}_1^{(0)}(t))^{+}]^{+}. \end{aligned}\tag{4.23}$$

It follows from the above theorem and definitions that

$$\hat{Q}_1^{(1)}(t) = A^{(1)}(t) - \Delta^{(1)}(t). \tag{4.24}$$

Before the asymptotic result for the virtual waiting time distribution can be stated, a few more definitions and assumptions must be given. The *potential service initiation* process D^η for the server is defined as:

$$D^\eta(t) = \Delta^\eta(t) + \eta n,\ t \geq 0. \tag{4.25}$$

Recall that $A^\eta(t) - \Delta^\eta(t) = \hat{Q}^\eta(t), t \geq 0$. So, if $\hat{Q}^\eta(t) < \eta n$, then $A^\eta(t) < D^\eta(t)$. Thus, by Theorem 4.3,

$$\lim_{\eta\to\infty} \frac{1}{\eta} D^\eta(\cdot) = D^{(0)}(\cdot) \text{ a.s.}, \tag{4.26}$$

where the convergence is uniform on compact sets of t and $D^{(0)}(t) = \Delta^{(0)}(t) + n, t \geq 0$. Note that $D^{(0)}(t)$ is continuously differentiable because $\Delta^{(0)}(t)$ is continuously differentiable as the fluid limit of the departure process. Thus, the derivative of $D^{(0)}(t)$ is denoted by $d^{(0)}(t)$. The following assumption for $D^{(0)}(t)$ is important, but not too restrictive for the virtual waiting time result [51]:

$$\lim_{t\to\infty} D^{(0)}(t) > A^{(0)}(\tau), \tag{4.27}$$

where $D^{0)}(t)$ is continuously differentiable with *strictly positive* derivative. Note that, based on previous definitions, $A^{(0)}(\cdot)$ and $A^{(0)}(\tau)$ are constant on the interval $[\tau, \infty)$. Also, it is convenient to assume that all processes are defined on the interval $[-T, \infty)$ where $T = n/d^{(0)}(0)$ instead of $[0, \infty)$. This interval extension assumes that there are no arrivals or departures within the interval $[-T, 0)$.

Now, Theorem 4.3 and Theorem 4.4 can be written in terms of D as follows:

$$\lim_{\eta\to\infty} \frac{1}{\eta} (Q^\eta, A^\eta, D^\eta) = \left(\hat{Q}^{(0)}, A^{(0)}, D^{(0)}\right) \tag{4.28}$$

and,

$$\lim_{\eta\to\infty}\sqrt{\eta}\left(\frac{1}{\eta}\hat{Q}_1^{\eta}-\hat{Q}_1^{(0)},\frac{1}{\eta}A^{\eta}-A^{(0)},\frac{1}{\eta}D^{\eta}-D^{(0)}\right)\stackrel{d}{=}\left(\hat{Q}^{(1)},A^{(1)},D^{(1)}\right), \quad (4.29)$$

where $D^{(1)}=\Delta^{(1)}$ and $t\geq -T$.

Note that $A^{(0)},D^{(0)},A^{(1)},D^{(1)}$ are continuous and $D^{(0)}(-T)=D^{(1)}(-T)=0$ [51]. Let the *first attainment* process, $S_1^{(\eta)}(t)$, be defined for all $t\geq -T$ as:

$$S_1^{\eta}(t)=\inf\{s\geq -T: D^{(\eta)}(s)>A^{\eta}(t)\}, \quad (4.30)$$

and,

$$S_1^{(0)}(t)=\inf\{s\geq -T: D^{(0)}(s)>A^{(0)}(t)\}. \quad (4.31)$$

Similarly, define the *attainment* waiting time process as:

$$W_1^{\eta}(t)=S^{\eta}(t)-t, \quad (4.32)$$

and,

$$W_1^{(0)}(t)=S^{(0)}(t)-t. \quad (4.33)$$

The conventions and assumption defined above allow the previous processes to be well-defined and finite with probability 1 for sufficiently large η.

Now, define the *virtual* waiting time at τ_i, $\hat{W}_1^{\eta}(\tau_i)$, as the time a customer arriving to the queueing service node at time τ_i would have to wait until its service starts, assuming that customer *does not* leave the queue [51]. Thus, the virtual waiting time, $\hat{W}_1^{\eta}(\tau_i)$, and the attainment waiting time, $W_1^{\eta}(t)$, are related as:

$$\hat{W}^{\eta}(\tau_i)=W^{\eta}(\tau_i)^{+}. \quad (4.34)$$

So, if $\hat{Q}_1^{\eta}(\tau_i)<\eta n$, then $W_1^{\eta}(\tau_i)$ (and $W_1^{(0)}(\tau_i)$) will be negative. Therefore, by definition, $\hat{W}_1^{\eta}(\tau_i)=0$. If $W_1^{\eta}(\tau_i)$ is non-negative, then $\hat{W}1^{\eta}(\tau_i)$ will have the same value as $W_1^{\eta}(\tau_i)$.

The next theorem follows directly from Equations (4.28), (4.29), and the theorem in Puhalskii [60]. Those results yield the following convergence theorem:

Theorem 4.5

$$\lim_{\eta\to\infty}\frac{1}{\eta}\left(\hat{Q}^{\eta},A^{\eta},D^{\eta},W_1^{\eta}\right)=\left(\hat{Q}^{(0)},A^{(0)},D^{(0)},W_1^{(0)}\right),\ \textit{a.s.,} \tag{4.35}$$

and,

$$\lim_{\eta\to\infty}\sqrt{\eta}\left(\frac{1}{\eta}\hat{Q}_1^{\eta}-\hat{Q}^{(0)},\frac{1}{\eta}A^{\eta}-A^{(0)},\frac{1}{\eta}D^{\eta}-D^{(0)},W_1^{\eta}-W_1^{(0)}\right)\stackrel{d}{=}\left(\hat{Q}^{(1)},A^{(1)},D^{(1)},W_1^{(1)}\right), \tag{4.36}$$

where

$$W_1^{(1)}=\frac{A^{(1)}(t)-D^{(1)}(S_1^{(0)}(t))}{d^{(0)}(S_1^{(0)}(t))}\ \textit{and}\ S_1^{(0)}(t)=\textit{inf}\{s\geq -T: D^{(0)}(s)>A^{(0)}(t)\}. \tag{4.37}$$

Since the processes $A^{(1)}(t)$, $D^{(1)}(t)$, $\hat{Q}_1^{(1)}(t)$, $W_1^{(1)}(t)$ are continuous a.s., their finite dimensional distributions converge [51]. In particular, consider the non-trivial case $S_1^{(0)}(\tau_i)\geq\tau_i$, which is equivalent to $\hat{Q}_1^{(0)}(\tau_i)\geq n$. Moreover, assume that the set of points $\left\{t\mid \hat{Q}_1^{(0)}(t)=n\right\}$ has measure zero on $[0,\tau_i]$. Then:

$$\lim_{\eta\to\infty}W_1^{\eta}(\tau_i)=W_1^{(0)}(\tau_i)\ \text{a.s.} \tag{4.38}$$

and

$$\lim_{\eta\to\infty}\sqrt{\eta}(W_1^{\eta}(\tau_i)-W_1^{(0)}(\tau_i))\stackrel{d}{=}W_1^{(1)}(\tau_i)=\frac{\hat{Q}_1^{(1)}(S_1^{(0)}(\tau_i))}{d^{(0)}(S_1^{(0)}(\tau_i))} \tag{4.39}$$

where $\hat{Q}_1^{(1)}(S_1^{(0)}(\tau_i))$ is a Gaussian process with a mean and variance defined by the following procedure below. First, solving Equation (4.21) for $\hat{Q}_1^{(0)}(\cdot)$ in the interval $[\tau_i,\infty]$ yields:

$$\frac{d}{dt}\hat{Q}_1^{(0)}(t)=-\mu_1(Q_1^{(0)}(t)\wedge n)-\beta[Q_2^{(0)}(t)-(n-Q_1^{(0)}(t))^{+}]^{+},\ t\geq\tau_i. \tag{4.40}$$

Now, by definition,

$$S_1^{(0)}(\tau_i) = \min\left\{t \geq \tau_i \mid \hat{Q}_1^{(0)}(t) = n\right\}. \tag{4.41}$$

Second, the mean, $\mathsf{E}\left[\hat{Q}_1^{(1)}(S^{(0)}(\tau_i))\right]$, and variance, $\mathsf{Var}\left[\hat{Q}_1^{(1)}(S^{(0)}(\tau_i))\right]$, are computed as the solutions to the following equations:

$$\frac{d}{dt}\mathsf{E}[\hat{Q}_1^{(1)}(t)] = \beta Q_2^{(1)}(t)\mathsf{E}[\hat{Q}_2^{(1)}(t)],\ t \geq \tau_i, \tag{4.42}$$

and

$$\begin{aligned}\frac{d}{dt}\mathsf{Var}[\hat{Q}_1^{(1)}(t)] &= 2\beta\hat{Q}_2^{(1)}(t)\mathsf{Cov}[\hat{Q}_1^{(1)}(t), \hat{Q}_2^{(1)}(t)] + \mu(\hat{Q}_1^{(0)}(t) \wedge n) \\ &\quad +\beta[\hat{Q}_2^{(0)}(t) - (n - \hat{Q}_1^{(0)}(t))^+]^+,\ t \geq \tau_i. \end{aligned}\tag{4.43}$$

Because zero is a solution to Equation (4.42), we assume the mean is zero. Finally, since $d^{(0)}\left(S_1^{(0)}(\tau_i)\right) = n\mu$ when $S_1^{(0)}(\tau_i) \geq \tau_i$, we can compute the following:

$$\mathsf{Var}\left[W_1^{(1)}(\tau_i)\right] = \frac{\mathsf{Var}\left[\hat{Q}_1^{(1)}(S_1^{(0)}(\tau_i))\right]}{(n\mu)^2}. \tag{4.44}$$

Low Priority Customers

The computation for the low priority customers' waiting time is more complex than for the high priority customers. Since these customers can be preempted and abandon to the high priority queue, their waiting time before they complete service is a combination of three processes. The first one is the waiting-time process in the low priority queue for some random amount of time. The second one is the partial-service process that customers receive before being preempted by a high priority customer, if a server is idle and no high priority customer is present in the system. This amount of time is also random. The last process is the waiting-time process in the high priority queue, if the low priority customer

abandons the low priority queue before completing service while waiting there or being preempted. This time is a random period as well.

After some fixed time τ_i, we set the low priority arrival rate, λ_2, equal to 0. Next, we can only conjecture about the virtual waiting time distribution for the low priority customers. The fact that some of these customers abandon their queue, and complete service as a high priority customer complicates the limit theorem results stated earlier as Theorem 4.3, Theorem 4.4, Theorem 4.28, and Theorem 4.29. Thus, we can not state, presently, a true asymptotic limit theorem for the low priority waiting-time distribution. But, for those customers that do abandon, the asymptotic limit theorems do apply separately to their waiting-time processes observed in both queues. In others words, the distribution of the low priority customers' virtual waiting time in the low priority queue (ignoring preemption for the moment) would be a Gaussian distribution where the mean and variance are solutions to a differential equation. Similarly, once the low priority customers abandoned to the high priority queue, the distribution of the low priority customers' virtual waiting time in the high priority queue would also be a Gaussian distribution. However, it is not the case that the virtual waiting time processes in the low and high priority queues are independent. This is true because the upgraded low priority customers in the high priority queue will occupy all the servers, and thus, may increase the virtual waiting time of the low priority customers in the low priority queue. Therefore, since the two low priority waiting time processes are not independent, we cannot state that their joint waiting time distribution is also Gaussian, which is opposite the limit theorem results in Mandelbaum et al. [51].

In addition, the virtual waiting time process in the low priority queue might

have an additional component due to possible preemption by high priority customers (which we ignored in the above paragraph). If a preemption occurs, the low priority customers service is incomplete and their waiting time in the low priority increases based on their remaining service time. Since the service time distribution is exponential, the remaining service time will be exponentially distributed. The additional waiting time component is this remaining service time after each preemption. Thus, the complete virtual waiting time for low priority customers consists of their waiting time in the low priority customer, their remaining service after being preempted, and their waiting time in the high priority queue. Therefore, we can conjecture that the joint virtual waiting-time distribution for low priority customers is a composition of at most two Gaussian distributions (if an "upgrade" occurs) and an exponential distribution (if a preemption occurs).

Although we can not completely describe the virtual waiting time distribution, we can determine the fluid approximation for the mean virtual waiting time of a virtual low priority customer arriving at time τ_i using Theorem 4.5 above. First, we compute the fluid approximations of the mean number in system at time τ_i, which is used to determine the mean virtual waiting time fluid approximation. We develop a "pseudo-deterministic" fluid approximation algorithm to compute these performance estimates.

Our algorithm can be summarized in the following way. First, the low priority arrival rate λ_2 is set equal to 0, so that the arriving low priority customer at τ_i is the last to enter the system. Second, the "abandonment" time (i.e., upgrade time) and service time for this customer are generated. We model the abandonment rate, β_2, from the low priority queue as an exponential random variate with mean

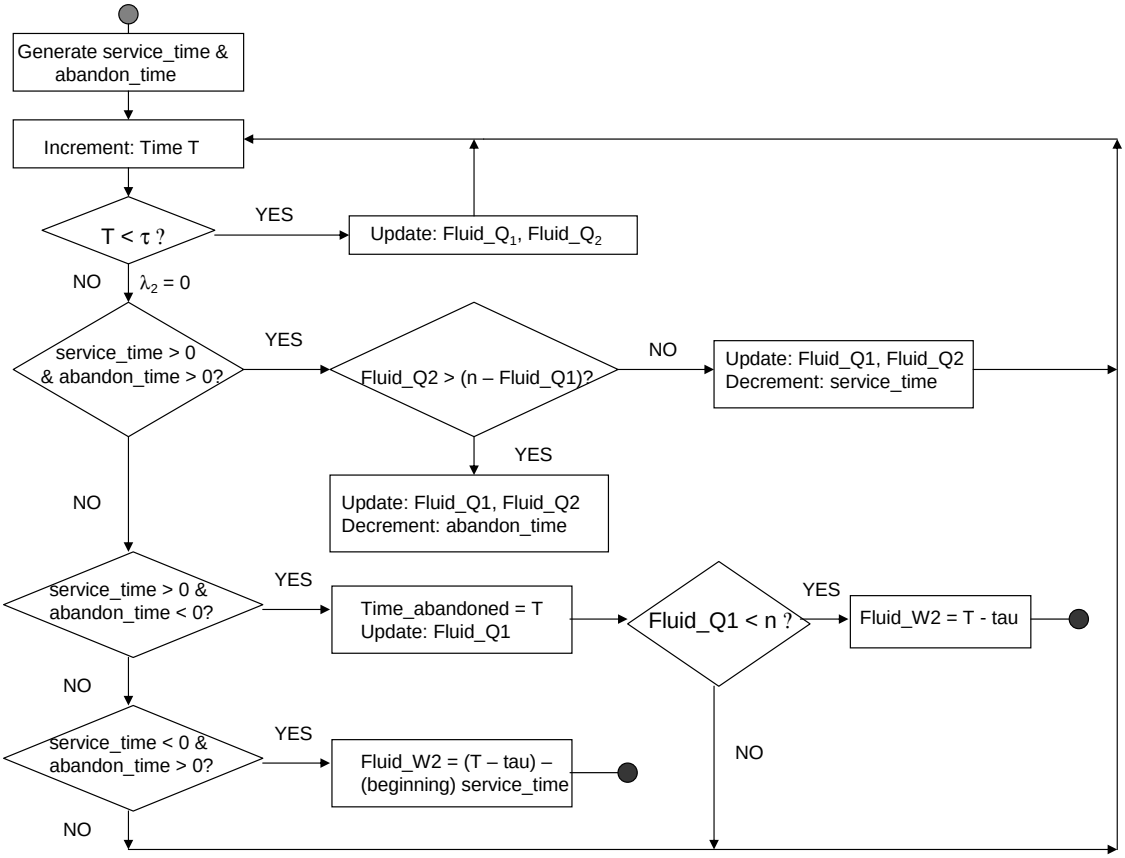

Figure 4.2: Outline of Low Priority Fluid Approximation Algorithm

$1/s$, where s corresponds to the low priority service level time requirement. Since we set the abandonment time equal to $1/\beta_2$, each low priority customer in the queue will have a different abandonment time until time τ_i. We model the service time also as an exponential random variate with rate μ_2, where μ_2 is the low priority service rate. Since we must introduce random quantities into the usually deterministic fluid method, our algorithm is actually a "pseudo-deterministic" one.

We show the progression of the fluid computations in our algorithm in Figure 4.2.

Third, our pseudo-deterministic algorithm models the rest of the low priority waiting time process by "tracking" this virtual customer as a discrete entity through the system. (We realize, however, that a true deterministic fluid method models only the average, or aggregate, movement of entities.) Thus, as long as its abandonment time has not been attained, the customer waits in the low priority queue for a server to become available. While he waits in queue, we allow other high priority customers to enter the system, creating additional low priority customer delay. If the number of high priority customers in the system is less than the number of servers, n, and the low priority customer is at the head of its queue, then the low priority customer enters service. Once at a server, the customer must receive service for the duration of its service time before completing service. Note that the low priority customer can not abandon the server in order to be upgraded to the high priority queue. If the customer is preempted while in service, the customer returns to the head of low priority queue. It remains in queue until a server becomes available again, or it abandons to the high priority queue if its abandonment time is achieved. The low priority customer can not abandon the server while in service in order to be upgraded to the high priority queue.

Fourth, if the abandonment time is attained before the customer completes service, then the customer abandons its queue and enters the end of the high priority queue. There, the customer waits for service in the same manner as any other high priority customer. Then, the "high priority waiting-time" part for the upgraded customer is computed in the following way. The high priority *external* arrival rate is set equal to 0. When the high priority fluid level equals or drops below the number of servers, the upgraded customer's delay in the high priority

queue is computed. Finally, the total waiting time in queue for the virtual low priority customer arriving at τ_i is equal to the sum of their waiting time in the both the high and low priority queues. Since we introduce some randomness into the computation, we compute the mean waiting time estimate for a number of independent replications. We use the average value of the mean virtual waiting times over these replications as our low priority mean virtual waiting time fluid approximation. Therefore, we can compute the fluid approximation to the virtual waiting time using our algorithm.

Finally, we can summarize our "pseudo-deterministic" algorithm in the following manner. We use a randomly generated abandonment time and service time for a virtual low priority arriving at time τ to determine which of the following "events" occurs next in time:

1. the customer waits in low priority queue until entering service, or
2. the customer is preempted from service by a high priority customer, or
3. the customer is upgraded to end of high priority queue and eventually completes service, or
4. the customer completes service from the low priority queue.

Using our low priority algorithm and the fluid approximation of the virtual waiting time, we compute the low priority variance of the virtual waiting time. However, we must first derive equations similar to Equation (4.42), Equation (4.43), and Equation (4.44) above for the high priority virtual waiting time. Thus, we consider the case $S_2^{(0)}(\tau_i) \geq \tau_i$ as before, which is equivalent to $\hat{Q}_2^{(0)}(\tau_i) \geq (n - \hat{Q}_1^{(0)}(\tau_i)$. First, solving Equation (4.21) for $\hat{Q}_2^{(0)}(\cdot)$ in the interval $[\tau_i, \infty]$

yields:

$$\frac{d}{dt}\hat{Q}_2^{(0)}(t) = -\mu_2[Q_2^{(0)}(t) \wedge (n - \hat{Q}_1^{(0)}(t))^+]^+ - \beta[\hat{Q}_2^{(0)}(t) - (n - Q_1^{(0)}(t))^+]^+, \; t \geq \tau_i. \tag{4.45}$$

Now, by definition,

$$S_2^{(0)}(\tau_i) = \min\left\{t \geq \tau_i \mid hatQ_2^{(0)}(\tau_i) = (n - \hat{Q}_1^{(0)}(\tau_i)\right\}. \tag{4.46}$$

The mean, $\mathsf{E}\left[\hat{Q}_1^{(1)}(S^{(0)}(\tau_i))\right]$, and variance, $\mathsf{Var}\left[\hat{Q}_1^{(1)}(S^{(0)}(\tau_i))\right]$, are computed as the solutions to the following equations:

$$\frac{d}{dt}\mathsf{E}[\hat{Q}_2^{(1)}(t)] = (-\beta\hat{Q}_2^{(0)}(t))\mathsf{E}[\hat{Q}_2^{(1)}(t)], \; t \geq \tau_i, \tag{4.47}$$

and

$$\begin{aligned}\frac{d}{dt}\mathsf{Var}[\hat{Q}_2^{(1)}(t)] &= -2\beta Q_2^{(0)}(t)\mathsf{Var}[Q_2^{(1)}(t)] + \mu[Q_2^{(0)}(t) \wedge (n - Q_1^{(0)}(t))^+]^+ \\ &\quad +\beta[Q_2^{(0)}(t) - (n - Q_1^{(0)}(t))^+]^+, \; t \geq \tau_i. \end{aligned} \tag{4.48}$$

$$\tag{4.49}$$

Since zero is a solution to Equation (4.47), we assume that the mean is zero. Finally, we compute that the derivative of $D^{(0)}(t)$ is $d^{(0)}(t)$, where:

$$\begin{aligned} d^{(0)}\left(S_2^{(0)}(\tau_i)\right) &= \mu_2(Q_2^{(0)}(t) \wedge (n - Q_1^{(0)}(t))^+) + \beta(Q_2^{(0)}(t) - (n - Q_1^{(0)}(t))^+) \\ &\quad +\mu_1(Q_1^{(0)}(t) \wedge n) \end{aligned} \tag{4.50}$$

$$\tag{4.51}$$

when $S_2^{(0)}(\tau_i) \geq \tau_i$. Therefore, we have:

$$\mathsf{Var}\left[W_2^{(1)}(\tau_i)\right] = \frac{\mathsf{Var}\left[\hat{Q}_2^{(1)}(S_2^{(0)}(\tau_i))\right]}{\left(d^{(0)}\left(S_2^{(0)}(\tau_i)\right)\right)^2} \tag{4.52}$$

In Figure 4.1.2, we show the pseudo-code used in implementing our algorithm and computing the fluid approximations using the C-programming language.

```
if (simulation time <= tau)
   compute high and low priority fluid level;
else  /* Simulation time > tau -- Virtual customer arrives. */
     if (abandonment time > 0 AND service time > 0) {
        if (low priority fluid level >= number servers - high priority fluid level)
            AND (abandonment time > 0)
            /* Low priority customer waits in queue (possibly preempted) until
               either entering service or abandoning to high priority queue. */
            Decrement abandonment time;
            Update high and low priority fluid level (with low priority arrival rate = 0);
     else if (abandonment time > 0)
                 /* Low priority customer enters and receives service until
                    high priority preempts that customer from service. */
                 Decrement service time;
                 Update high and low priority fluid level
                        (with low priority arrival rate =0);
     else if (service time > 0 AND abandonment time < 0)
                 /* Low priority customer abandons queue2 and enters
                    end of queue1 as high priority customer. */
                 /*-----------------------------------------------*/
                 /* Compute part1 of low priority waiting time. */
                 /*-----------------------------------------------*/
                 Compute waiting time in low priority queue for
                      "abandoning" (i.e. upgraded) low priority customer;
                 /*-------------------------------------------------------------*/
                 /* Compute part2 of low priority waiting time:                 */
                 /* Begin high priority waiting time process for upgraded customer.*/
                 /*-------------------------------------------------------------*/
                 if (high priority fluid level < number of servers - 1);
                    Compute waiting time for upgraded low priority customer;
     else if (service_time[i][j] < 0 && abandon_time1 > 0)
                /* Low priority customer completed service from low priority
                   queue (without abandoning). */
                Compute waiting time of low priority customer;
```

Figure 4.3: Pseudo-code for Low Priority Mean and Variance of Virtual Waiting Time at Tau Computation.

4.2 Staffing Algorithm

We use the fluid approximations for the mean virtual waiting time to predict an actual staffing level for our call center model. Our criteria for changing the staffing level, or number of servers, in our model uses a comparison of the mean virtual waiting-time for each customer class to their corresponding target waiting-time. The simple staffing algorithm is the following:

1. Choose an initial staffing level, or value for the number of servers, and target service level for the high and low priority customers. These values are determined from our actual call center data.

2. Compute the mean virtual waiting-time using the fluid and diffusion approximations for each customer class.

3. If the percentage of mean virtual waiting-times is greater than the target service level for either class, then increment the number of servers by 1.

4. Repeat the second step until the target service level is satisfied for both classes of customers.

This generates a staffing level, which we use as one of the inputs into our simulation model. Then, we compute the percentage of mean virtual waiting-times that are below the target waiting-time for each class. By comparing this percentage with the corresponding one from the fluid approximations, we can measure the accuracy of the fluid approximations estimate for the staffing level in the real system.

It is important to note that this staffing level is an estimate for the minimum staffing required to meet a target service level. In a real call center setting, a

manager would use this estimate as a constraint in a larger business scheduling optimization model. Such a model optimizes call center costs subject to constraints such as the number of consecutive employee work shifts, scheduled breaks, vacations, etc. Our predicted staffing level is basically an input into this larger model. Therefore, the actual staffing level in a call center might be greater than our estimated level to handle agents finishing their shifts or taking breaks during the day.

4.3 Model Verification

The fluid and diffusion approximations model are verified using three methods. The first method compares a well-known queue result with our estimate for the same queue result. This method uses the stationary, $M/M/1$ preemptive-resume, priority queue where the arrival rates are constant. The mean waiting-time times for high and low priority customers of this queue are known results. Thus, we can compute the mean virtual waiting-time at each τ_i for this queue using our fluid and diffusion approximations. Therefore, our fluid model is verified by matching the mean virtual waiting-time at each τ_i for each customer class with the corresponding mean virtual waiting-time know result.

The second method uses the differential equations that model the change in the fluid level, or mean number in system, for both high and low priority customers in the non-stationary, $M_t/M_t/n$ preemptive-priority queue. At the maximum and minimum number in system values, the derivative of the number in system with respect to time will be zero. We compute the number in system fluid approximations over the entire time horizon, and determine the minimum and maximum values. Then, we set the right-hand side of the differential equations in

Section 4.1.1 equal to zero. Finally, we substitute, into the right-hand-side of the equations, our mean number in system estimates and corresponding parameter values. If the right-hand side of the differential equations are very close to zero, then our mean number in system fluid approximations are accurate. Therefore, our fluid approximations model are verified.

Finally, the last method is model validation, which involves convincing a specific call center manager that our model is accurate. Since the arrival rates, service rates, and number of agents in our model pertain to only one call center, a manager of a different call center can test our model by inputting their own parameters and data. Thus, they will run our model to generate approximations for their call center operations. Then, they can compare our fluid and diffusion approximations with estimates computed using their own approximating scheme. If our model is accurate, then our approximations will be very close to (if not better) than their estimates, upon comparison of both sets of estimates with those from a discrete-event simulation model.

Consequently, we have three different methods for verifying the accuracy of our fluid approximations model.

Chapter 5

Simulation

5.1 Simulation Model

We use a discrete-event simulation to compute estimates for the mean number in the system, mean virtual waiting time, and the variance of the number in system and virtual waiting time, and waiting time tail distribution values for high and low priority customers. The simulation models our call center defined in Chapter 3. The details of the simulation model are outlined below. Our C program code for our non-stationary $M_t/M/n$ model is an extension of Law and Kelton's C program code for the stationary $M/M/1$ queue model found in [47].

We estimate the performance measures, such as the number in system and virtual waiting time, for both customer classes. To obtain these estimates, we first compute the values of the performance measures at distinct time points τ_i, where $i = 1, 2, \ldots, m$. Here, m is the number of subintervals contained in the overall time interval $[0, T]$ of the simulation. Second, we compute these values again at each τ_i for a large number of independent replications. Note that for each τ_i, we repeat a full simulation run, or replication, which causes the run time of our simulation to increase significantly. Finally, we determine the performance

estimates by averaging each value at τ_i over this set of replications.

5.2 Simulation Components

The computer program used to implement the simulation model consists of several components. The simulation starts in the empty-and-idle state, where no customers are present and all of the servers are idle. The basic inputs to the simulation are the arrival, service and abandonment rates for each customer class, the number of servers, the stopping time, and the target waiting-times for each customer class. One run of the simulation is repeated until a given stopping criterion is reached. Here, one run of the simulation is stopped after a finite horizon time is reached, corresponding, for example, to a 20 hour period of real-world a call center operations. However, it can also be stopped after a certain number of customer completions. Independent replications of the simulation are performed until a certain precision of the performance measures is attained. In our case, replications of the simulation are performed until the standard error of the number in system and virtual waiting-time estimates reaches a small precision compared to the estimated values, usually three or four orders of magnitude. Finally, the random numbers, which model the stochastic nature, are generated using a pseudo-random number generator.

5.2.1 Random Number Generator

A simulation of any system, which operates in a random, or unpredictable, manner, requires a method of generating random numbers. In our call center, the arrival, service, abandonment, and priority assignment processes are the source

of the randomness of the system. Thus, we discuss and employ a convenient and efficient method to generate these parameters from their corresponding probability distribution. The samples generated from a specific distribution will be described as random variates.

The methodology of generating random numbers has a long and interesting history. The earliest methods were carried out by hand, by throwing dice or dealing cards, for example [47]. As computers, and thus simulation, became more widely used, research began focusing on random number generation methods that were compatible with the way computers work [47]. In the 1940's, the first *numerical*, or *arithmetic*, generator was proposed from research by Von Neumann and Metropolis. A carefully designed generator can produce numbers that *appear* to be independent draws from the $U(0, 1)$ distribution, in that the sequence of numbers pass a series of statistical tests.

The last statement is a useful definition of random numbers. Additionally, Law and Kelton state that a good arithmetic random number generator has the following properties: [47]

1. The numbers produced should be uniformly distributed on $[0, 1]$ and should be independent of each other.

2. The generator should be fast and efficient storage-wise, in practice.

3. We should be able to reproduce a given stream of random numbers, which aids in debugging and simulating different systems with identical random numbers.

4. The generator should be able to easily produce separate streams of random numbers, where a stream is a subsegment of numbers produced by the

generator and where the next stream begins where the last one ended.

Linear Congruential Generators

Many random number generators today are linear congruential generators (LCGs) introduced by Lehmer [48]. A sequence of integers $Z_1, Z_2, \ldots$ is defined by the recursive formula:

$$Z_i = (aZ_i + c) \,(\text{mod } m) \tag{5.1}$$

where m, the modulus, a, the multiplier, c, the increment, and Z_0, the seed or starting value, are nonnegative integers. From equation 5.1, $0 \leq Z_i \leq m-1$, based on modulo arithmetic. To obtain the random numbers U_i, $i = 1, 2, \ldots$, we let $U_i = Z_i/m$.

If $c = 0$ in equation 5.1, then the generator is a multiplicative linear congruential generator. The majority of LCGs today are multiplicative, to avoid the addition of c. However, this type of generator cannot have full period, m. A generator has full period m if the number of distinct Z_i's generated before any one is repeated is m. Full-period generators are desirable for large-scale simulations using hundreds of thousands of random numbers. However, if m is a prime number, then the period is $m-1$ if a is a primitive element for modulo m. Note that a is a prime element modulo m if the smallest integer l for which $a^l - 1$ is divisible by m is $l = m - 1$ [47]. With m prime and a a prime element modulo m, the generator becomes a prime modulus multiplicative linear congruential generator (PMMLCG).

We use a PMMLCG based on the portable FORTRAN code of Marse and Roberts in our simulation, where $m = 2,147,483,647$ and $a = 630,360,016$ with period $m-1$ corresponding to the maximum number of variates used for a random

variable. There are multiple (100) streams that are supported in the PMMLCG where seeds are spaced $100,000$ numbers apart. In our simulation model, a few thousand independent replications can be made, depending on the values of the input parameters.

Generating Random Variates

There are many techniques that can be used for generating random variates from a given distribution. The first part of every technique is finding a source of $U(0,1)$ random numbers. The LCGs are designed to generate these random sequences of $U_i's$ as described previously.

Once a random number generator is chosen, several algorithms for generating random variates can be used. Some of these are the inverse transform method, the composition method, and the acceptance-rejection method. We use the inverse transform method to generate random variates for our arrival, service, abandonment, and call type assignment processes. We use an exponential distribution (with different rates) to model each of the four random processes.

The inverse transform method can be applied to continuous or discrete probability distributions. Since our distribution is exponential, we use the continuous version of the method. Now, let X be a continuous random variate with distribution function F. Assume that F is continuous and strictly increasing when $0 < F(x) < 1$. Thus, if $x_1 < x_2$ and $0 < F(x_1) \leq F(x_2) < 1$, then $F(x_1) < F(x_2)$. Also, let F^{-1} denote the inverse of the function F, where $F^{-1}(y) = \{x : F(x) = y\}$. Then, Law and Kelton [47] define the general inverse transform algorithm as the following:

1. Generate $U \sim U(0,1)$.

2. Return $X = F^{-1}(U)$.

Note that $F^{-1}(U)$ will always be defined, since $0 \leq U \leq 1$ and the range of F is $[0, 1]$. Now, to demonstrate that the X computed from the algorithm, we must show that for all real numbers x, $P(X \leq x) = F(x)$. Thus, since F is invertible, we have:

$$P(X \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(X)$$

where last equality follows since $U \sim U(0, 1)$ and $0 < F(x) < 1$ [47].

Since we require exponential random variates for our random processes, $F(x)$ in our simulation has the form:

$$F(x) = \begin{cases} 1 - e^{-\frac{x}{\beta}} & \text{if} \quad x \geq 0; \\ 0 & \text{if} \quad x < 0. \end{cases}$$

Note, however, that in our simulation, we use $1/\beta \cdot ln(U)$ instead of $1/\beta \cdot ln(1-U)$ to generate our exponential random variates.

5.2.2 Timing Process

The timing process is used to determine which event occurs next. The possible events are an arrival to the system, an abandonment from the low priority queue (queue 2), or a departure from one of the servers. A random arrival and departure time are computed from the random number generator using different streams. Also, a deterministic abandonment time is computed for all low priority customers in the queue. The event with the corresponding minimum time, among all these events, becomes the next event. Finally, the simulation clock time is advanced to the time of this next event and any performance measure statistics are updated.

5.2.3 Arrival Process

One of the main components of the stochastic simulation is the arrival process. We choose to approximate the true arrival rate function as a piecewise linear function over a set of disjoint 30-minute time subintervals, $(t_a, t_{a+1}]$, which partition the overall finite-time horizon interval $[0, T]$, where $a = 1, 2, \ldots, m-1$ and m represents the number of 30-minute subintervals. Thus, the arrival time of the k-th customer with priority i, $i = 1, 2$, A_{ik}, is used to advance the overall simulation time, S, where $S \leq T$, into the next time subinterval. We compute the arrival time, A_{ik}, by generating a random inter-arrival time, X_{ik}, between customer $k-1$ and k, and adding X_{ik} to the current simulation time S. Since the arrival process for each priority class is Poisson, X_{ik} has an exponential distribution with arrival rate($\lambda_i(t)$. Therefore, it can be generated using the inverse transform method.

Since our model supports two types of customers, λ_t is the overall arrival rate and is define as:

$$\lambda_t = \lambda_{1t} + \lambda_{2t} \tag{5.2}$$

where the arrival rates for the high priority customers, λ_{1t}, and the low priority customers, λ_{2t}, also vary with time t. Now, we randomly determine the call type of each customer upon their arrival. Here, based on Poisson thinning, a customer will have call type i with time-varying probability $\lambda_{it}/\lambda_t \ \forall t \epsilon [0, T]$.

Now, an arriving customer who finds at least one server idle enters service immediately at some server, n_i, $i = 1, 2, \ldots, n$, where n is the total number of servers. Server n_i is chosen from all the other idle server using an ordered search algorithm. In other words, if servers 1 and 2 are both idle, then server 1 is chosen to provide service. In practice, calls are switched to agents in this manner,

although more efficient methods exist depending on the type of call centers. Once the customer enters service, we generate an exponentially distributed service time, Y_k, with mean service time $1/\mu$ (independent of time), for the k-th customer using the inverse transform method.

If all the servers are busy, the customer either enters the appropriate queue, or preempts lower priority customers already in service. Now, if a low priority customer enters the queue, then its "abandonment" time, or the time until it leaves the low priority queue and enters the end of the high priority queue, is computed. If the arriving customer preempts a lower priority customer at some server, n_i, then the preempted customer is placed at the head of the appropriate queue. The preempted customer, and its server, is chosen in an ordered search algorithm of all low priority customers in service at the time of the high priority customer's arrival. For example, if there are 3 low priority customers at servers, n_4, n_7, and n_{10}, respectively, then the low priority customer at server n_4, will be preempted. Next, the arriving high priority customer is sent to the now-vacant server. In practice, this preemption approach is used; however, more complicated, efficient algorithms can be implemented, especially when the skills sets of agents vary. Finally, the departure time for the arriving customer, as well as the arrival time of the next customer, is generated.

5.2.4 Abandonment Process

There is the "abandonment", or upgrade, process from the low priority queues. In the real-world system, a call center manager would allow the low priority e-mails to be upgraded only after their waiting time has reached its limit, or abandonment time. Consequently, in our simulation model, an e-mail cannot

be upgraded before another e-mail that arrived previously to the system. Thus, each low priority customer has the same abandonment time and its abandonment rate, β, equals $1/s$, where s is the target service level time for the low priority customers, i.e., 2 hours. Thus, the abandonment rate is constant for all low priority customers. These low priority customers would receive service If a low priority customer abandons queue 2, then it is "upgraded" to the high priority queue, i.e., queue 1. Note that the upgraded customer is placed at the end of queue 1, and its priority changes from low to high. In this manner, the low priority customer's priority becomes *dynamic*. A customer that abandons queue 2 for queue 1 receives service as a high priority customer, but its performance, such as virtual waiting time in queue, is measured as if it is still a low priority customer. However, upgraded customers are added to the count for the number of high priority customers in the system.

5.2.5 Departure Process

The other main component is the departure process. Here, a customer leaves the system after completing service at some server, n_i. Thus, the server n_i is now available, and the next customer to enter service at server n_i is chosen based on its priority. If there are any customers in the high priority queue (queue 1), then the customer at the head of the queue will enter into service. If, however, there are no customers in queue 1, then the customer at the head of the low priority queue, queue 2, will enter into service. Of course, if there are no customers in either queue, then server n_i remains available, or idle, until a new customer enters the system.

The total waiting time in queue, or delay, for the customer entering service at

server n_i is computed based on its priority. If it is a low priority customer, then a high priority customer may preempt it from service. This low priority customer might have entered service (not necessarily at server n_i) several times before it completed service and departed the system. Therefore, its class 2 delay, D_2, is defined as:

$$D_2 = \sum_{i=1}^{p} D_2^i \ , \tag{5.3}$$

where D_2^i is the i-th class 2 delay, and p is the total number of previous class 2 delays. Note that p varies for each low priority customer. However, a high priority customer only enters the queue once (upon arrival). Thus, it has a single delay term, D_1.

Now, after a customer's delay is computed, its value is compared to a target delay for that customer class. If the customer's delay is less than or equal to the target delay, then for the appropriate customer class, the number of delays meeting the target are incremented by one. This statistic is required to compute the empirical delay, or waiting time in queue, distribution for each class.

5.2.6 Delay Process

Another main component of our simulation is the delay, or waiting time in queue, computation process. For the high priority customers, the delay computation is relatively straight-forward. If there is an idle server, or a server busy with a low priority customer, available upon arrival to the system, then the customer's delay equals 0. However, the customer might have to enter the high priority queue before an idle server becomes available and the customer can receive service. Once the customer departs the high priority queue, its arrival time to the queue is subtracted from its departure time from the queue. If all the servers were busy

upon the customer's arrival, then its arrival time to the queue is equivalent to its arrival time to the system. Finally, the departure time from queue is actually the current simulation, or system, time. Thus, the two quantities used to compute the delay are relatively easy to determine.

For the low priority customers, the overall delay computation can have two additional parts. One part is exactly as discussed above for the high priority customers. However, if the low priority customer is preempted from service, then it reenters the low priority queue. This new delay, or "preemption" delay, is added to its previous delay. Now, if the customer is upgraded to the high priority queue, before completing service, then its delay in the high priority queue is added to its preemption delay, and any other previous delays before preemption to accurately compute its overall delay. Although the overall delay process for the low priority customers can be complicated, the quantities required to track and update all the delay parts can be computed in our discrete-event simulation.

5.2.7 Virtual Waiting Time Methodology

The final main component in our simulation is the virtual waiting time in queue, as defined in Chapter 4. The actual customer waiting time in queue is used to compute the main performance estimator of the simulation, namely the virtual waiting time in queue.

In the simulation, the actual waiting time is computed by tracking each customer as it moves through the system. Once a customer arrives, it is assigned a priority level and either receives service immediately, or waits in queue before receiving service. Low priority customers are allowed to abandon their queue and enter the high priority queue after waiting beyond a given amount of time.

Also, high priority customers, upon their arrival, are allowed to preempt a low priority customer from service. Now, from the simulation, each customer's arrival time, queue time, queue abandonment time, service time, and departure time are computed. Therefore, we can compute each customer's waiting time in queue.

The virtual waiting time is computed for both high and low priority customers at a fixed sequence of time points, τ_i where $i = 1, 2, \ldots, m$ that do not necessarily correspond to the random arrival times of our simulation. Thus, for any given simulated sample path, we compute the virtual waiting time, $V(\tau_i)$, at each τ_i as a function of the actual waiting time, $\hat{W}_i$, for the first customer arrival after τ_i. Before we explain the virtual waiting time computation, we define some variables. Let $N_1(t)$ and $N(t)$ represent the number of high priority customers and the total number of customers in the system, respectively. Also, let $\hat{\tau}_i$ be the first arrival epoch after τ_i. Note that if $N_1(\tau_i) < n$, then $V(\tau_i) = 0$ for a high priority customer. Similarly, if $N(\tau_i) < n$, then $V(\tau_i) = 0$ for a low priority customer. For both high and low priority customers, there are two other cases for computing $V(\tau_i)$. These cases are basically identical, except that the term $N_1(t)$, used for high priority customers, is replaced by $N(t)$ for low priority customers. Therefore, we define only the cases for high priority customers below:

Let $\tau_i^* = min\{t \geq \tau_i : N_1(t) < n\}$. Then:

1. Case 1: If $\tau_i^* < \hat{\tau}_i$, then $V(\tau_i) = \tau_i^* - \tau_i$.

2. Case 2: If $\tau_i^* \geq \hat{\tau}_i$, then $V(\tau_i) = \hat{W}_i + \hat{\tau}_i - \tau_i$.

We provide a graphical representation of the two cases for a generic customer in Figure 5.1 and Figure 5.2.

Consequently, we can compute the virtual waiting time of a customer arriving at τ_i by adjusting the actual waiting time for the first arrival after τ_i. In our

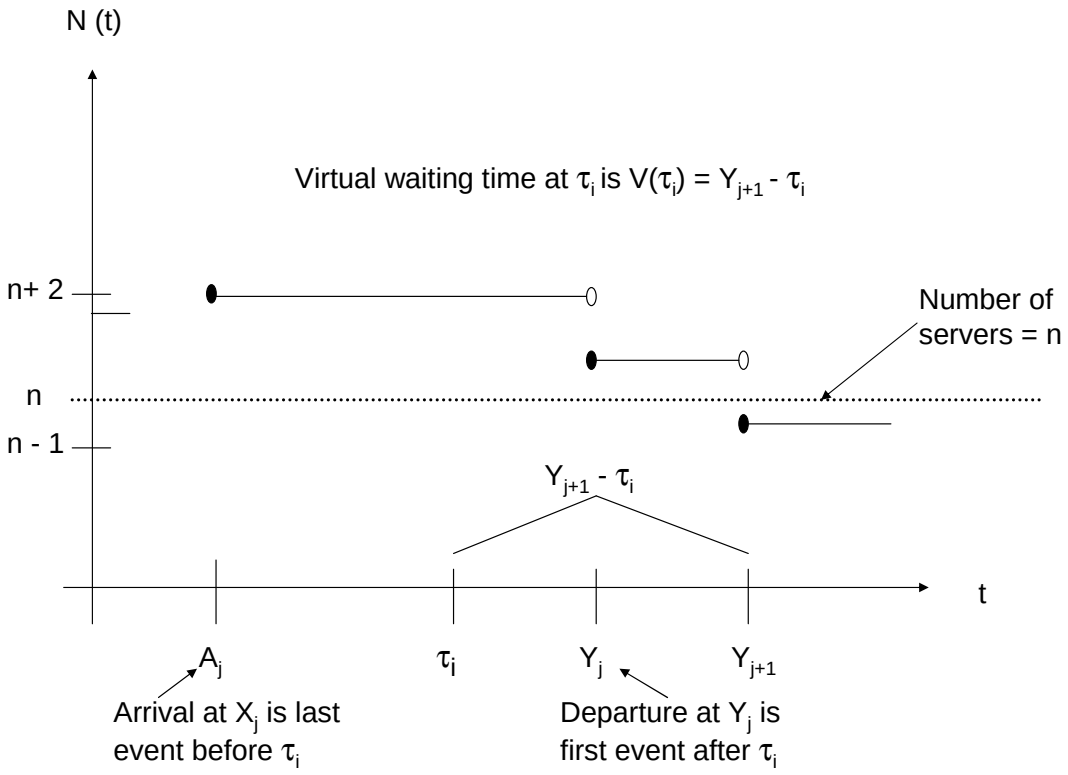

Figure 5.1: Virtual Waiting Time Computation–Case 1

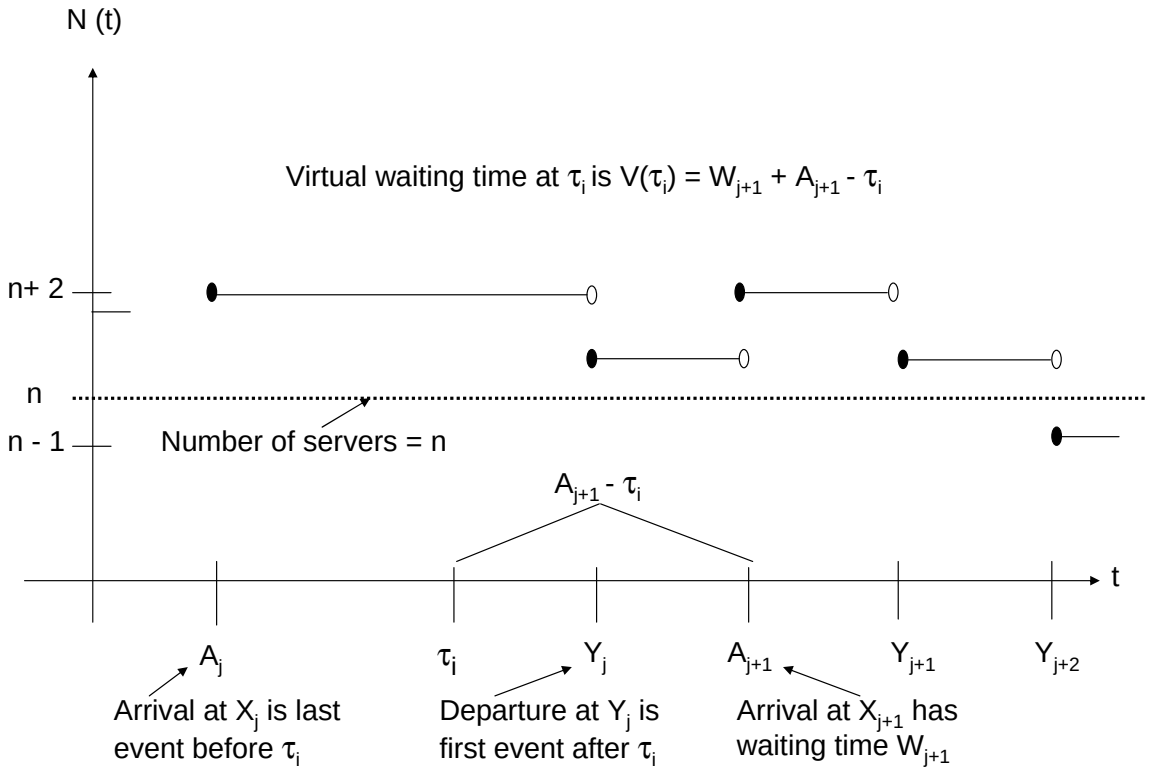

Figure 5.2: Virtual Waiting Time Computation–Case 2

simulation code, we check the number of customers in the system left after each departure epoch greater than τ_i, and less than $\hat{\tau}_i$. We mark τ_i^* as the first of those departure epochs where the number in system drops below the number of servers. Depending on whether any such epochs exist, we compute $V(\tau_i)$. Consequently, we can compare these values with those from the fluid and diffusion approximations.

5.2.8 Performance Estimation

In the report process, the estimates of the performance measures, such as the mean customer delay and the probability that the mean delay is less than (or equal to) some target probability, are computed. Here, the estimates are derived from statistics computed in the timing, arrival, and departure processes. Some of these statistics are the following:

1. Arrival time, A_{jk}, for k-th customer, C_k, with priority j.
2. Service time, Y_{jk}, for k-th customer, C_k, with priority j.
3. Abandonment time, B_k, for k-th customer, C_k, with priority j.
4. Time k-th customer with priority j enters queue, QIn_{jk}, and exits queue, $QOut_{jk}$.
5. Time k-th customer with priority j departs system, D_{jk}.
6. Number of class j arrivals during one simulation replication, NA_j.
7. Number of class j virtual customer delays, ND_j, $j = 1, 2$.
8. Number of class j virtual delays less than given class j target delay, $NDBelow_j$, $j = 1, 2$.

9. Total number of class j customer delays, TD_j, $j = 1, 2$.

We now define mathematically our two main estimators. The first one estimates the expected mean delay of the customers with arrival priority j, $j = 1, 2$, computed over R independent replications. Remember that low priority customers have a dynamic priority. If they are upgraded, then they have a low priority and high priority component to their expected mean delay estimator. Note that the delay, D_k, of the first arrival after time τ_i is used to compute the virtual delay, $V(\tau_i)$ at time τ_i as defined in Section 5.2.7. The second one estimates the expected mean value of the probability that the mean delay at time τ_i is less than a given target delay. In other words, it estimates the $P(V(\tau_i) < v^*)$ where v^* is the given target delay. Now, we define the high priority mean delay estimator as the following:

$$W_1 = \sum_{r=1}^{R} \frac{\sum_{k=1}^{NA_1} \frac{D_{1k}}{NA_1}}{R} \tag{5.4}$$

where W_1 is the expected mean high priority delay, and D_{1k} is the kth high priority customer delay, where $D_{1k} = QOut_{1k} - QIn_{1k}$, and R is the total number of replications. Next, we define the low priority mean delay estimator as the following:

$$W_2 = \sum_{r=1}^{R} \frac{\sum_{k=1}^{NA_1} \frac{D_{2k}}{NA_1}}{R} \tag{5.5}$$

where W_2 is the low priority delay, and D_{2k} is the kth low priority customer delay, where $D_{2k} = \sum_{\mathcal{C}} QOut_{2k} - QIn_{2k} - Y_{2k}$, where $\mathcal{C}$ is total number of times the low priority customer is preempted. Next, we use the general well-known formula to compute the variance estimate, $S^2(n)$, of our two main performance estimators:

$$S^2(n) = \sum_{i=1}^{n} \frac{(X_i - \bar{X})^2}{n-1} \tag{5.6}$$

where X_i is the i-th of a general random variable and n is the number of observations. We use Equation 5.6 to estimate the variance of our number in system and virtual waiting time estimators at time τ_i over the R replications. In addition, we compute the standard error for our simulation estimates by taking the square root of $S^2(n)$. Finally, we define the delay distribution estimator as:

$$ED_j = \sum_{r=1}^{R} \frac{P_{jr}}{R}. \tag{5.7}$$

Here, ED_j is the expected mean value of the empirical delay distribution for class j customers, $j = 1, 2$, and P_{jr} is the probability that the mean virtual delay at τ_i is below a given target for replication r, where $P_{jr} = \frac{NDBelow_{jr}}{ND_{jr}}$. Note that, by definition, ND_{jr} and $NDBelow_{jr}$ are the number of class j customer delays and number of class j delays less than the given class j target delay for replication j. Also, ND_{jr} and $NDBelow_{jr}$ are each bounded above by the total number of τ_i's in the simulation.

5.3 Model Verification

We use a $M/M/n$, two-class, preemptive-resume priority model to verify the basic operation of our simulation model, independent fluid approximations model. Thus, using stationary results for the mean waiting time from Wolff [73], we compared our mean waiting time estimates for high and low priority customers to those of Wolff's, after converting our non-stationary simulation model to a stationary one. In addition, we computed a log file tracking each customer through the simulation model. For example, each customer's arrival, waiting (queue), and departure time is reported. Also, for each customer, the number of customers in the system upon their arrival and after their departure is reported, as is the num-

ber of the server which provided them service. Finally, low priority customers are marked in the log file if they were preempted and/or upgraded to high priority status. Using all of the above detailed data, we are able to verify the number in system and virtual waiting time computations, and overall performance of the simulation.

We also verify our simulation results with those from a different call center simulation model developed by Rodney Wallace [69]. Wallace uses the method of batch means to estimate his performance measures, whereas we use the method of independent replications. His simulation models a stationary, skill-based routing, $M/M/n/L$ call center, where each server can handle only certain types of customers. However, our simulation models can be synchronized to compute comparable estimates of the mean waiting time in queue for a high priority and low priority customer class. Mainly, he allows his servers to handle any type of customer and we change our arrival process from a stationary one to a non-stationary one. After increasing the run length in our models to reduce the impact of any initial transient behavior, we obtain estimates very close to those from Wallace's simulation. Therefore, we verify our simulation model in several different ways.

Chapter 6

Results of Model Comparison

6.1 Overview

We perform numerical computations to compare the performance measure estimates from our fluid approximations model and simulation model. First, we show the difference in performance among different service disciplines in our simulation models. Next, we discuss our main comparison of our fluid approximations to our simulation estimates for performance measures from the $M_t/M/n$, two-class, preemptive-resume, dynamic priority queue model. Stochastic simulation results require a number of independent replications, whereas the fluid approximations require the numerical integration of a set of seven ordinary differential equations. Although we also state the diffusion approximations and simulation estimates for the variance of the number in system and virtual waiting time, we only present the comparisons between the fluid approximations and simulation estimates for the following group of performance measures:

- mean number in system for the high and low priority customers, and
- mean virtual waiting time for the high and low priority customers.

We computed the numerical results by implementing both our fluid approximations and discrete-event simulation estimates using the C programming language code compiled on a Windows-based operating system. We use the Microsoft Visual $C++$ package to edit and compile our C code.

6.1.1 Call Center Data

We begin our computations by defining the values of our queueing model arrival rates, service rates, abandonment rates, number of servers, and time horizon. The parameter values, such as the arrival rates for each customer class, were taken from a real-world, help desk call center, in which calls represent requests for IT support (e.g., network support, password resets, application support, etc.). We simulated the help desk over a 12-hour, or 720-minute, day, starting at 6:00 AM and ending at 6:00 PM, in our models. Thus, each independent replication in the simulation and fluid approximations represents the performance of the help desk over the course of a day. In the fluid and simulation methods, we modelled the time-varying arrival rate function, λ_t as a piecewise constant function, and use per-minute rates for all relevant parameters. Thus, if T is our 720 minute time interval, then λ_t varies every 30 minutes of the time horizon, where $t = 0, \ldots, T$.

The individual arrival rate functions, λ_i, $i = 1, 2$, for each customer class are shown in Figure 6.1. In practice, arrival data is collected by recording the number of customers who contact the call center during each 30-minute interval of the day. The data we use was averaged over a period of a week, or 5 business days. The high priority customer calls are telephone, or voice, calls, while the low priority customer calls are e-mails. The arrival process of calls is modelled as a non-stationary Poisson process, where the Poisson process is "thinned" into

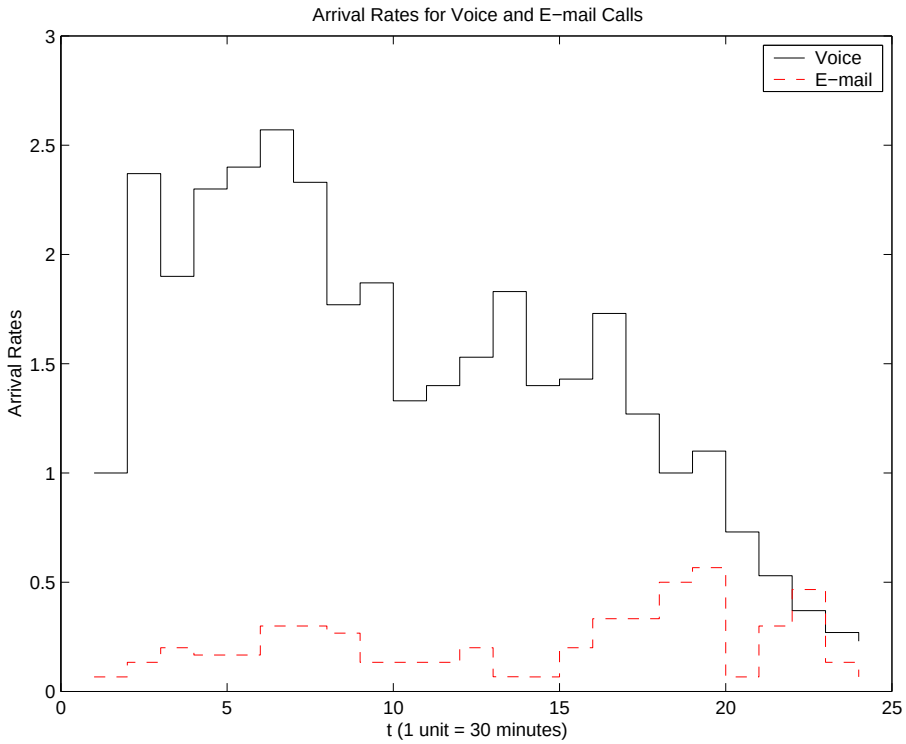

Figure 6.1: Arrival Rates for High (Voice) and Low (E-mail) Priority Customers

two streams for the high and low priority customers.

Many types of call centers today process both voice calls and e-mails. However, there are some challenges in gathering information about e-mail customer interaction with call centers. For example, managers collect more detailed information on parameters for telephone calls than for e-mails. Thus, parameters, such as service rates, are not often collected for each e-mail that arrives to a call center. Some call center managers simply use a "best effort", non-scientific approach to respond to e-mails whenever agents have free time. Some separate the group of agents that respond to e-mails from the group that respond to voice calls in order to limit training and other costs. In our model, we assume that a single group of agents has the necessary skills to respond to both voice calls

and e-mails. Large call centers can usually afford to handle multiple types of customer, i.e., blend their customer types, with a single group of multiple-skilled agents.

We use a mean service time of 8.69 minutes, or 521.29 seconds, for high priority calls. Since the service rate, μ_1, is the reciprocal of the mean service time, we have $\mu_1 = 1/8.69 = 0.1151$ customers per minute. In our help desk, the mean service times are not reported for the low priority, or e-mail, customers. Thus, we set the service rate, μ_2, for the low priority customers equal to that of the high priority customers, which is not an unreasonable assumption in practice. Therefore, $\mu_2 = \mu_1 = 0.1151$ customers per minute.

Our abandonment time for low priority customers is based on the target service level of the class. There is no industry standard for target service levels for non-voice, or low priority, customers, such as e-mails. Some call center managers assign a target service level of either 2, 8 or 24 hours to e-mails [61]. The chosen service level depends on the individual manager's commitment to satisfying low priority customers. In our model, we used a target service level time of 2 hours. Therefore, we allow low priority customers to abandon their queue after waiting for 90 minutes, in an attempt to satisfy their service level requirement of 2 hours.

Finally, we set the number of agents, or servers, $n = 20$. Since we are using asymptotic limits for the fluid approximations, we must scale both the arrival rates and the number of agents towards infinity in order to compute accurate fluid estimates. We use a scale factor of 35 to compare our fluid and simulation estimates. Scale factors above 35 did not produce estimates that were appreciably more accurate. We used server utilizations, $\rho_t = \lambda_t/n\mu$, that vary over time between 0.1302 and 1.245, where the maximum value occurs between 8:30 and

9:00 AM and the minimum value occurs from 5:30 to 6:00 PM. Therefore, our system progressed through over-loaded ($\rho_t > 1$, i.e., unstable) and under-loaded ($\rho_t < 1$, i.e., stable) phases, as the arrival rates vary over time.

6.2 Numerical Results

We present several types numerical results in this chapter. First, for our $M_t/M/n$, two-class, priority simulation model, we show a comparison of three different service disciplines. Thus, we compare the non-preemptive, static priority service discipline with both the preemptive-resume, static and preemptive-resume, dynamic priority service disciplines. Second, for our $M_t/M/n$, two-class, priority fluid and diffusion model, we examine the importance of scaling and its impact on the accuracy of our performance measure approximations. Next, we show the main comparison of our fluid and diffusion approximations to our simulation estimates after using am appropriately scaled system. Finally, we exhibit the results of the staffing algorithm used to optimize the number of agents required to satisfy the given service levels for both classes of customers.

6.2.1 Non-preemption vs. Preemption Priority

We compare four different types of service disciplines for our $M_t/M/n$, two-class queue using our simulation model. The four service discipline types are:

- non-preemptive, static priority,
- non-preemptive, dynamic priority,
- preemptive-resume, static priority, and

- preemptive-resume, dynamic priority.

The non-preemptive priority discipline is the simplest of the four priority disciplines to model. Kleinrock [41] and Wolff [73] both state analytical results for the number in system and waiting time performance measures for non-preemptive priority queues. Kleinrock even provides some analytical results for non-preemptive, dynamic priority queues.

Under the non-preemptive, static priority service discipline, high priority customers are allowed to receive service before low priority ones, without being preempted. Airlines, for example, use this type of system to seat passengers, with first-class customers being seating before lower priority customers. The low priority customers receive service either when the number of high priority customers in the system is less than the number of available servers, or all the high priority have been serviced. If all available servers are busy with high priority customers, then the low priority customers must wait in queue, i.e., line. Also, when a low priority customer begins their service, the customer can not be preempted from service by a high priority customer. Thus, a low priority customer is guaranteed to complete service once service begins. However, if there is a constant presence of a large number of high priority customers, then a low priority customer may have to wait an extremely long time before beginning service. If this happens too often, then customer satisfaction would suffer. Under the non-preemptive, dynamic priority discipline, low priority customers can upgrade their status to high priority, after waiting a fixed amount of time. Therefore, these customers would not be subjected to extremely long waiting times.

Compared to the non-preemptive, priority disciplines, the preemptive-resume, priority disciplines are more complex to model. The main difference between the

two is that a low priority customer can be preempted from service by a high priority customer. Again, if there is a large number of high priority customers in the system over an extended period of time, then the low priority customer would be preempted often. In such cases, under the preemptive-resume, static discipline, low priority customers may have to wait an extremely long time to complete service, which would lower customer satisfaction. In addition, the preempted customer's remaining service time must be tracked so that the customer may resume service from the time at which their initial service ended. However, under the preemptive-resume, dynamic priority discipline, the low priority customer is upgraded to high priority status, when their waiting time has an exceeded a given amount of time. Now, the customer is guaranteed to complete service at some point on time. Thus, the preemptive-resume, dynamic priority discipline is the most difficult of the four service disciplines to model.

Using our simulation model, we compute the mean number in system and mean virtual waiting time under the four service disciplines. We then perform the following comparisons of the performance estimates between service disciplines:

- non-preemptive, static priority vs. preemptive-resume, static priority,
- preemptive-resume, static priority vs preemptive-resume, dynamic priority,
- non-preemptive, static priority vs. non-preemptive, dynamic priority, and
- non-preemptive, dynamic priority vs. preemptive-resume, dynamic priority.

Our goal is to quantify how the non-preemption and preemptive-resume service disciplines, and the static and dynamic priority types, affect the performance estimates.

Non-Preemption, Static vs. Preemptive-Resume, Static Priority

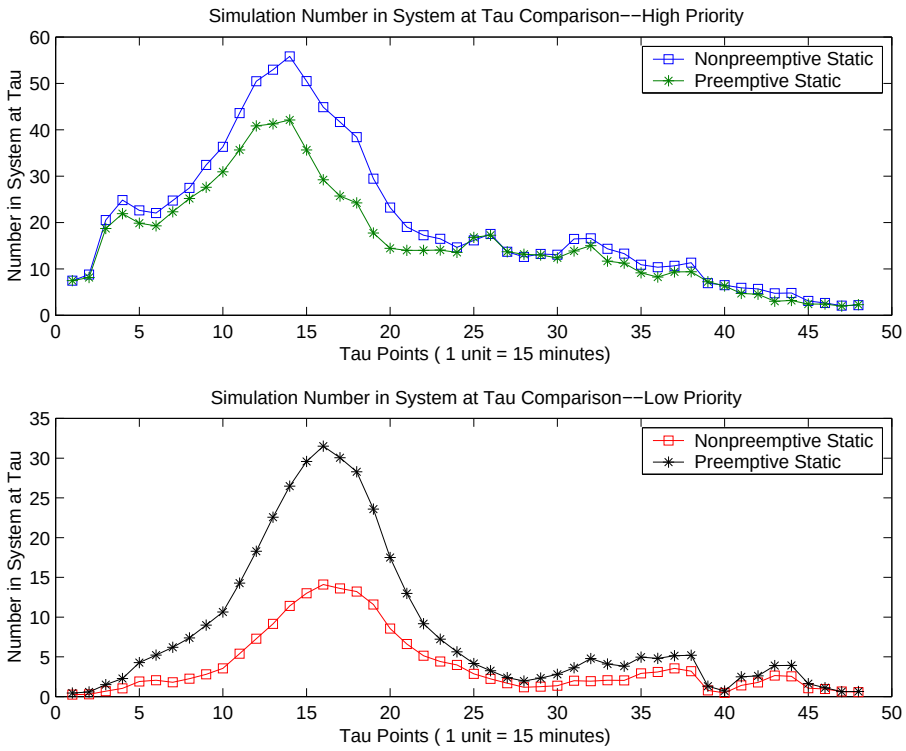

Figure 6.2: Simulation Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Non-preemptive, Static vs. Preemptive-resume, Static Comparison

We show the difference in the mean number in system for high and low priority customers between the service disciplines in Figure 6.2. We report the estimates at time points τ_i, $i = 1, \ldots, m$, where $m = 48$ is the number of 15 minute subintervals within our total time interval of 720 minutes. We use the arrival rate function discussed in Figure 6.1 above. In this comparison, the mean number in system for both priority classes is higher under the non-preemptive, static discipline than the preemptive-resume, static discipline.

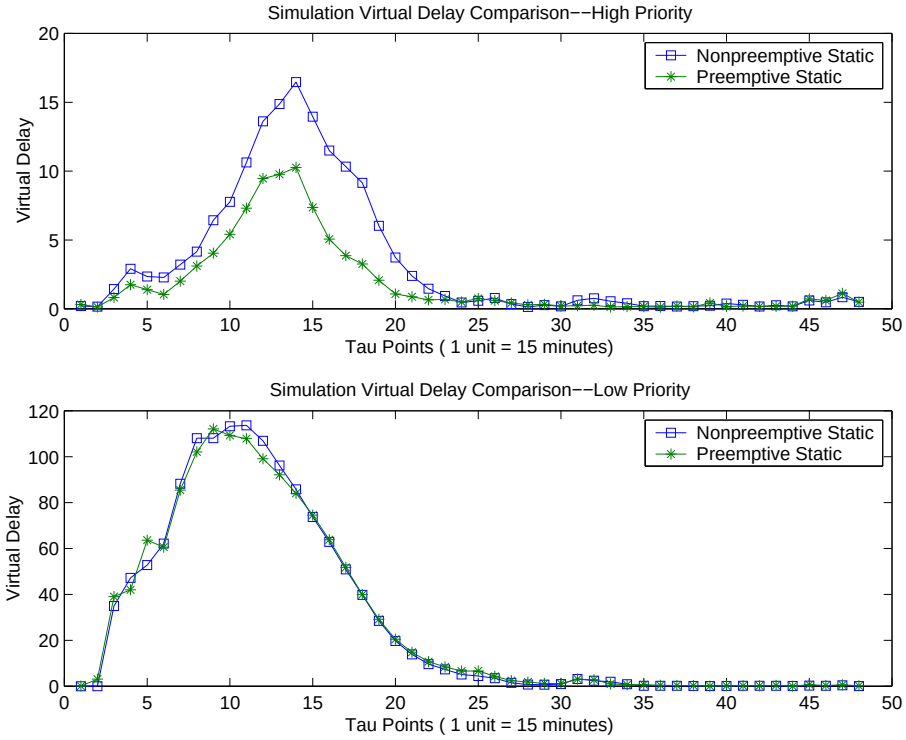

Figure 6.3: Simulation Estimates of the Virtual Delay for High and Low Priority Customers for the Non-preemptive, Static vs. Preemptive-resume, Static Comparison

In Figure 6.3, we show the difference in the mean virtual waiting time for high and low priority customers between the service disciplines. For this comparison, the mean virtual waiting time for the high priority customers is shorter under the preemptive-resume, static discipline than the non-preemptive, static discipline. In addition, the mean virtual waiting time for the low priority customers is basically the same for both disciplines. Thus, the higher priority customers receive better service under the preemptive-resume, static priority scheme, without significantly affecting quality of service of the low priority customer service.

Preemptive-Resume, Static vs. Preemptive-Resume, Dynamic Priority

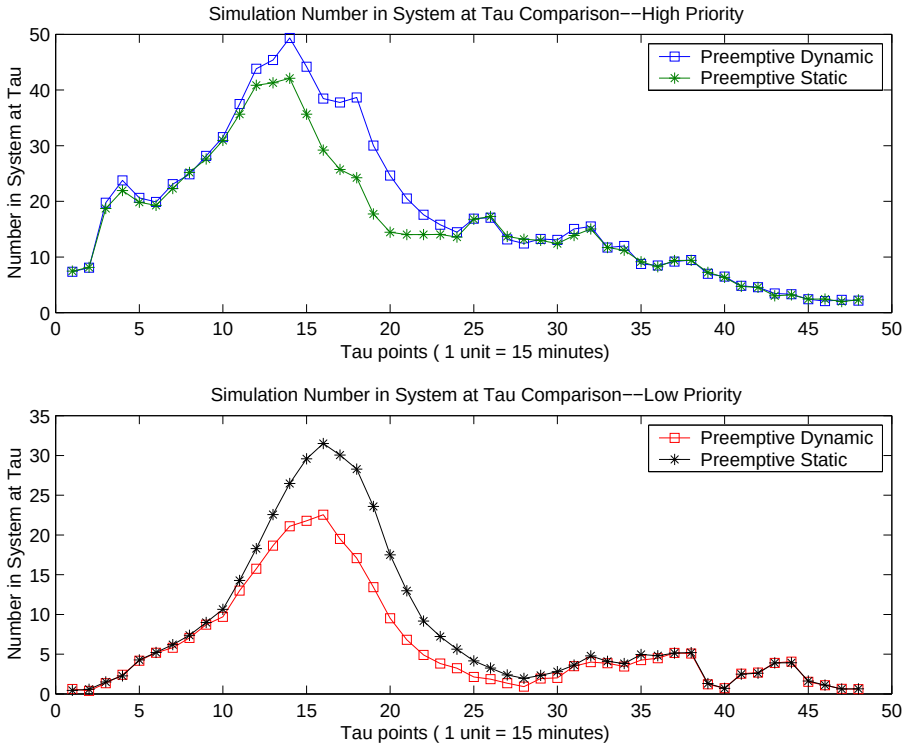

Figure 6.4: Simulation Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Preemptive-Resume, Static vs. Preemptive-resume, Dynamic Comparison

We show the difference in the mean number in system for high and low priority customers between the service disciplines in Figure 6.4. Here, the mean number in system is larger under the preemptive-resume, dynamic discipline for the high priority customers. However, the mean number in system is smaller under the preemptive-resume, static discipline for the low priority customers.

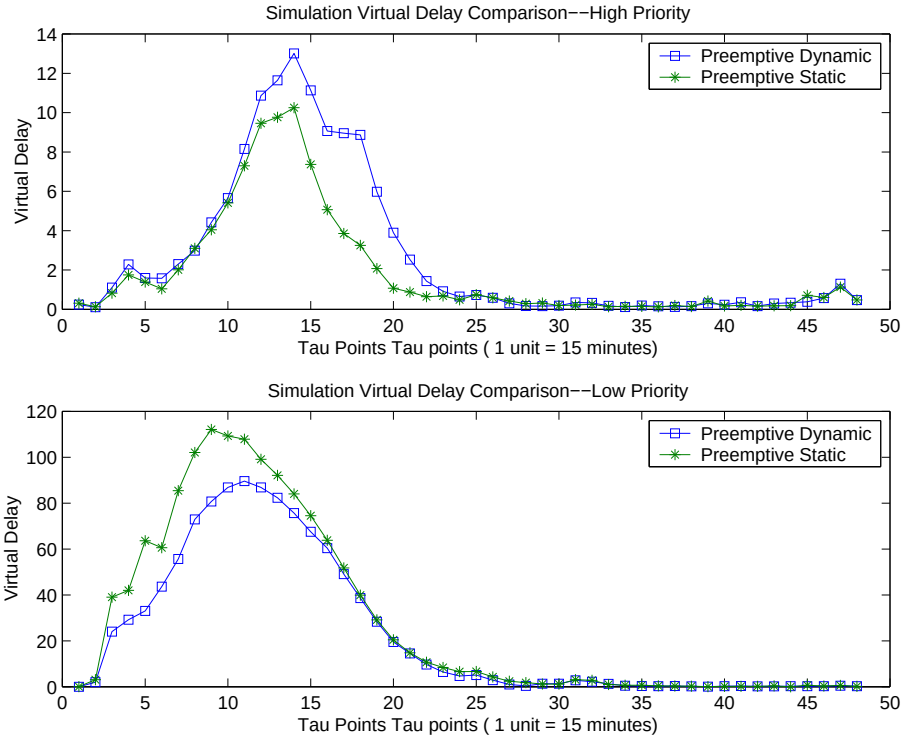

Figure 6.5: Simulation Estimates of the Virtual Delay for High and Low Priority Customers for the Preemptive-Resume, Static vs. Preemptive-resume, Dynamic Comparison

In Figure 6.5, we show the difference in the mean virtual waiting time for high and low priority customers between the service disciplines. The mean virtual waiting time is lower under the preemptive-resume, static discipline for the high priority customers, while the mean virtual waiting time is lower under the preemptive-resume, dynamic discipline for the low priority customers. Thus, the higher priority customers receive better service under the preemptive-resume, static priority scheme, while the low priority customers receive better service under the preemptive-resume, dynamic service.

Non-Preemption, Dynamic vs Preemptive-Resume, Dynamic Priority

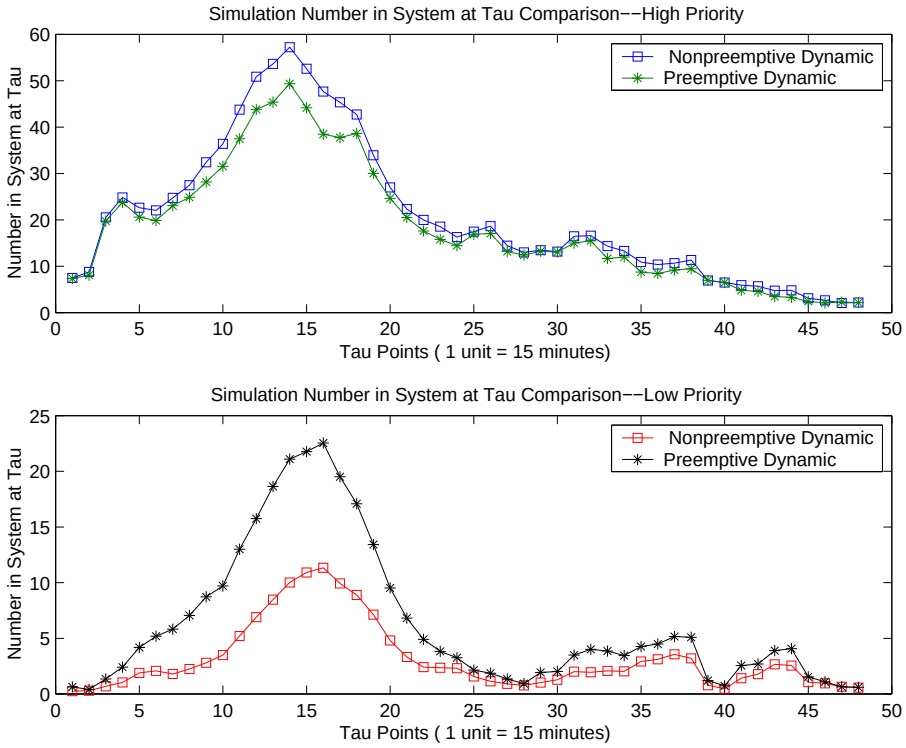

Figure 6.6: Simulation Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Non-preemptive, Dynamic vs. Preemptive-resume, Dynamic Comparison

We show the difference in the mean number in system for high and low priority customers between the service disciplines in Figure 6.6. In this comparison, the mean number in system is larger under the non-preemptive, dynamic discipline for the high priority customers, while the mean number in system is smaller under the same discipline for the low priority customers.

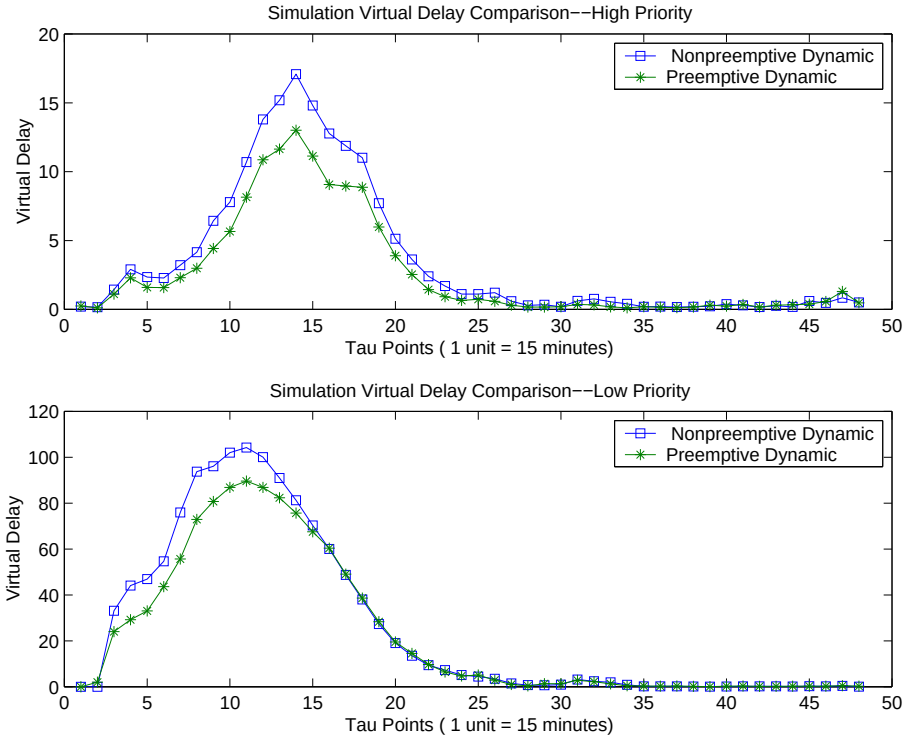

Figure 6.7: Simulation Estimates of the Virtual Delay for High and Low Priority Customers for the Non-preemptive, Dynamic vs. Preemptive-resume, Dynamic Comparison

In Figure 6.7, we show the difference in the mean virtual waiting time for high and low priority customers between the service disciplines. For this comparison, the mean virtual waiting time is shorter for both the high and low priority customers under the preemptive, dynamic discipline. Thus, both classes of customers receive a better quality of service under the preemptive-resume, dynamic priority scheme.

Non-Preemption, Static vs Non-Preemption, Dynamic Priority

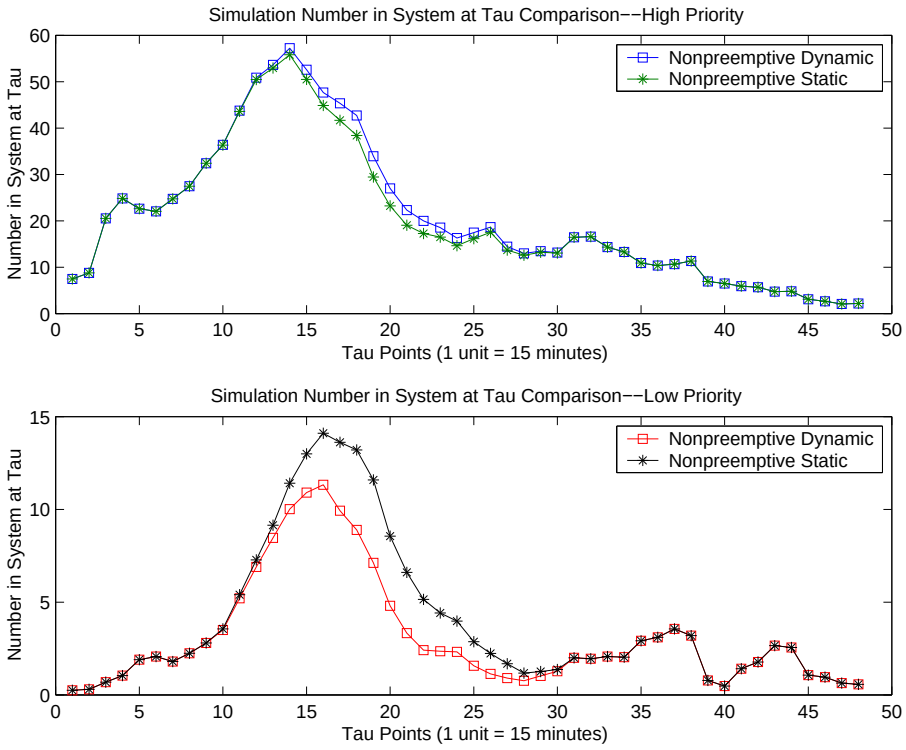

Figure 6.8: Simulation Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Non-preemptive, Static vs. Non-preemptive, Dynamic Comparison

We show the difference in the mean number in system for high and low priority customers between the service disciplines in Figure 6.8. Here, the mean number in system is slightly larger under the non-preemptive, dynamic discipline for the high priority customers, and smaller under the non-preemptive, dynamic discipline for the low priority customers.

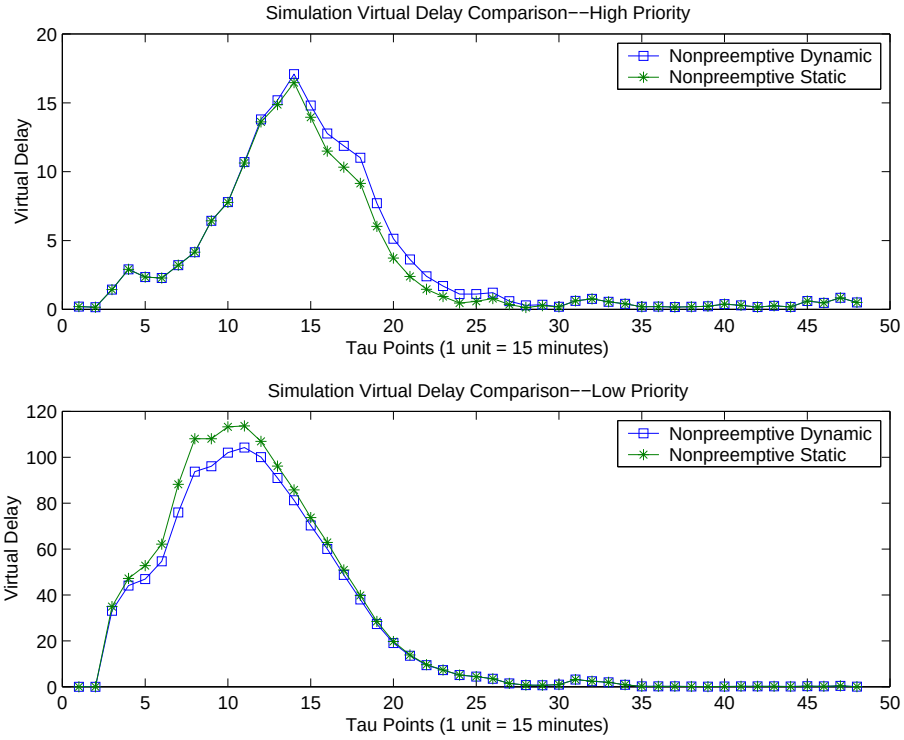

Figure 6.9: Simulation Estimates of the Virtual Delay for High and Low Priority Customers for the Non-preemptive, Static vs. Non-preemptive, Dynamic Comparison

In Figure 6.9, we show the difference in the mean virtual waiting time for high and low priority customers between the service disciplines. In this comparison, the mean virtual waiting time is slightly longer for the non-preemptive, dynamic discipline for the high priority customers. Also, the mean virtual waiting time is shorter for the non-preemptive, dynamic discipline for the low priority customers. Thus, the lower priority customers receive better service under the non-preemptive, dynamic priority scheme, without significantly affecting the high priority customer quality of service.

Finally, we summarize the effects of the non-preemption and preemption-resume service disciplines, and the static and dynamic priority types. For our model, the preemptive-resume discipline provides at least the same, and sometimes better, quality of service than the non-preemptive discipline for both customer classes. Also, the dynamic priority type provides a better service for both customer classes under the non-preemptive discipline. However, under the preemptive-resume discipline, the dynamic priority type provides better service only for the low priority customers, while the low priority type provides better service for only the high priority customers. Therefore, both the preemptive-resume discipline and dynamic priority type can provide a significant impact on the quality of service of the two customer classes.

6.2.2 Importance of Scaling

Recall that the fluid and diffusion approximation method is an asymptotic method. Thus, its estimates converge asymptotically to those of the simulation under the Halfin-Whitt regime [27]. In this regime, the offered load ρ^* remains fixed as λ_t and n are increased by a factor η, where η approaches ∞. This maintains the

offered load ratio while creating an asymptotic limit. We use a scale factor of $\eta = 35$. Without this scaling regime, our fluid estimates for the mean number in system and mean virtual waiting time would not be accurate.

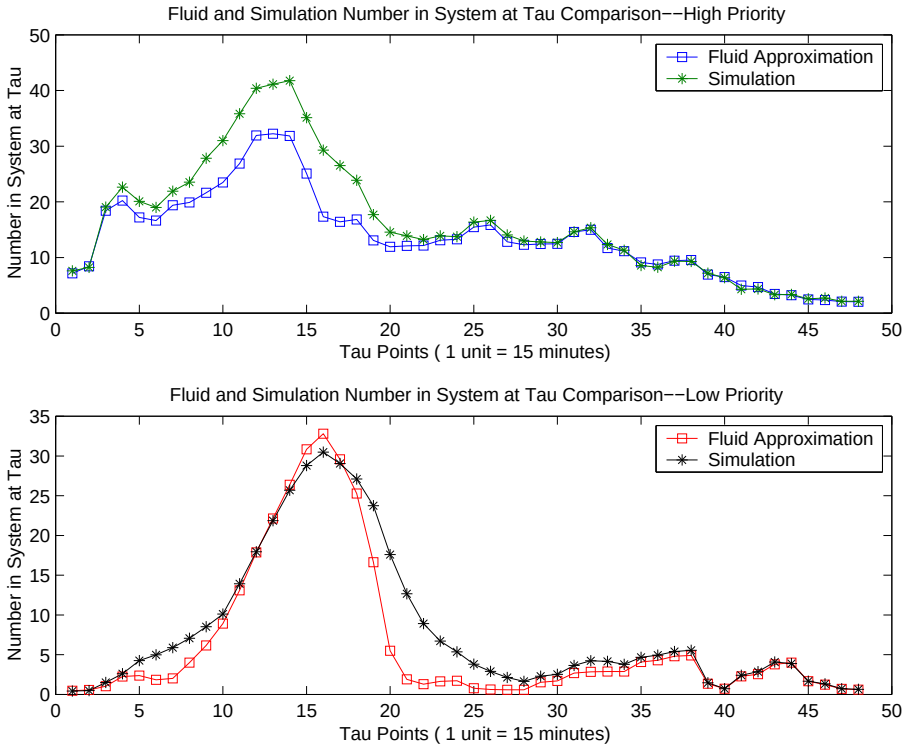

Figure 6.10: Unscaled Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Fluid vs. Simulation Comparison

In Figure 6.10 and Figure 6.11, we show the importance of scaling for the mean number in system and mean virtual waiting time estimates for high and low priority customers. We use the arrival rate function discussed in Figure 6.1. Without scaling, there is a significant gap between the fluid and simulation estimates.

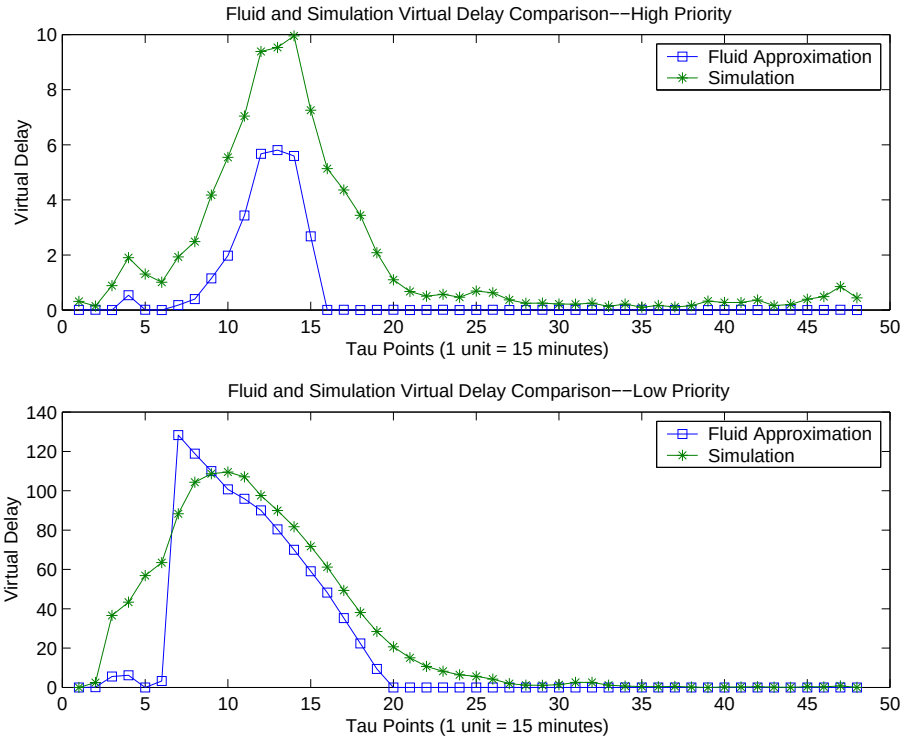

Figure 6.11: Unscaled Estimates of the Virtual Delay for High and Low Priority Customers for the Fluid vs. Simulation Comparison

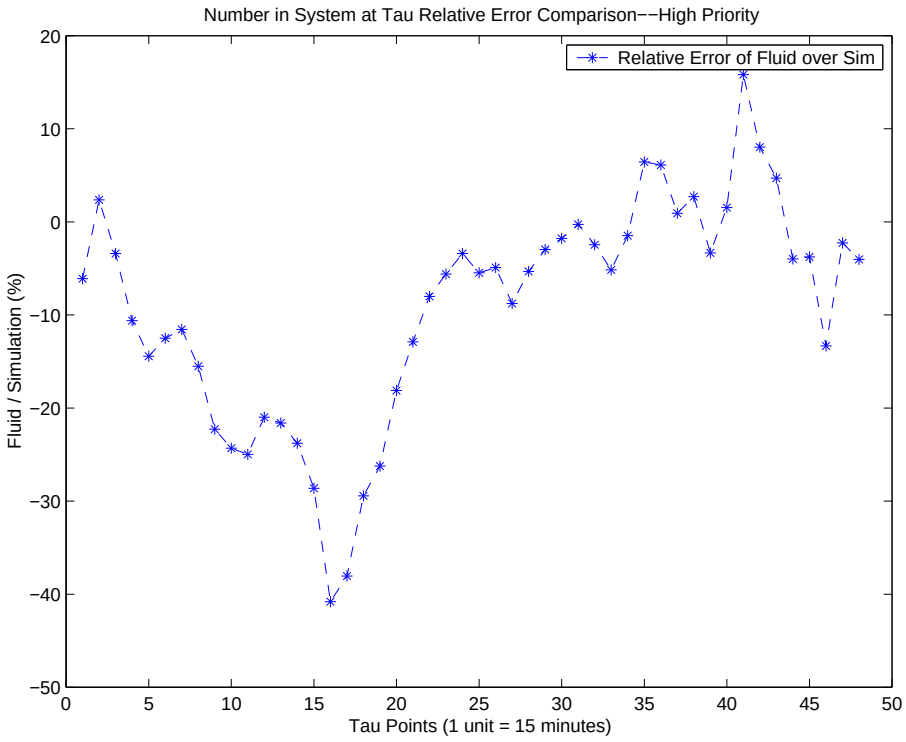

Figure 6.12: Relative Error for the Unscaled Estimates of the Number in System at Time τ_i for High Priority Customers for the Fluid vs. Simulation Comparison

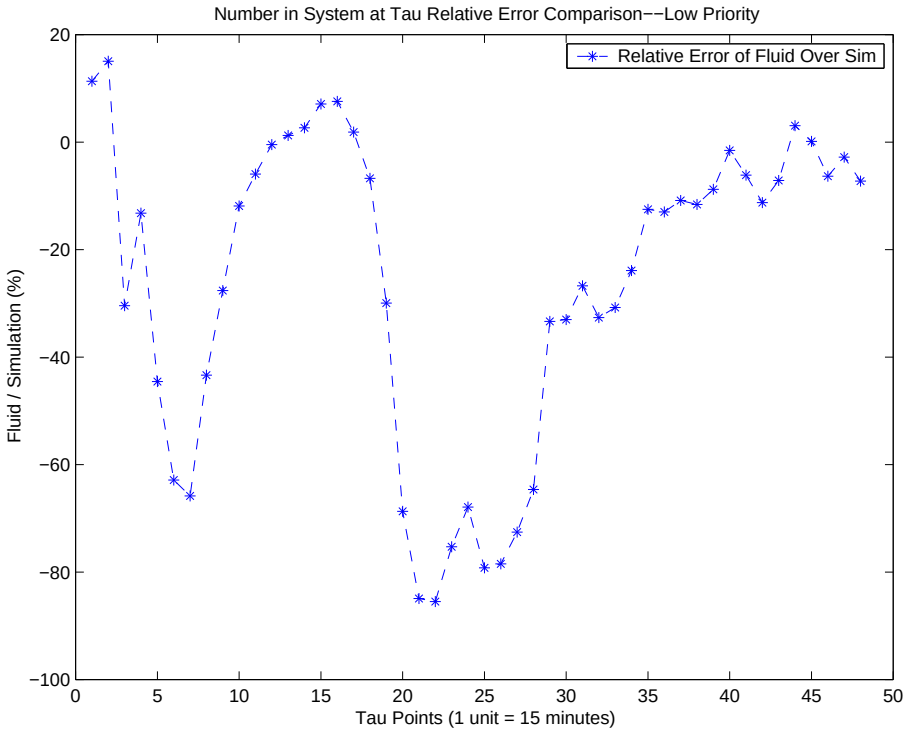

Figure 6.13: Relative Error for the Unscaled Estimates of the Number in System at Time τ_i for Low Priority Customers for the Fluid vs. Simulation Comparison

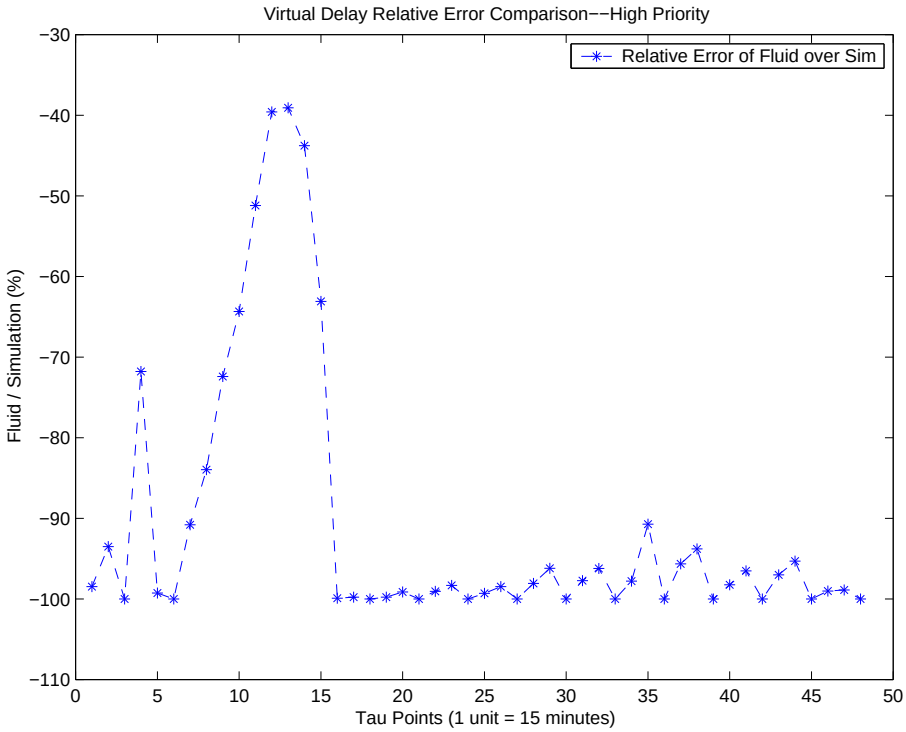

Figure 6.14: Relative Error for the Unscaled Estimates of the Virtual Delay for High Priority Customers for the Fluid vs. Simulation Comparison

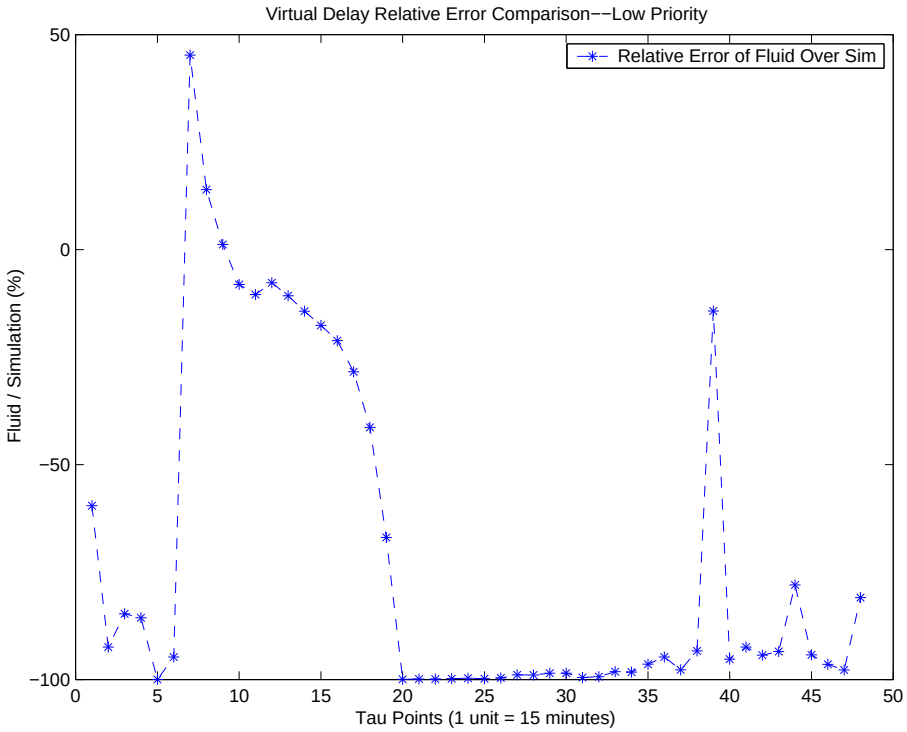

Figure 6.15: Relative Error for the Unscaled Estimates of the Virtual Delay for Low Priority Customers for the Fluid vs. Simulation Comparison

We show the relative error computations for the mean number in system and mean virtual waiting time estimates for high and low priority customers in Figure 6.12, Figure 6.13, Figure 6.14, and Figure 6.15. The relative error further quantifies the gap between the fluid and simulation estimates without using any scaling.

Next, we show the effects of increasing the level of scaling on the difference between the fluid approximations and simulation estimates. As we increase our scale factor η from 5 to 30 in increments of 5, the difference between the performance measures becomes smaller.

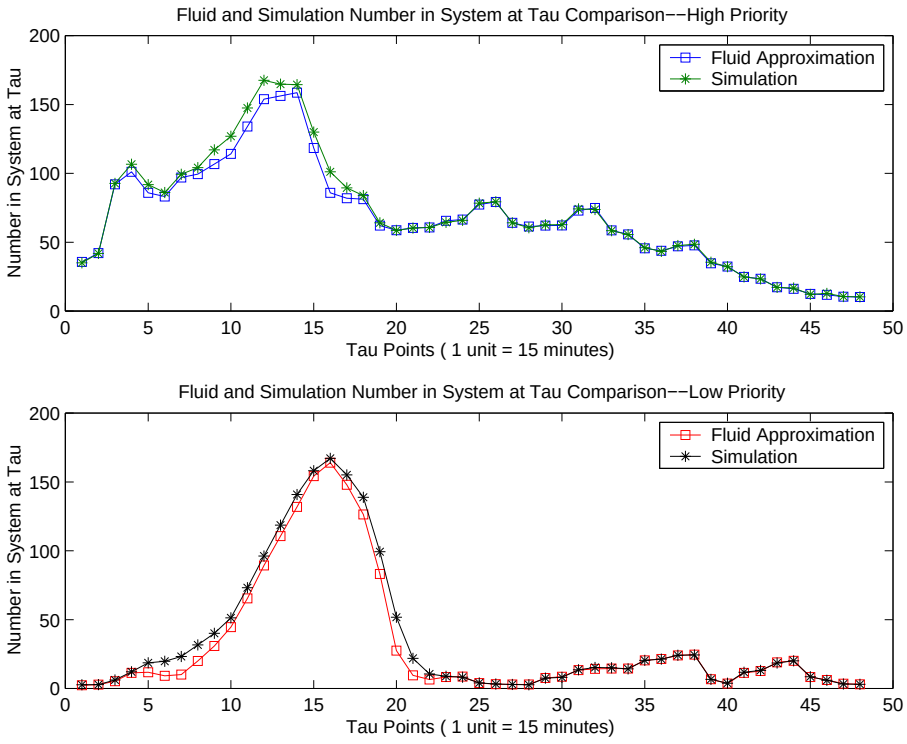

Figure 6.16: Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 5$

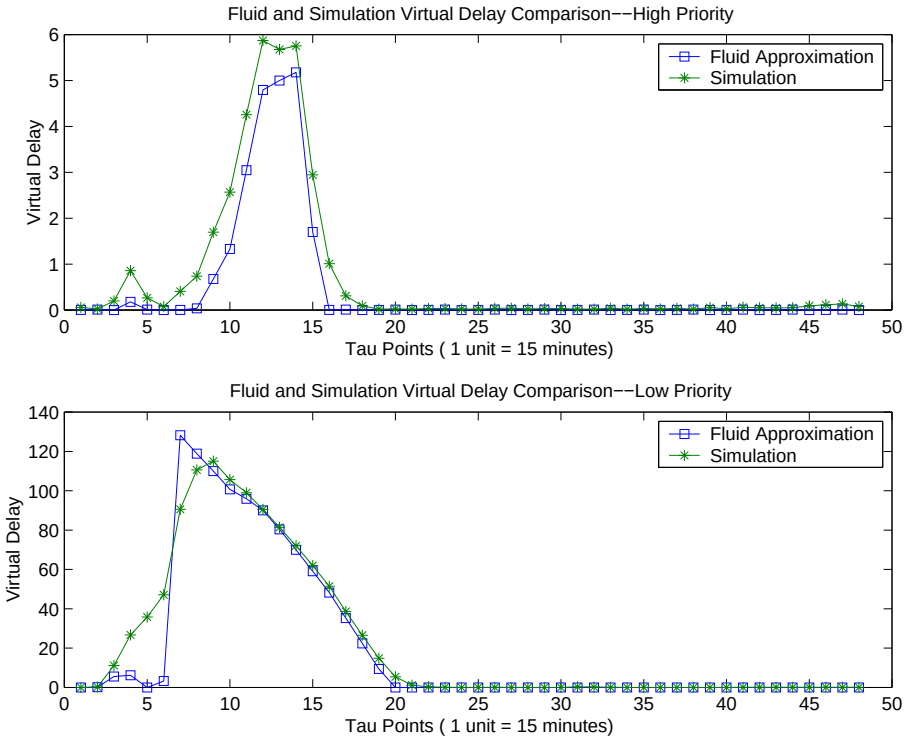

Figure 6.17: Estimates of the Virtual Delay for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 5$

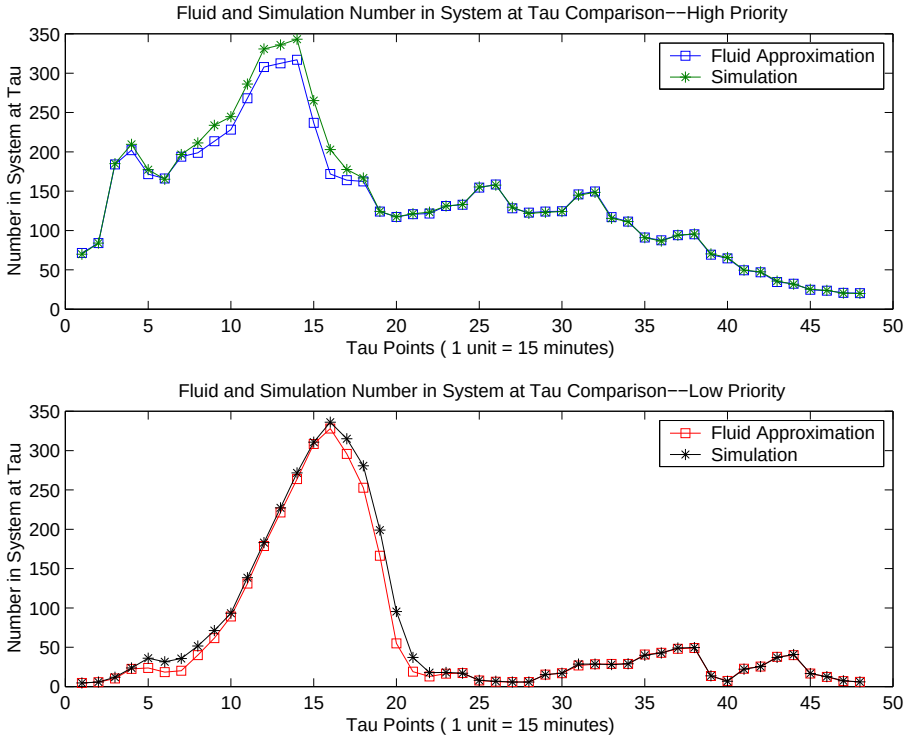

Figure 6.18: Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 10$

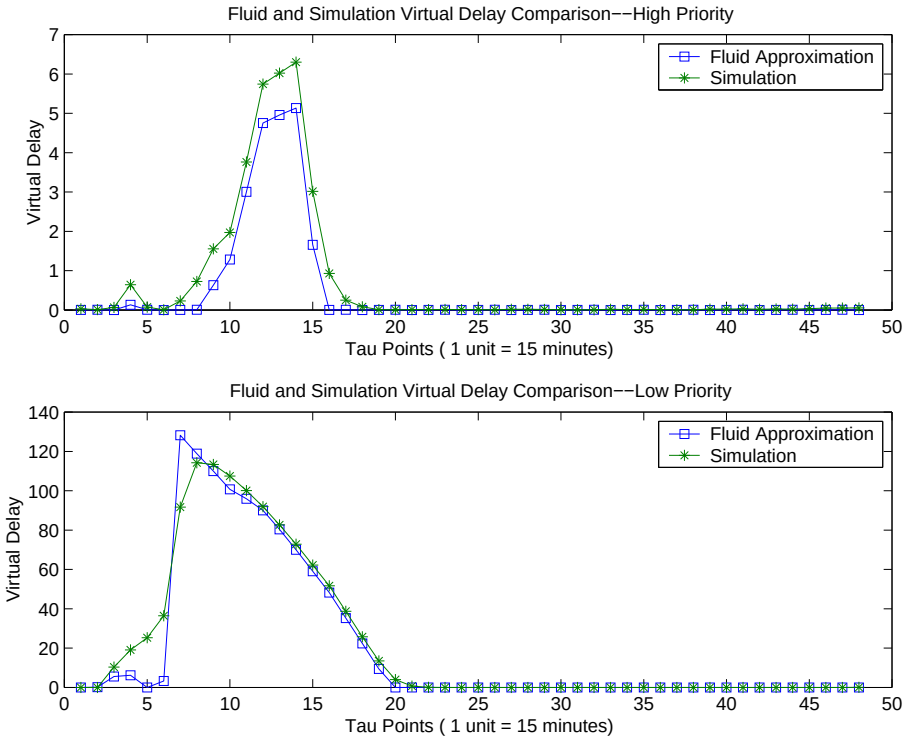

Figure 6.19: Estimates of the Virtual Delay for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 10$

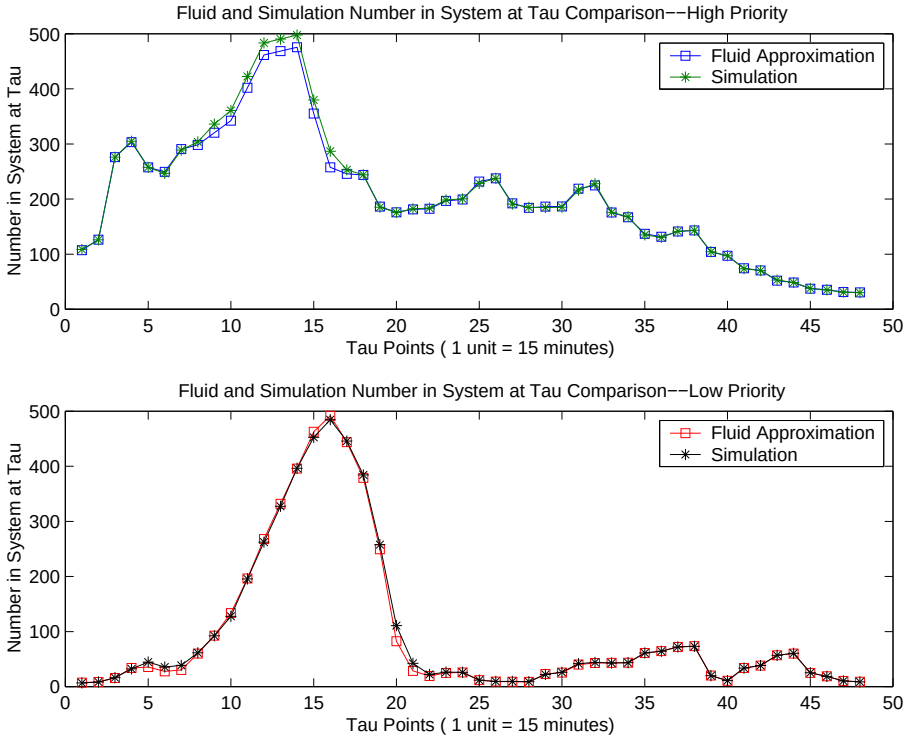

Figure 6.20: Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 15$

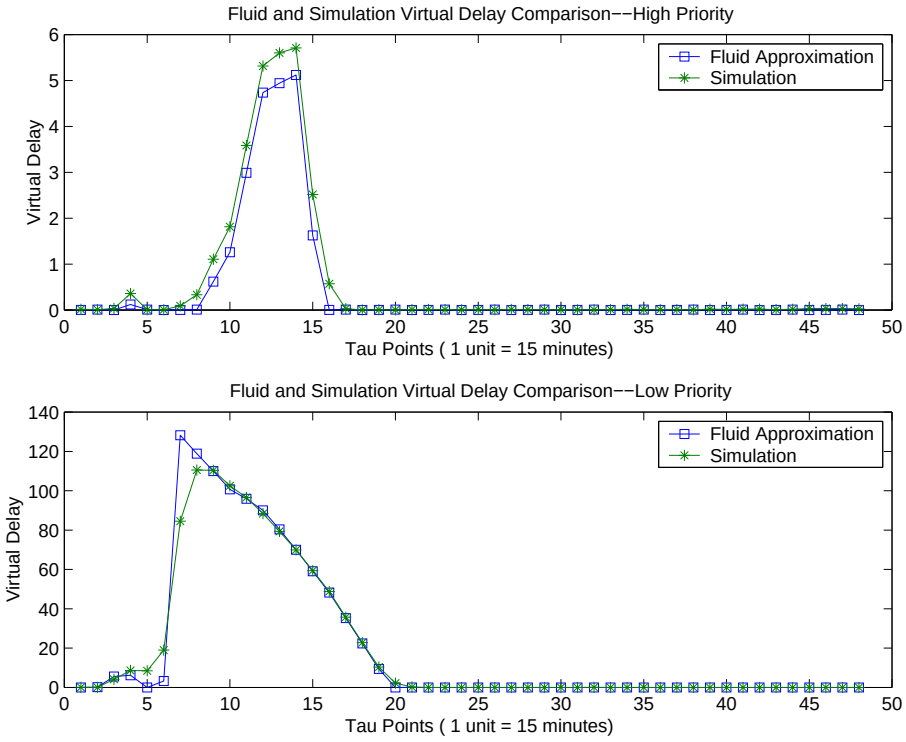

Figure 6.21: Estimates of the Virtual Delay for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 15$

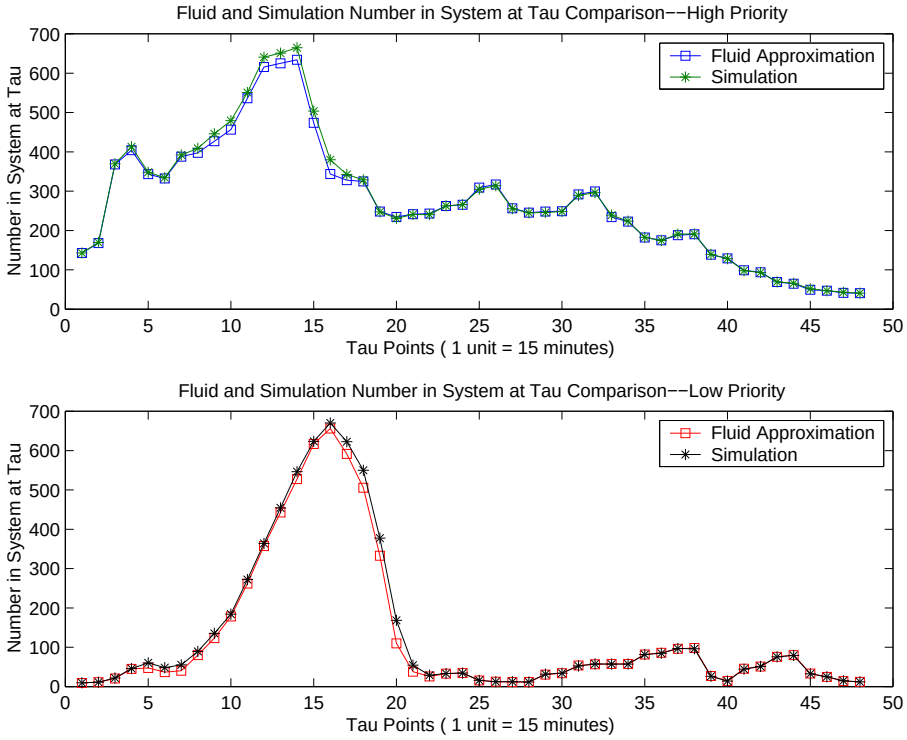

Figure 6.22: Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 20$

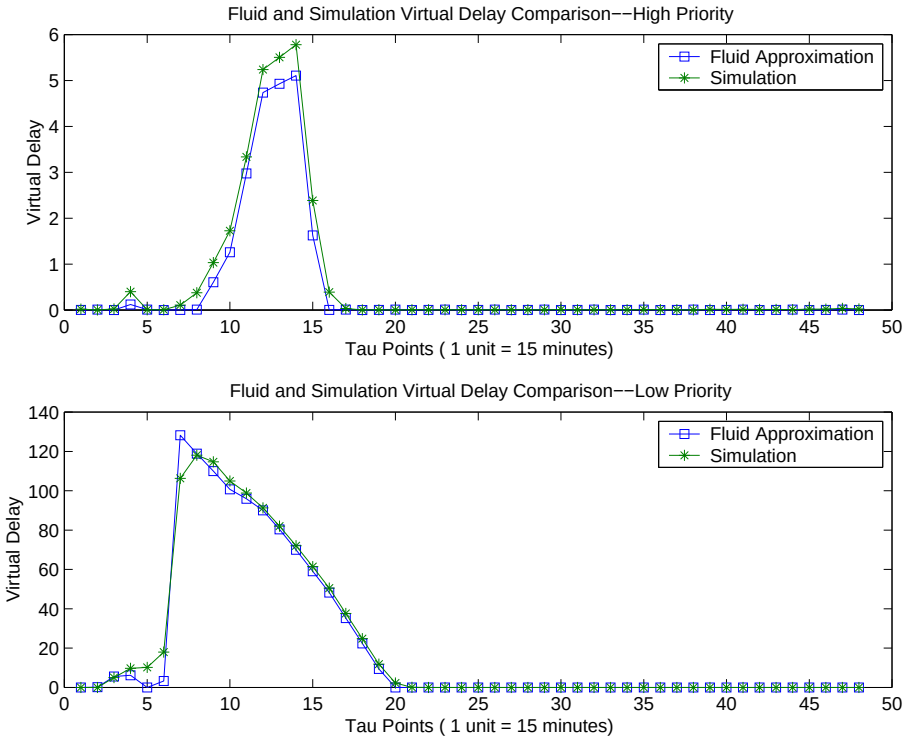

Figure 6.23: Estimates of the Virtual Delay for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 20$

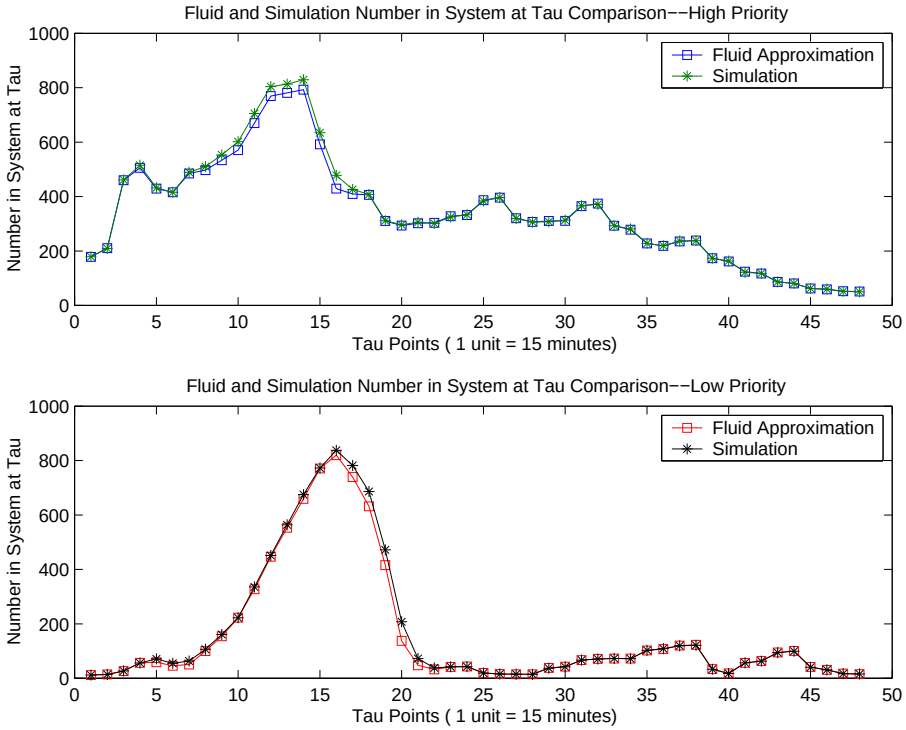

Figure 6.24: Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 25$

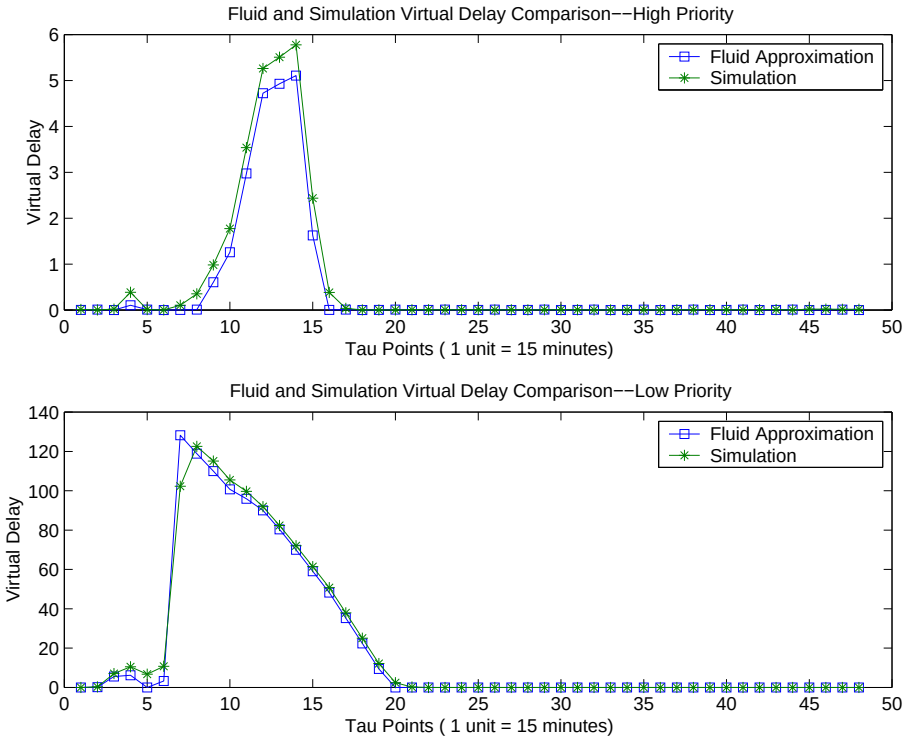

Figure 6.25: Estimates of the Virtual Delay for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 25$

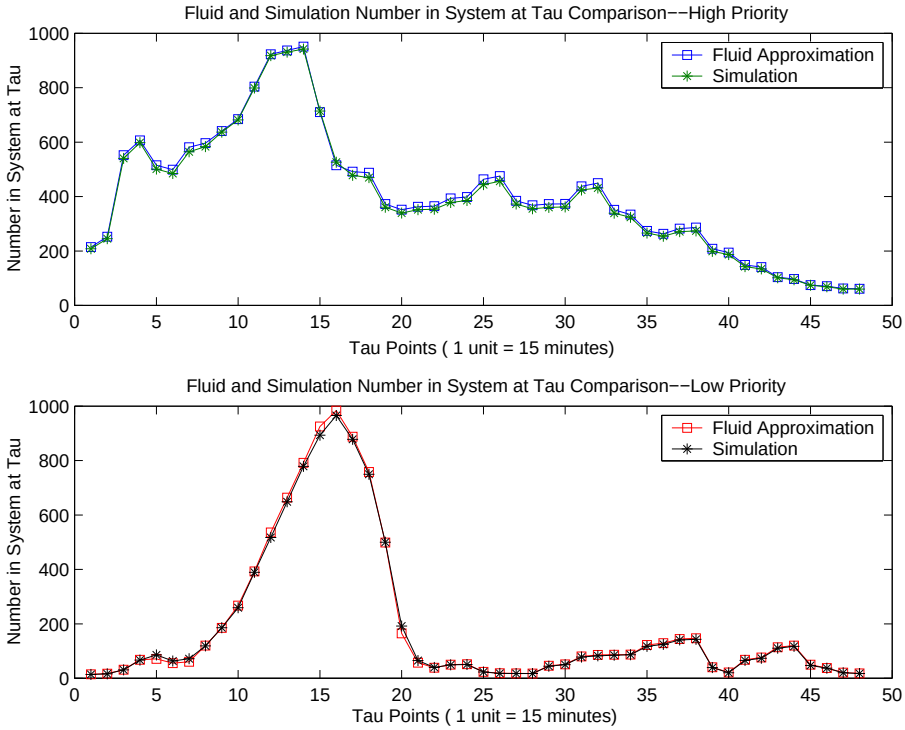

Figure 6.26: Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 30$

We show the importance of scaling for the mean number in system and mean virtual waiting time estimates for high and low priority customers in Figure 6.16 through Figure 6.27. Without proper scaling, there is a significant difference between the fluid approximations and simulation estimates. The difference does, however, greatly decrease at $\eta = 5$ for both performance measures. For further values of η, the difference continues to decrease, albeit slowly. After $\eta = 30$, there is no appreciable difference in the difference between both performance measures.

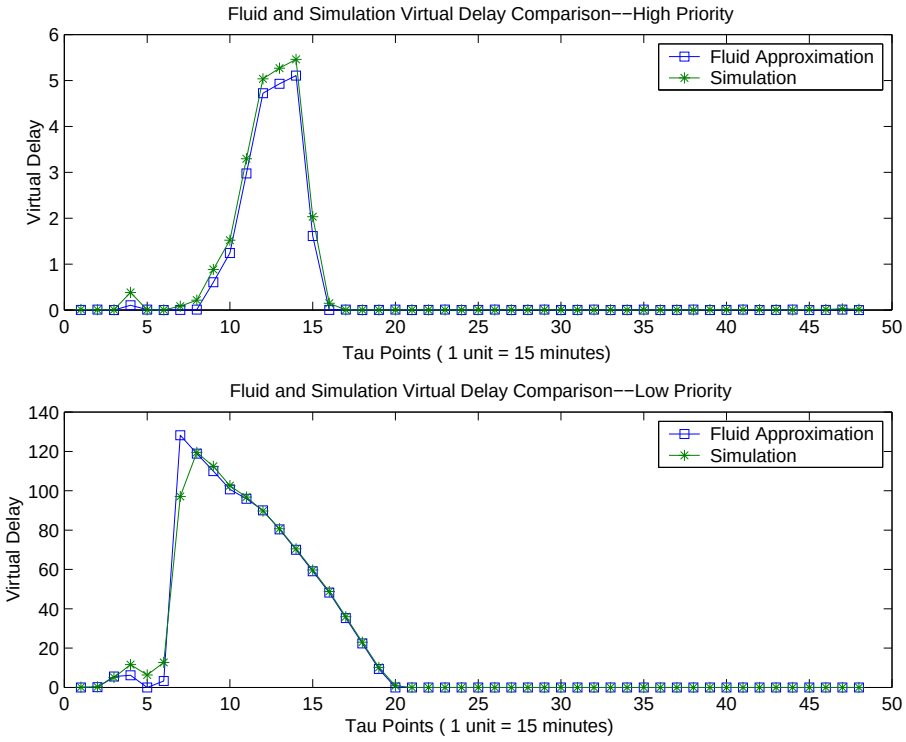

Figure 6.27: Estimates of the Virtual Delay for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 30$

6.2.3 Fluid and Diffusion vs. Simulation

Fluid Approximations vs Simulation

We compare the estimates of the mean number in the system and mean virtual waiting time for each customer class between the fluid approximations and the simulation methods. Here, we use a scale factor of $\eta = 35$ for our final comparisons. Beyond a factor of 35, the fluid approximation estimates were not much closer to the simulation estimates, which suggested that 35 was a good stopping point for our scaling process. As stated earlier, both methods model the $M_t/M/n$, preemptive-resume, dynamic priority, two-class queue. Again, we use the arrival rate function shown in Figure 6.1. The performance measures were estimated at 48 distinct time points, τ_i, which are spaced 15 minutes apart over the 12, or 720 minute time horizon.

In Figure 6.28 and Figure 6.29, we show the comparison of the mean number in the system and mean virtual waiting time estimates between our fluid and simulation models for the high and low priority customers. The fluid approximations are very close to the simulation estimates at all time points τ_i for the high priority and low priority calls. In fact, as the offered load, which is a measure of the arrival intensity of customers to the call center, varied with time, the accuracy of our fluid approximations remained good. For example, the largest loads occurred from 6 : 30 to 9 : 00 AM and the smallest loads occurred after 3 : 30 PM. However, our approximations were very good in all time periods. As shown above, without using a scaling regime, these estimates would not be as close.

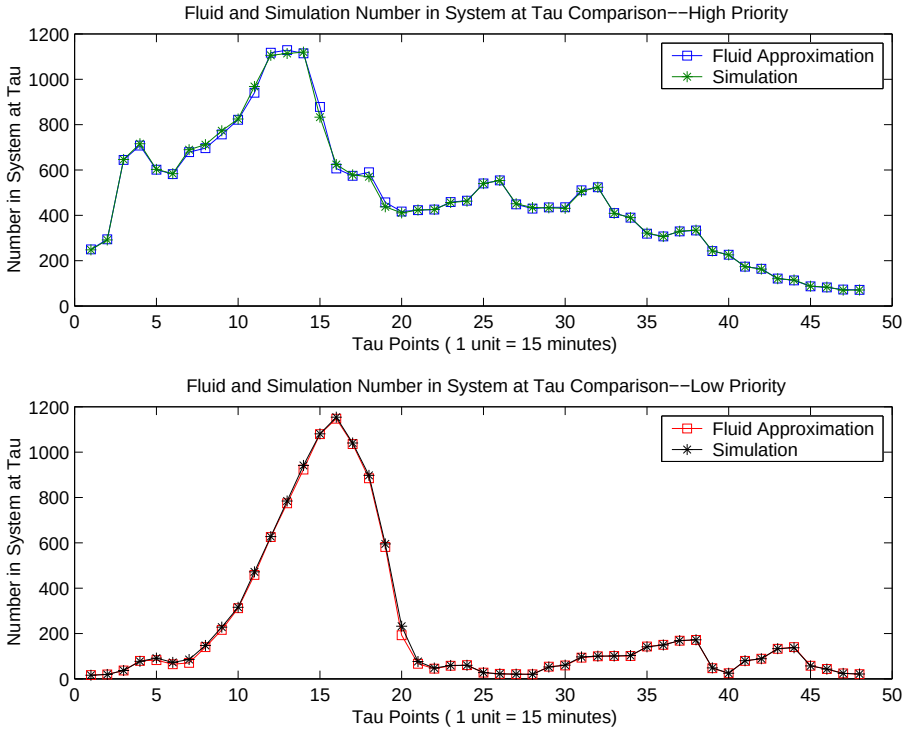

Figure 6.28: Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

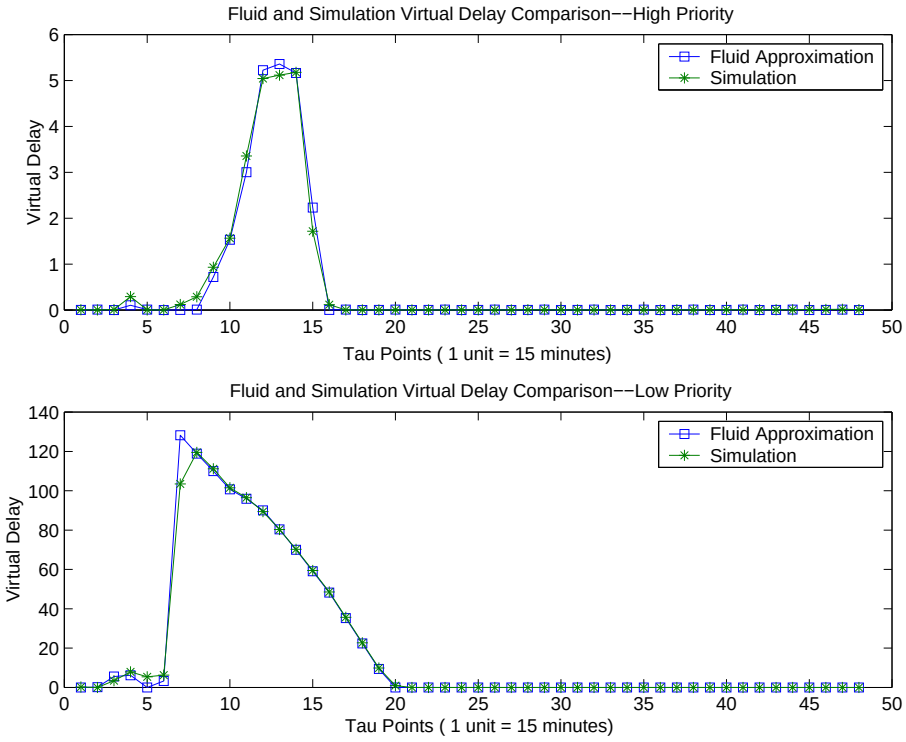

Figure 6.29: Estimates of the Virtual Delay for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

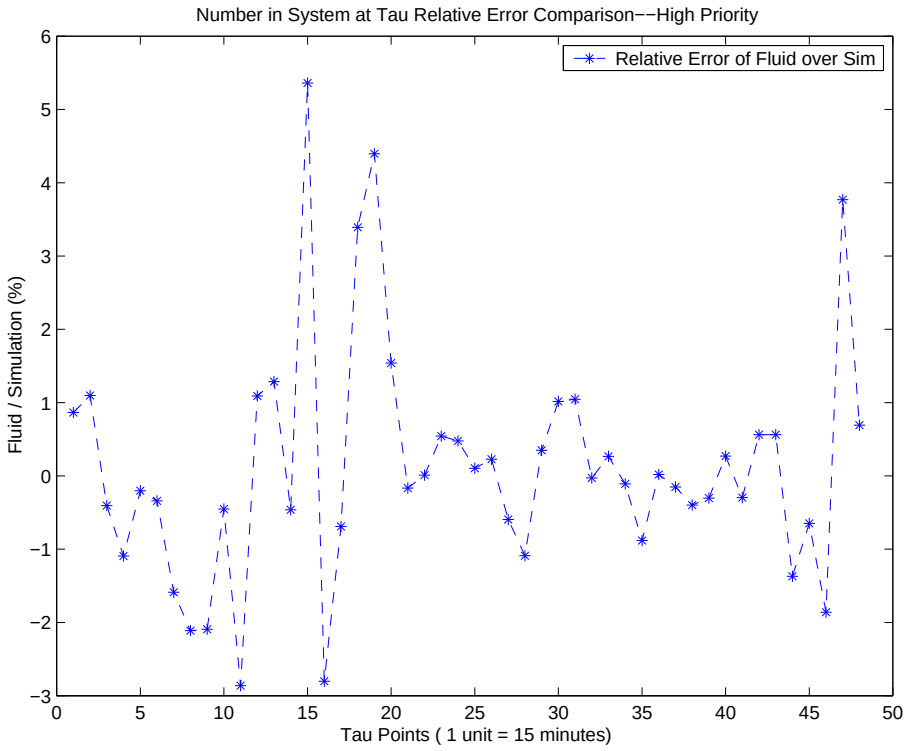

Figure 6.30: Relative Error for the Estimates of the Number in System at Time τ_i for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

We display the relative error in both performance measure estimates for the high and low priority customers. The graphs show the relative error, or percent difference between the fluid and simulation estimates in Figure 6.30, Figure 6.31), Figure 6.32, and Figure 6.33. Thus, most of the fluid estimates are within 10 percent of the simulation estimates.

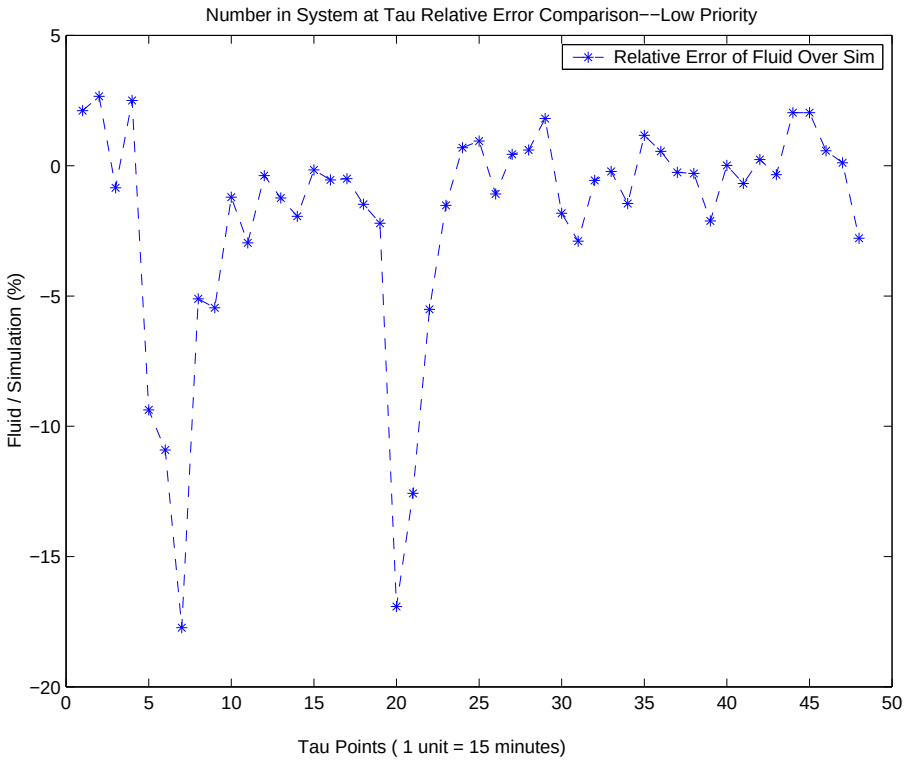

Figure 6.31: Relative Error for the Estimates of the Number in System at Time τ_i for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

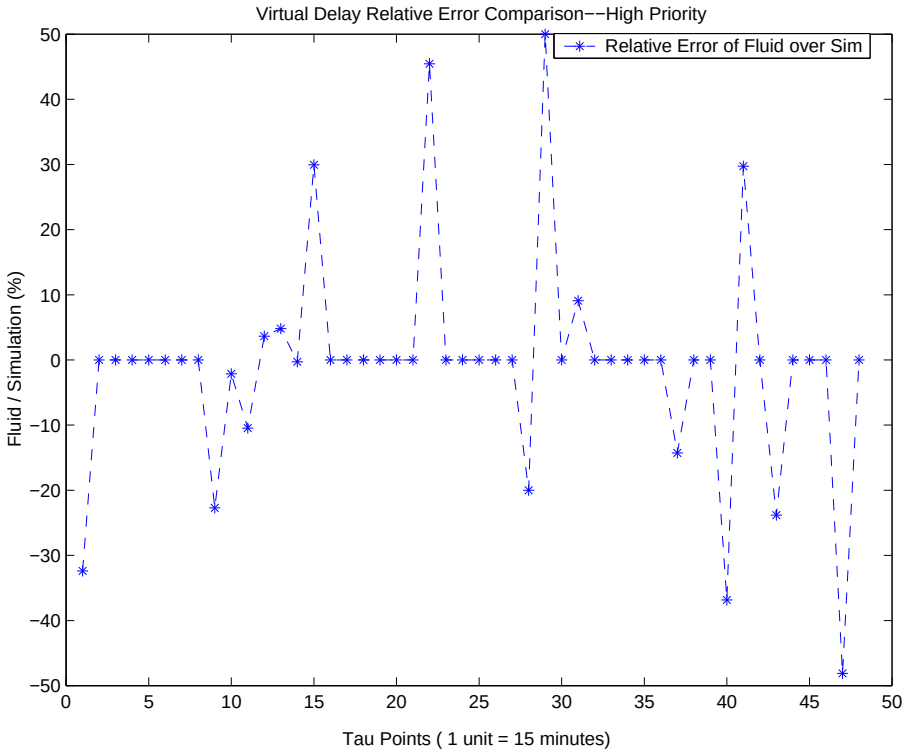

Figure 6.32: Relative Error for the Estimates of the Virtual Delay for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

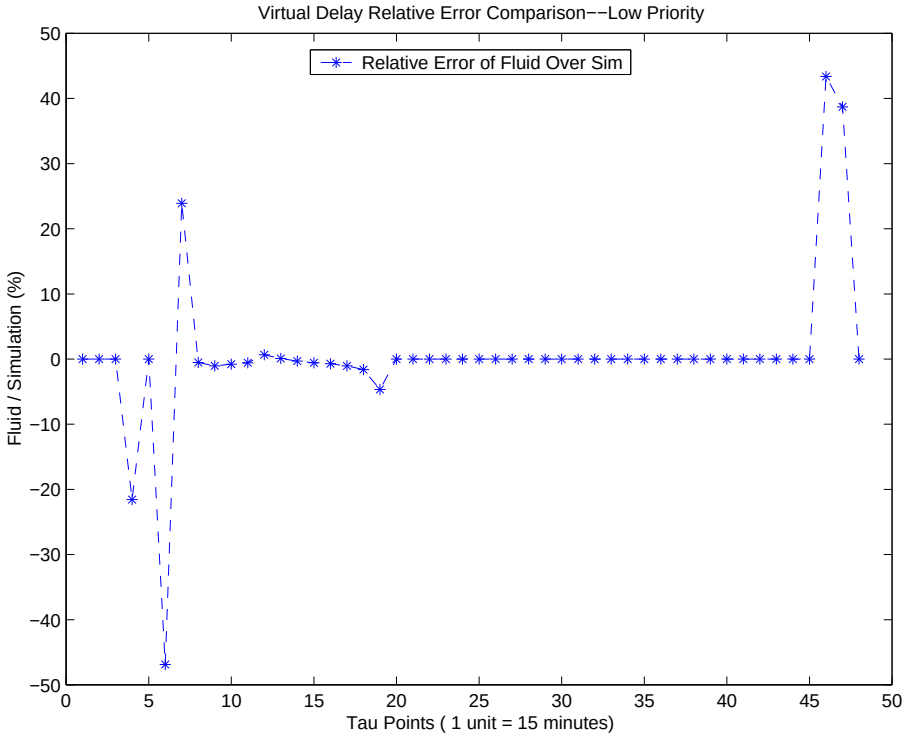

Figure 6.33: Relative Error for the Estimates of the Virtual Delay for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

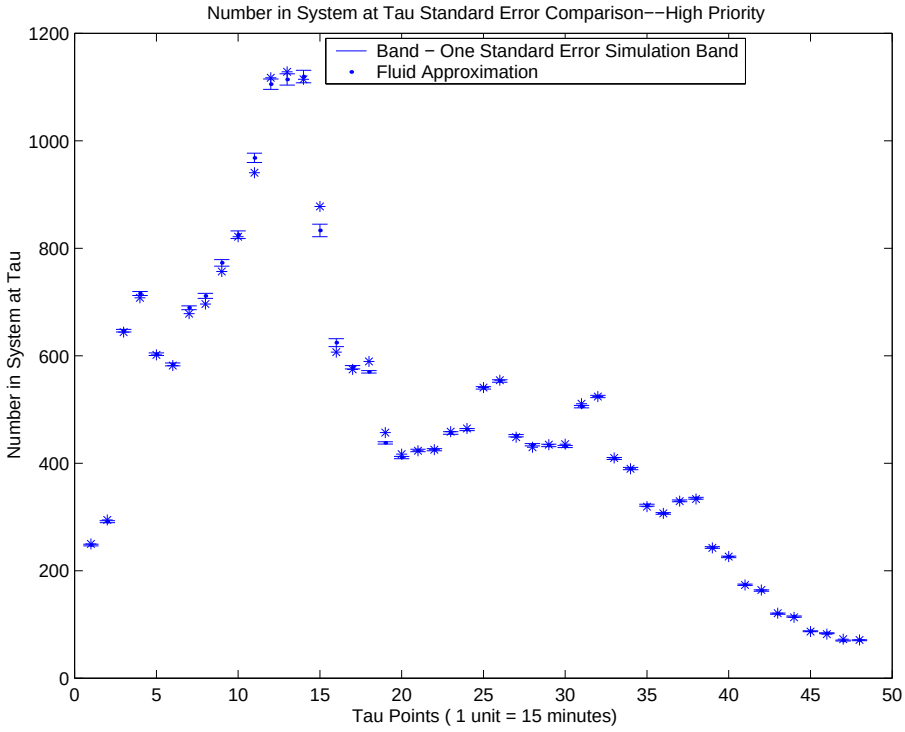

Figure 6.34: Standard Error Band for the Estimates of the Number in System at Time τ_i for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

In Figures 6.34, 6.35, 6.36, and 6.37, we show the comparison of the standard error in the performance measures between our fluid and simulation methods for the high and low priority customers. We compute a simulation band by forming an one standard error, i.e., one sample standard deviation, interval around the simulation estimate. In the graph, we show whether the fluid estimate is within the standard error band of the simulation estimate.

In Table 6.1, we display the computer run times for our scaled fluid approximation and simulation C-programs. The unscaled run times for our simulation C-program is 65 minutes for the desktop computer and 25 minutes for the laptop

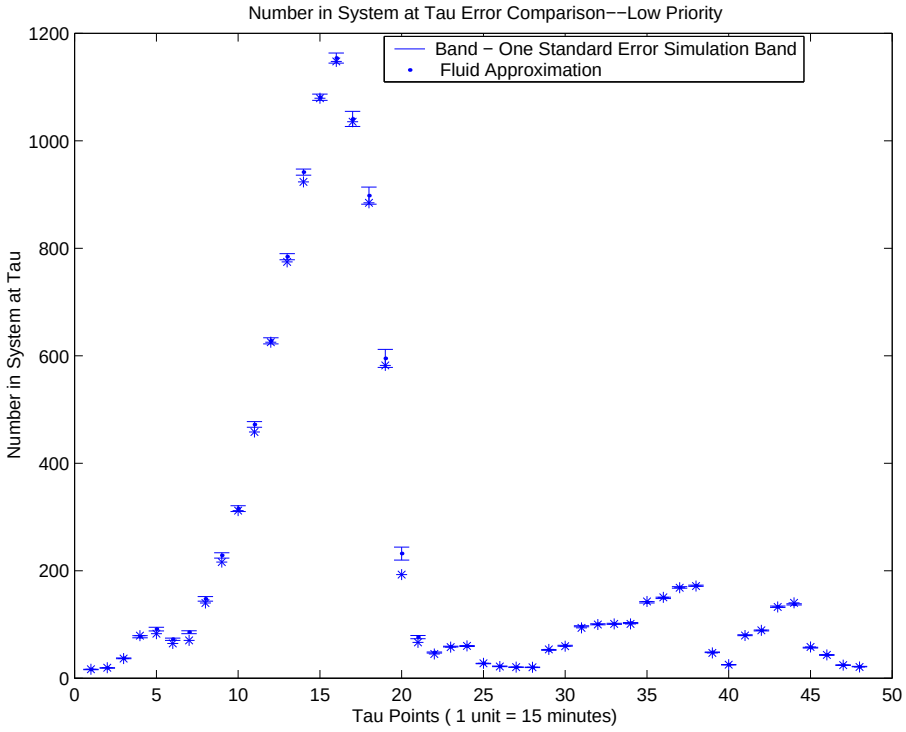

Figure 6.35: Standard Error Band for the Estimates of the Number in System at Time τ_i for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

computer. As we increase the scale factor η from 1 to 35, the run time increases greatly. (We do not show the progression of individual run times over each scale factor though.) The run times are listed for two different computers using the Windows operating system. The desktop computer has a 550 MHz processor and 320 MB of RAM, while the laptop computer has a 1.6 GHz processor and 512 MB of RAM. There is a reduction in the run times for the scaled and unscaled programs on the laptop computer. Also, the scaled fluid program runs significantly faster than the scaled simulation program on both computers. We also show that there is a significant difference between the scaled and unscaled simulation run

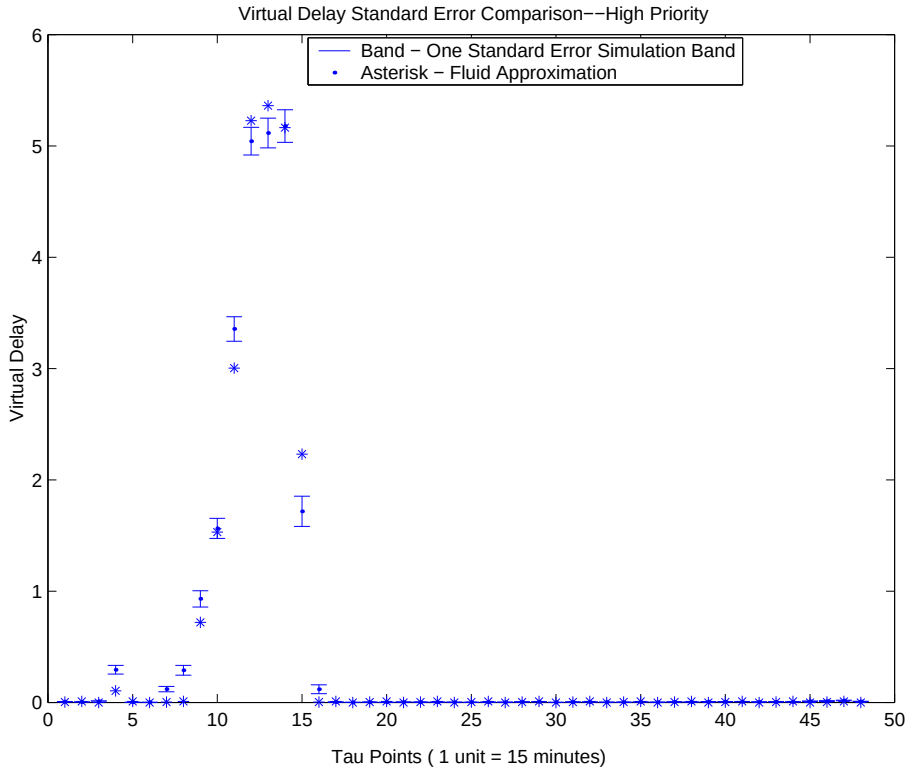

Figure 6.36: Standard Error Band for the Estimates of the Virtual Delay for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

times, which displays the affect of scaling. However, for each τ_i, we repeat a full simulation run, or replication, which causes the run time of our scaled simulation program to increase significantly. Therefore, a more efficient simulation program would reduce the run time, but the fluid program would still run faster.

Finally, we test the importance of the choice of the probability distribution function for the service times in our model. We use the exponential distribution for the service times in the simulation model. Thus, the high priority service times are exponentially distributed with mean $\mu_1 = 0.1151$, and the low priority

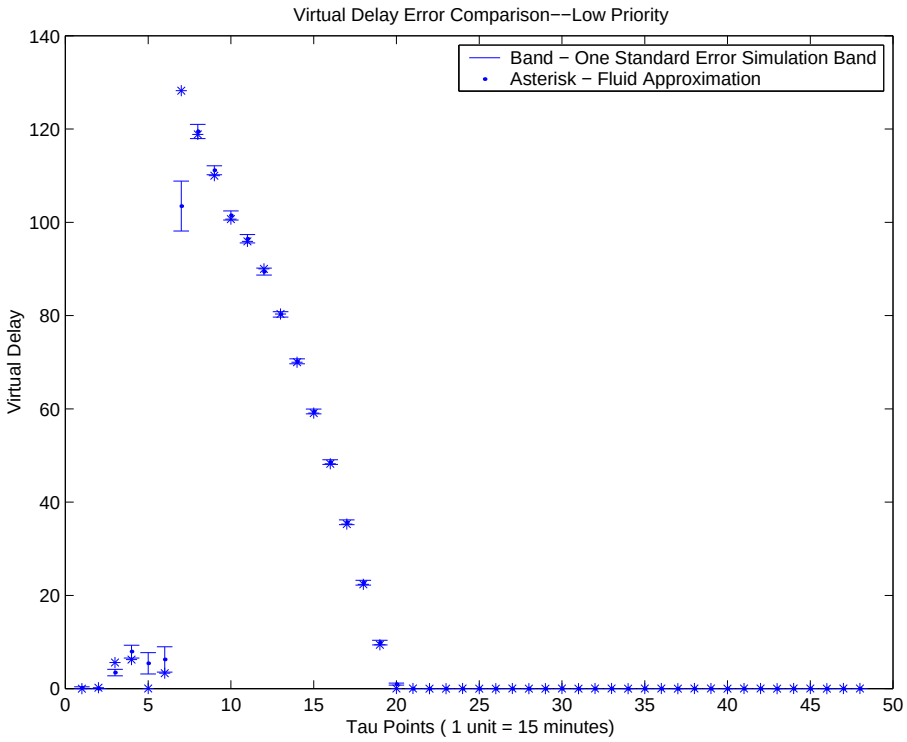

Figure 6.37: Standard Error Band for the Final Estimates of the Virtual Delay for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

service time distribution is also exponentially distributed with mean $\mu_2 = 0.1151$. Since the fluid approximations only depend on the mean of the service times, we change the service time distribution in our simulation model to a deterministic one and repeat our fluid and simulation comparisons. Thus, the service times are now constant with rate $\mu_1 = \mu_2 = 0.1151$ customers per minute. We attempt to quantify any difference in the accuracy of our fluid approximations to our simulation estimates after such a change in the service time distribution. Again, we use a fully scaled (i.e., scale factor of $\eta = 35$) system for our comparisons.

In Figure 6.38 and Figure 6.39, we show that fluid approximations of mean

C-Program Run Times (Minutes)

Computer	Scaled Fluid and Diffusion	Scaled Simulation
Windows Desktop PC	45	960
Windows Laptop PC	35	180

Table 6.1: C-Program code Run Times for Our Fully-Scaled Models

number in system and mean virtual waiting time are still close to the simulation estimates for the high priority customers. However, the fluid approximations of the same two performance measures are not as close to the simulation estimates for the low priority customers. Therefore, the choice of service distribution in the simulation model does affect the quality of the approximations of the low priority customers' service.

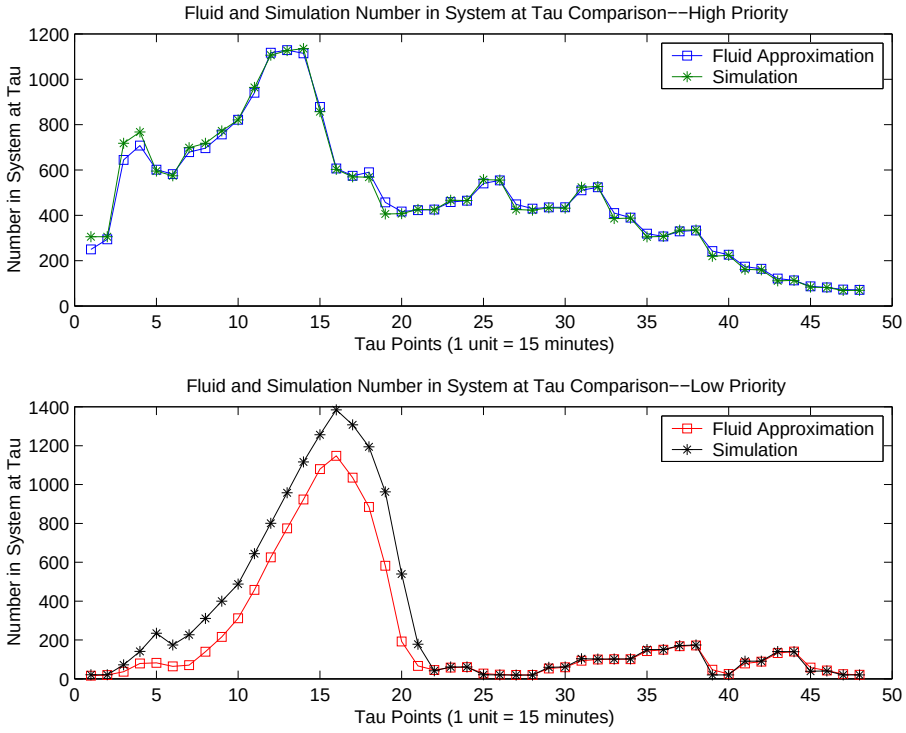

Figure 6.38: Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

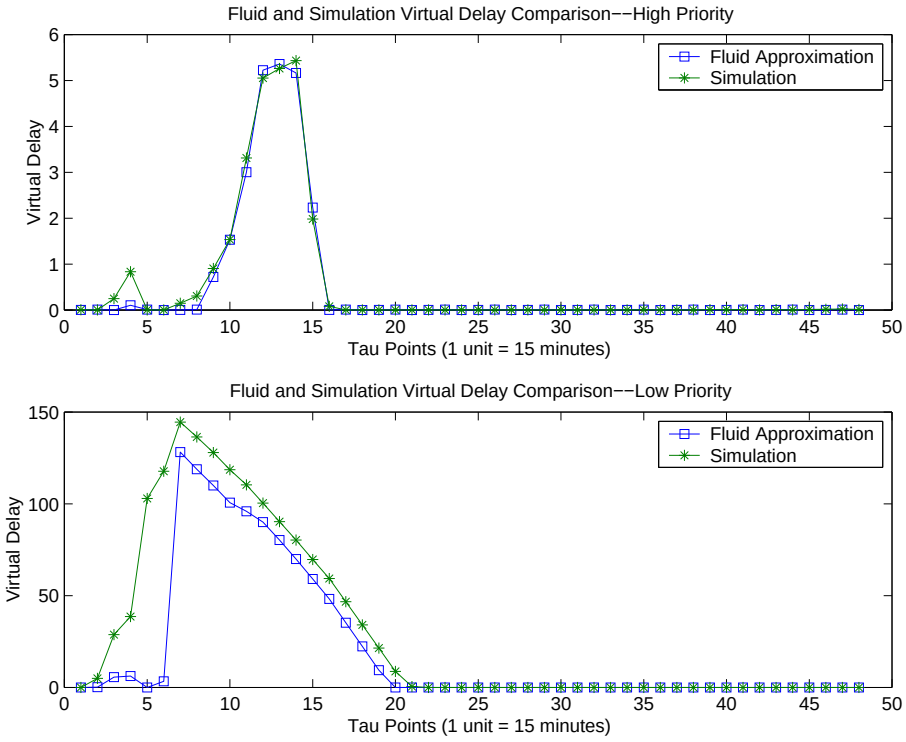

Figure 6.39: Estimates of the Virtual Delay for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

6.2.4 Fluid vs. Simulation - Case 1 Arrival Rates

We compute the fluid and simulation estimates for the mean number in system and mean virtual waiting time using a somewhat different set of inter-arrival rates. These rates are also time-varying; however, the high and low priority inter-arrival rates vary between only two values. For the first 6 hours of the time horizon, the inter-arrival rates are chosen such that the system is under-loaded, or stable, i.e. $\rho < 1$. Conversely, in the last 6 hours the inter-arrival rates are such that the system is over-loaded, or unstable, i.e., $\rho > 1$.

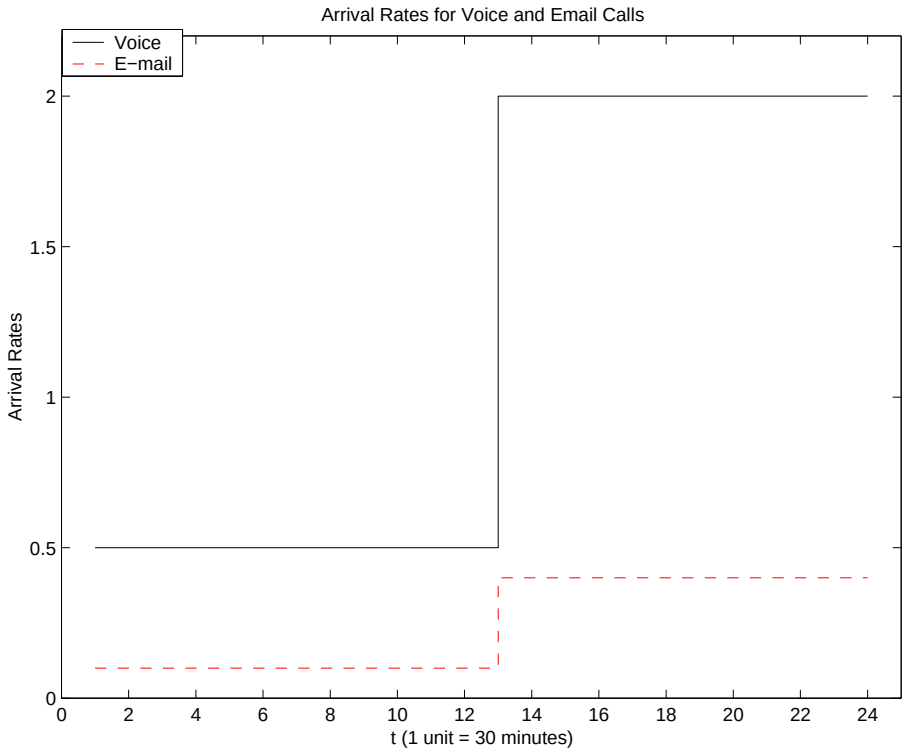

Figure 6.40: Piecewise Constant Arrival Function with Rates Varying at Time τ_i - Case 1

We show these rates in Figure 6.40.

In Figure 6.41 and Figure 6.42, we show the comparison of the two perfor-

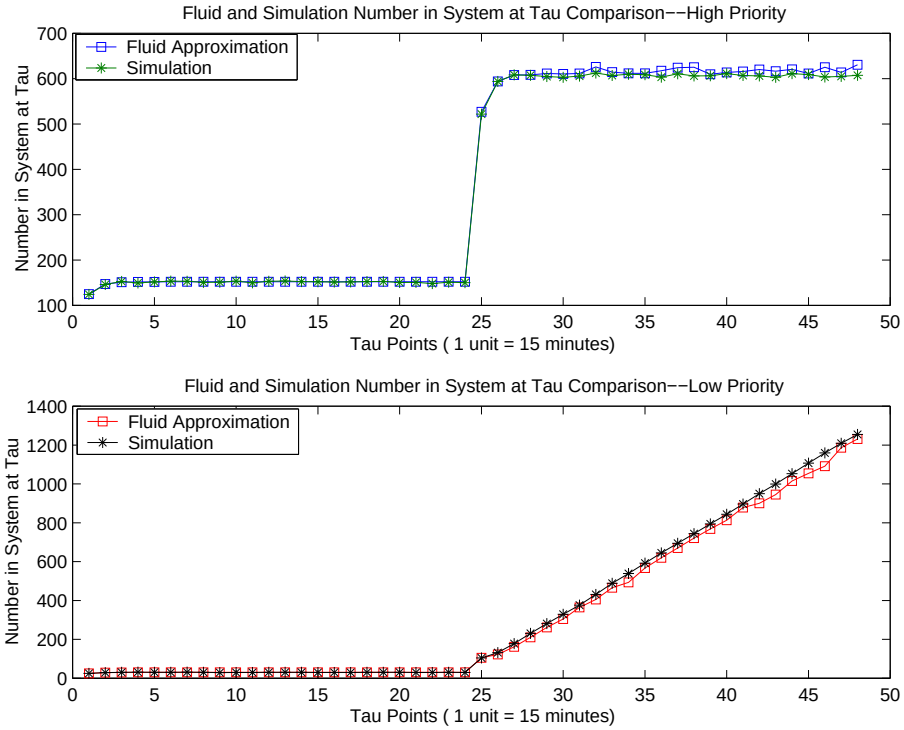

Figure 6.41: Case 1 - Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

mance measure estimates for the high and low priority customers between our fluid and simulation models. As the system passes through the under-loaded to the over-loaded phase, the fluid estimates remain close to the simulation estimates for the mean number in system and the mean virtual waiting time.

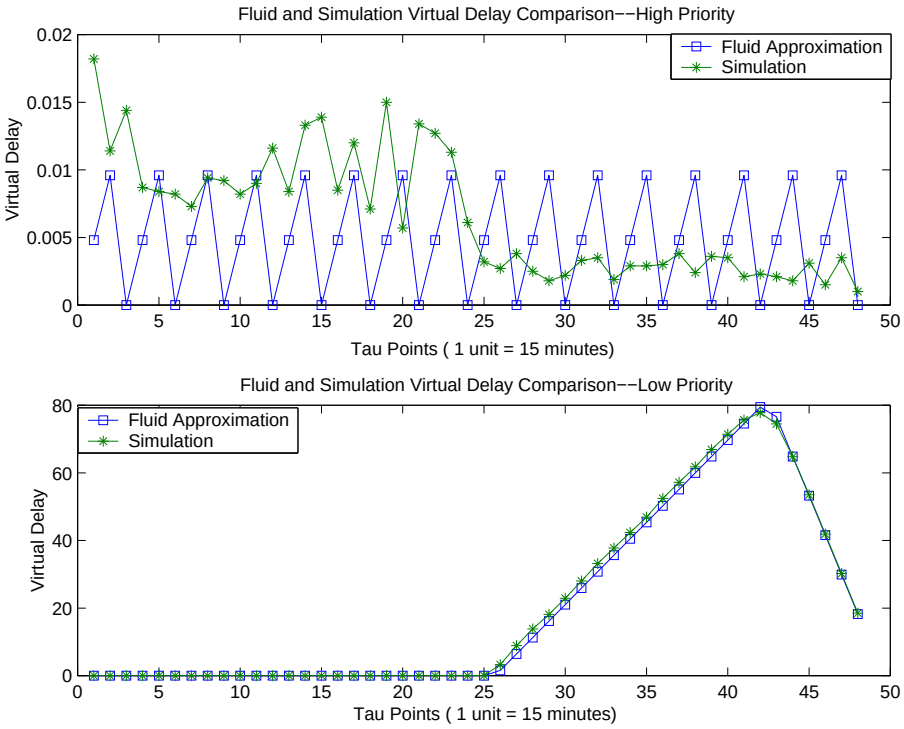

Figure 6.42: Case 1 - Estimates of the Virtual Delay for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

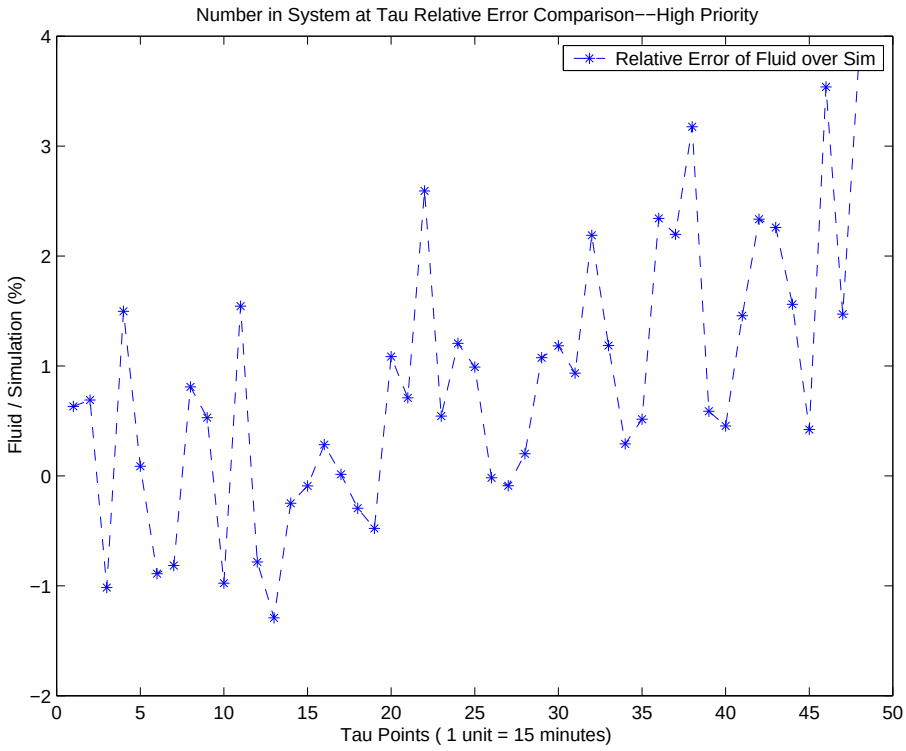

Figure 6.43: Case 1 - Relative Error for the Estimates of the Number in System at Time τ_i for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

In Figure 6.43, Figure 6.44, Figure 6.45, and Figure 6.46, we show the relative error in the two performance measure estimates for the high and low priority customers. As the system passes through the under-loaded to the over-loaded phase, most of the relative error values remain relatively small, i.e. less than 10 percent.

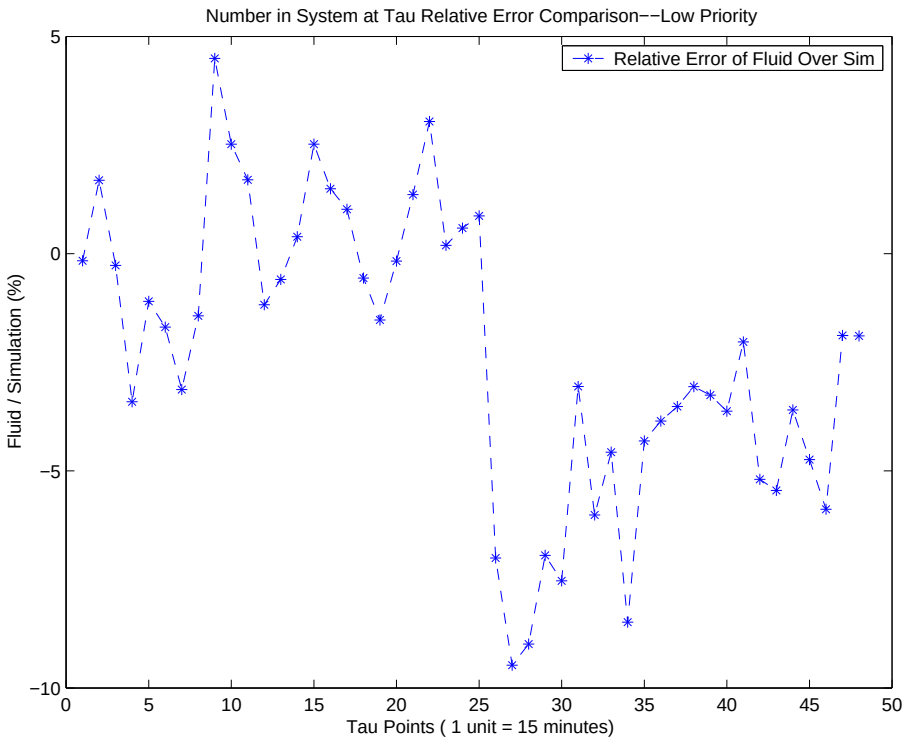

Figure 6.44: Case 1 - Relative Error for the Estimates of the Number in System at Time τ_i for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

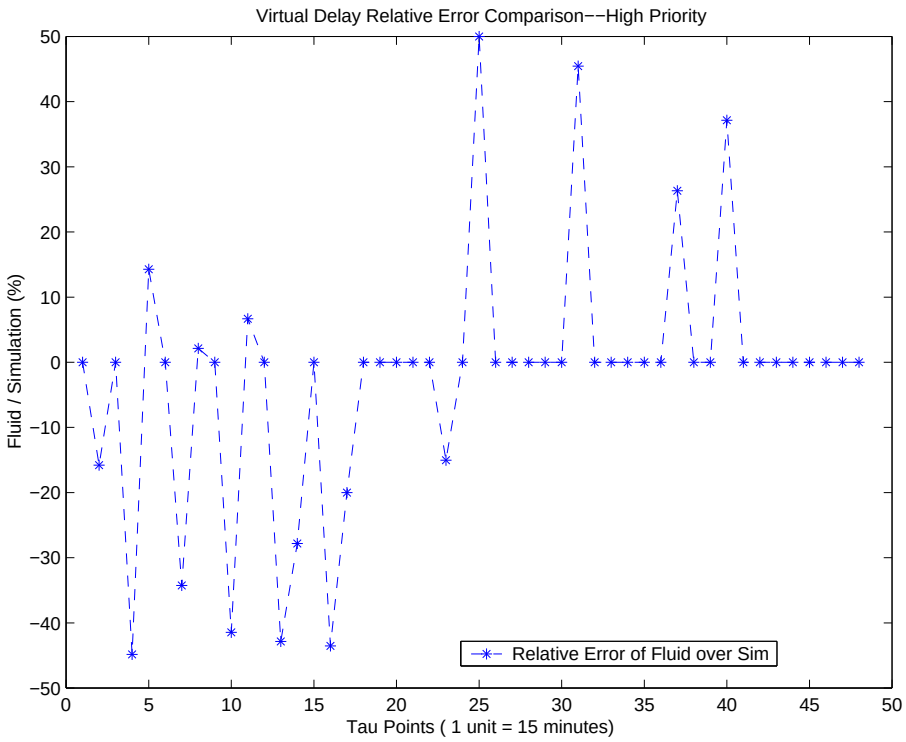

Figure 6.45: Case 1 - Relative Error for the Estimates of the Virtual Delay for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

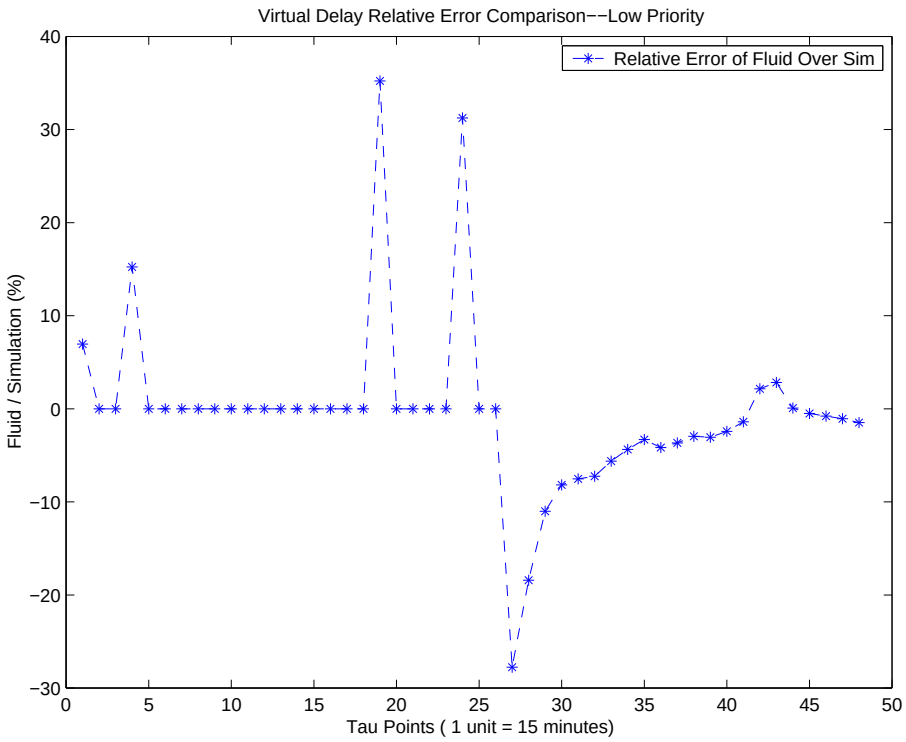

Figure 6.46: Case 1 - Relative Error for the Estimates of the Virtual Delay for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

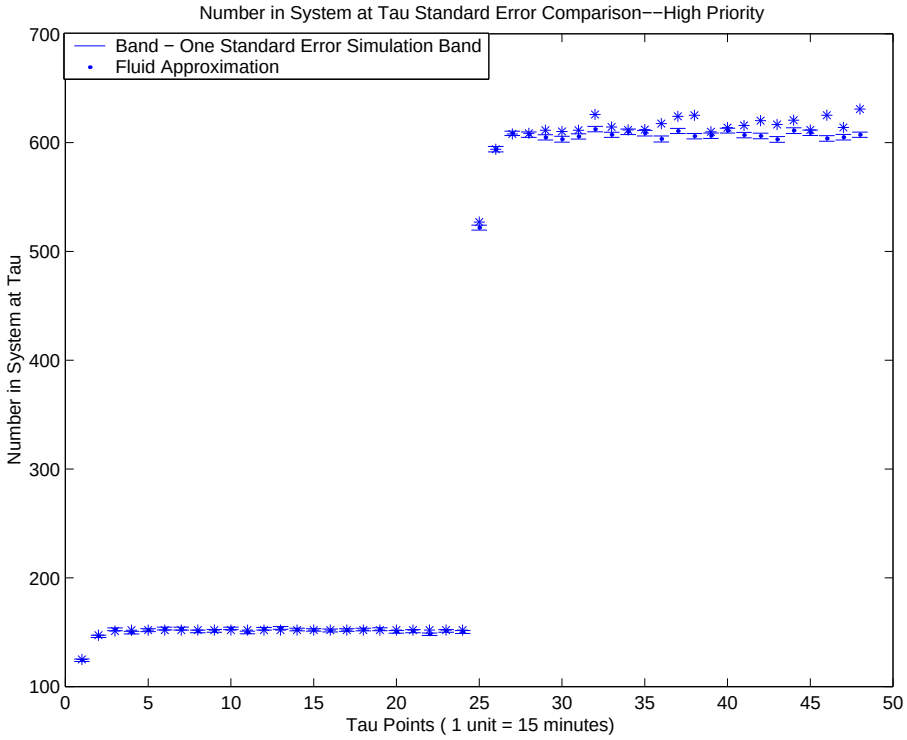

Figure 6.47: Case 1 - Standard Error Band for the Estimates of the Number in System at Time τ_i for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

In Figure 6.47, Figure 6.48, Figure 6.49, and Figure 6.50, we show the standard error in the two performance measure estimates for the high and low priority customers. As the system passes through the under-loaded to the over-loaded phase, either the fluid estimate is within the standard error band of the simulation, or relatively close to the simulation estimate itself.

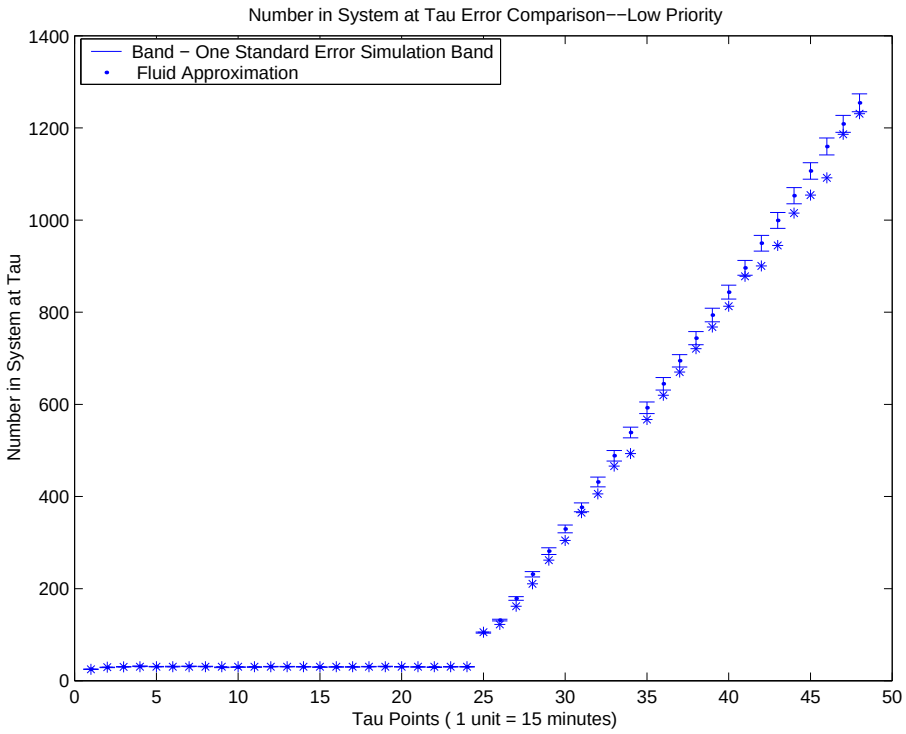

Figure 6.48: Case 1 - Standard Error Band for the Estimates of the Number in System at Time τ_i for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

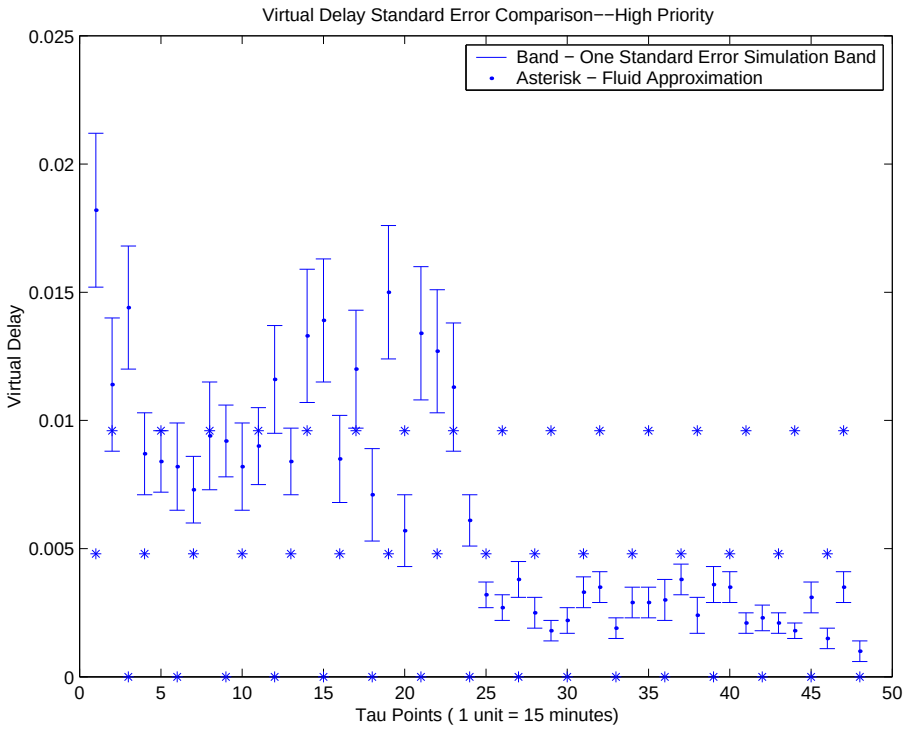

Figure 6.49: Case 1 - Standard Error Band for the Estimates of the Virtual Delay for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

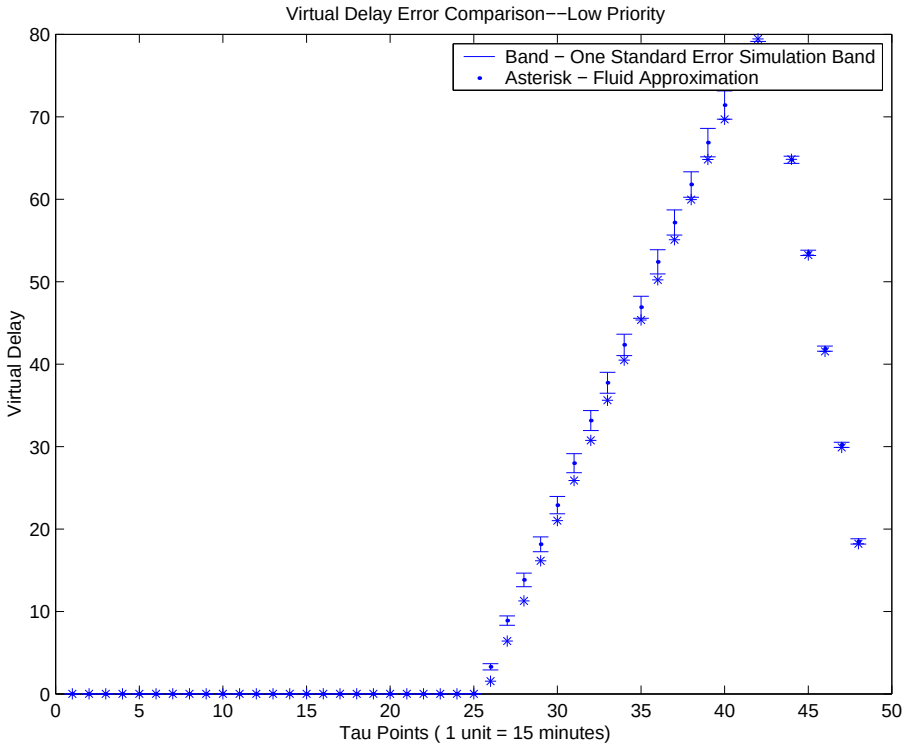

Figure 6.50: Case 1 - Standard Error Band for the Estimates of the Virtual Delay for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

6.2.5 Fluid vs. Simulation - Case 2 Arrival Rates

We computed the fluid and simulation estimates for the mean number in system and mean virtual waiting time using a somewhat different set of inter-arrival rates. These rates are also time-varying; however, the high and low priority inter-arrival rates vary between only two values. For the full 12 hours of the time horizon, the inter-arrival rates are chosen such that the system is under-loaded, or stable, i.e. $\rho < 1$. However, over the last 6 hours, the system remains close to the under-loaded/over-loaded boundary , i.e., $\rho \approx 1$.

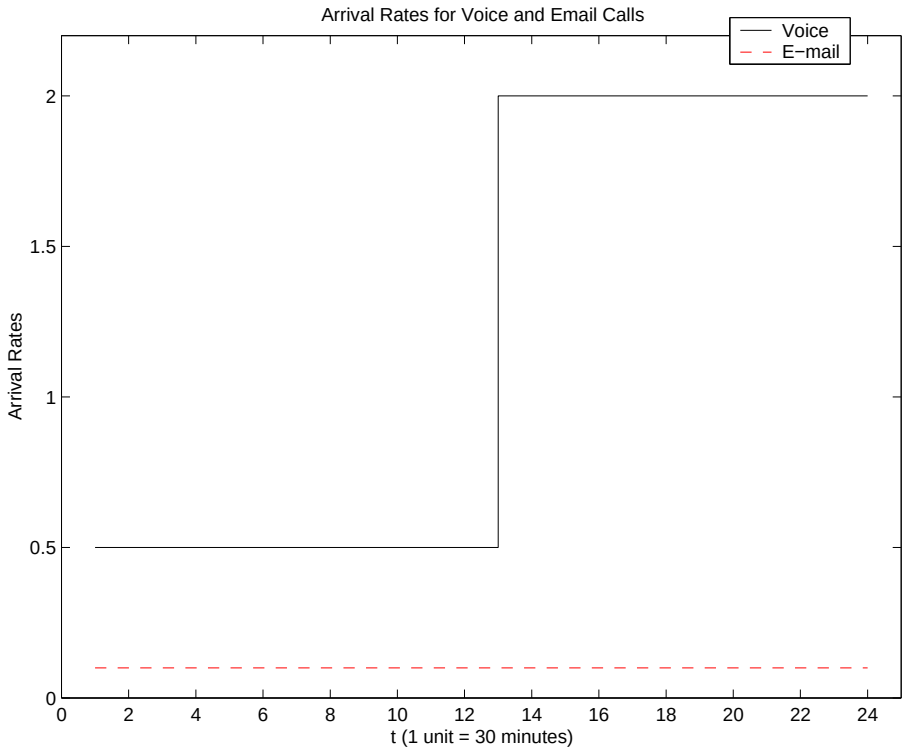

Figure 6.51: Piecewise Constant Arrival Function with Rates Varying at Time τ_i - Case 2

We show these rates in Figure 6.51.

In Figure 6.52 and Figure 6.53, we show the comparison of the two perfor-

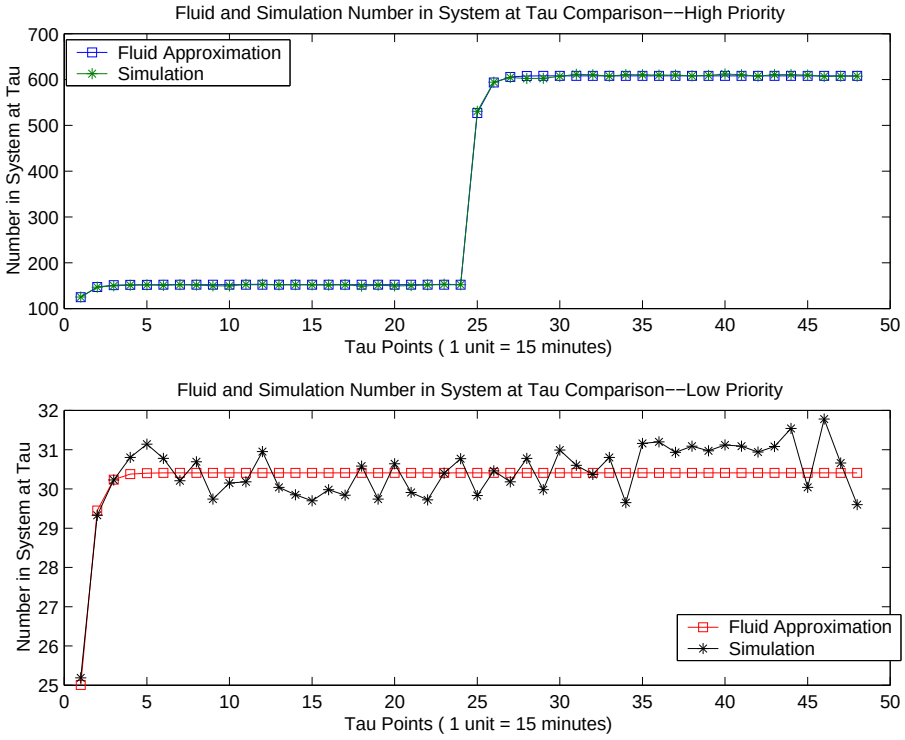

Figure 6.52: Case 2 - Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

mance measure estimates for the high and low priority customers between our fluid and simulation models. The fluid estimates remain close to the simulation estimates for the mean number in system and the mean virtual waiting time over the total time horizon.

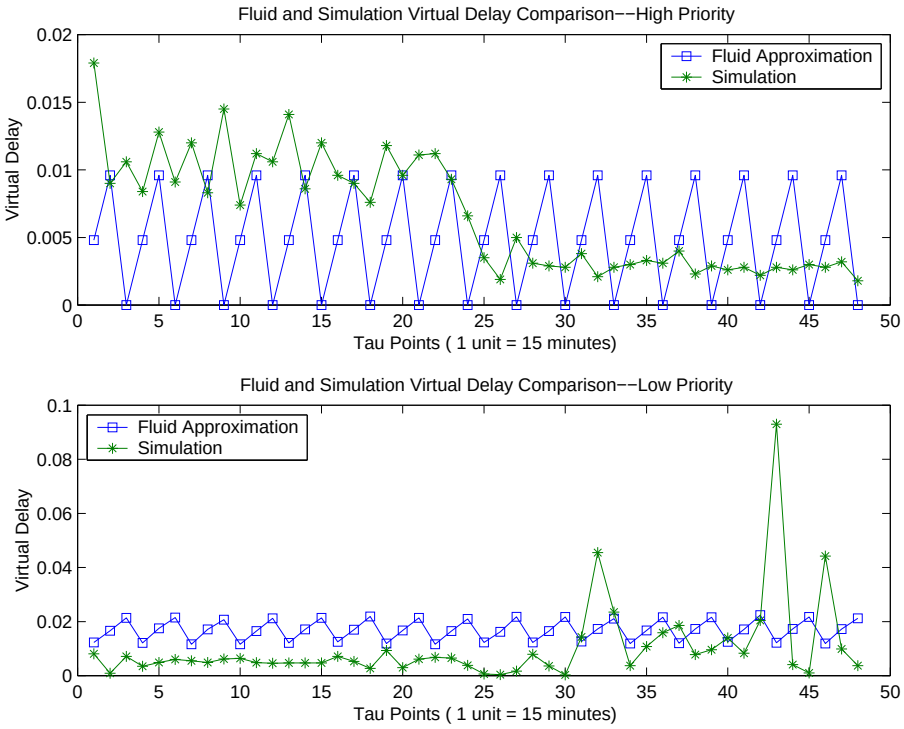

Figure 6.53: Case 2 - Estimates of the Virtual Delay for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

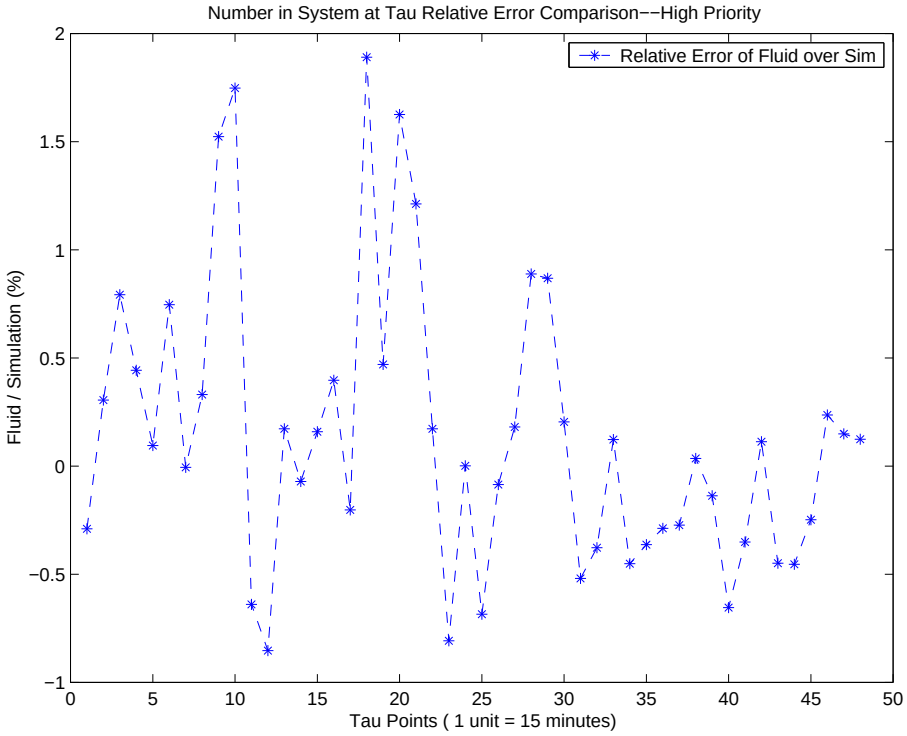

Figure 6.54: Case 2 - Relative Error for the Estimates of the Number in System at Time τ_i for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

In Figure 6.54, Figure 6.55, Figure 6.56, and Figure 6.57, we show the relative error in the two performance measure estimates for the high and low priority customers. Most of the relative error values remain relatively small, i.e. less than 10 percent throughout the total time horizon.

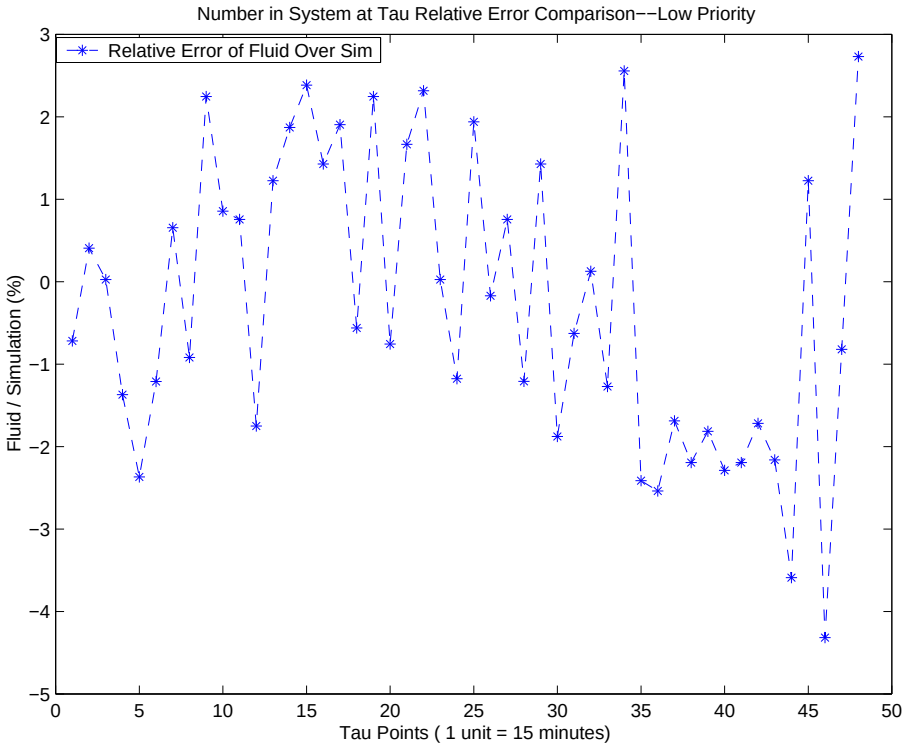

Figure 6.55: Case 2 - Relative Error for the Estimates of the Number in System at Time τ_i for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

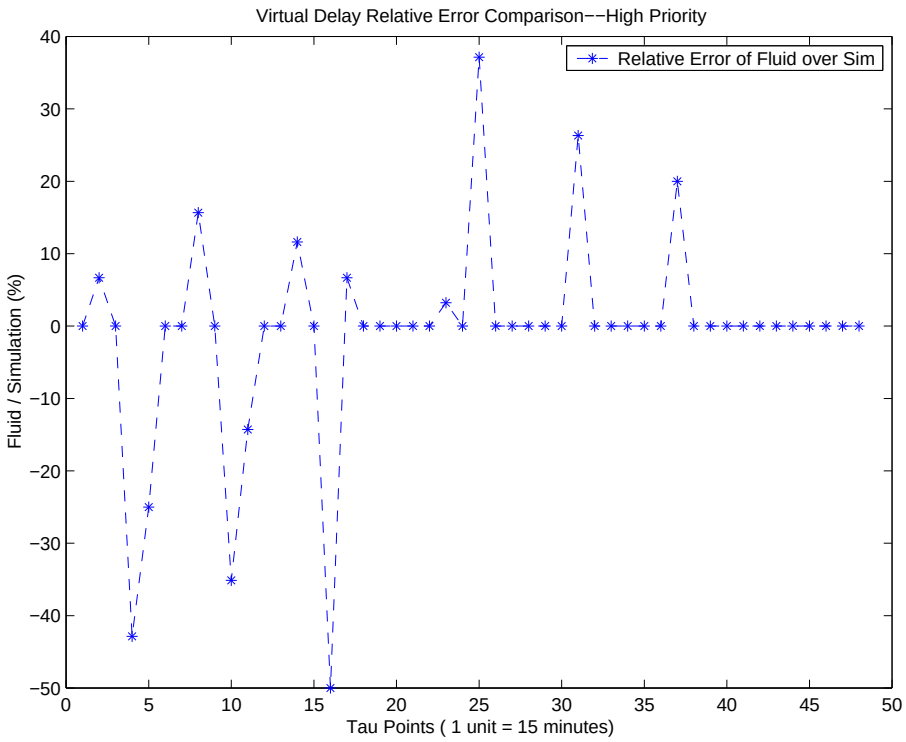

Figure 6.56: Case 2 - Relative Error for the Estimates of the Virtual Delay for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

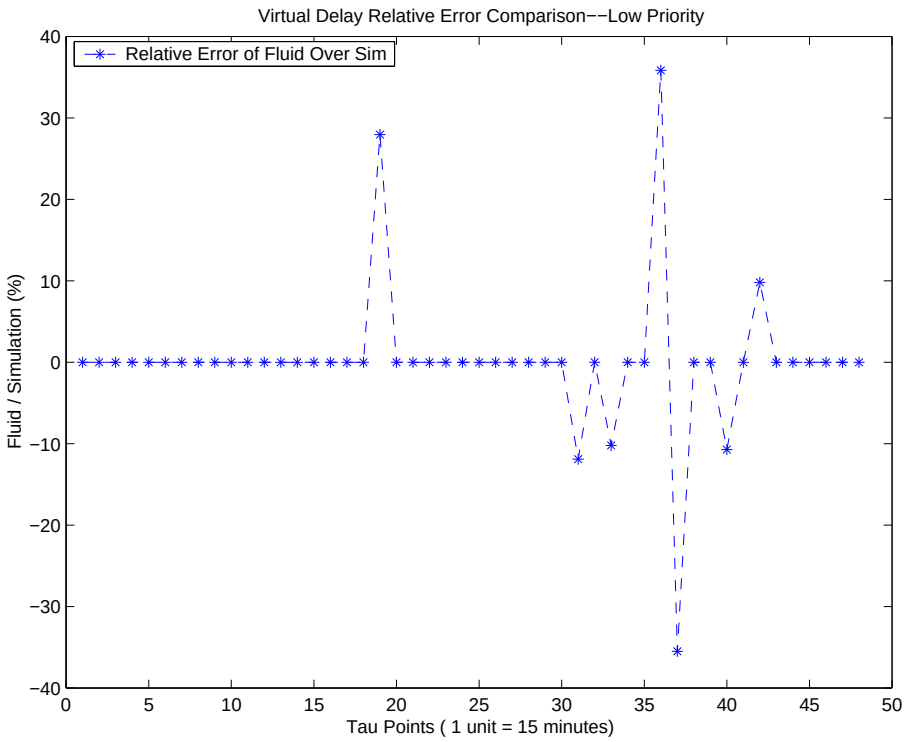

Figure 6.57: Case 2 - Relative Error for the Estimates of the Virtual Delay for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

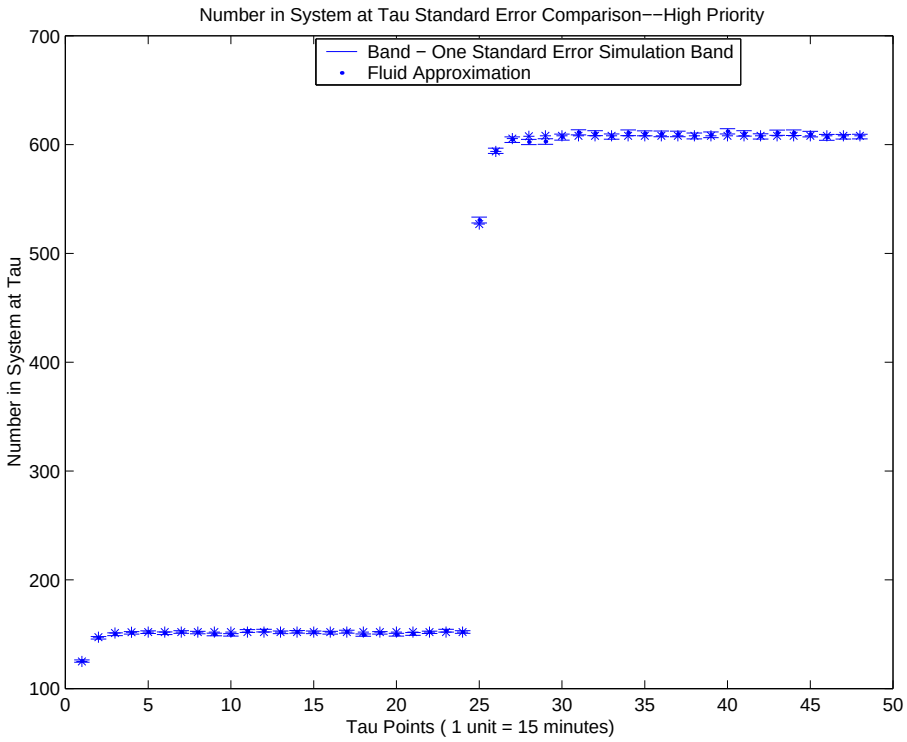

Figure 6.58: Case 2 - Standard Error Band for the Estimates of the Number in System at Time τ_i for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

In Figure 6.58, Figure 6.59, Figure 6.60, and Figure 6.61, we show the standard error in the two performance measure estimates for the high and low priority customers. The fluid estimate remains close to the simulation estimate itself throughout the total time horizon.

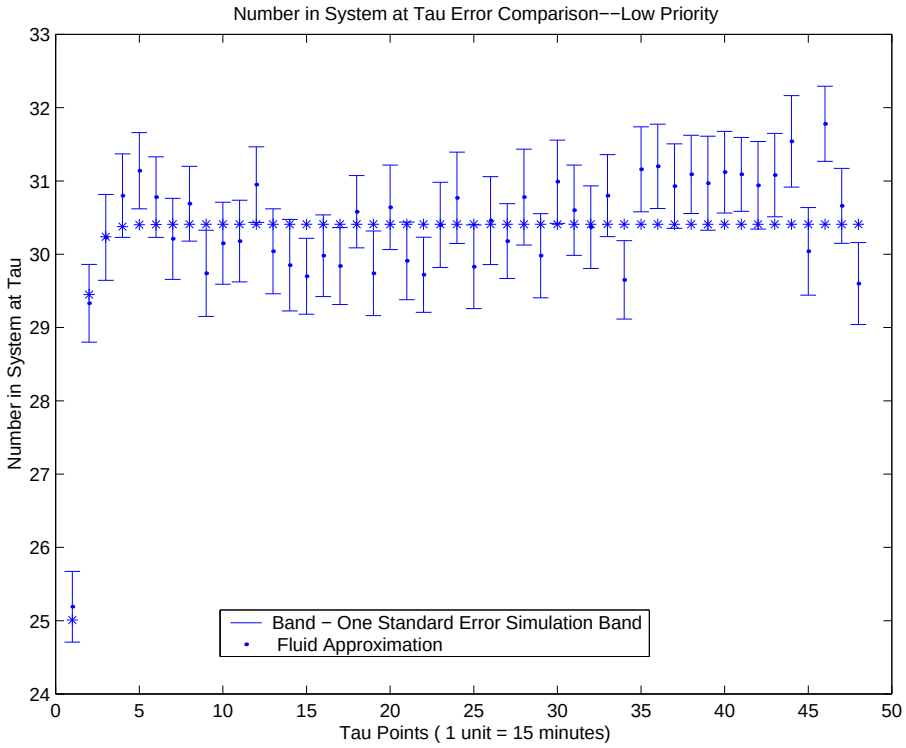

Figure 6.59: Case 2 - Standard Error Band for the Estimates of the Number in System at Time τ_i for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

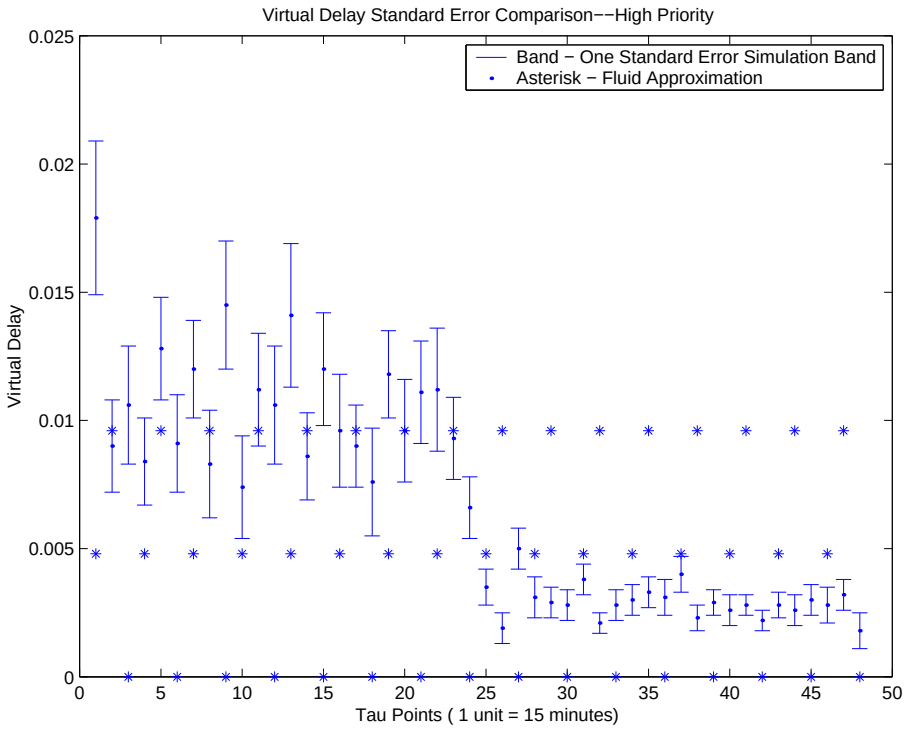

Figure 6.60: Case 2 - Standard Error Band for the Estimates of the Virtual Delay for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

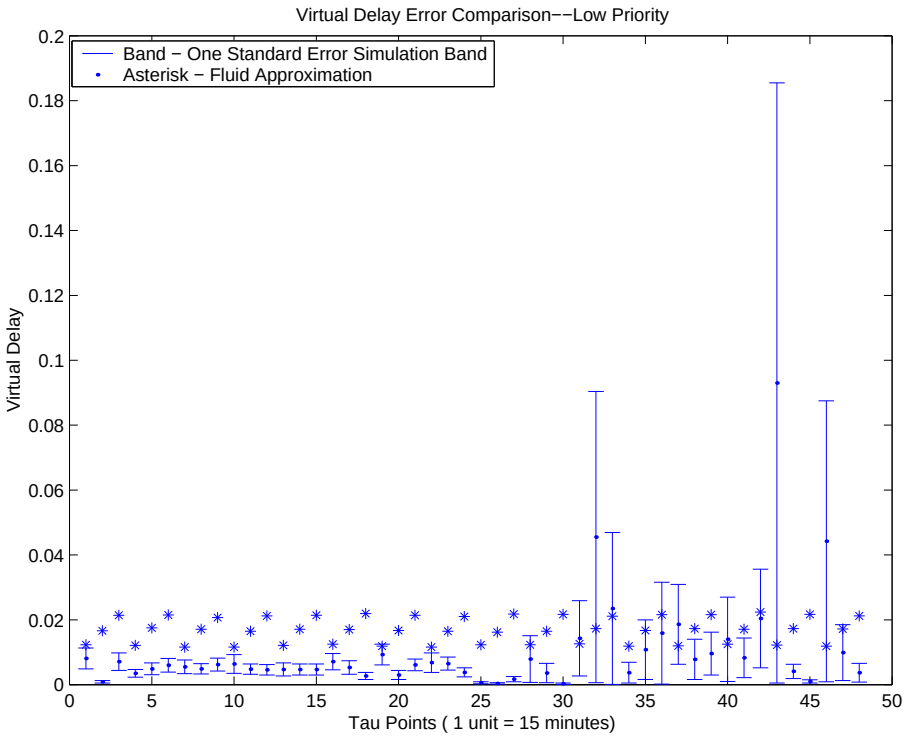

Figure 6.61: Case 2 - Standard Error Band for the Estimates of the Virtual Delay for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

6.2.6 Fluid vs. Simulation - Case 3 Arrival Rates

We computed the fluid and simulation estimates for the mean number in system and mean virtual waiting time using a somewhat different set of inter-arrival rates. These rates are also time-varying; however, the high and low priority inter-arrival rates vary between only two values. For the full 12 hours of the time horizon, the inter-arrival rates are chosen such that the system is under-loaded, or stable, i.e. $\rho < 1$. Here, over the 12 hours, the system remains well-within the under-loaded phase.

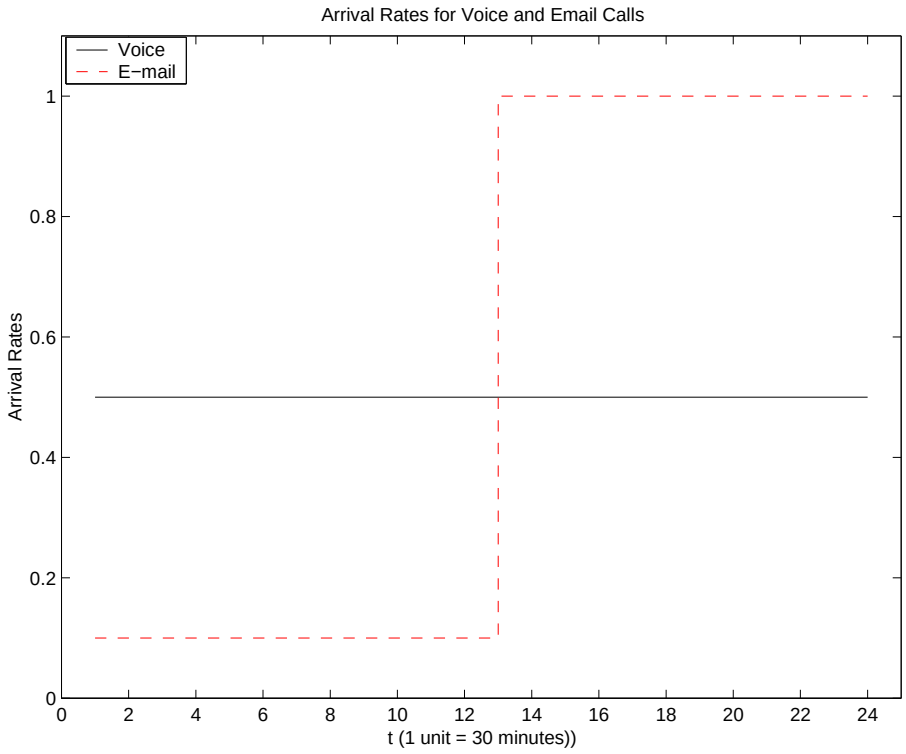

Figure 6.62: Piecewise Constant Arrival Function with Rates Varying at Time τ_i - Case 3

We provide these rates in Figure 6.62.

In Figures 6.63 and Figure 6.64, we show the comparison of the two perfor-

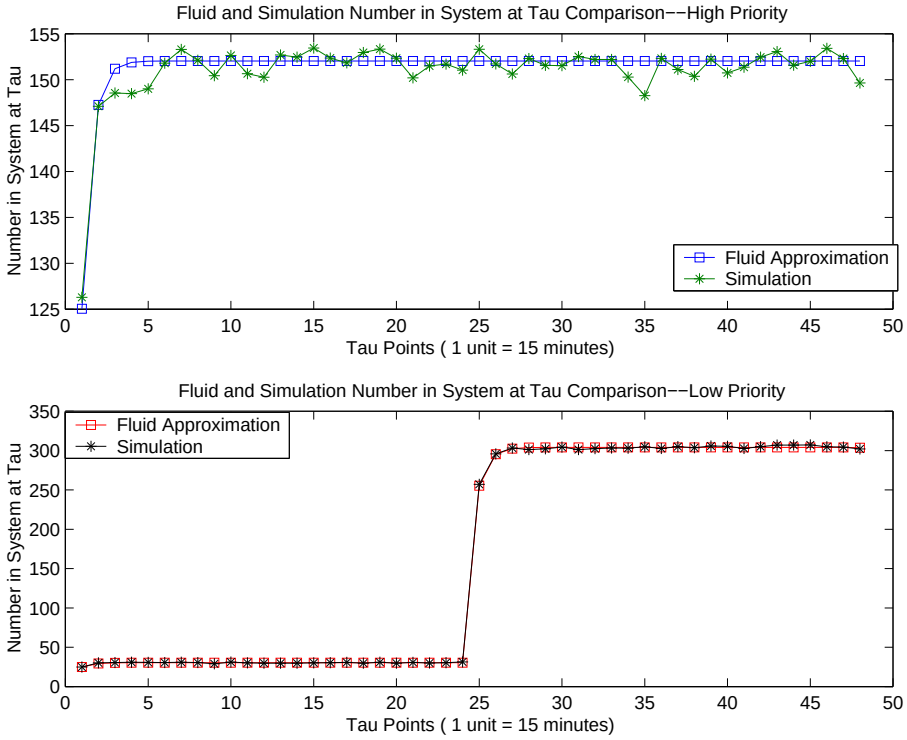

Figure 6.63: Case 3 - Estimates of the Number in System at Time τ_i for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

mance measure estimates, namely the number in system and virtual waiting time, for the high and low priority customers between our fluid and simulation models. The fluid estimates remain close to the simulation estimates for the mean number in system and the mean virtual waiting time over the total time horizon.

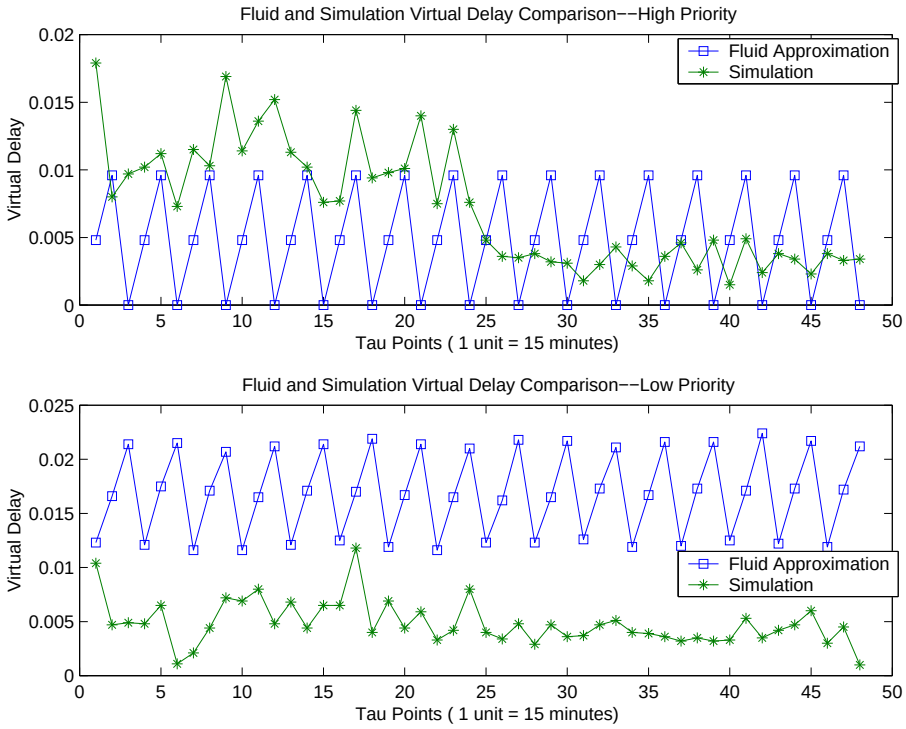

Figure 6.64: Case 3 - Estimates of the Virtual Delay for High and Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

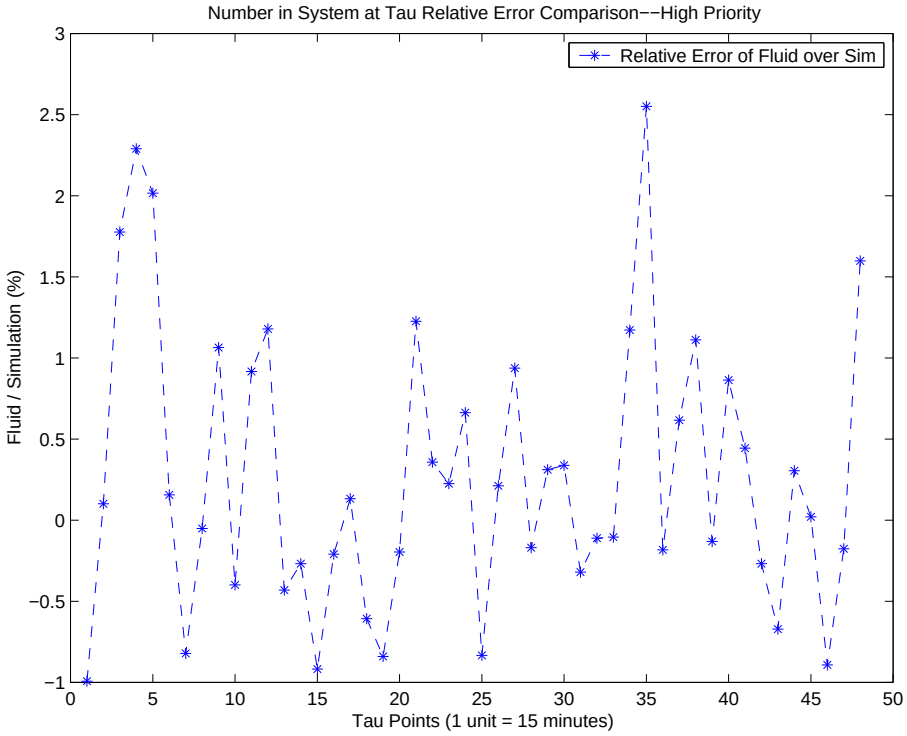

Figure 6.65: Case 3 - Relative Error for the Estimates of the Number in System at Time τ_i for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

In Figure 6.65, Figure 6.66, Figure 6.67, and Figure 6.68, we show the relative error in the two performance measure estimates for the high and low priority customers. Most of the relative error values remain relatively small, i.e. less than 10 percent throughout the total time horizon.

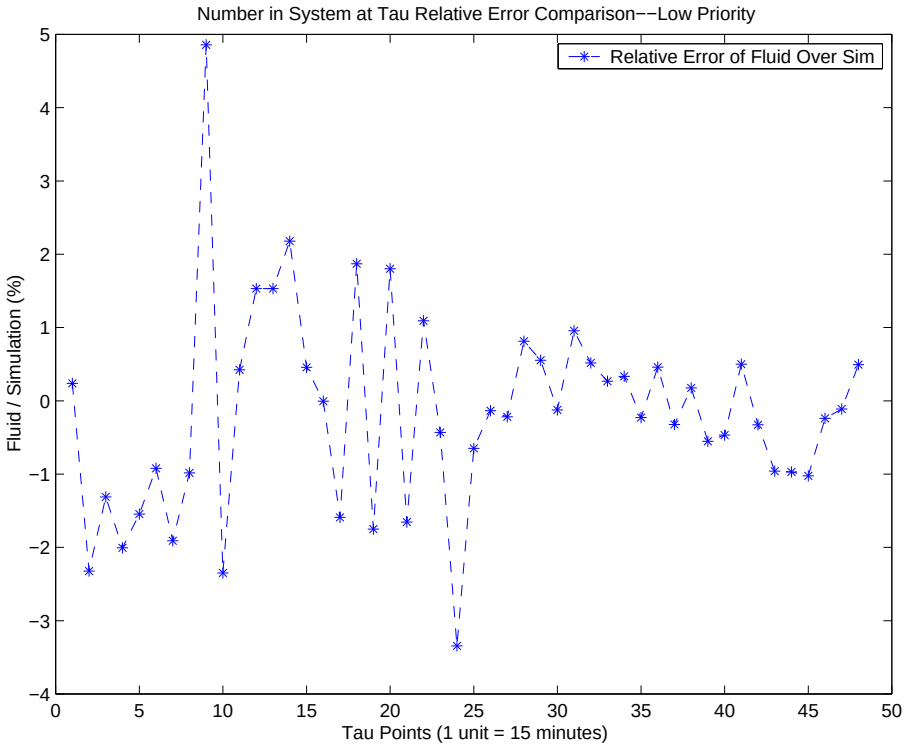

Figure 6.66: Case 3 - Relative Error for the Estimates of the Number in System at Time τ_i for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

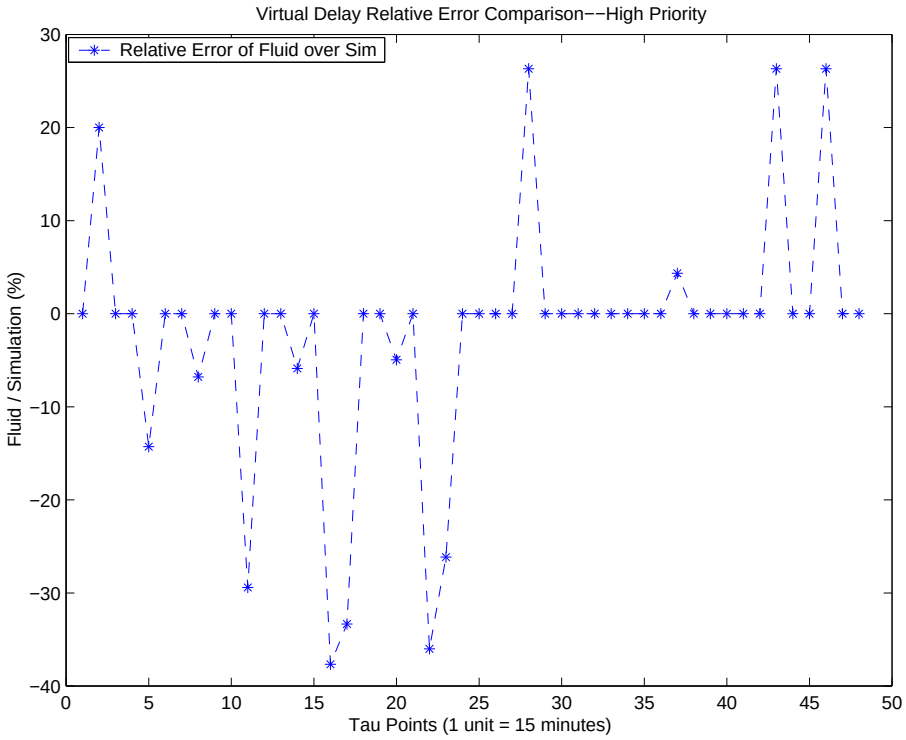

Figure 6.67: Case 3 - Relative Error for the Estimates of the Virtual Delay for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

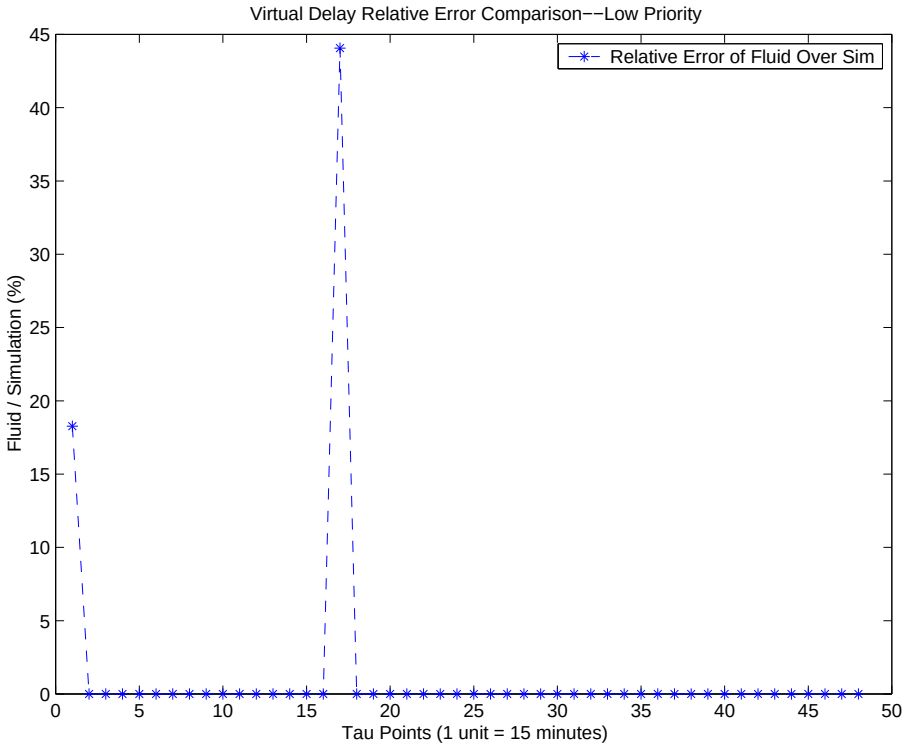

Figure 6.68: Case 3 - Relative Error for the Estimates of the Virtual Delay for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

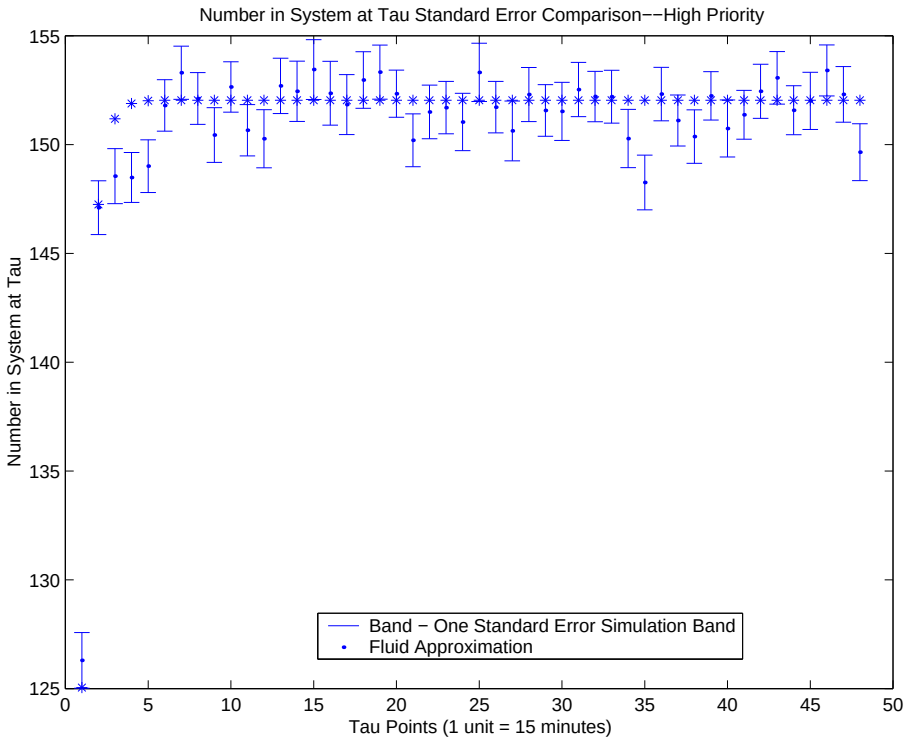

Figure 6.69: Case 3 - Standard Error Band for the Estimates of the Number in System at Time τ_i for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

In Figure 6.69, Figure 6.70, Figure 6.71, and Figure 6.72, we show the standard error in the two performance measure estimates for the high and low priority customers. The fluid estimate remains close to the simulation estimate itself throughout the total time horizon.

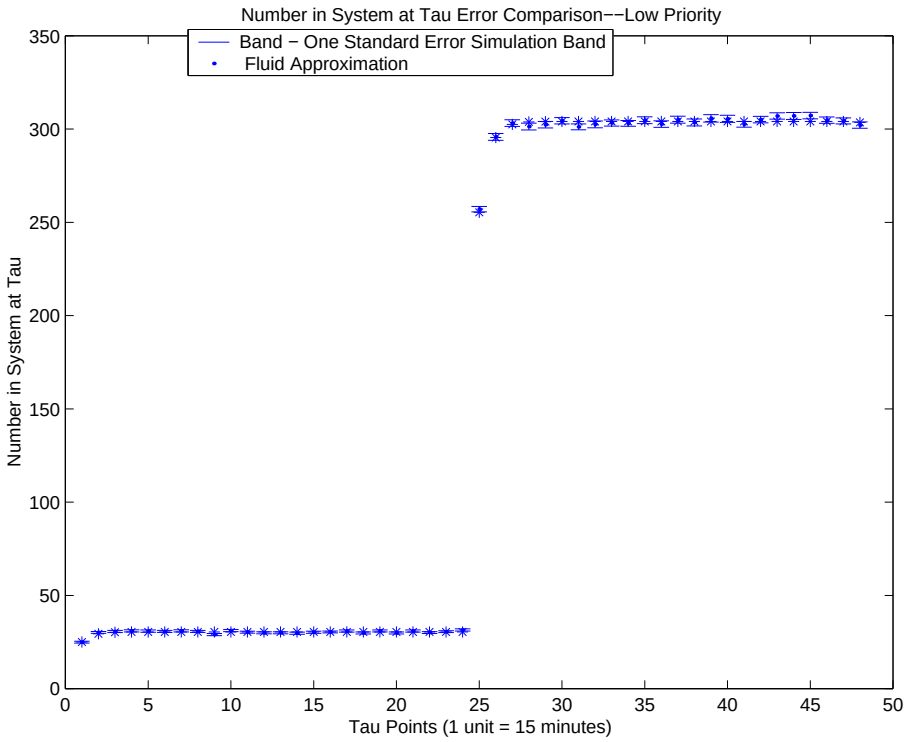

Figure 6.70: Case 3 - Standard Error Band for the Estimates of the Number in System at Time τ_i for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

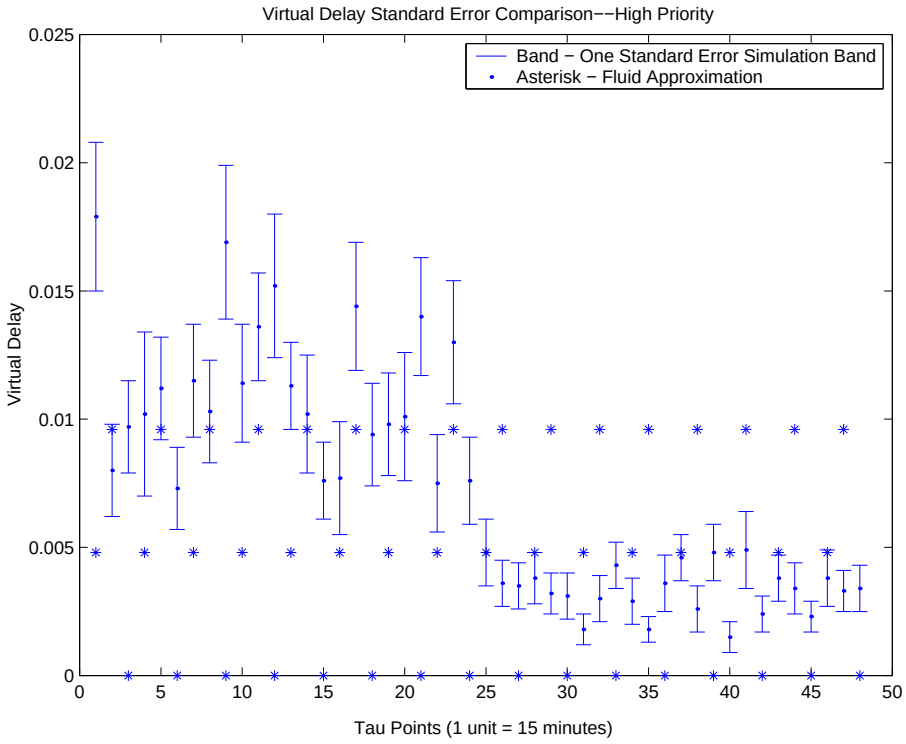

Figure 6.71: Case 3 - Standard Error Band for the Estimates of the Virtual Delay for High Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

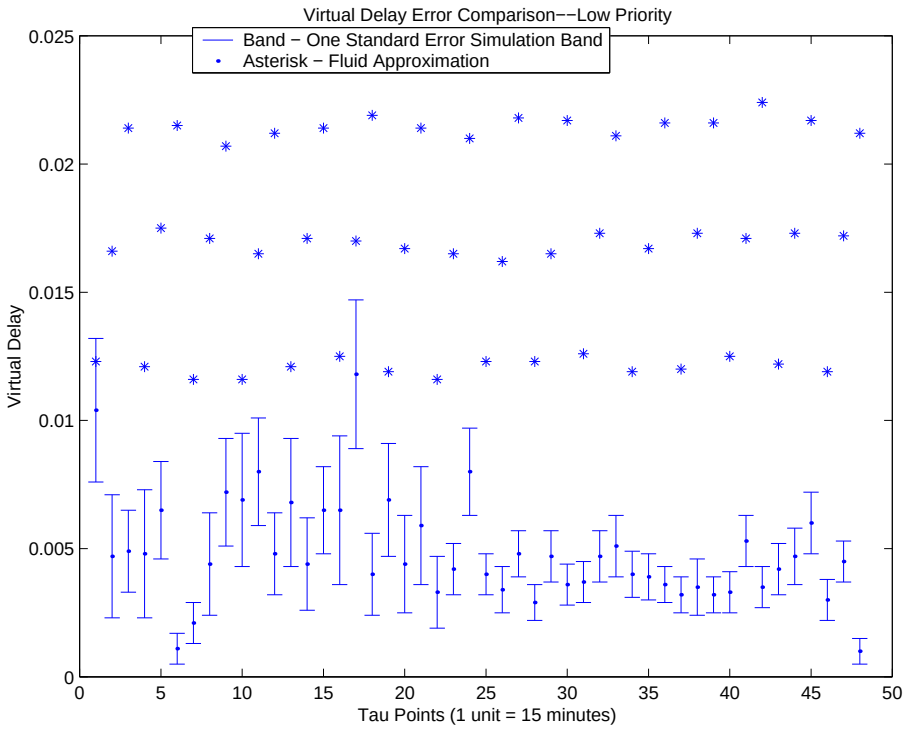

Figure 6.72: Case 3 - Standard Error Band for the Estimates of the Virtual Delay for Low Priority Customers for the Fluid vs. Simulation Comparison - Scale Factor $\eta = 35$

6.2.7 Optimal Staffing Level

With an initial number of servers of 700 servers, we estimate the optimal number of servers required to satisfy both the high and low priority customers' service levels simultaneously using our fluid approximations. The service level for the high priority customers is a waiting time of 30 seconds, while the service level for the low priority customers is a waiting time of 90 minutes. Using the fluid approximations, we computed the percentage of mean virtual waiting time estimates at τ_i that were below the target waiting time for both high and low priority customers. When 90 percent of the mean virtual waiting time estimates for each customer class satisfy the service levels for the first time, then we use the current number of servers as the optimal servers estimate. If not, we increment the current number of servers by 1, and repeat the process.

Using the methodology above, we compute an optimal servers estimate of 730, which corresponds to an unscaled estimate of 20.85, or 21 servers. This fluid estimate is substituted into the discrete-event simulation. We then use the simulation to compute the percentage of mean virtual waiting time estimates at τ_i that were below the target waiting time for both customer classes are computed. For 730 servers, the percentages for both classes did not satisfy the service level. Thus, we increased the number of servers by one, and repeated the computations until the first time the service levels for both classes are satisfied. For 755, or roughly 22 servers, the service levels for both classes of customers are satisfied for the first time using the simulation.

In Table 6.2 and Table 6.3, we summarize some of the percentage computations for a given number of servers. The optimal number of servers estimate using the fluid approximations is close to the optimal number using the simulation as

Optimal Number of Servers - Fluid Estimate

Servers	Percent High Priority Delays	Percent Low Priority Delays
700	83.33	89.58
705	85.42	94.79
710	87.50	94.79
720	89.58	97.92
730	91.67	98.96

Table 6.2: Optimal Number of Servers Computations - Fluid

Optimal Number of Servers - Simulation Estimate

Servers	Percent High Priority Delays	Percent Low Priority Delays
730	80.18	95.50
735	83.25	96.54
740	87.58	97.25
745	90.50	99.04

Table 6.3: Optimal Number of Servers Computations - Simulation

well.

The optimal search method for the number of servers dramatically increases the computation time of our fluid approximations model. By continuing to increase the number of servers based on the virtual waiting time meeting the service level criteria, the run time of the fluid method is significantly increased. The method normally runs in about 35 minutes. However, by checking if the high and low priority waiting times meet their service levels, the optimal search method now runs in about 2 hours, or 120 minutes.

The optimal scaled number of servers, 745, from our simulation model, corresponds to 21.3 or 22 "actual" servers, or agents. Using this actual number of servers in our simulation model, we find that the percentage of high and low priority customer delays that simultaneously meet both target service levels is 75.4 and 95.5, respectively. In our unscaled simulation model, both the high and low priority service levels are satisfied simultaneously for 25 servers. Here, the percentage values for the high and low priority customers are 90.2 and 98.2, respectively. This true optimal number of actual servers is still relatively close to our predicted optimal value of 22 from our fluid model.

6.3 Conclusions

We obtained fairly accurate fluid approximations to simulation estimates for modelling the performance of a two-class $M_t/M/n$ preemption-resume, dynamic priority queue. We demonstrated that our fluid estimates can be used to provide an accurate optimal staffing level. Also, the fluid approximations of the mean number in system and mean virtual waiting time at various time points τ_i were close to the simulation estimates for the high and low priority customer classes.

We measured the accuracy of our fluid estimates for time-varying inter-arrival rates, and under-loaded and overloaded system conditions. In all cases, our fluid estimates matched closely to the simulation ones. Lastly, we showed that the preemption-resume, dynamic priority service discipline can provide better performance for both customer classes.

Finally, the number of differential equations in our fluid approximations method is independent of the number of servers in the call center. Thus, the complexity of our fluid approximations method does not increase as the call center increases in size, e.g., in the number of agents. However, the simulation estimates will probably increase in complexity as the call center becomes larger. Therefore, our fluid approximation is a much more scalable solution than the simulation.

Chapter 7

Future Research

7.1 Model Variations

For a two-customer class, preemptive-resume priority $M_t/M/n$ queue, we measured the performance of a call center using fluid and simulation models to compute the mean number in system and mean virtual waiting time. We could extend our research by presenting a comparison of the diffusion approximations and simulation estimates for the variance of the number in system and virtual waiting time for our current model. However, we could vary our model characteristics in several ways to incorporate other aspects of a real-world call center. First, we could use a $M_t/M/n/L$ queueing model, where L denotes a finite limit on the trunk capacity, or number of telephone lines, available to a call center our model. Although such a model would restrict, or block, a number of incoming calls, a manager can control the costs of the telephone lines, which is sometimes important. With this model, we could optimize trunk capacity with respect to meeting target service levels. Also, we would be able to measure the blocking probability of arriving calls, which is another performance measure of interest to call center mangers.

Second, we could also vary our model to allow parameters, such as mean service times and number of servers, to vary over time. This would allow us to solve agent shift-scheduling problems more accurately, where managers sometimes vary the number of available agents in real-time to respond to varying system performance levels. Third, we could expand our analysis to p_i-classes of customers, $i = 1, \ldots, m$, $m \geq 3$, where the p_i-th customer has higher priority than the p_{i+1}-st customer. Fourth, we could vary the skill-levels of the agents. Thus, specific call types would be routed to agents with the skill-level to handle the call. An example of a real-world "skill-based routing" call center is one employing bilingual agents. Wallace [69] modelled a call center with multi-skilled agents to measure the effects of resource pooling. Since he did not consider priority classes of customers, we could combine our research to model a preemptive priority, multi-class call center with time-varying arrival rates and multi-skilled agents. We would be able to research resource pooling problems concerning the mixture of groups of agents with different skill-sets. Fifth, we could expand our analysis to virtual, or networks of call centers, instead of a single call center. Finally, we could determine the asymptotic waiting time distributions for both the high and low priority customer for our current model. We then could measure the service level in terms of a certain percentage of waiting times being below a target level using the tail of the waiting time distribution. Therefore, we can extend our research in several ways to the study other kinds of call centers.

7.2 Alternate Fluid and Diffusion Model

In addition to extending our current research to other types of call centers, we can also improve upon our current model. To better approximate the "abandon-

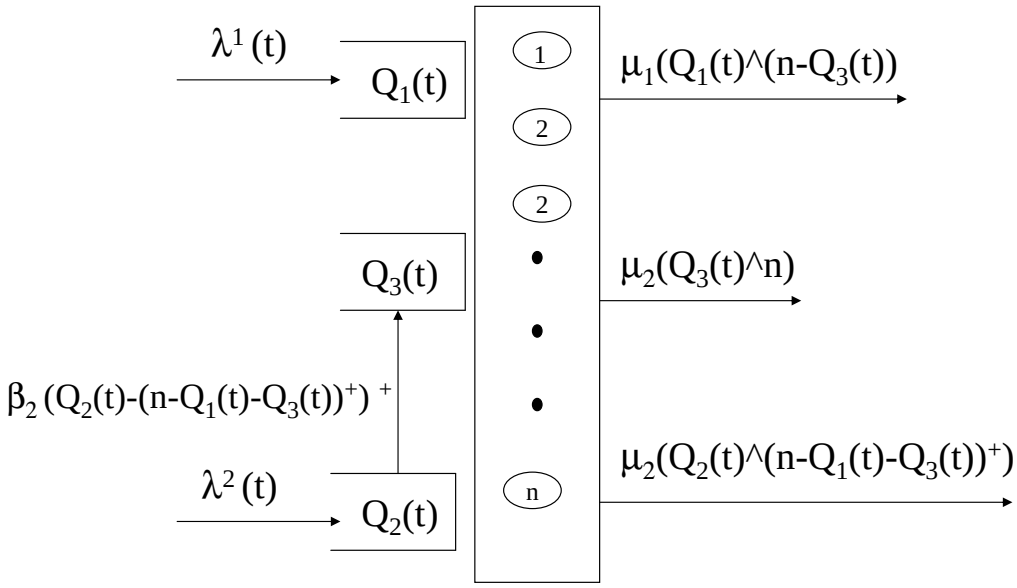

Figure 7.1: The Two-Customer Class, three-queue $M_t/M/n$ model with Abandonment

ment", or upgrade, process of low priority customers from their queue, we propose an alternate model. This alternate model allows low priority customers to abandonment to a third queue, completely separate from the high priority queue. The customers in this new queue will have the highest priority of any customer in the system. Thus, they will receive service before any high priority customers or low priority customers remaining in the low priority queue. However, they are not permitted to preempt any other customer from service.

We give a graphical representation of the model in Figure 7.1.

The differential equations governing the change in fluid levels for this process are similar to those of our current model. However, there is an additional equation for the third queue. The assumptions for functional strong law of large numbers theorem on the mean number in system fluid approximations remain the same, but the corresponding differential equations are now:

$$\begin{aligned}
\frac{d}{dt}Q_1^{(0)}(t) &= \lambda_1(t) - \mu[Q_1^{(0)}(t) \wedge (n - Q_3^{(0)}(t))]; && (7.1)\\
\frac{d}{dt}Q_2^{(0)}(t) &= \lambda_2(t) - \mu[Q_2^{(0)}(t) \wedge (n - Q_1^{(0)}(t) - Q_3^{(0)}(t))^+] \\
&\quad -\beta[Q_2^{(0)}(t) - (n - Q_1^{(0)}(t) - Q_3^{(0)}(t))^+]^+; && (7.2)\\
\frac{d}{dt}Q_3^{(0)}(t) &= \beta[Q_2^{(0)}(t) - (n - Q_1^{(0)}(t) - Q_3^{(0)}(t))^+]^+ - \mu(Q_3^{(0)}(t) \wedge n), && (7.3)
\end{aligned}$$

where we define $\lambda_3(t) = \beta[Q_1^{(0)}(t) - (n - Q_1^{(0)}(t) - Q_3^{(0)}(t))^+]^+$, representing the arrival rate of abandoning customers from the low priority queue into the third queue. The corresponding differential equations for the virtual waiting-time fluid approximations must be changed accordingly.

BIBLIOGRAPHY

[1] O.Z. Aksin and P.T. Harker. Capacity sizing in the presence of a common shared resource: Dimensioning an inbound call center. Working paper, 2001.

[2] E. Altman, T. Jimenez, and G.M. Koole. On the comparison of queueing systems with their fluid limits. *Probability in the Engineering and Information Sciences*, 15:165–178, 2001.

[3] R. Anupindi and B.T. Smythe. Call centers and rapid technology change. Teaching note, 1997.

[4] M. Armony and C. Maglaras. Customer contact centers with multiple service channels. Working paper, 2001.

[5] Avaya. Definity servers - product architecture. Downloadable from www.avaya.com/ac/common/index.jhtml?location=M1H1005G1002F2013P3044N4318, 2003.

[6] T. Aykin. Optimal shift scheduling with multiple break windows. *Management Science*, 42:591–602, 1996.

[7] F. Baccelli and G. Hebuterne. On queues with impatient customers. *Performance '81*, pages 159–179, 1981.

[8] H. G. Bernett, M. J. Fischer, and D. M. B. Masi. Internet Protocol/Public Switched Telephone Network blended call center performance analysis. *The Telecommunications Review*, 12:51–60, 2001. Mitretek Systems,Inc.

[9] H. G. Bernett, M. J. Fischer, and D. M. B. Masi. Web-enabled call centers - a progress report. *Business Communciations Review*, pages 38–42, 2002. July 2002 edition.

[10] S.C. Borst, A. Mandelbaum, and M.I. Reiman. Dimensioning large call centers. Working paper, 2000.

[11] A. Brandt, M. Brandt, G. Spahl, and D. Weber. Modelling and optimization of call distribution systems. *Proceedings of the 15th International Teletraffic Conference*, pages 133–144, 1997. V. Ramaswami and P.E. Wirth, editors, Elsevier Science.

[12] A.J. Brigandi, D.R. Dargon, M.J. Sheehan, and T. Spencer III. At&t's call processing SIMULATOR operational design for inbound call centers. *Interfaces*, 24(1):6–28, 1994.

[13] A. Cobham. Priority assignment in waiting line problems. *Operations Research*, 2:70–76, 1954.

[14] Geotel Communications Corporation. CTI: Re-engineering call centers. Technical report, 1998. White Paper.

[15] Response Design Corporation. Call center metric database-questions and answers. Downloadable from responsedesign.com/qa/index.asp, 2003.

[16] R. H. Davis. Waiting time distribution of a multiserver, priority queueing system. *Operations Research*, 14:133–136, 1966. Presented at 10th Annual

1st International Meeting of Operations Reserach Society of America, Western Section.

[17] D. Duxbury, R. Backhouse, M. Head, G. Lloyd, and J. Pilkington. Call centres in BT UK customer service. *British Telecommunications Engineering*, 18:165–173, 1999.

[18] M.C. Fu, S.I. Marcus, and I.J. Wang. Monotone optimal policies for a transient queueing staffing problem. *Operations Research*, 48:327–331, 2000.

[19] N. Gans, G. Koole, and A. Mandelbaum. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing and Service Operations Management*, 5:79–141, 2003.

[20] O. Garnett and A. Mandelbaum. An introduction to skills-based routing and its operational complexities. Downloadable from ie.technion.ac.il/serveng/References, 2001.

[21] O. Garnett, A. Mandelbaum, and M. Reiman. Designing a call center with impatient customers. Working paper, 2001.

[22] J.J. Gordon and M.S. Fowler. Accurate force and answer consistency algorithms for operator services. *Proceedings of the* 14*th International Teletraffic Conference*, pages 339–348, 1994. J. Labetoulle and J.W. Roberts, editors.

[23] L. Green and P. Kolesar. Testing the validity of a queueing model of police patrol. *Management Science*, 37:84–97, 1989.

[24] L. Green and P. Kolesar. The pointwise stationary approximation for queues with non-stationary arrivals. *Management Science*, 37:84–97, 1991.

[25] G.R. Grimmett and D.R. Stirzaker. *Probability and Random Processes*. Oxford University Press, New York, 1992.

[26] B. Halachmi and W.R. Franta. A diffusion approximation to the multiserver queue. *Management Science*, 24:522–529, 1978.

[27] S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations Research*, 29:567–587, 1981.

[28] R.W. Hall. *Queueing Methods for Services and Manufacturing*. Prentice Hall, Englewood Cliffs, New Jersey, 1991.

[29] C.M. Harris, K.L. Hoffman, and P.B. Saunders. Modeling the IRS telephone taxpayer information system. *Operations Research*, 35:504–523, 1987.

[30] W.B. Henderson and W.L. Berry. Heuristic methods for telephone operator shift scheduling: An experimental analysis. *Management Science*, 22:1372–1380, 1976.

[31] P. Hokstad. Approximations for the M/G/m queue. *Operations Research*, 26(3):511–523, 1978.

[32] J. R. Jackson. Queues with dynamic priority discipline. *Management Science*, 8(1):18–34, 1961.

[33] N.K. Jaiswal. *Priority Queues*. Academic Press, New York, 1968.

[34] O.B. Jennings, A. Mandelbaum, W.A. Massey, and W. Whitt. Server staffing to meet time-varying demand. *Management Science*, 42:1383–1394, 1996.

[35] G. Jongbloed and G.M. Koole. Managing uncertainty in call centers using poisson mixtures. Tech. Report 2000-3, Dept. of Stochastics, Vrije Universiteit Amsterdam, www.cs.vu.nl/ koole/papers/S2000-3.ps, 2000.

[36] O. Kella and U. Yechiali. Waiting times in the non-preemptive priority M/M/c queue. *Communications in Statistics-Stochastic Models*, 1(2):257–262, 1985.

[37] H. Kesten and J. Th. Runnenburg. Priority in waiting-line problems I and II. *Indag. Mathematik*, 19:312–336, 1957.

[38] A. Y. Khinchin. Mathematical theory of stationary queues. *Mathematik Sbornik*, 39:73–84, 1932.

[39] G. Kim. Call centers: Present and future. *X-Change*, pages 26–29, 1997.

[40] J. K. C. Kingman. Inequalities in the theory of queues. *Journal of Royal Statistics Society*, B32:102–110, 1970.

[41] L. Kleinrock. *Queueing Systems, Volume I: Theory*, volume 1. John Wiley & Sons, New York, Chichester, Brisbane, Toronto, 1975.

[42] L. Kleinrock. *Queueing Systems, Volume II: Computer Applications*, volume 2. John Wiley & Sons, New York, London, Sydney, Toronto, 1976.

[43] Y. Kogan, Y. Levy, and R.A. Milito. Call routing to distributed queues: Is FIFO really better than MED? *Telecommunications Systems*, 7:299–312, 1997.

[44] G. Koole and A. Mandelbaum. Queueing models of call centers: An introduction. Downloadable from www.cs.vu.nl/obp/callcenters and il.techion.ac.il/serveng, 2001.

[45] B.W. Kort. Models and methods for evaluating customer acceptance of telephone connections. *IEEE*, pages 706–714, 1983.

[46] M.S. Lane, A.H. Monsour, and J.L. Harpell. Operations research techniques: A longitudinal update 1973-1988. *Interfaces*, 23:63–68, 1993.

[47] A. Law and W. D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, Boston, New York, Toronto, 2000.

[48] D.H. Lehmer. Mathematical methods in large-scale computing units. *Annals of Computational Laboratories - Harvard University*, 26:141–146, 1951.

[49] A. Mandelbaum, W.A. Massey, and M.I. Reiman. Strong approximations for markovian service networks. *Queueing Systems*, 30:149–201, 1998.

[50] A. Mandelbaum, W.A. Massey, M.I. Reiman, and R. Rider. Time-varying multiserver queues with abandonments and retrials. *Proceedings of the Sixteenth International Teletraffic Conference*, 1999. P. Key and D. Smith, editors.

[51] A. Mandelbaum, W.A. Massey, M.I. Reiman, R. Rider, and A. Stolyar. Queue lengths and waiting times for multiserver queues with abandonments and retrials. *Proceedings of the Fifth INFORMS Telecommunications Conference*, 2001.

[52] A. Mandelbaum, A. Sakov, and S. Zeltyn. Empirical analysis of a call center. Downloadable from ie.technion.ac.il/serveng/References, 2001.

[53] A. Mandelbaum and N. Shimkin. A model for rational abandonments from invisible queues. *Queueing Systems*, 36:141–173, 2000.

[54] E. Zohar A. Mandelbaum and N. Shimkin. Adaptive behavior of impatient customers in tele-queues: Theory and empirical support. Working paper, 2000.

[55] G.F. Newell. *Approximate Stochastic Behavior of n-Server Service Systems with Large n.* Springer-Verlag, Berlin, 1973.

[56] G.F. Newell. *Applications of Queueing Theory.* Chapman and Hall, London, 1982.

[57] C. Palm. Methods of judging the annoyance caused by congestion. *Tele*, 4:189–208, 1953.

[58] M. Pinedo, S. Seshadri, and J.G. Shanthikumar. *Creating Value in Financial Services: Strategies, Operations, and Technologies.* Kluwer, 1999. E.L. Melnick, P. Nayyar, M.L. Pinedo, and S. Seshadri, editors.

[59] F. Pollaczek. I-ii $\ddot{U}$ber eine aufgabe dev wahrscheinlichkeitstheorie. *Mathematik Zeitschrift*, 32:64–100, 1930.

[60] A. A. Puhalskii. On the invariance principle for the first passage time. *Mathematics of Operations Research*, 19:946–954, 1994.

[61] Prosci Research. Web-enabled call centers. Downloadable from call-center.net/web-customer-service.htm, 2001.

[62] J. Riordan. *Stochastic Service Systems.* Wiley, 1961.

[63] M. Segal. The operator scheduling problem: A network-glow approach. *Operations Research*, 24:808–823, 1974.

[64] J. Stoer and R. Bulirisch. *Introduction to Numerical Analysis*. Springer-Verlag, New York, 1980.

[65] D.Y. Sze. A queueing model for telephone operator staffing. *Operations Research*, 32:229–249, 1984.

[66] L. Takacs. Priority queues. *Operations Research*, 12(1):63–74, 1963.

[67] H. Takagi. *Queueing Analysis: A Foundation of Performance Evaluation*, volume 1. North-Holland, Amsterdam, New York, Oxford, Tokyo, 1991. Vacation and Priority Systems, Part 1.

[68] G.M. Thompson. Improved implicit optimal modeling of the labor shift scheduling problem. *Management Science*, 41:595–607, 1995.

[69] Rodney B. Wallace. *Performance Modelling of Call Center with Skill-Based Routing*. PhD thesis, George Washington University, 2003.

[70] W. Whitt. Improving service by informing customers about anticipated delays. *Management Science*, 45:192–207, 1999.

[71] W. Whitt. Predicting queueing delays. *Management Science*, 45:870–888, 1999.

[72] T. M. Williams. Non-preemptive multi-server priority queues. *Journal of Operational Research Society*, 31:1105–1107, 1980.

[73] R. Wolff. *Stochastic Modeling and the Theory of Queues*. Prentice Hall, Englewood Cliffs, New Jersey, 1989. W.J. Fabrycky and J.H. Mize, editors.