INTEGRATED ANALYTICAL PERFORMANCE EVALUATION MODELS OF WAREHOUSES

By

KARTHIK AYODHIRAMANUJAN

Bachelor of Engineering (Mechanical)
S. V. National Institute of Technology
Surat, Gujarat, India
1998

Master of Science (Mechanical Engineering)
Oklahoma State University
Stillwater, Oklahoma, USA
2001

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2009

INTEGRATED ANALYTICAL PERFORMANCE EVALUATION MODELS OF WAREHOUSES

Dissertation Approved:

Dr. Manjunath Kamath

Dissertation Advisor

Dr. Ricki G. Ingalls

Dr. David B. Pratt

Dr. Dursun Delen

Dr. Gordon Emslie

Dean of the Graduate College

Acknowledgments

There is an Indian adage about one's spiritual evolution, "Maata, Pitha, Guru, Deivam"; Maata (mother) identifies the child to its Pitha (father), who shows the child to the Guru (teacher), who leads one to the path to Deivam (God). First and foremost, I wish to dedicate my work to my father (Late) Mr. Ayodhiramanujan and mother, Mrs. Ganapriya Ayodhiramanujan.

There are no words that can express my gratitude to my advisor, Dr. Manjunath Kamath, who is not only my teacher but also my mentor, for providing me with valuable guidance and immense support at all times. I would like to thank my wife, Janani Murali, who by just being there for me gave me great strength in the last three years.

Special thanks go to my committee members, Dr. Ricki Ingalls, Dr. David Pratt and Dr. Dursun Delen for their suggestions in simulation modeling, many aspects of warehouse processes and in general sharing their experiences with me, thereby bolstering my interest and confidence.

I would like to take this opportunity to thank my friends and colleagues at the Center for Computer Integrated Manufacturing enterprises who made the research center, a home away from home. I would like to thank Sandeep Srivathsan for his tremendous help especially during my work from Dallas.

Finally, I would like to acknowledge the financial support provided by Center for Engineering Logistics and Distribution, and Department of Biosystems Engineering at Oklahoma State University.

As always, "*Sarvam Krishnarpanam*", everything at the lotus feet of Lord Krishna.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With rapidly changing business scenarios, the role of warehouses is becoming increasingly critical in the efficient management and success of supply chains. With respect to their position in a supply chain, Frazelle (2001) classified the warehouses as

- Production warehouses
- Finished goods warehouses and fulfillment centers
- Distribution warehouses
- Contract warehouses

Production warehouses hold raw materials or work-in-process inventory for use by manufacturing facilities. Finished goods warehouses store finished products, typically in pallet loads that serve as a buffer against uncertainties in customer demand. With the proliferation of e-commerce, fulfillment centers are shipping small quantities or individual items to end customers directly. Distribution warehouses accumulate and consolidate products from multiple manufacturing facilities for multiple customers. Contract warehouses are operated by a third party organization for one or more customers.

Two major factors that have caused a change in the focus of warehousing systems are the evolution of manufacturing concepts like Just-In-Time and the evolution of information systems and technology. Mass customization and global competition are requiring supply

chain partners to be more flexible with respect to product demand and product mix. Frequent delivery in low volumes for a wide range of products is the focus of current supply chains, thus moving many small, but important value-added services closer to the customer. With the renewed emphasis on customer satisfaction and integrated supply chain management, warehouses are not the traditional storage locations they once used to be. Today's warehouses are responsive to customer demands by providing value-added services such as last minute customization, small assembly, labeling, kitting, and special packaging. Hence, warehouse operations are not only more productive but also more complicated than ever before.

Modern information systems have enabled the traditional warehouses plan their operations more effectively. Concepts such as cross-docking have received more attention; the results include the reduction of the time a product spends at a warehouse and the elimination of some storage and double-handling of products. With customer demand-patterns evolving continuously, the drive to reduce cost and extreme competition have forced warehouses to devote a lot of effort to constantly improving their methods and systems. In such a dynamic environment, modeling and analysis of the underlying warehouse systems and continuous improvement of their operations becomes critical for their effective and efficient design and control.

1.1 Warehousing Systems

Warehousing systems are one of the most researched components of a supply chain. Many authors have provided excellent reviews of warehousing systems, see for example, Berg (1999), Berg & Zijm (1999), Yoon & Sharp (1996), and Tompkins et al. (2003). Depending on the position of the warehouse in the supply chain, the activities within the warehouse and the form of material handled are determined. The typical activities in a warehouse are summarized in Figure 1.1.

Receiving is the collection of all the activities related to the orderly receipt of goods, inspection (for quantity and quality) and disbursing to storage or cross-docking for immediate shipping.

Repackaging is the process of splitting the products that are ordered in bulk quantities and repacking to customer specifications (single or carton/case), or assembling to form kits with other parts of a customer shipment. Some part of the load might also be held in storage for future shipment. This function is also called break-bulk operation.

Putaway is the process of placing the merchandise in either long-term storage (reserve) or short-term storage (forward).

Order-picking is the process of retrieving items from the storage area to meet a specific demand. Many classifications of warehousing systems exist based on the type of order picking which is the most cost intensive process in the warehouse.

Sortation is the process of sorting the accumulated batch picks into individual orders.

Cross-docking is the process of staging the inbound goods directly to shipping without sending it to storage.

Replenishment is the process of refilling the primary and secondary picking areas from long-term storage.

Shipping includes all the activities related to checking the order for completeness and appropriate packaging; determining shipping charges; accumulating orders by outbound trailer; and loading the trailers.

1.2 Warehouse Activity Description

The basic functions common to all the warehouses are receiving, putaway or storage, picking, and shipping. A typical warehouse with reserve and forward storage areas, together with the material flow is shown in Figure 1.2. In this section, we will describe the configuration of warehouses with particular attention to storage and retrieval operations.

A typical material flow in a warehouse starts with the incoming trucks arriving into the yard. The trucks either deliver the trailers directly to the dock or wait in a queue till a dock door is available. Once the trailer occupies a dock door, a worker crew (strippers) is assigned

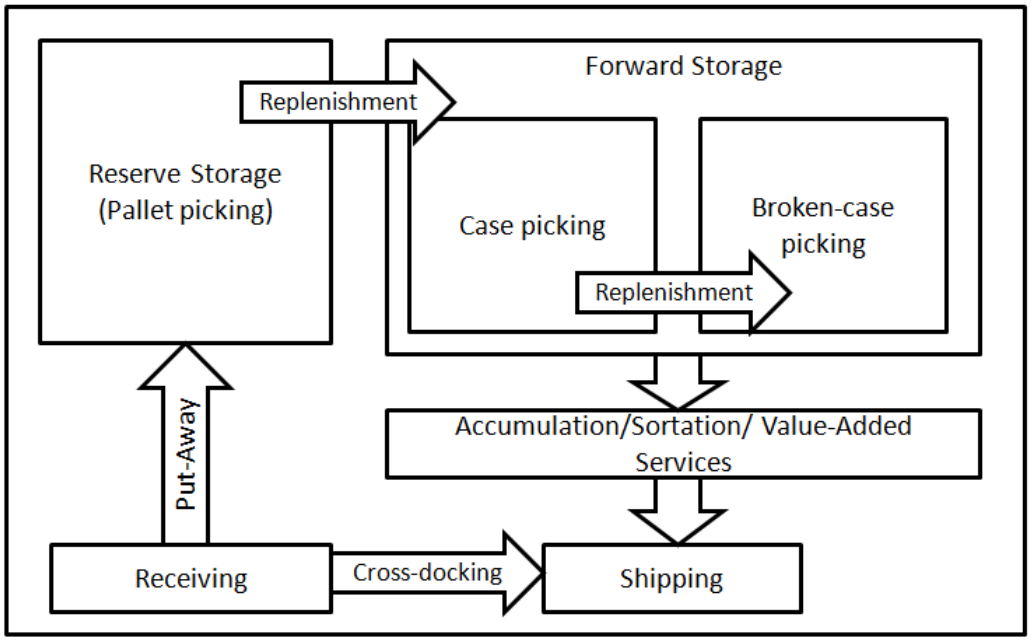


Figure 1.1: Functional areas and product flow in a typical warehouse

to unload the trailer. In this dissertation, we assume that trailers contain pallet loads. The workers unload all the contents of the trailer and place the items in the receiving/staging area for further processing (e.g. inspection). The dock door is then scheduled to unload the next waiting trailer. The stripper or another employee verifies the contents of the trailer for quality in the receiving/staging area.

The stripper may place the items for cross-docking or long-term storage. Discrete or continuous material handling devices may be used to move the items from the receiving area. The items meant for long-term storage usually do not alter their unit-load configuration. Workers/fork lifts move the pallets into the reserve storage area. This process is automated in some warehouses using Automated Guided Vehicles (AGVs). The reserve storage may be as simple as a rack storage system or an Automated Storage and Retrieval System (AS/RS). In the case of AS/RS, the storage into the racks and retrievals from the racks are performed by Storage/Retrieval (S/R) machines.

A warehouse dealing with less-than-pallet-loads may have two different storage areas – reserve or long term storage for pallets and forward or short term storage for cases. The presence of forward and reserve areas necessitates an internal replenishment policy. There

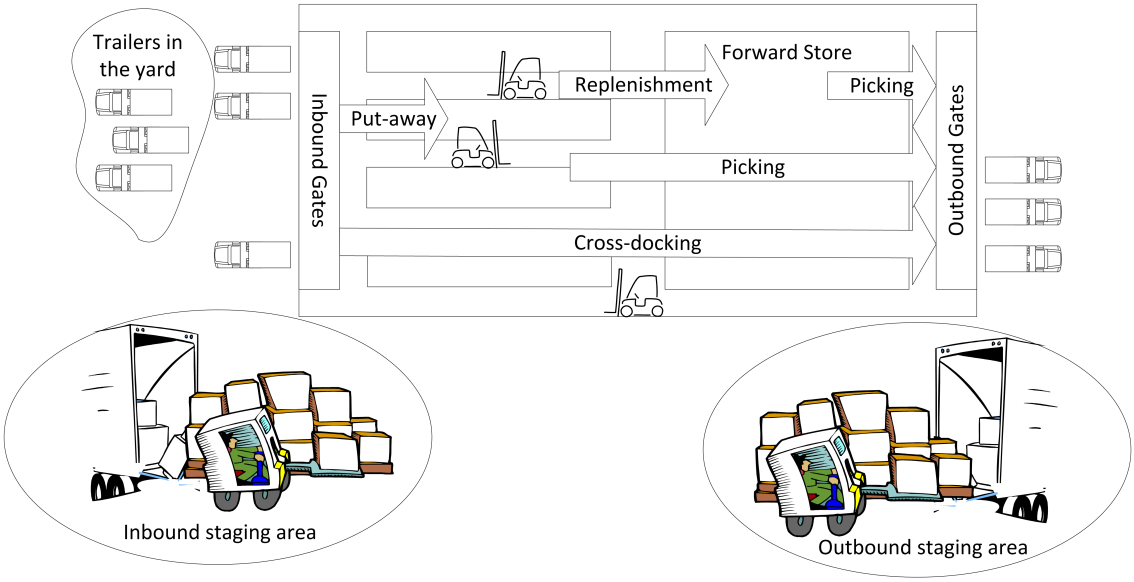


Figure 1.2: A typical warehouse with forward-reserve inventory

is a break bulk operation, i.e. pallets from the reserve storage are broken into individual cases. Typically the order pickers picking from the forward area pick individual cases from the replenished pallet loads.

Some warehouses may deal with individual items. In such scenarios, there is another break bulk like operation called the split-case operation. The order pickers may pick individual items from the cases; sort and assemble an order before shipping. Each order is defined by number of unique line items and related quantities.

Once an order is received for an item, orders are picked either from the forward or reserve storage. Items are accumulated in a shipping/staging area to be loaded on to the trailers. The orders are verified to ensure quality and items are loaded by a worker crew (stackers). The items are assembled to form a tight packing and order integrity is preferred, i.e., items in the same order are shipped together. Some of the factors that influence the retrieval of items include:

- Order picking method – single, dual or multiple command
- Material handling equipment properties – capacity of the carts, fork lifts or pallet jacks

- Layout of the terminals – multiple cross aisles
- Storage assignment policies – random, dedicated, or class based
- Clustering of items in storage
- Order batching and hence, associated sortation process if necessary
- Presence of forward-reserve storage areas

In some warehouses, palletization may be an additional process to build unit load pallets that could consist of similar items or a mixed load.

1.3 Performance Evaluation of Warehouses

Suri et al. (1993) defined performance evaluation (PE) as "a methodology (including techniques and tools) for determining the performance measures that can be expected to result from a given set of decisions."

Performance evaluation usually employs simulation models or analytical models. Simulation models are dynamic in nature and model the evolution of the system over time. These are detailed models and model development takes considerable effort. Analytical models, also called as aggregate dynamic models, account for some uncertainties and interactions in the system using mathematical or symbolic relationships. These models can be used for rapid analysis of many design configurations albeit at an aggregate level. Analytical models based on stochastic Petri nets, Markov chains, and queueing theory provide rapid analysis capability and can provide insights into the behavior of the system. As with any performance evaluation model, there is a trade-off between model detail and tractability.

Some of the performance measures of interest in a warehouse environment are throughput, average response time, fill rate, and utilization of space, equipment, and human resources, in addition to financial metrics. Schefczyk (1990) provides a comprehensive list of performance measures related to warehouses.

Many authors have developed performance evaluation models for warehousing systems. A detailed review is provided in the following chapter. To a great extent, these models

focus on a particular system or class of systems within a bigger facility, for example End-of-Aisle order picking systems (Bozer & White (1990)), AS/RS (Abdelkrim et al. (2003), and Lee (1997)), and sortation systems (Bozer et al. (1988) and Johnson and Meller (2002)), which are important sub-systems of a warehouse. Some of the models focus on limited and well-defined isolated problems like routing and sequencing of order pickers or dwell point determination, neglecting the interaction amongst system components.

1.4 Motivation for the Current Research

Warehouse system design is a complex process with numerous alternatives at all design levels for the designer to consider and evaluate. For example, a warehouse might deal with more than one product configuration (pallet, case, or item), choice of storage systems (AS/RS, carousels, or bin shelves), and storage policies (random, class-based or dedicated). Warehouse design decisions typically focus on three important aspects; the throughput capacity, the size of the inventory to be stored and the material handling equipment requirements. Enumerating all feasible solutions that satisfy the throughput and storage capacity requirements and finding an optimal solution is not practical. Until now, the decision-makers have relied on experience and descriptive, systematic procedures to select a set of feasible candidates of warehouse design.

The literature on integrated models focuses either on descriptive design methodologies (e.g., Ashayeri & Goetschalckx (1988)) or sequential solution approaches. In addition, the warehouse managers have limited ready-made tools to evaluate the warehouse performance or resource requirement if a new operation is to be incorporated into the material flow, for example, repackaging.

Simulation is the preferred tool for evaluating the designs. Building a warehouse simulation model takes considerable time and effort, though computing power is no longer a constraint. Analytical models for performance evaluation, on the other hand may be approximate but enable quick evaluation and offer insight into the system behavior. Analytical models can help in examining a larger set of alternatives initially, thereby reducing the number of candidates for detailed simulation analysis. A useful tool in the design pro-

cess or in the evaluation of current warehousing systems would be an integrated model that can capture the interactions among material handling and storage processes that span receiving, inspection, storage/putaway, picking, shipping and value-added services.

Isolated analysis of warehouse sub-systems, though valuable and important, is not sufficient at the overall system design stage. Rouwenhorst et al. (2000) point out that the design decisions at the strategic and tactical levels are interrelated. For example, a decision to have separate forward and reserve areas (strategic) leads to an inventory replenishment policy between the storage areas (tactical). Also, the inventory decisions (size of the warehouse) are made independently from the individual sub-systems such as AS/RS. But the performance of the sub-systems is affected by the storage size. Such interrelated decisions need to be modeled jointly. To support decision making for the new generations of warehouses, there is a need to include a larger set of issues such as inventory and capacity/congestion, within a single analytical model, so that their impact on the total system performance can be evaluated.

1.4.1 The Problem Statement

With the changing role of warehouses, the ability to model multiple decisions simultaneously, especially inventory and throughput decisions, will complement the warehouse design process and aid in analyzing existing operations. Queues and queueing networks have been applied in the performance evaluation of warehouses, but the focus has been more on isolated systems like AS/RS and its related decisions like throughput and storage size estimation, separately.

Production-inventory networks, i.e., queueing network models that address both capacity/congestion and planned inventory issues have been successfully applied in manufacturing and supply chain systems. But very limited literature is available on their application in the warehouse domain. This dissertation is the first step towards addressing this gap. Hence, the problem statement can be described as "developing analytical models of queueing-inventory systems that address capacity/congestion and inventory issues simultaneously in the context of a warehouse system."

To this end, the dissertation first focuses on the development of performance evaluation

models of sub-components that are representative of AS/RS type systems (e.g., a server that stores and retrieves material from the same storage area) and order-picking systems where unit-load configurations vary between two successive material movements. These models address both material handling and material storage operations in a manner similar to the production-inventory models. The dissertation also demonstrates the applicability of these individual models in building comprehensive end-to-end warehouse performance evaluation models.

1.5 Overview of the Document

In this chapter, we introduced warehousing systems in general, and commented on the current status of performance evaluation of such systems. We described some of the characteristics of warehousing systems and provided the motivation for the research effort. Chapter 2 gives an overview of the warehouse performance evaluation models and a review of the literature on production-inventory networks. The research objectives, general assumptions and research limitations are summarized in Chapter 3. In Chapter 4, we focus on the development and analysis of the shared-server system, a key building block in the performance evaluation of warehouses. We extend the shared-server model to the multi-server case in Chapter 5. We then focus on the development and analysis of models that accommodate changing unit-load configuration (single and multi-server cases) in Chapter 6. Chapter 7 illustrates the development of a proof-of-concept end-to-end model for warehouses. Finally, we conclude the dissertation with a prospectus for future research with respect to warehouse performance evaluation models.

Chapter 2

Literature Review

In this chapter, we focus on the review of literature pertinent to warehouse performance evaluation and production-inventory models. The analytical models in this review tend to focus more on the application of queueing or queueing network models.

2.1 Review of Warehouse Performance Evaluation Models

The major sources of randomness in a warehouse are the demand for the items to be retrieved from storage, the arrival of the items to be stored, the material handling times and the inherent reliability of the servers (human and machine). Sophisticated simulation models that address some of these sources of randomness have been developed and can evaluate the performance of different configurations of warehouses (see for example, Linn & Wysk (1984), Berg & Gademann (2000) among others).

The analytical performance evaluation models of warehouses can be classified as throughput capacity models, storage capacity models, and warehouse design models (Cormier & Gunn (1992)). Throughput capacity models are mostly tactical and operational models that evaluate the throughput of the warehouse, where throughput is defined as the number of storage/retrieval operations per unit time. The storage capacity models, which are usually strategic models, focus on the determination of the size of the warehouse to satisfy a minimum service level commitment. The warehouse design models focus on the overall design involving decisions related to space allocation among storage systems, cross-docking,

and other value-added services.

2.1.1 Throughput Capacity Models

Throughput capacity issues have received considerable attention in the warehouse literature, mainly because the order-picking costs constitute a major portion of the total operational costs (Tompkins et al. (2003)). In this section, we will summarize the modeling approach and the decisions considered in the throughput models for two important sub-systems, namely Automated Storage and Retrieval Systems (AS/RS) and Order Accumulation and Sortation Systems (OASS).

AS/RS performance evaluation models have focused on the development of travel-time models for both unit-load AS/RS and miniload AS/RS. For a detailed literature survey on stochastic modeling of AS/RS and travel time models, the reader is referred to Johnson and Brandeau (1996) and Sarker & Babu (1995) respectively.

Lee (1997) presented the first stochastic analysis of a unit-load AS/RS by using a single-server queueing model. He assumed aisle captive S/R machines and modeled each aisle as a single server with two queues: a storage queue for incoming unit loads and a retrieval queue. Both the queues have finite capacity. The storage and retrieval arrivals are lost when the queues are full. The S/R machine always returned to the I/O point and the FIFO policy was followed for the queues, except when both the queues had transactions waiting. In the later case, a dual command cycle is performed. The proposed model could be viewed in part as an assembly-like queue and in part as a polling queue. Lee (1997) further assumed independent Poisson arrivals for storage and retrieval queues, and exponential service times for single and dual command cycles. Using a Continuous Time Markov Chain (CTMC) to represent the queue, Lee derived many useful performance measures including system throughput, turn around time for the requests and S/R machine utilization. The limited queue capacity and high variance assumption in the previous model underestimated the throughput and the S/R machine utilization. Lee (1997) had also assumed equal arrival rates for storage and retrieval.

Hur et al. (2004) relaxed some these assumptions and modeled the S/R machine as a M/G/1 queuing system with separate queues for storage and retrieval requests. There was

no capacity limit on the queues and the arrivals were independent with different arrival rates. They assumed that the S/R machine could start and end the single command (SC) and dual command (DC) cycles at the I/O point or at the rack. Because of this assumption, they also assumed that the travel time for the SC and DC followed the same distribution with a single service rate. They also proposed a state space for the S/R machine by defining the state as (i, j) where i, j are the number of requests in the storage and retrieval queues, respectively, after a service completion. The resulting CTMC was solved to derive system performance measures. They compared their solution with that of Lee (1997) and found it to outperform Lee's in many instances.

Bozer & Cho (2005) assumed separate travel times for SC and DC cycles. They assumed the dwell point as the last known storage location of the S/R machine. The S/R machine always tried to perform a DC violating the FIFO policy for storage or retrieval request arrivals. They still assumed independent Poisson arrivals. For random storage assignment and different configurations of the rack, they derived closed form equations to determine whether the AS/RS meets the required throughput. They compared the S/R machine utilization for balanced and unbalanced systems (when storage requests exceeded retrieval requests or vice versa, which is possible in a warehouse) with simulation. Their results are also valid for other storage assignment policies and I/O point locations as long as the mean interleaving time (time between drop-off and pick-up) in a DC cycle is smaller than mean SC cycle time.

Hur & Nam (2006) extended the models of Hur et al. (2004) by considering separate service times for single and dual command cycles for the S/R machine with Poisson arrivals for service requests. They assumed finite queue capacity only for storage requests. The state space of the system is defined as the number of requests in the queues at the completion of a service request or the start of a busy period, similar to the earlier model. A Semi-Markov Process (SMP) is generated from the Markov Chain to obtain the time-average probabilities, which is later used to obtain the system performance measures, including the probability that an arbitrary arrival is lost. All the above models for AS/RS performance evaluation considered unit-load systems with equal sized storage spaces.

Lee et al. (1999) presented models for AS/RS with unequal sized cells, i.e. cells within

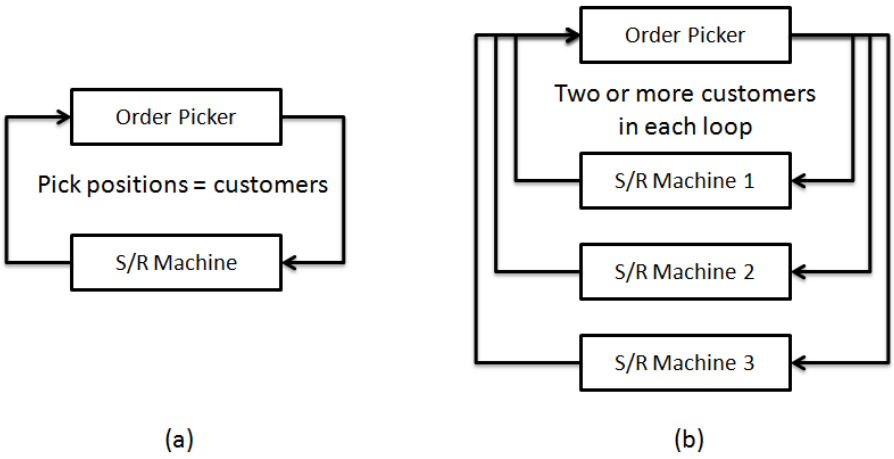


Figure 2.1: End-of-Aisle System with (a) dedicated and (b) multiple aisles per picker

a zone have the same size but differ in size between zones. They derived travel time models for SC and DC cycles, including interleaving time between different zones.

An example of end-of-aisle order picking systems is a miniload AS/RS. In such systems, stored material is delivered by the S/R machine to the order picker located at the end of the aisle. While the order picker picks from the storage container, the S/R machine returns the previous container and retrieves the next order. These systems operate predominantly DC cycles. There are at least two pick positions at the end of each aisle. Bozer & White (1990) presented a design algorithm for such end-of-aisle order picking systems. They modeled each aisle as a closed queueing network with two nodes, the S/R machine and the order picker, as shown in Figure 2.1(a). The number of pick locations was the number of customers in the system. They assumed random storage policy with dedicated pickers for each aisle. They presented an iterative algorithm to find the minimum number of aisles necessary to meet the storage and throughput requirements. The throughput constraints were based on the picker utilization and the S/R machine utilization was only an additional measurement. Using simulation, they obtained an approximation for the standard deviation of the DC cycle travel time and approximated the DC cycle time with a uniform distribution. Their model confirmed that as the rack became more non-square-in-time, the number of aisles increased to meet the required throughput.

Bozer & White (1996) extended their work to model multiple pick positions per aisle.

They relaxed their assumptions about aisle-restricted order-picker. The more general closed queueing network model is shown in Figure 2.1(b). Using diffusion approximations, they derived the expected utilization of the S/R machine and the order picker. They modified their design algorithm to include the expressions for utilization. They further experimented with sequencing of the retrieval requests. Only when the variance of pick time is low, significant improvements to throughput were achieved.

Park (1999) used a similar closed queueing network model to study the impact of buffer sizes (number of pick locations per aisle for storage and retrieval queues) on the throughput of the system. They found that the maximum throughput obtainable by increasing the queue capacity is less than or equal to twice the throughput with a single space for storage and retrieval. They also analyzed the conditions under which the S/R machine can be blocked ("production blocking") because of limited queue capacity. The literature on performance evaluation of AS/RS also considers operational details such as dwell point and order batching, and operating characteristics such as acceleration and deceleration of the S/R machines.

Order Accumulation and Sortation Systems (OASS) find applications in both manufacturing and warehousing systems. The basic components in an OASS are the input conveyors; induction, spacing, and merge units; sortation mainline; and diverter modules. An example OASS is given in Figure 2.2 with one induction point, a re-circulating conveyor, and multiple accumulation lanes (Johnson and Meller (2002)). In distribution centers, when orders are picked at the case and item level, orders are dispatched to the conveyor that sorts the items to different chutes assigned to a particular order or outbound truck.

Throughput of the OASS is an important performance measure and it depends on many factors including the speed of the conveyors, induction process, sorting strategies, the upstream order picking process and downstream stacking/palletizing process amongst others. For wave picking (each wave consists of many orders and each order consists of many line items) and re-circulating sortation conveyors, Johnson (1998) developed analytical models that study the impact of sorting strategies on the throughput of the system. He developed expressions for the expected time to sort a wave with and without blocking at the accumulation conveyors for Fixed Priority Rule (FPR) (smallest order first and largest order first)

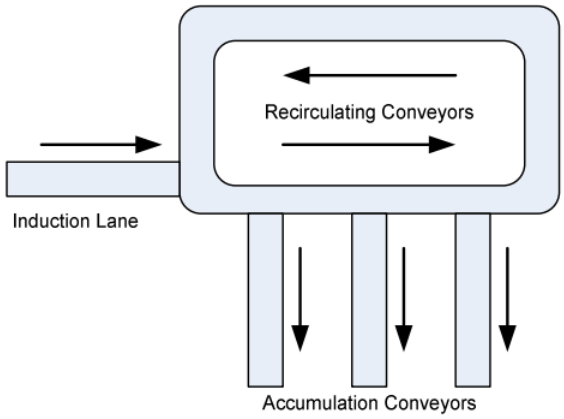


Figure 2.2: An example Order Accumulation and Sortation System (Johnson and Meller, 2002)

and Next Available Rules (NAR) for sorting. In the NAR, the orders are sorted based on the order in which the boxes pass through the scanner that initiates the sorting process. Eldemir (2003) developed another strategy based on the earliest completion time of an order, which reduced the wave sortation time.

Johnson and Meller (2002) developed analytical models of an OASS with multiple induction points in the main conveyor. They assumed no recirculation of orders. When there is no blocking at the accumulation conveyors and the number of orders is less than the number of accumulation conveyors, the OASS performance is dependent only on the induction process. The authors analyze systems with side-by-side induction points and split induction points. In the side-by-side induction systems, the inductors compete for the same scanner thereby creating interference like phenomenon that tends to reduce the system throughput. The authors also address the impact of presorting orders on the OASS system performance.

Eldemir (2003) proposed an open queueing network model for the design of OASS. The network consists of three processes (induction, sortation, and shipping) with corresponding queues (induction lane, main conveyor, and accumulation line). He derived expressions for the blocking probability and the lengths of the main sortation and accumulation conveyors by approximating each queue as an M/G/n/N queue. Russell & Meller (2003) developed descriptive and prescriptive design models for order fulfillment systems that are integrated order picking and order-sortation systems. They developed throughput simulation models

to compare the different configurations of wave picking and manual and automated OASS systems. Apart from these studies on OASS, there is a huge body of literature on conveyor theory, see for example, Bastani (1988); Arantes et al. (1998) and Bozer & Hsieh (2005) amongst others.

2.1.2 Storage Capacity Models

The purpose of storage capacity models is to determine the number of warehouses, the size of each warehouse and any additional space that could be leased by minimizing the total discounted costs and/or to achieve predetermined service level.

Cormier & Gunn (1992) categorized the models as static and dynamic models. In static models, the demand is assumed stationary and a warehouse size is determined. In dynamic models, the demand is assumed to be non-stationary and the warehouse is allowed to expand and contract i.e. size of the warehouse at different periods is determined. The authors also review models related to performance evaluation and maximization of space utilization through unitization & block stacking methods.

Roll & Rosenblatt (1983) compared the effect of random and grouped storage policies on the warehouse capacity. In general, the random storage policy offers higher space utilization (assumption that the demand for each pallet is independent and all storage spaces are equally likely to be occupied) compared to grouped storage policy. In addition, the authors clearly noted that grouped storage policies offered operational and administrative advantages over the random storage policy. The authors defined Nominal Capacity Requirements (NCR) as the product of average throughput and the average storage time for each pallet. It is the lower bound on the required warehouse capacity and is the average size necessary subject to random throughput factors. Because of the stochastic nature of the arrivals of order and number of items per order, the warehouse may not be able to store the entire shipment and has to lease space outside. The effects of number of items and their characteristics, and operational issues (travel distances, order-picking policies) were not considered.

Rosenblatt & Roll (1988) later studied the factors that influence the storage capacity of the warehouse; a) number of items stored, b) demand characteristics of the items

(distribution of orders and items in an order) c) replenishment policy (order quantity and order point) for the item. They performed simulation experiments assuming a (r, Q) replenishment policy and random storage policy. They derived an approximate multiplicative expression using regression analysis for the deviation from the NCR for 95% service level as follows:

$$Y_{95} = 34 \frac{Q^{0.16} D_1^{0.22}}{N^{0.62} r^{0.06} D_2^{0.02}} \tag{2.1}$$

where, Q is the order quantity, r is the reorder point, D_1 is the average demand (orders/day), D_2 is the percentage deviation from D_1, and N is the number of items. We can see that reorder point and the variance of the demand have very little effect on capacity. They tested the demand for different distributions (uniform, normal and exponential) and found that the maximum deviation was around 10% of the NCR capacity. They assumed the same inventory policy for all items and that the items have similar physical and economical characteristics. They also claimed that changes in the above have little effect on the storage capacity. Roll et al. (1989) found a suitable size of a warehouse container and used simulation to find the optimal combination of warehouse capacity and container size.

Sung & Han (1992) extended the queuing model of Schwarz et al. (1978) to determine the size of AS/RS for single and multiple item storage scenarios. In the case of single item storage, the AS/RS is treated as an M/M/m/m or M/G/m/m model. For multiple item storage, a single class closed queueing network model is developed to determine the storage size. Both the models were extended to include blocking (when an arriving item does not find a storage space in the rack) and batch arrivals. The important assumption is that the items spend a known but random amount of time at the racks.

Cormier & Gunn (1992) formulated the warehouse sizing problem as a cost minimization problem considering inventory policy costs, warehouse construction/operating costs and the cost of leasing for constant product demand (static conditions) for a single period, assuming a continuous review policy without the possibility of backorders.

Rao & Rao (1998) presented a modified formulation for warehouse sizing under static and dynamic conditions. They provided three extensions to the static conditions involving

varying cost over time, economies of scale in capital expenditure and/or operating cost and stochastic version. The dynamic version of the problem with stochastic demand was shown to be a network flow problem and its concave cost version could be solved efficiently using dynamic programming methods.

Huang et al. (2003) simultaneously selected distribution centers and their capacities by solving a 2-stage distribution network. They modeled the distribution center as an M/G/c queue where each storage space represents a server. They consider the case of discrete and continuous racks. They also studied the difference between a stepwise approach (site selection and space determination) and the integrated approach.

2.1.3 Integrated Design Methods

The warehouse design procedure is a complex process because the number of available design alternatives is large and hence, the choice of a particular design depends on the experience of the designers. Ashayeri & Goetschalckx (1988) presented a systematic planning and designing procedure for order-picking systems. Their stepwise design procedure consists of nine steps starting with external strategic planning considering market information to selecting operational policies for the order picking system.

Gray et al. (1992) proposed a multi-stage hierarchical decision approach. The approach consists of three levels; facility design and technology selection, item allocation, and operating policy decisions. Each level has a set of mathematical models that evaluates the major trade-offs to obtain a set of feasible design alternatives. The authors suggest the use of simulation to fine tune the design and operating policies. They applied the methodology successfully to design a spare parts distribution center.

Yoon & Sharp (1995, 1996) present a cognitive design procedure for an order picking system (OPS). They present a general framework for the OPS design and analysis that consists of a general structure of the OPS and a conceptual design procedure. The general structure illustrates all the functional areas and material flows (pallets, cases, and items) in an OPS. The conceptual design procedure consists of input, selection and evaluation stages. An alternative design methodology was proposed by McGinnis et al. (2000) based on a functional flow network. The activities are represented as nodes and flows are represented

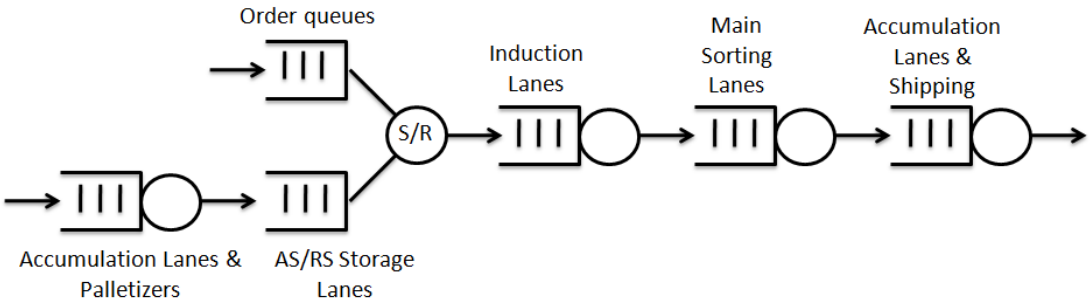


Figure 2.3: A simple high-level queueing network model of a warehouse

as arcs. Once a flow network for a particular configuration of the warehouse is established, the functions are assigned to spaces in the warehouse. Goetschalckx et al. (2002) applied this methodology for a small parts warehousing system.

Bodner et al. (2002) developed a process model to assist in the development of computational tools for warehouse design.

Many researchers have used simulation to analyze tactical and operational decisions simultaneously. Petersen & Aase (2004) compared picking, storage and routing policies on warehouse throughput. Manzini et al. (2005) used simulation to develop an expert system by performing a comprehensive set of designed experiments. Berg & Gademann (2000) compared control policies like storage location assignment and sequencing together using simulation.

Eldemir (2003) suggested two approaches for modeling an integrated system based on queuing network and material flow diagrams. Material flow diagrams are graphical representations of the movement of materials used in a process. They are a useful aid in identifying the source, stages and sink including the quantities and losses at each stage/process. Using a set of standard procedures at the process and routing probabilities after the process, system throughput, and subsystem throughput can be calculated. This approach cannot capture the stochastic nature of the processes in the system. The second approach is modeling the system as a network of queues. The author had modeled individual systems (palletizer, AS/RS, and sortation) with buffer capacities. Eldemir (2003) suggested the use of Jackson network type models or Queueing Network Analyzer (QNA) based models proposed by Whitt (1983). An example of such a system model is presented in Figure 2.3.

Jackson network assumes exponential service times and Poisson arrivals. In addition, the extensions for multiple classes assume that the service time is the same for all the classes, which severely restricts the model use. QNA provides a more flexible framework for modeling such a system.

Our approach is based on the parametric-decomposition approach presented by Whitt (1983). In the queueing network approach suggested by Eldemir (2003), the storage rack configuration is modeled implicitly in the service time of the S/R machines operating in a single or dual command mode. Other authors who explicitly consider the rack and the storage process, assume that each rack space/bay as a server (Huang et al. (2003)) or the rack as a queue space. The disadvantage of the first approach is that the service time of the racks is not known or can only be assumed. Some retail warehouses are very large with thousands of bays effectively modifying the rack to be an infinite server. In the case where rack is treated as a queue space, the potentially large queue space will tend to behave like an infinite queue.

2.2 Review of Production-Inventory Models

General queueing network models are readily applicable for make-to-order systems, where the only inventory is the work-in-process due to the parts waiting to be processed. In make-to-stock systems, finished goods and intermediate items are produced and stored in anticipation of customer demand. The demand is satisfied immediately reducing the overall waiting time of the customer. Such holding of both finished goods and intermediate parts in anticipation of demand is called planned inventory. In addition to providing better customer service, the planned inventories act as a buffer against uncertainties like machine failure. Some of the relevant literature in production-inventory networks includes Buzacott & Shantikumar (1993), Lee and Zipkin (1992), Sivaramakrishnan & Kamath (1997), Sivaramakrishnan (1998) and Zipkin (1995). A review of the relevant work is presented in the following paragraphs.

Consider an M-stage production-inventory model as shown in Figure 2.4. Each stage is represented by a queue-server-output store. Each stage operates under a base-stock

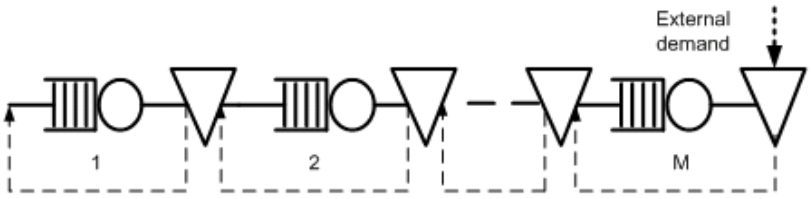


Figure 2.4: M-stage Production-Inventory model

policy. The base stock level is the maximum planned inventory at the output a stage. The demand process, occurring at the M^{th} stage is assumed to be a renewal process and for one unit at any given time. If a finished item is available, the order is fulfilled immediately and a replenishment order is placed at stage $M-1$. Such a policy is called a one-for-one replenishment policy. If the item is not available, the demand is backordered. The inventory is replenished until the base stock level is reached.

The replenishment order at stage $M-1$ looks at the output store of that stage. If a part is available, it immediately moves to the processing queue of stage M and an upstream replenishment order is placed else it is backordered. In the first stage, orders join the processing queue immediately. Unlimited supply of raw material is assumed at stage 1. In all the stages, a backorder is fulfilled first before the planned inventory is replenished. In the one-for-one replenishment policy, the demand arrivals are reflected at all the stages of the network. Hybrids of make-to-order and make-to-stock systems can be analyzed by constraining some of the base-stock levels to be zero.

Lee and Zipkin (1992) modeled such a tandem production line with Poisson arrivals and exponential service times. They modeled each stage as an M/M/1 queue and applied the approximations of Svoronos & Zipkin (1991) for a multi-echelon inventory system. Zipkin (1995) extended these results to tandem queues with feedback.

Single stage make-to-stock systems were studied extensively by Buzacott & Shantikumar (1993). They modeled such systems using the Production-Authorization (PA) card concept. When each item in the manufacturing facility is produced, a tag is associated with the item. When the demand consumes an item in the output store, the tag is removed and is converted into a Production-Authorization card. They vary the rules governing the transmission of

PA cards to authorize production. The variations are a) immediate transmittal of PA cards into the facility as soon as it is generated, b) fixed batch of PA cards, say q, when at least q PA cards are accumulated, and c) all the PA cards when at least q PA cards are accumulated. These variations represent the one-for-one replenishment base stock policy, reorder point/order quantity and reorder point/order up to inventory policies, respectively. Buzacott & Shantikumar (1993) modeled single stage systems with unit demand, bulk demand, and interruptible demand, backlogging and lost sales, yield loses and multiple classes of customers.

Sivaramakrishnan & Kamath (1997) analyzed multi-stage tandem make-to-stock systems using a node decomposition approach. Each stage in the model had a delay node that captured the effects of backorder delay. A processed item at each stage would satisfy any backorders in that stage or replenish the inventory. The output store in each stage was controlled by a base stock policy with one-for-one replenishment. Using the parametric-decomposition approach (Whitt, 1983), Sivaramakrishnan (1998) extended the results to include general arrivals and service times, multiple servers, batch service, limited raw material supply, multiple classes, service interruptions and feedback. Feed forward networks were also considered.

Liu et al. (2004) modeled a similar tandem network with a base-stock policy and one-for-one replenishment. The difference with the Liu et al. (2004) is in modeling the departure process from the output store of a stage. In each stage, the input buffer consists of two queues; material queue (orders for which material was available at the output store of the previous stage) and backorder queue (orders for which material was not available at the output store). In general, some of the performance measures considered were expected inventory levels at the finished goods stores, average work-in-process, fill rate and average number of backorders.

Dong & Chen (2005) developed an approximate model for a (q, S) inventory policy for a single stage system. They adopted the target level PA cards mechanism with fixed lot size model in Buzacott & Shantikumar (1993). They used the $GI^X/G/1$ model, where X is the fixed batch size q. They transform the bulk arrival queue into an equivalent $GI/G/1$ queue by modifying the service time to obtain performance measures similar to Buzacott

& Shantikumar (1993).

Srivathsan (2005) developed production-inventory models of supply chain networks. He developed models for convergent (two suppliers, supplying a manufacturer) and divergent (one manufacturer supplying two retailers) aspects of a supply chain. He modeled the lead time/transit time using a delay node and analyzed a larger network with multiple suppliers, manufacturers and retailers.

2.3 Summary

In this chapter, we have provided a detailed literature review of the status of performance evaluation models for warehouses and general production-inventory systems. In this context, we note that

- Majority of performance evaluation models for warehouses are specific and analyze specific warehouse systems in isolation. A majority of the studies focused on aisle-based automated warehouses, while literature on carousel systems and Autonomous Vehicle Storage Retrieval Systems (AVS/RS) are emerging (Fukunari, 2003). One of the main assumptions in the performance evaluation of these systems is that there is always space available for the waiting storage requests and similarly, a customer demand can always be satisfied. Hence, the performance measures and improvements focus more on improving the material handling aspects of the storage and retrieval systems.

- New approaches for systems design and evaluation have started emerging such as those based on process modeling techniques but lack the analytical evaluation capability. The current systematic procedures for warehouse design are mostly descriptive with some prescriptive steps where specific optimization models can be applied for evaluating economic trade-offs.

- In the limited applications of queueing or queueing network models to warehouse performance evaluation, only the congestion effects in the warehouse storage systems have been analyzed. In addition, many models assume that the service provided by

the storage system is known and can be approximated by the exponential distribution, which in fact may not be realistic.

- In general, the models developed do not provide an analysis framework to include value-added services in the warehouse.

In the next chapter, we will discuss the research goals and objectives and the contribution made by this dissertation.

Chapter 3

Statement of Research

The overall goal of this research was to develop analytical models for warehouse performance evaluation that can simultaneously deal with inventory and capacity/congestion issues. In a warehouse system, the primary storage function of the warehouse and the inbound/outbound configuration of the unit-loads give rise to two important configurations; the shared-server system and the order-picking system. The first two objectives in this dissertation can be thought of as focusing on queueing-inventory models of these two configurations which are seen as key building blocks of a warehouse system. The final research objective focuses on building a proof-of-concept end-to-end warehouse system model using these building blocks.

3.1 Research Objectives

Research objectives 1 and 2 focus on the development of the shared-server and order-picking system respectively while research objective 3 focuses on the development of end-to-end models.

Objective 1: To develop and investigate the accuracy of an approximate analytical model of the shared-server system i.e., an inventory store with a server performing both storage and retrieval operations (hence the name, shared-server). The storage operation increases the inventory level and the retrieval operation decreases the inventory level. The analytical model explicitly considers the presence or absence of items in the inventory store

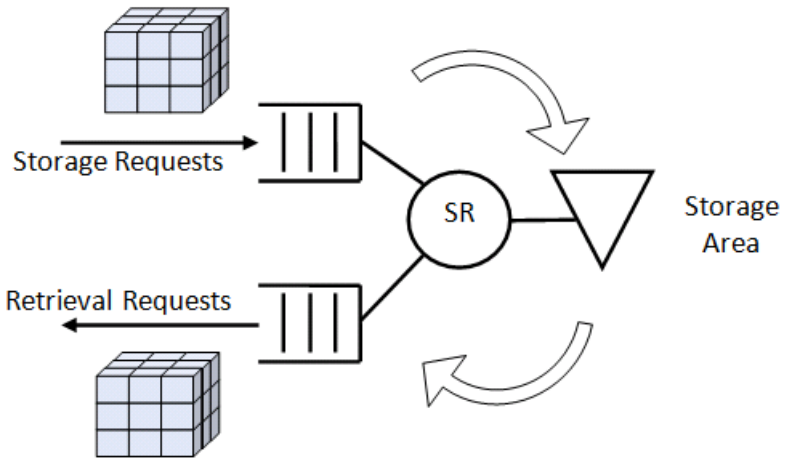


Figure 3.1: A shared-server system

and its size.

In this objective, we study the system independent of the rest of the operations in the warehouse and we make the following assumptions.

- We assume independent arrivals for the storage/retrieval requests.
- The configuration of the unit-load is maintained during storage/retrieval operation and the system operates under FCFS discipline.
- The server operates in a single-command mode and the storage/retrieval operations have identical service time distributions.
- The storage request or retrieval request is for a single item.

A typical configuration of such a system is illustrated in Figure 3.1.

Sub-objective 1.1: The shared-server is studied under Markovian assumptions – Poisson arrivals for storage/retrieval requests and exponential service times for the S/R machine.

Sub-objective 1.2: The Markovian assumption is relaxed and the shared-server system is modeled under general arrival and service time distributions.

Sub-objective 1.3: This objective extends the general model to account for parallel aisles in the storage system with dedicated S/R servers.

Warehouses that deal with different unit-load configurations (pallets and cases) will have separate storage areas allocated to a particular configuration; reserve storage for pallets

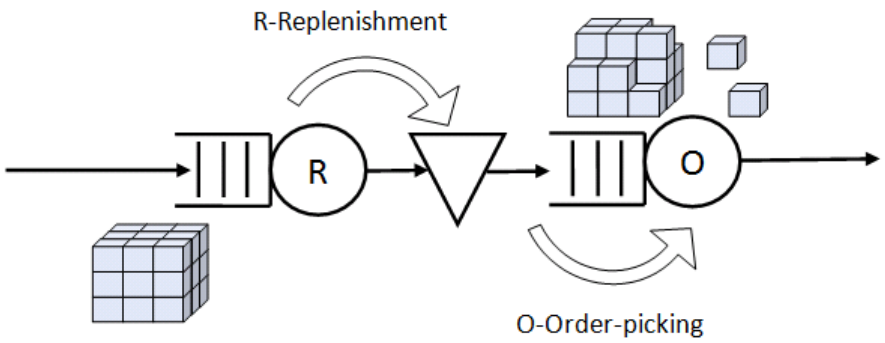


Figure 3.2: An order-picking system

and forward storage for cases, in general. Whenever the inventory in the forward area is depleted because of order-picking, an internal replenishment occurs from the reserve area. In objective 2, our focus is on this forward storage area, where the replenishment orders are in pallet loads and order-picking is in case loads.

Objective 2: To develop and analyze an approximate analytical model for an order-picking system; a single storage area with a server picking in less-than-unit-load quantities (cases) and a separate server replenishing the inventory in unit-loads (pallets). A tandem model representing such an order-picking operation is illustrated in Figure 3.2. We assume unit order-picking quantity and unit replenishment–order quantity. We also assume that the order-picker and replenishment server have unit capacity.

Sub-objective 2.1: The order-picking system is studied under general arrivals and general service time distributions for the single server case.

Sub-objective 2.2: The model is extended to include multi-server cases.

Objective 3: To demonstrate the applicability of the models developed in objectives 1 and 2 as building blocks to develop comprehensive end-to-end models of the warehouse system. The proof-of-concept system includes a reserve storage area, a forward storage area and a downstream shipping operation. The reserve storage area is modeled using the shared-server system and the forward storage area is modeled using the order-picking system. Hence, this objective demonstrates the applicability of models developed in previous objectives in building end-to-end warehouse models.

3.2 Research Scope and Limitations

The scope of this research was limited by the following assumptions.

- The analytical models developed are for a single class of customers and the customer demand for storage and retrieval are for unit order-quantity. Also, the servers are assumed to be reliable.

- Design characteristics of the storage area such as warehouse layout, zoning (assigning workers to particular sets of aisles), slotting (assigning products to individual bays), and operational characteristics such as order scheduling and order sequencing are not modeled. Orders are mostly satisfied on a FCFS basis. The storage rack is treated as a single inventory location for modeling purposes.

- This framework does not model the inventory staggering decisions that have proven to reduce the maximum inventory levels in the warehouse (Hariga & Jackson, 1996). We assume that the capacity of the inventory store is given as the inventory sizing decisions are now made during the design of distribution network itself, which is outside the scope of this research.

- Travel time models for the storage systems are abundant and they consider the physical characteristics of the storage racks (square-in-time and non-square-in-time) and storage assignment policies. We do not consider such decisions and policies explicitly in this study.

In the next chapter, we focus on the development of the shared-server system in detail.

Chapter 4

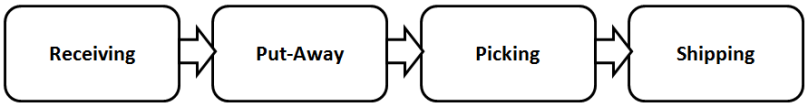
Shared-Server System

In this chapter, we focus on the development of an analytical model of a shared-server system, a key building block for developing end-to-end performance evaluation models of a warehouse. We described the activities and the material flow within a typical warehouse in chapter 1. Here, we focus on the shared-server system, describe our modeling assumptions, and develop an approximate analytical model of the shared-server. We perform a detailed evaluation of the shared-server model by comparing its results with equivalent simulation results. We conclude the chapter by discussing how the shared-server model can be part of an end-to-end warehouse model.

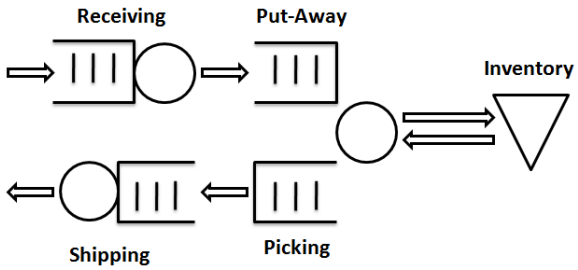
4.1 Shared-Server System Development

A queueing-inventory model of a warehouse illustrating a subset of warehouse operations with respect to a single storage area is illustrated in Figure 4.1. From the perspective of process flow, the warehouse operations follow a sequential flow. But from a resource centric view, the queueing-inventory model of the same is not necessarily tandem.

The distinction comes from the fact that resources are shared between activities/operations. In a traditional production-inventory model of manufacturing system, such as the one shown in Figure 4.2(a), each resource/processing unit has its own output store. When a demand consumes an inventory at the output store of stage 2, it triggers a replenishment order immediately. This order then looks for a part at the output store of stage 1, and if available



(a) Process-centric view of warehouse operations



(b) Resource-centric view of warehouse operations

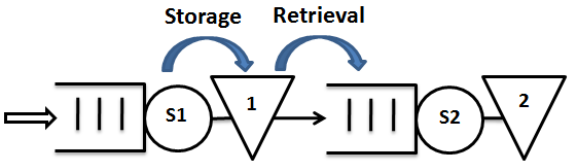
Figure 4.1: Process and resource centric views of warehouse operations

goes and waits for processing at stage 2. Parts after processing move to the output store. This process is repeated at each stage. Hence, at each inventory store, parts are put into the store by an upstream machine and parts are retrieved for processing by a downstream machine.

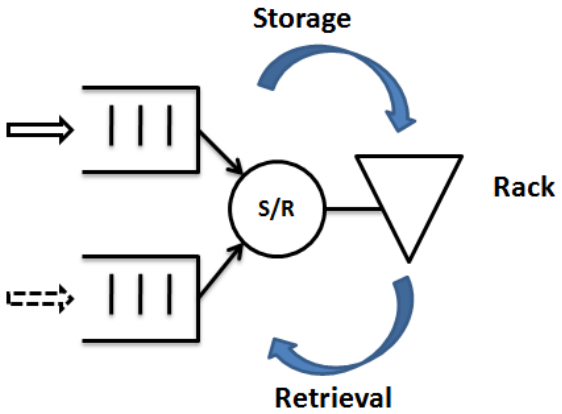
In warehouses, material handling resources such as cranes and S/R machines that are assigned to a particular storage area perform both storage and retrieval operations. Hence, the pallets that need to be stored (similar to upstream operation) and the orders that need to be retrieved (similar to downstream operation) share the same resources. We call this resource/server that is shared between storage and retrieval operations as the Shared-Server. This chapter of the dissertation focuses on the performance evaluation of the shared-server system for the single server case and how it can be used as a building block in a comprehensive end-to-end model of the warehouse.

4.1.1 Description of the Shared-Server System

A typical AS/RS consists of multiple parallel aisles, one or more S/R machines that can travel simultaneously in horizontal and vertical directions, and an input/output station. The S/R machine can operate in a single command mode (either storage or retrieval operation in a single cycle) or in a dual command mode (a storage operation and a retrieval operation in the same cycle). There is a buffer in front of the AS/RS where the requests wait in



(a) PI model of a tandem manufacturing system



(b) PI model of a warehouse rack storage

Figure 4.2: Production-Inventory (PI) models of (a) manufacturing system (b) warehouse rack storage

a queue to be serviced. It is a physical queue in case of storage requests and a (virtual) information queue, in case of retrieval requests. A shared-server is representative of AS/RS type of systems. The model consists of a server (S/R machine), separate queues for storage and retrieval requests, and physical inventory store (rack). The following assumptions are made about the shared-server system.

4.1.2 Assumptions

- Storage and retrieval requests arrive independently of each other and join separate queues. The storage requests are say, pallets waiting to be stored and hence, the storage queue has a physical limit because of limited warehouse space. The retrieval requests are information, and hence, the maximum backlog is limited by design decisions. In this dissertation, we assume that both the physical queue and the information queue are finite and are equal in capacity. Similarly, the rack has a finite capacity. Requests that arrive when the storage (request) queue is full will be lost.

- The shared server is assumed to operate in a single command mode and follow a first

come first served service discipline as long as the request can be serviced. Because of the limited capacity of the rack, the FCFS discipline may be violated. When a storage request arrives before a retrieval request and there is no space in the rack, the request is blocked (storage blocking). The blockage is resolved when the retrieval request is serviced before the storage request. Similar blockage occurs when a retrieval request arrives before a storage request, and there is no item to retrieve (retrieval blocking).

- Each request (storage or retrieval) is for a unit-load item only and the server can handle only one unit-load at a time.

The following notation is used in this chapter.

λ_S^{-1}, C_S^2 - mean and SCV of the inter-arrival times of storage requests

λ_R^{-1}, C_R^2 - mean and SCV of the inter-arrival times of retreival requests

μ_{SC}^{-1}, C_{SC}^2 - mean and SCV of the service times

B_S, B_R - Queue capacities for storage and retrieval requests respectively

Z - Rack size

$L_Q(S), L_Q(R)$ - mean queue length of storage and retrieval requests respectively

$L(RACK)$ - average inventory level in the rack

$\lambda_{dR}^{-1}, C_{dR}^2$ - mean and SCV of the inter-departure times of retrieval requests

The squared coefficient of variation (SCV) of a random variable (rv) is defined as the variance of the rv divided by the square of its mean.

We believe that this dissertation effort is the first analytical model of the shared server system where the inventory store or the rack size is explicitly modeled. As is customary with any new performance modeling research, we first model the shared-server system under the Markovian assumption.

4.2 CTMC Model and Analysis

Modeling queues using Continuous Time Markov Chains (CTMC) is a widely used performance evaluation technique because it provides us with an exact method of analysis under exponential assumptions. Algorithms exists to solve the CTMC model and to compute

performance measures that can be used to understand system behavior. In our case, we also use the CTMC model to verify the simulation model that is used to evaluate the more general model of the shared-server system which is the subject of section 4.3.

We assume that the single command service time follows an exponential distribution with the service rate ($\mu_S = \mu_R$) for storage and retrieval requests. The arrivals of storage and retrieval requests are independent of each other and are Poisson processes with mean arrival rates of λ_S and λ_R, respectively. We also assume limits, namely, B_S and B_R on the capacities of storage and retrieval request queues, respectively. Storage and retrieval arrivals are lost when the queues are full.

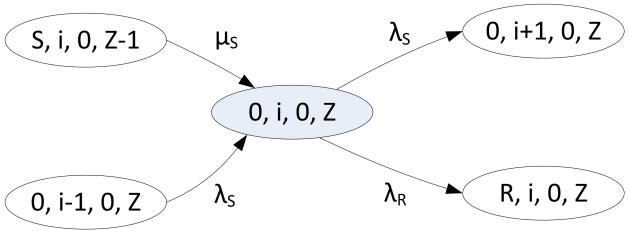
The state of the system at time t is then defined as,

$$X(t) = \{m, i, j, k\}$$

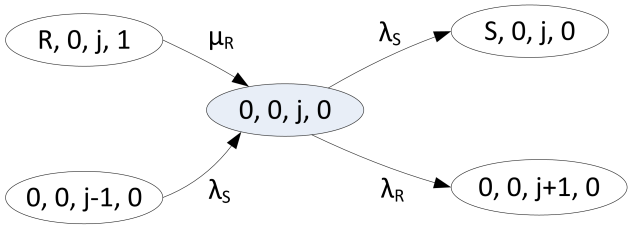
where m represents the current mode of the server (0 idle, S serving a storage request, R serving a retrieval request); i, j, k are non-negative integers representing the number of storage requests waiting in queue, number of retrieval requests waiting in queue and the inventory level in the rack, respectively; $0 \leq i \leq B_S$, $0 \leq j \leq B_R$ and $0 \leq k \leq Z$.

The server becomes idle when both the request queues are empty ($i = 0$, $j = 0$). The server is blocked when $i = 0$, $j > 0$ and $k = 0$ (retrieval blocking) and when $i > 0$, $j = 0$ and $k = Z$ (storage blocking). Figure 4.3 provides examples of system states together with their transitions. In the CTMC model, when the server is capable of servicing either request, as in Figure 4.3(c) we assumed that with a probability p_S ($= \lambda_S/(\lambda_S + \lambda_R)$), the server satisfies a storage request and with a probability p_R ($= 1 - p_S$), the server satisfies a retrieval request.

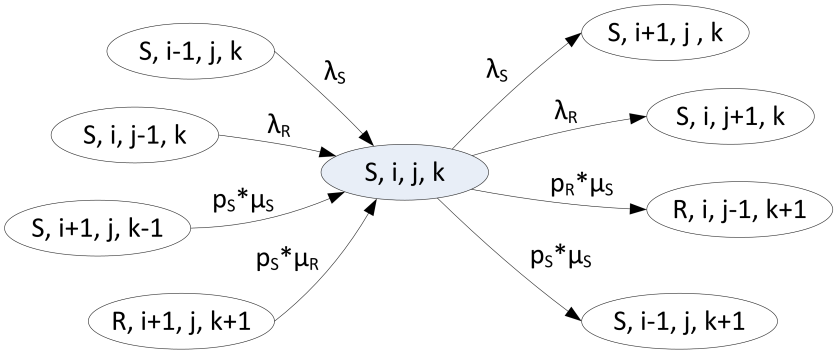
Using the memory-less property of the exponential distribution, the behavior of the queueing model can be represented as a Continuous Time Markov Chain. The stationary equations are presented in Appendix A.1. The stationary equations were solved numerically using Xpress-MP (Heipcke, 2000) and compared against simulation estimates. The experimental setup to verify the analytical model is presented in Table 4.1. The arrival rates ($\lambda_S = \lambda_R$) were set at 1 / time unit and the service times are set such that the utilization



(a) state illustrating storage blocking



(b) state illustrating retrieval blocking



(c) state illustrating a storage operation

Figure 4.3: Example state-transitions for the CTMC model of the shared-server system

Parameter	Levels (values)
Service Times	1 (corresponding to 80% utilization level)
Rack Size (Z)	10 (1 - 10)
Buffer Size ($B_S = B_R$)	$B_S = B_R = Z$ and $B_S = B_R > Z$
Number of Servers	1

Table 4.1: Design of experiments for shared server system (Markovian case)

of the shared-server is 0.8. While our model can handle any limit on the queue capacity, in our experiment we present two scenarios; one where the queue capacity is equal to the rack size, and the other where the queue capacity is greater than the rack size.

Let $P_{m,i,j,k}$ be the steady-state probability that CTMC will be in state (m, i, j, k) which we can obtain by solving the balance equations given in Appendix A.1. With the solution of $P_{m,i,j,k}$, we can derive some useful performance measures, such as:

1. Utilization of the server

$$P(S/R) = 1 - \sum_{k>=0} P_{0,0,0,k} \tag{4.1}$$

2. Probability of storage blocking

$$P(Storage\ Blocking) = \sum_{i>0} P_{0,i,0,Z} \tag{4.2}$$

3. Probability of retrieval blocking

$$P(Retrieval\ Blocking) = \sum_{j>0} P_{0,0,j,0} \tag{4.3}$$

4. Effective throughput of the server (can be less than the total arrival rate because of lost arrivals and blocking)

$$\lambda_{eff} = \mu_{SC} \left(1 - \left(\sum_{k\geq 0} P_{0,0,0,k} + \sum_{i>0} P_{0,i,0,Z} + \sum_{j>0} P_{0,0,j,0} \right) \right) \tag{4.4}$$

5. Expected number of retrieval requests waiting to be serviced

$$L_Q(R) = \sum_{j\geq 0} j.P_{m,i,j,k} \tag{4.5}$$

6. Expected number of storage requests waiting to be serviced

$$L_Q(S) = \sum_{i \geq 0} i.P_{m,i,j,k} \tag{4.6}$$

4.2.1 Numerical Experiments

In order to verify the results of the CTMC model, we compare the output of the CTMC model to the performance measure estimates obtained by simulating the shared-server. In the case of utilization and queue length performance measures, the relative percentage difference is defined as

$$Rel.\,Diff\% = \frac{Analytical - Simulation}{Simulation} * 100$$

We present the absolute difference when the quantities involved are small (typically less than 1) (Whitt, 1983). Tables 4.2 and 4.3 summarize the results when the buffer size is equal to the rack size and greater than the rack size, respectively. From the tables, we see that the CTMC model tracks the simulation model closely. The maximum relative percentage difference on the storage (retrieval) queue is 4.84% (4.97%) in the first scenario and 2.53% (2.68%) in the second scenario. In the case of utilization and average inventory level in the rack, the differences are very small, and only absolute differences are reported.

The difference between the analytical and simulation model can be explained by the following assumption in the CTMC model. In the CTMC model, when the server is capable of servicing either request, as in Figure 4.3(c) we have assumed that with a probability p_S $(= \lambda_S/(\lambda_S + \lambda_R)$, the server satisfies a storage request and with a probability p_R $(= 1 - p_S)$, the server satisfies a retrieval request. But in the simulation model we enforce the FCFS discipline strictly and the shared-server services the request that arrived first into the system.

The results indicate that the utilization is an increasing function of the rack size and it is less than the expected utilization level of 80%, because of 1) the blocking of requests when the rack is full or empty and 2) the loss of requests, when the storage or retrieval queues are full. The average number of items in the rack was maintained close to half of

the rack size, since the arrival rates were same for the storage and retrieval requests. We note that the average inventory level in the rack is the same in both cases, namely, "buffer size = rack size" and "buffer size > rack size".

The above analysis also provides confidence in the simulation model that will be used for evaluation of the approximate analytical model developed in the next section. As the rack size or the queue capacities increase, the CTMC state space grows rapidly and the numerical solution to the balance equations becomes challenging. If both queues have no capacity limits, then we also need to investigate the conditions under which the CTMC will have a steady state.

		Utilization			Storage Queue			Retrieval Queue			Rack		
Rack Size (Z)	Buffer Size ($B_S = B_R$)	A	S	%Diff	A	S	%Diff	A	S	%Diff	A	S	%Diff
1	1	0.500	0.500	0.000	0.374	0.374	0.000	0.374	0.374	0.000	0.500	0.500	0.000
2	2	0.614	0.615	0.001	0.731	0.737	0.006	0.731	0.738	0.007	1.000	1.000	0.000
3	3	0.670	0.671	0.001	1.070	1.093	2.10%	1.070	1.095	-2.28%	1.500	1.497	0.003
4	4	0.702	0.704	0.002	1.386	1.430	3.08%	1.386	1.432	3.21%	2.000	1.999	0.001
5	5	0.723	0.725	0.002	1.680	1.746	3.78%	1.680	1.750	4.00%	2.500	2.495	0.005
6	6	0.738	0.740	0.002	1.955	2.040	4.17%	1.955	2.046	4.45%	3.000	2.994	0.006
7	7	0.748	0.750	0.002	2.213	2.317	4.49%	2.213	2.320	4.61%	3.500	3.497	0.003
8	8	0.756	0.757	0.001	2.457	2.573	4.51%	2.457	2.580	4.77%	4.000	3.989	0.011
9	9	0.762	0.763	0.001	2.688	2.821	4.71%	2.688	2.828	4.95%	4.500	4.488	0.012
10	10	0.767	0.767	0.000	2.908	3.056	4.84%	2.908	3.060	4.97%	5.000	4.985	0.015

Table 4.2: Results for the shared-server CTMC model ($B_S = B_R = Z$) $\rho = 0.8$ and $\lambda_S = \lambda_R = 1$

		Utilization			Storage Queue			Retrieval Queue			Rack		
Rack Size (Z)	Buffer Size ($B_S = B_R$)	A	S	%Diff	A	S	%Diff	A	S	%Diff	A	S	%Diff
1	2	0.543	0.543	0.000	0.501	0.501	0.000	0.501	0.501	0.000	0.500	0.500	0.000
2	4	0.651	0.650	0.001	0.972	0.965	0.007	0.972	0.966	0.006	1.000	0.999	0.001
3	6	0.698	0.696	0.002	1.388	1.365	1.68%	1.388	1.367	1.54%	1.500	1.499	0.001
4	8	0.724	0.722	0.002	1.753	1.713	2.34%	1.753	1.715	2.22%	2.000	1.999	0.001
5	10	0.740	0.738	0.002	2.078	2.025	2.62%	2.078	2.029	2.41%	2.500	2.497	0.003
6	12	0.751	0.749	0.002	2.369	2.305	2.78%	2.369	2.308	2.64%	3.000	2.999	0.001
7	14	0.759	0.757	0.002	2.634	2.561	2.85%	2.634	2.561	2.85%	3.500	3.501	0.001
8	16	0.764	0.762	0.002	2.880	2.801	2.82%	2.880	2.800	2.86%	4.000	3.996	0.004
9	18	0.769	0.767	0.002	3.110	3.025	2.81%	3.110	3.026	2.78%	4.500	4.489	0.011
10	20	0.772	0.770	0.002	3.328	3.246	2.53%	3.328	3.241	2.68%	5.000	4.993	0.007

Table 4.3: Results for the shared-server CTMC model ($B_S = B_R > Z$) $\rho = 0.8$ and $\lambda_S = \lambda_R = 1$

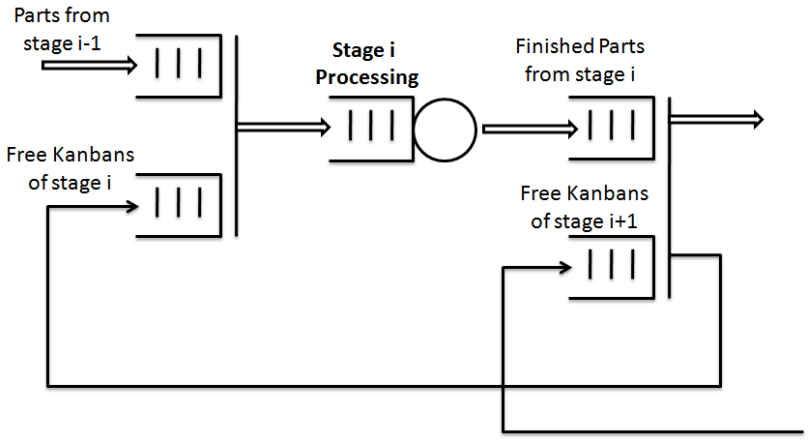


Figure 4.4: A single stage kanban system (Krishnamurthy, 2002)

4.3 Queueing Network Model of the Shared-Server

In this section, we present an approximate analytical model of the general shared-server system. Our modeling and solution approach uses previous work on the parametric-decomposition method (Whitt, 1983) and the modeling of synchronization operations (Krishnamurthy, 2002). The need for modeling synchronization operations can be explained by the observation that for a storage (retrieval) operation to begin a storage (retrieval) request must be waiting and an empty space (item) must be present in the rack.

The analytical model presented here is based on the material control model developed for a single stage kanban system by Krishnamurthy (2002). In a kanban controlled production system, kanbans (cards) are used to control material flow and trigger production. In a multi stage production system, each stage has a fixed number of kanbans. A part can be processed at given stage i, if the corresponding stage i kanban is attached to the part. Upon completion of the process, both the finished part and the kanban wait at the output buffer of stage i. The part is transferred to the next stage, stage i+1, as soon as a kanban from stage i+1 is available. The stage i kanban is then returned to the input buffer of stage i enabling new parts to enter stage i. Such a kanban system can be represented as a queueing model using fork/join synchronization stations and is shown in Figure 4.4. A similar material control modeling approach is followed in the development of the approximate analytical model of the shared-server system.

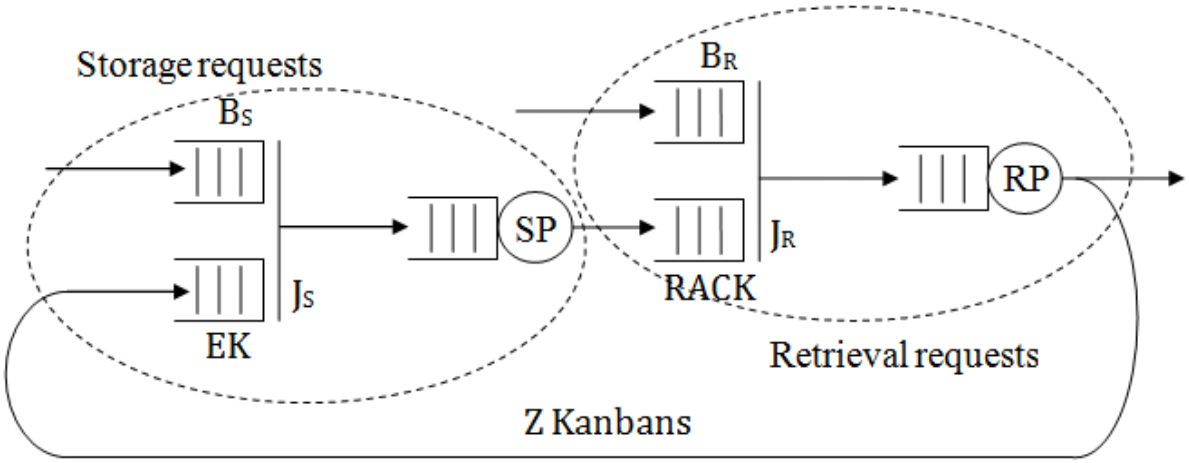


Figure 4.5: Queueing network model of a shared server

Figure 4.5 represents a queueing network model of a single class, single shared-server system. The shared-server is represented as two independent servers, serving the storage and retrieval requests, at the storage processing (SP) and retrieval processing (RP) stations, respectively. The synchronization stations J_S and J_R model the material control mechanism at the processing stations. A closed loop system is formed by the sync stations J_S and J_R and processing stations SP and RP as shown. Associated with this closed loop are Z kanbans which represent the number of rack spaces in the system. Hence, this part of the queueing network model can be viewed as a closed queueing network where the kanbans act as customers, circulating in the network formed by queues EK, SP, $RACK$ and RP. The arrival processes of storage and retrieval requests are external processes that need to be synchronized with the internal flow of the kanbans in the closed loop.

The sync station J_S models the synchronization of the storage requests (that arrive from the upstream or external processes) with that of kanbans that represent the empty spaces in the rack. J_S has two input queues, the storage request queue (B_S) and the empty space/kanban queue (EK). The sync station J_R models the synchronization of the retrieval requests (that arrive from downstream or external processes) with that of kanbans that represent items in the rack. J_R has two input queues, the retrieval request queue (B_R) and the rack queue ($RACK$).

The material control model works as follows. Unit-load items wait in the queue, RACK,

to be retrieved and satisfy a retrieval request. These items in the $RACK$ have a kanban attached to it. As soon as a retrieval request arrives in queue B_R, an item from the $RACK$ and the request is joined together and released from the sync station to join the retrieval processing queue. Upon completion of the service (the item was successfully retrieved from the rack), the kanban is returned to the queue EK, that represents the empty spaces in the rack. The queue EK is a part of the storage synchronization station (J_S). At this station, when a storage request arrives at the queue B_S, it is immediately joined with the kanban in queue EK and sent to the storage processing station (SP) to be stored in the rack. As we can see from the operating mechanism, the kanban cards are either in one of the four queues or at the servers; EK representing empty spaces, SP representing unit-loads waiting to be stored and in process, $RACK$ representing unit-loads in the rack and RP representing unit-loads waiting to be retrieved and in process. Hence, these four queues and the two servers can have a maximum of Z kanbans/customers at any time.

The arrival processes at the synchronization stations and service processes at the processing stations are assumed to be general and hence, the queueing network model is a non-product form. Therefore, approximation techniques must be used for performance analysis. The solution approach to solving the queueing network consists of four steps and is based on the parametric-decomposition approach (Whitt (1983, 1994); Krishnamurthy (2002)). The two main features of this approach are that the departure and arrival processes within the network are approximated as renewal processes and such a renewal process is described by its first two moments, mean and squared coefficient of variation. In reality, the successive departures or arrivals in the closed queueing network are not independent, and hence not a renewal process. Earlier works on open and closed queueing network models (Kuehn (1979); Whitt (1983); Kamath (1989); Krishnamurthy (2002)) have shown that such an approximation is effective and yields reliable estimates of the desired performance measures without much computational effort. In this solution approach, we extend the application of this technique to solve the shared-server system model. The approach consists of four steps: Decomposition, Characterization, Linkage, and Solution. An overview of these steps is given below and illustrated in Figure 4.6.

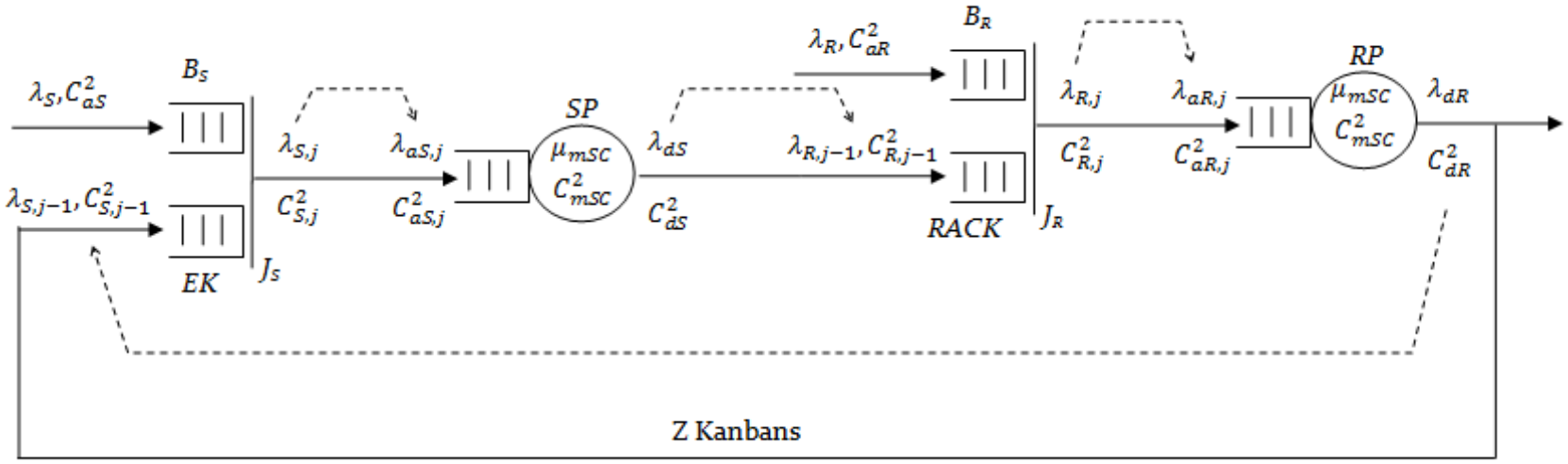


Figure 4.6: Overview of parametric-decomposition approach for the queueing network model of the shared-server

- **Decomposition**: The queueing network representation of the shared server system is decomposed into individual components; storage and retrieval synchronization stations (J_S and J_R), and storage and retrieval processing stations (SP and RP).

- **Characterization**: Each component/station (J_S, SP, J_R and RP) obtained from the decomposition step is analyzed in isolation. We assume that the arrival process and the service process (if any) are known and are renewal processes. We also assume that the renewal process is adequately quantified by two parameters; the mean and squared coefficient of variation (SCV) of the inter-renewal times. In this queueing network, we know the external arrival processes for storage and retrieval requests, and the single command service process but we do not know the internal departure processes from each of the components. By analyzing the components/stations independent of the rest of the network, we obtain the mean and SCV of the departure process and estimate the performance measures of each of the components. The details of this characterization step, especially the storage and retrieval processing stations are described in detail in the following sections.

- **Linkage**: In this step, the traffic equations from the individual stations are linked together in the closed loop part of the queueing network. We make use of the expressions derived in Krishnamurthy & Suri (2006) that link the traffic processes at the stations (J_S and SP, SP and J_R, J_R and RP, and RP and J_S). The resulting sets of non-linear equations are then solved numerically to obtain the parameters of the internal traffic processes.

- **Solution**: The system of non-linear equations, linking the traffic processes of the individual components is iteratively solved to determine the internal traffic parameters. Once those parameters are determined, the network performance measures as well as the station performance measures can be obtained easily.

4.3.1 Characterization of the Synchronization Station

The decomposition step presents two synchronization stations, storage and retrieval synchronization stations. The storage sync station (J_S) consists of two input queues, one for

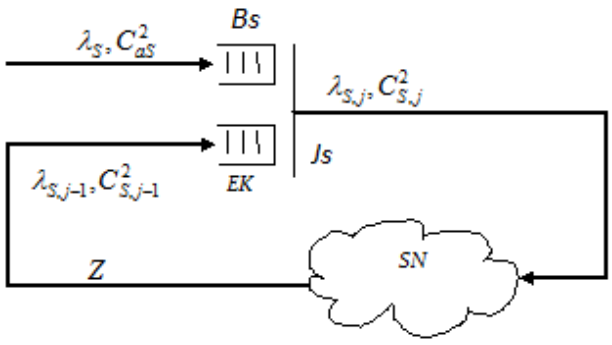


Figure 4.7: Characterization of storage synchronization station (J_S)

the storage requests that come from upstream stages (B_S) and the other for kanbans representing empty storage spaces (EK). The sub-network SN in Figure 4.7 represents the downstream stages where the Z kanbans circulate. In the queue EK, kanbans representing empty storage spaces wait for the storage requests. Each storage request is then attached with the kanban and they proceed together to be processed, i.e., the storage request and the kanban are routed together in the sub-network SN. Because the rack has a finite capacity represented by the Z kanbans, the sum of kanbans in queue EK and sub-network SN will always be equal to Z. Additionally, the arrival process to the queue B_S will shut-off as soon as its capacity is reached, which is set at B_S.

In line with two moment approximations, we assume that the arrival processes to queues EK and B_S are renewal processes characterized by the mean and SCV of the inter-arrival times; λ_S^{-1}, C_{aS}^2 to the queue B_S and $\lambda_{S,j-1}^{-1}$, $C_{s,j-1}^2$ to the queue EK. There are only Z kanbans circulating in the sub-network and queue EK, the arrivals to the queue EK shuts off once all the kanbans are in the queue EK. Hence, we really assume that the traffic process conditioned on the event that it is not shutdown is a renewal process. Thus, the synchronization station J_S is characterized by the 6-tuple $(\lambda_S^{-1}, C_{as}^2, B_S, \lambda_{s,j-1}^{-1}, C_{s,j-1}^2, Z)$. The characterization of J_S will be complete when the parameters for the departure process are specified, which were derived by Krishnamurthy (2002). Let $r = {}^{\lambda_S}/_{\lambda_{S,j-1}} | r \leq 1$ and $C^2 = 0.5(C_{aS}^2 + C_{s,j-1}^2)$.

The rate of the departure process is given by

$$\lambda_{S,j} = \begin{cases} \lambda_S \left[\frac{1-r^{Z+B_S}}{1-r^{Z+B_S+1}}\right] \left[1 - 0.5(C^2-1)\left(\frac{(1-r)r^{Z+B_S}}{1-r^{2(Z+B_S)+1}}\right)\right] & r < 1 \\ \lambda_S \left(\frac{Z+B_S}{Z+B_S+1}\right)\left(1 - \frac{0.5(C^2-1)}{2(Z+B_S)+1}\right) & r = 1 \end{cases} \tag{4.7}$$

As we can see from the above expression, for finite values of Z and B_S, the rate of the departure process is always less than $min(\lambda_S, \lambda_{S,j-1})$. The SCV of the departure process is given by (Krishnamurthy, 2002)

$$C^2_{S,j} = \left[\left(\frac{\lambda^5_S}{\lambda^5_S + \lambda^5_{S,j-1}}\right) C^2_{S,j-1} + \left(\frac{\lambda^5_{S,j-1}}{\lambda^5_S + \lambda^5_{S,j-1}}\right) C^2_{aS}\right]\left[1 - \frac{1}{(Z+B_S+1)} - \frac{1}{(Z+B_S+1)^2}\right] \tag{4.8}$$

The expressions for queue length parameters are (Krishnamurthy, 2002)

$$L_{BS} = \begin{cases} \lambda_S \left[\frac{1-r^{Z+B_S}}{1-r^{Z+B_S+1}}\right] \left[1 - 0.5(C^2-1)\left(\frac{(1-r)r^{Z+B_S}}{1-r^{2(Z+B_S)+1}}\right)\right] & r < 1 \\ \left(\frac{B_S}{2}\right)\left(\frac{B_S+1}{Z+B_S+1}\right) & r = 1 \end{cases} \tag{4.9}$$

$$L_{EK} = \begin{cases} \lambda_S \left[\frac{1-r^{Z+B_S}}{1-r^{Z+B_S+1}}\right] \left[1 - 0.5(C^2-1)\left(\frac{(1-r)r^{Z+B_S}}{1-r^{2(Z+B_S)+1}}\right)\right] & r < 1 \\ \left(\frac{Z}{2}\right)\left(\frac{Z+1}{Z+B_S+1}\right) & r = 1 \end{cases} \tag{4.10}$$

The characterization of the retrieval synchronization station (J_R) is very similar to that of storage synchronization station. The station J_R is characterized by two queues, the retrieval request queue (B_R) and the queue representing the items in storage ($RACK$).

4.3.2 Characterization of the Processing Station

Figure 4.8 shows the storage processing station obtained by the decomposition of the queueing network. The processing station can be any configuration, and in this section we assume a single server storage processing station operating under a FCFS discipline. The sub-network SN represents the rest of the queueing network in which the Z kanbans circulate. The arrivals to the SP station are the storage requests with kanbans, i.e., each storage request will have a space reserved for it when it joins the queue. Upon completion of the storage processing operation, the kanbans are routed back to the sub-network SN where they are subject to random delays. In the sub-network, the kanban stays in the $RACK$

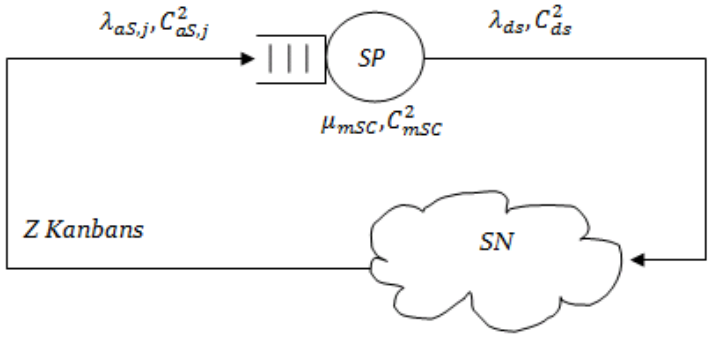


Figure 4.8: Characterization of the Storage Processing station (SP)

until it is matched to a retrieval request and the completion of the retrieval operation will release the kanban to the queue EK. Matching a storage request with a kanban in queue EK results in the kanban revisiting the storage processing station. The number of kanbans at the SP station and sub-network will always be equal to Z and consequently, the arrival process to the SP station shuts off when all the kanbans are in the SP station.

The arrival process to the SP station can be fairly complex. Hence, in line with the two-moment approximations, we assume that the arrival process to the SP station is a renewal process conditioned on the event that the arrival shuts off when all the customers are in the station. The arrival process is characterized by the mean and SCV of the inter-arrival times $(\lambda^{-1}_{aS,j}, C^2_{aS,j})$. Together with the parameters describing the service process, the SP station can be represented by the 5-tuple $(\lambda^{-1}_{aS,j}, C^2_{aS,j}, \mu^{-1}_{mSC}, C^2_{mSC}, Z)$. The service times are modified single command cycle times since the SP station and RP station are both serviced by a single shared server. The details of the modification are presented in the next subsection. Meanwhile, we assume that the service times are i.i.d with mean (μ^{-1}_{mSC}) and SCV (C^2_{mSC}). The characterization of the SP station will be complete with the description of the departure process and the performance measure of interest, namely the mean queue length.

By flow conservation principle, the mean of the inter-departure time is given by

$$\lambda^{-1}_{ds} = \lambda^{-1}_{aS,j} \tag{4.11}$$

The estimation of the SCV of the inter-departure times is based on the approximation by Whitt (1983) for a $GI/G/1$ queue. Let $\rho_S = {}^{\lambda_{S,j}}/_{\mu_{mSC}}$be the utilization of the shared server to account for servicing the storage requests. Then, the SCV (C_{ds}^2) is given by

$$C_{dS}^2 = (1 - \rho_S^2)C_{aS,j}^2 + \rho_S^2 C_{mSC}^2 \tag{4.12}$$

To obtain the expression for mean queue length of the SP station, we first obtain the expression for mean waiting time ($W_{q,SP}$). Following the approach by Kamath et al. (1988), the approximate mean waiting time is given by $Cf * W_q(GI/G/1)$, where Cf is a correction factor and $W_q(GI/G/1)$ is the waiting time in a $GI/G/1$ queue. Based on the work of Kuehn (1979) and Whitt (1983), the mean waiting time in a $GI/G/1$ queue is given by

$$W_q = g(\rho_S, C_{aS,j}^2, C_{mSC}^2)\left(\frac{C_{aS,j}^2 + C_{mSC}^2}{2}\right)\left(\frac{\rho_S}{1-\rho_S}\right)\mu_{mSC}^{-1} \tag{4.13}$$

The above equation assumes that the customers to the queue arrive from an infinite population. The correction factor (Cf) accounts for the finite population of Z kanbans in the closed loop part of the queueing network, and has been derived by Kamath et al. (1988) as,

$$Cf = \left(\frac{Z-1}{Z}\right)\left(\frac{1}{1 + \frac{W_q}{Z\,\mu_{mSC}^{-1}}}\right) \tag{4.14}$$

Then, using Little's law (Little, 1961), we obtain the mean queue length at the storage processing station as

$$L_{q,SP} = \lambda_{S,j} * Cf * W_q \tag{4.15}$$

Modifying the Single Command Service Time

In the characterization of the SP station, we had mentioned that the single command service time was modified. The reason behind this modification is to account for a single server that is shared by the storage processing station (SP) and the retrieval processing station (RP). We model this shared server as two independent servers and then account for the time spent on each other's activities. The service time spent by the shared-server in performing the

storage activity accounts for the time it spends in performing any retrieval activity before the next storage activity and vice versa. This is similar to the approach used by Segal & Whitt (1989) to model the service interruptions within the QNA framework (Whitt, 1983). To modify the service time for the storage processing station, we assume that the down time is the time spent on performing retrieval operations between two storage operations. Let λ_S and λ_R be arrival rates for storage and retrieval requests, and $\tau_{SC}(= 1/\mu_{SC})$ be the mean single command service time for storage/retrieval requests. Let S' be the random variable representing the modified service time, which is given by

$$S' = S + \sum_{N_R} R \tag{4.16}$$

where $S(R)$ is the random variable representing original storage (retrieval) time and N_R is the random variable representing the number of retrieval operations performed until the next storage operation. The number of retrieval requests completed before servicing another storage request is a modified geometric random variable, whose parameter is the probability of that the next request is a storage request and is given by $p_S = \lambda_S/(\lambda_R + \lambda_S)$. $\sum_{N_R} R$ is a random sum of identical random variables R.

The mean of the modified single command storage service time is given by

$$\begin{aligned} E[S'] &= E[S] + E[N_R] * E[R] \\ E[N_R] &= \tfrac{p_S}{1-p_S} \\ E[S] = E[R] &= \mu_{SC}^{-1} = \tau_{SC} \\ E[S'] &= \tau_{mSC} = \tfrac{\tau_{SC}}{1-p_S} \end{aligned} \tag{4.17}$$

The variance of the modified single command storage service time is given by

$$\begin{aligned} Var[S'] &= Var[S] + Var[\textstyle\sum_{N_R} R] \\ Var[S'] &= Var[S] + Var[N_R] * (E[R])^2 + E[N_R] * Var[R] \\ Var[R] &= C_{SC}^2 * \tau_{SC}^2 \\ Var[N_R] &= \tfrac{p_S}{(1-p_S)^2} \\ C_{mSC}^2 &= \tfrac{Var[S']}{\tau_{mSC}^2} \end{aligned} \tag{4.18}$$

Since, we assume that the arrival rates for storage and retrieval requests are equal ($\lambda_S = \lambda_R$), the expressions 4.17 and 4.18 will also apply for modifying the service time for the retrieval requests. We also note that the two independent servers can be 'active' at the same time in the queueing network model. The specification of the traffic processes for each of the components/stations and the performance measures of interest complete the characterization step. In the following section, we link the traffic processes of the decomposed components.

4.3.3 Linking the Stations

In the characterization step, the parameters describing the input processes to the processing stations and the synchronization stations are assumed to be given, which in fact need to be determined. In this section, we will determine the relationship between the inter-departure times from a station and the inter-arrival times to a downstream station, thereby explicitly incorporating the effects of shut downs in the arrival process in the closed part of the queueing network. We need to determine the following four linkages (see Figure 4.6).

1. Linking the departure processes of the storage synchronization station (J_S) at the arrival process of the storage processing station (SP)

2. Linking the departure process of the storage processing station (SP) with the retrieval synchronization station (J_R)

3. Linking the departure processes of the retrieval synchronization station (J_R) at the arrival process of the retrieval processing station (RP)

4. Linking the departure process of the retrieval processing station (RP) with the storage synchronization station (J_S).

Linking the Departure Process from J_S to Arrival Process at SP

We now describe the procedure to link the mean and SCV of the departure process of the station J_S ($\lambda_{S,j}^{-1}$ and $C_{s,j}^2$) to the mean and SCV of the arrival process at the SP ($\lambda_{aS,j}^{-1}$ and $C_{as,j}^2$) (see Figure 4.9). We note that the parameters of the arrival process to queue

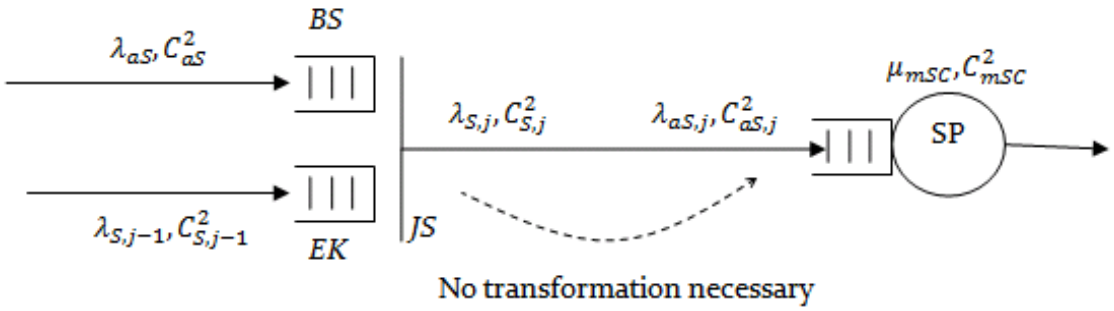


Figure 4.9: Linking storage synchronization and storage processing stations

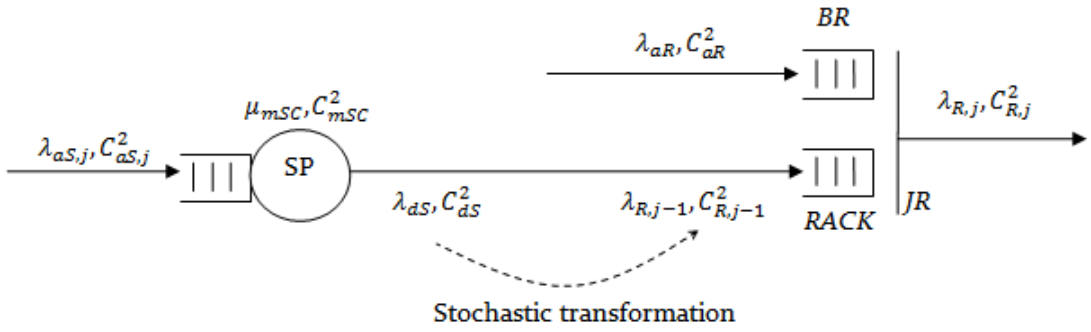


Figure 4.10: Linking storage processing and retrieval synchronization stations

B_S correspond to external demand and hence, the λ_{aS}^{-1}, C^2_{aS} and B_S are user inputs. The parameters of the inter-arrival process at SP, mean ($\lambda_{aS,j}^{-1}$) and SCV ($C^2_{aS,j}$) can be equated directly to the parameters of the inter-departure process from J_S. In the characterization step, the correction factor incorporates the finite population nature of the closed part of the queueing network and hence, we do not have to modify the parameters of the inter-arrival process at SP. Then, using flow conservation principle,

$$\begin{aligned} \lambda_{aS,j} &= \lambda_{S,j} \\ C^2_{aS,j} &= C^2_{S,j} \end{aligned} \tag{4.19}$$

We can provide a similar argument to link the departure process from the retrieval synchronization station (J_R) to the arrival process to the retrieval processing station (RP).

Linking the Departure Process from SP to Arrival Process at J_R

We now describe the procedure to link the mean and SCV of the departure process of the station SP (λ_{ds}^{-1} and C_{ds}^2) to the mean and SCV of the arrival process at the J_R ($\lambda_{R,j-1}^{-1}$ and $C_{R,j-1}^2$) (see Figure 4.10). We note that the parameters of the arrival process to queue B_R could correspond to external retrieval request arrival process and hence, λ_{aR}^{-1}, C_{aR}^2 and B_R are user inputs to this model. In the characterization step of the synchronization station, the parameters of the arrival process to the queue RACK ($\lambda_{R,j-1}^{-1}$ and $C_{R,j-1}^2$) are not conditioned on the event that the arrival process is shut-down when all the kanbans are in RACK. Hence, we need to incorporate this fact when we link these two stations. Krishnamurthy & Suri (2006) developed a procedure to develop the linkage equations analyzing the arrival point process to the queue RACK. We will describe the procedure next. Let π_{RACK} denote the long run proportion of time that arrivals to RACK are shut down. Then,

$$\begin{aligned} \lambda_{R,j-1} &= \tfrac{\lambda_{dS}}{1-\pi_{RACK}} \\ C_{R,j-1}^2 &= \tfrac{C_{dS}^2}{(1-\pi_{RACK})^2} - \left(\tfrac{\pi_{RACK}}{(1-\pi_{RACK})^2}\right)\left(\tfrac{\lambda_{dS}}{\lambda_{aR}}\right)\left(\tfrac{2C_{aR}^2}{1+C_{aR}^2}\right) \end{aligned} \tag{4.20}$$

By the principle of flow conservation, we have $\lambda_{R,j}^{-1} = \lambda_{dS}^{-1}$. Together, the three equations provide the necessary stochastic transformation required to relate the traffic processes of the stations SP and J_R. We assume that parameters of the departure process from the station SP ($\lambda_{dS}^{-1}, C_{dS}^2$) are given and proceed to develop a numerical procedure to find the arrival process to the station J_R. We know that the arrival parameters and the size of the queue B_R ($\lambda_{aR}^{-1}, C_{aR}^2, B_R$) are the user inputs. The iterative numerical procedure, based on the bisection search method is given in Algorithm 4.3.1.

Algorithm 4.3.1: LINKPROCTOSYNC$(\lambda_{dS}, C_{dS}^2, Z, \lambda_{aR}, C_{aR}^2, B_R)$

comment: Links the storage processing station and retrieval synchronization station

Initialize: $low = 0, high = 1, \varepsilon$

while $|\delta| > \varepsilon$

do
- **Step1:** *Set* $\pi_{RACK}^{(k)} = (low + high)/2$
- **Step2:** *Compute* $\lambda_{R,j-1}$ *and* $C_{R,j-1}^2$
- **Step3:** *Compute* $\lambda_{R,j}$ *using synchronization characterization equations*
- **Step4:** *Compute* $\delta = \lambda_{R,j}^{(k)} - \lambda_{dS}$
- **if** $\delta < -\varepsilon$
  - **then** $low = \pi_{RACK}^{(k)}$
  - **else if** $\delta > \varepsilon$
  - **then** $high = \pi_{RACK}^{(k)}$

In the algorithm 4.3.1, $\pi_{RACK}^{(k)}$ represents the estimate for π_{RACK} in the k^{th} iteration. In each iteration, we compute the difference between the estimated throughput ($\lambda_{R,j}$) and the required throughput (λ_{dS}). If the estimated throughput is lower than the required throughput, then we update the interval of the bisection search to increase the value of π_{RACK} in the next iteration, and vice-versa. The algorithm terminates when the estimate of π_{RACK} meets the required tolerance level (ε) in the throughput. Convergence property of the algorithm is discussed in a later section. Using a similar argument, we can develop a numerical procedure to solve the linkage equations connecting the retrieval processing station (RP) and the storage synchronization station (J_S).

4.3.4 Solution Approach

The solution step involves solving the set of non-linear equations linking the traffic process of the individual stations in the closed part of the queueing network. We increase the user input (Z) by one kanban to account for the fact that the two independent servers can be active at the same time in the queueing network model.

We initialize the algorithm by first modifying the single command storage (retrieval) processing time using equations 4.16 - 4.18. Then, we proceed with an initial estimate of

parameters for the departure process of one of the four component stations, in our case it is the storage processing station SP. The algorithm then iteratively estimates the internal traffic process parameters, updating the initial estimates until they are consistent with the user inputs. The solution procedure is described in Algorithm 4.3.2.

After computing the modified service time parameters for the processing stations, we obtain initial estimates of λ_{dS}, and C_{dS}^2 in step 1 of loop1. As the initial estimates may be inconsistent with each other, we update the value of C_{dS}^2 in loop2 for given value of λ_{dS}. To update C_{dS}^2, we make use of the characterization and linking equations derived in earlier sections, and solve for the traffic parameters for each of the stations sequentially (steps 2.1 to 2.8). Upon solving the SP, we obtain a new value for C_{dS}^2 . We repeat this procedure until the difference between the new and old estimates of C_{dS}^2 are within the set tolerance limit (ε).

In step 3, algorithm verifies if these values of λ_{dS} and C_{dS}^2 are consistent with the user input values for the number of kanbans. To do so, we calculate the difference between the user input value (increased by one) and sum of the kanbans at each of the stations including the servers.

We update λ_{dS} using a bisection search approach in step 4, until the difference between current and previous estimates are within a predefined tolerance limit. If the sum of mean queue lengths is more than the number of kanbans, then the current estimate of λ_{ds} is too high; the bisection search algorithm accordingly updates the interval for the next iteration. At the end of loop1, we would have obtained λ_{dS} and C_{dS}^2 that are consistent with each other, and consistent with user input & modified service times.

As a last step, we update the modified service time based on the effective internal arrival process to the processing stations. We repeat the entire procedure starting with step 1, until the difference between the current and previous estimate of the modified service times are within the specified tolerance limits. Once, the algorithm converges, we can obtain the performance measures of interest such as the throughput and mean queue lengths at the various processing and synchronization stations.

The average inventory at the rack is the sum of the mean queue lengths of $RACK$, and mean number of customers at the retrieval processing station. The average number of

storage requests waiting to be serviced is the sum of the mean queue length of B_S and mean queue length at the storage processing station. The average number of retrieval requests waiting to be serviced is the sum of the mean queue length of B_R and mean queue length at the retrieval processing station.

Algorithm 4.3.2: SUPERSOLVE($\lambda_S, C_{aS}^2, B_S, \lambda_R, C_{aR}^2, B_R, \mu_{SC}, C_{SC}^2, Z$)

Modify: $No\ of\ kanbans\ Z' = Z + 1$

Compute: $Modified\ single\ command\ service\ time\ at\ SP\ and\ RP\ (\mu_{mSC}, C_{mSC}^2)$

Initialize: $Low = 0, High = min(\lambda_S, \lambda_R, \mu_{mSC})$

while $|\delta_1| > \varepsilon$

do

Step1: $Let\ \lambda_{ds}^{(k)} = (Low + High)/2,\ C_{ds}^{2(k)} = 1.0(say)$

while $|\delta_2| > \varepsilon$

do

Step2.1: $Solve\ \pi_{RACK}, \lambda_{R,j-1}, and\ C_{R,j-1}^2\ using\ Algorithm$ 4.3.1
$setting\ \lambda_{dS} = \lambda_{dS}^{(k)}\ and\ C_{dS}^2 = C_{dS}^{2(k)}$

Step2.2: $Calculate\ \lambda_{R,j}, C_{R,j}^2\ and\ L_{q,BR}\ and\ L_{q,RACK}\ using$
$the\ characterization\ equations$

Step2.3: $Compute\ input\ parameters\ to\ the\ retrieval\ processing\ station$

Step2.4: $Calculate\ \lambda_{dR}, C_{dR}^2\ and\ L_{q,RP}\ using$
$the\ RP\ characterization\ equations$

Step2.5: $Solve\ \pi_{EK}, \lambda_{S,j-1}, and\ C_{S,j-1}^2\ using\ Algorithm$ 4.3.1

Step2.6: $Calculate\ \lambda_{S,j}, C_{S,j}^2\ and\ L_{q,BS}\ and\ L_{q,EK}\ using$
$the\ characterization\ equations$

Step2.7: $Compute\ input\ parameters\ to\ the\ storage\ processing\ station$

Step2.8: $Calculate\ \lambda_{dS}, C_{dS}^2\ and\ L_{q,SP}\ using$
$the\ SP\ characterization\ equations$

Step2.9: $Compute\ \delta_2 = |C_{dS}^{2(k)} - C_{dS}^2|; C_{dS}^{2(k)} = C_{dS}^2$

Step3: $Compute\ \delta_1 = L_{q,RACK} + L_{q,RP} + \rho_R + L_{q,EK} + L_{q,SP} + \rho_S - Z'$

Step4: **if** $\delta_1 < -\varepsilon$

then $Low = \lambda_{dS}^{(k)}$

else if $\delta_1 > \varepsilon$

then $High = \lambda_{dS}^{(k)}$

4.3.5 Computational Effort and Convergence

We note that the number of unknown parameters in this analysis is independent of the number of customers in the closed part of the queueing network model. Both numerical algorithms are based on the bisection search procedure, and it is assumed that the solution lies within the specified intervals. In the Algorithm 4.3.1, bisection search is used to estimate the probability of the shut downs, (π_{RACK}) to the queue $RACK$ and (π_{EK}) to the queue EK. Hence, the interval $[0, 1]$ is sufficient to search for the probabilities.

In the case of Algorithm 4.3.2, the bisection search is used to obtain the throughput of the storage processing station consistent with the user input values. We note that the throughput of the system, then must lie within the interval $[0, min(\lambda_{aS}, \lambda_{aR}, \mu_{mSC})]$ where the μ_{mSC} is the modified single command service rate at the storage or retrieval processing station. We cannot guarantee a unique solution within this interval or provide a bound on the number of iterations necessary for the model to converge. In all our experiments, the algorithm converged to a solution within a reasonable number of iterations ($<$20).

4.3.6 Performance Measures and Model Accuracy

The performance measures of interest for the shared server model are related to throughput, inventory, and warehouse resources. The performance measures related to the throughput are the throughputs for storage and retrieval requests. Throughput is defined as the number of requests served per unit time.

Other measures of interest are the waiting time and average number of storage and retrieval requests in the system. With respect to inventory, average number of items in storage is a measure of interest. With respect to resources, utilization is the major performance measure. In the following sections, we summarize the results for the throughput for retrieval requests, utilization of the shared server, the mean queue length of the storage & retrieval queues, and the average inventory level in the rack.

The accuracy of the models is tested by comparing the analytical results with simulation estimates. The simulation models were developed for the shared-server component using the Arena simulation software (Kelton et al., 2002). The steady state estimate of a performance

measure was obtained by averaging over appropriate number of replications after accounting for warm-up periods. The warm-up period was estimated using Welch's method (Welch, 1983) and set at 400,000 entities. The statistics were collected for 600,000 entities (retreival requests) and the performance measures were averaged for 10 replications.

The relative percentage error (RE), a common measure to test the accuracy of analytical models, was used in the case of throughput (λ_{dR}) and utilization (ρ_{SC}). When the magnitude of the performance measures is small (typically less than 1), absolute error is considered better than RE (Whitt, 1983).

$$RE(\lambda_{dR}) = 100 * \frac{\lambda_{dR}^{(Analytical)} - \lambda_{dR}^{(Simulation)}}{\lambda_{dR}^{(Simulation)}}$$

In the case of mean queue lengths and average inventory in the rack, normalized error is measured rather than relative percentage error. The normalized error is measured as the difference between the analytical and simulation model as a percentage of the rack size. Since the shared server system is modeled as a queueing network, the normalized error (NE) is measured as

$$NE(L_Q(S)) = 100 * \frac{L_Q(S)^{(Analytical)} - L_Q(S)^{(Simulation)}}{Rack\ Size}$$

$$NE(L_Q(R)) = 100 * \frac{L_Q(R)^{(Analytical)} - L_Q(R)^{(Simulation)}}{Rack\ Size}$$

$$NE(L(RACK)) = 100 * \frac{L(RACK)^{(Analytical)} - L(RACK)^{(Simulation)}}{Rack\ Size}$$

The normalized error is used for measuring queue length accuracy in order to avoid the small queue length effect. Robustness of the analytical models will be tested by varying the parameters of the inter-arrival time distributions for storage/retrieval requests, service time distribution, and rack size. The performance measures will be examined under low (SCV = 0.5), medium (SCV = 1) and high variability (SCV = 2) conditions.

An experimental design is provided in Table 4.4 for the shared-server model. The arrival rates for the storage and retrieval requests are fixed at one, and the mean service times are

Parameter	Levels (values)
Service Times	3 (corresponding to 70%, 80% and 90% utilization levels)
SCV of service time distribution	3 (0.5, 1.0 and 2.0)
SCV of inter-arrival time distribution	3 (0.5, 1.0 and 2.0)
Rack Size	3 (5, 25 and 125)
Number of servers	1

Table 4.4: Design of experiments for shared-server system: single server case

set such that the expected utilization of the shared server is 70%, 80% and 90%.

4.3.7 Accuracy of the Shared-Server Model

The input parameters to the queueing model are the arrival parameters of the storage and retrieval requests, the service parameters of the single command service time, the queue capacities and the rack size. The output parameters, namely the performance measures of interest are the mean queue lengths of the storage and retrieval requests, the throughput (which is the departure rate of the retrieval requests) and average inventory in the rack. In the case of the shared-server system, we are also interested in measuring the parameters of the departure process of the retrieval requests as they will become the inputs to the downstream stations in an end-to-end comprehensive model of the warehouse. In all our experiments in this section, the shared-server is a single server operating under a FCFS discipline. We also study the shared-server system when the variability in the arrival processes is the same (balanced case) and when the variability is different (unbalanced case).

Balanced Case

In the balanced case, the storage and retrieval processes have the same arrival rate and variability. The estimates of mean queue length (storage and retrieval requests) and the average inventory level in the rack at 70%, 80% and 90% expected utilization for the balanced case are given in Tables 4.5, 4.7, and 4.9 respectively. The estimates of throughput and utilization of the shared-server are reported in Tables 4.6, 4.8, and 4.10 for the three expected utilization levels. The results from Tables 4.5, 4.7, and 4.9 indicate that the maximum absolute error for the mean queue length of storage (retrieval) request is 5.84% (5.83%).

The maximum absolute error for the average inventory in the rack is 3.95%. We note that the above errors are found at the 90% expected utilization levels. In the case of throughput and actual utilization, the maximum absolute error is 13.84% for 90% utilization levels. The observed error percentages are in the range of good to acceptable for queueing models based on two moment approximations as noted in many previous studies (e.g. Whitt, 1983 and Suri et al., 1993). Next, we develop insights into the behavior of the shared-server model for the balanced system.

			Storage Queue			Retrieval Queue			Average Inventory		
$C_{aS}^2 = C_{aR}^2$	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	0.5	5	0.526	0.387	2.79%	0.527	0.387	2.79%	2.416	2.499	-1.65%
0.5	0.5	25	1.220	0.781	1.76%	1.220	0.786	1.74%	12.317	12.474	-0.63%
0.5	0.5	125	2.438	2.296	0.11%	2.438	2.252	0.15%	62.300	62.915	-0.49%
0.5	1	5	0.595	0.471	2.48%	0.595	0.472	2.47%	2.423	2.499	-1.53%
0.5	1	25	1.414	0.928	1.95%	1.414	0.932	1.93%	12.318	12.481	-0.65%
0.5	1	125	2.682	2.451	0.18%	2.682	2.402	0.22%	62.293	62.894	-0.48%
0.5	2	5	0.713	0.581	2.64%	0.713	0.581	2.65%	2.434	2.499	-1.31%
0.5	2	25	1.786	1.217	2.28%	1.787	1.218	2.27%	12.321	12.497	-0.70%
0.5	2	125	3.166	2.717	0.36%	3.166	2.691	0.38%	62.298	62.870	-0.46%
1	0.5	5	0.547	0.475	1.45%	0.548	0.475	1.45%	2.424	2.497	-1.46%
1	0.5	25	1.340	1.030	1.24%	1.340	1.030	1.24%	12.319	12.472	-0.61%
1	0.5	125	2.600	2.606	0.00%	2.600	2.676	-0.06%	62.291	62.631	-0.27%
1	1	5	0.611	0.534	1.53%	0.611	0.535	1.52%	2.430	2.498	-1.36%
1	1	25	1.888	1.178	2.84%	1.888	1.178	2.84%	12.321	12.469	-0.59%
1	1	125	2.841	2.776	0.05%	2.841	2.833	0.01%	62.296	62.646	-0.28%
1	2	5	0.721	0.611	2.20%	0.721	0.612	2.18%	2.440	2.497	-1.14%
1	2	25	1.888	1.457	1.72%	1.888	1.457	1.72%	12.321	12.474	-0.61%
1	2	125	3.321	3.091	0.18%	3.321	3.137	0.15%	62.298	62.569	-0.22%
2	0.5	5	0.580	0.560	0.40%	0.580	0.560	0.41%	2.439	2.498	-1.18%
2	0.5	25	1.559	1.354	0.82%	1.560	1.357	0.81%	12.325	12.496	-0.68%
2	0.5	125	2.909	2.993	-0.07%	2.909	2.999	-0.07%	62.318	62.883	-0.45%
2	1	5	0.635	0.602	0.66%	0.635	0.603	0.65%	2.444	2.498	-1.08%
2	1	25	1.740	1.506	0.94%	1.740	1.508	0.93%	12.324	12.491	-0.67%
2	1	125	3.153	3.182	-0.02%	3.153	3.179	-0.02%	62.299	62.906	-0.49%
2	2	5	0.733	0.655	1.55%	0.733	0.654	1.57%	2.453	2.502	-0.99%
2	2	25	2.083	1.776	1.23%	2.083	1.761	1.29%	12.327	12.614	-1.15%
2	2	125	3.633	3.596	0.03%	3.633	3.481	0.12%	62.302	63.030	-0.58%

Table 4.5: Comparison of mean queue lengths (storage and retrieval requests) and average inventory level in the rack for $\lambda_S = \lambda_R = 1$ and 70% expected utilization (A: Analytical, S: Simulation)

			Utilization			Throughput			SCV
$C_{aS}^2 = C_{aR}^2$	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	0.5	5	0.584	0.647	-9.72%	0.835	0.924	-9.71%	0.452
0.5	0.5	25	0.683	0.689	-0.87%	0.976	0.985	-0.97%	0.543
0.5	0.5	125	0.697	0.697	0.06%	0.996	0.997	-0.07%	0.556
0.5	1	5	0.578	0.642	-10.00%	0.826	0.917	-10.02%	0.567
0.5	1	25	0.682	0.689	-1.03%	0.974	0.985	-1.13%	0.686
0.5	1	125	0.697	0.697	0.04%	0.996	0.997	-0.09%	0.704
0.5	2	5	0.567	0.629	-9.94%	0.890	0.899	-1.00%	0.793
0.5	2	25	0.680	0.689	-1.36%	0.971	0.984	-1.36%	0.971
0.5	2	125	0.697	0.698	-0.14%	0.996	0.997	-0.12%	1.007
1	0.5	5	0.576	0.602	-4.25%	0.823	0.860	-4.24%	0.544
1	0.5	25	0.681	0.679	0.28%	0.973	0.971	0.19%	0.654
1	0.5	125	0.697	0.695	0.29%	0.996	0.994	0.18%	0.667
1	1	5	0.571	0.597	-4.42%	0.815	0.853	-4.39%	0.656
1	1	25	0.678	0.679	-0.22%	0.968	0.971	-0.31%	1.080
1	1	125	0.697	0.695	0.27%	0.996	0.994	0.16%	0.815
1	2	5	0.560	0.585	-4.26%	0.800	0.838	-4.47%	0.878
1	2	25	0.678	0.678	-0.07%	0.968	0.969	-0.11%	1.080
1	2	125	0.697	0.696	0.10%	0.995	0.994	0.13%	1.111
2	0.5	5	0.562	0.541	3.79%	0.802	0.773	3.71%	0.739
2	0.5	25	0.677	0.662	2.22%	0.967	0.946	2.18%	0.884
2	0.5	125	0.696	0.692	0.64%	0.995	0.989	0.57%	0.890
2	1	5	0.557	0.536	3.82%	0.795	0.766	3.83%	0.847
2	1	25	0.676	0.661	2.19%	0.965	0.945	2.11%	1.024
2	1	125	0.696	0.692	0.61%	0.995	0.989	0.55%	1.038
2	2	5	0.547	0.526	4.05%	0.782	0.751	4.07%	1.060
2	2	25	0.673	0.658	2.33%	0.962	0.941	2.25%	1.307
2	2	125	0.696	0.692	0.58%	0.994	0.988	0.62%	1.334

Table 4.6: Comparison of utilization and throughput of the shared server for $\lambda_S = \lambda_R = 1$ and 70% expected utilization (A: Analytical, S: Simulation)

			Storage Queue			Retrieval Queue			Average Inventory		
$C_{aS}^2 = C_{aR}^2$	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	0.5	5	0.688	0.550	2.76%	0.688	0.550	2.77%	2.354	2.499	-2.90%
0.5	0.5	25	1.907	1.122	3.14%	1.907	1.126	3.12%	12.223	12.474	-1.00%
0.5	0.5	125	3.373	2.683	0.55%	3.373	2.639	0.59%	62.190	62.921	-0.59%
0.5	1	5	0.775	0.655	2.41%	0.776	0.655	2.41%	2.363	2.499	-2.72%
0.5	1	25	2.247	1.399	3.39%	2.248	1.402	3.38%	12.228	12.475	-0.99%
0.5	1	125	3.859	2.974	0.71%	3.860	2.937	0.74%	62.213	62.815	-0.48%
0.5	2	5	0.918	0.764	3.08%	0.918	0.764	3.08%	2.379	2.498	-2.38%
0.5	2	25	2.870	1.932	3.75%	2.869	1.942	3.71%	12.228	12.442	-0.85%
0.5	2	125	4.817	3.535	1.03%	4.816	3.513	1.04%	62.206	62.694	-0.39%
1	0.5	5	0.708	0.622	1.72%	0.708	0.622	1.72%	2.362	2.498	-2.72%
1	0.5	25	2.095	1.504	2.37%	2.096	1.502	2.37%	12.226	12.483	-1.03%
1	0.5	125	3.670	3.210	0.37%	3.670	3.273	0.32%	62.195	62.644	-0.36%
1	1	5	0.789	0.690	1.97%	0.789	0.691	1.95%	2.370	2.498	-2.55%
1	1	25	2.419	1.760	2.64%	2.420	1.760	2.64%	12.230	12.489	-1.04%
1	1	125	4.151	3.528	0.50%	4.151	3.586	0.45%	62.214	62.617	-0.32%
1	2	5	0.922	0.767	3.11%	0.923	0.769	3.07%	2.385	2.498	-2.25%
1	2	25	3.015	2.234	3.12%	3.014	2.236	3.11%	12.230	12.494	-1.06%
1	2	125	5.098	4.145	0.76%	5.098	4.187	0.73%	62.206	62.608	-0.32%
2	0.5	5	0.739	0.698	0.82%	0.739	0.699	0.80%	2.377	2.498	-2.42%
2	0.5	25	2.433	2.004	1.72%	2.434	2.007	1.71%	12.233	12.494	-1.05%
2	0.5	125	4.242	3.949	0.23%	4.243	3.958	0.23%	62.217	62.928	-0.57%
2	1	5	0.809	0.741	1.36%	0.809	0.741	1.36%	2.384	2.500	-2.32%
2	1	25	2.732	2.232	2.00%	2.732	2.235	1.99%	12.233	12.498	-1.06%
2	1	125	4.723	4.283	0.35%	4.723	4.285	0.35%	62.206	62.943	-0.59%
2	2	5	0.929	0.790	2.77%	0.929	0.790	2.77%	2.397	2.501	-2.08%
2	2	25	3.287	2.603	2.73%	3.286	2.599	2.75%	12.237	12.506	-1.08%
2	2	125	5.661	5.067	0.48%	5.661	4.994	0.53%	62.207	63.208	-0.80%

Table 4.7: Comparison of mean queue lengths (storage and retrieval requests) and average inventory level in the rack for $\lambda_S = \lambda_R = 1$ and 80% expected utilization (A: Analytical, S: Simulation)

			Utilization			Throughput			SCV
$C_{aS}^2 = C_{aR}^2$	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	0.5	5	0.647	0.733	-11.80%	0.808	0.917	-11.91%	0.499
0.5	0.5	25	0.777	0.788	-1.40%	0.971	0.985	-1.42%	0.605
0.5	0.5	125	0.797	0.797	-0.03%	0.996	0.997	-0.10%	0.620
0.5	1	5	0.637	0.724	-11.99%	0.796	0.905	-12.00%	0.635
0.5	1	25	0.775	0.787	-1.55%	0.968	0.984	-1.61%	0.780
0.5	1	125	0.797	0.797	-0.05%	0.996	0.997	-0.12%	0.803
0.5	2	5	0.621	0.704	-11.78%	0.776	0.880	-11.80%	0.900
0.5	2	25	0.771	0.786	-1.97%	0.963	0.982	-1.96%	1.125
0.5	2	125	0.796	0.797	-0.10%	0.995	0.997	-0.17%	1.166
1	0.5	5	0.639	0.680	-6.10%	0.798	0.850	-6.06%	0.567
1	0.5	25	0.774	0.776	-0.22%	0.968	0.970	-0.22%	0.681
1	0.5	125	0.796	0.795	0.18%	0.996	0.994	0.15%	0.694
1	1	5	0.630	0.671	-6.13%	0.787	0.839	-6.14%	0.700
1	1	25	0.772	0.775	-0.37%	0.965	0.969	-0.40%	0.856
1	1	125	0.796	0.794	0.28%	0.995	0.994	0.13%	0.876
1	2	5	0.615	0.654	-5.96%	0.769	0.819	-6.14%	0.962
1	2	25	0.768	0.773	-0.65%	0.960	0.966	-0.64%	1.199
1	2	125	0.796	0.795	0.10%	0.995	0.993	0.18%	1.239
2	0.5	5	0.623	0.607	2.69%	0.779	0.759	2.61%	0.711
2	0.5	25	0.769	0.754	1.99%	0.961	0.943	1.89%	0.839
2	0.5	125	0.796	0.791	0.58%	0.995	0.989	0.54%	0.844
2	1	5	0.616	0.599	2.84%	0.770	0.749	2.80%	0.840
2	1	25	0.767	0.753	1.85%	0.959	0.942	1.81%	1.012
2	1	125	0.795	0.791	0.56%	0.994	0.989	0.52%	1.025
2	2	5	0.603	0.586	2.92%	0.754	0.733	2.91%	1.093
2	2	25	0.763	0.750	1.73%	0.954	0.936	1.86%	1.355
2	2	125	0.795	0.790	0.63%	0.994	0.987	0.67%	1.388

Table 4.8: Comparison of utilization and throughput of the shared server for $\lambda_S = \lambda_R = 1$ and 80% expected utilization (A: Analytical, S: Simulation)

			Storage Queue			Retrieval Queue			Average Inventory		
$C_{aS}^2 = C_{aR}^2$	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	0.5	5	0.877	0.780	1.94%	0.877	0.781	1.92%	2.301	2.498	-3.95%
0.5	0.5	25	3.364	2.029	5.34%	3.366	2.035	5.33%	12.137	12.484	-1.39%
0.5	0.5	125	6.114	3.833	1.83%	6.116	3.812	1.84%	62.117	62.806	-0.55%
0.5	1	5	0.978	0.887	1.82%	0.978	0.887	1.83%	2.313	2.499	-3.71%
0.5	1	25	3.930	2.607	5.29%	3.931	2.610	5.28%	12.140	12.485	-1.38%
0.5	1	125	7.299	4.591	2.17%	7.300	4.588	2.17%	62.098	62.550	-0.36%
0.5	2	5	1.136	0.971	3.30%	1.136	0.972	3.29%	2.334	2.500	-3.32%
0.5	2	25	4.883	3.518	5.46%	4.882	3.517	5.46%	12.146	12.506	-1.44%
0.5	2	125	9.573	6.107	2.77%	9.572	6.131	2.75%	62.094	62.395	-0.24%
1	0.5	5	0.893	0.803	1.79%	0.893	0.804	1.78%	2.309	2.497	-3.77%
1	0.5	25	3.637	2.600	4.15%	3.638	2.601	4.15%	12.140	12.487	-1.39%
1	0.5	125	6.775	4.976	1.44%	6.776	5.022	1.40%	62.100	62.688	-0.47%
1	1	5	0.987	0.872	2.30%	0.987	0.872	2.30%	2.320	2.498	-3.56%
1	1	25	4.161	3.022	4.56%	4.161	3.022	4.56%	12.143	12.476	-1.33%
1	1	125	7.933	5.721	1.77%	7.934	5.765	1.73%	62.115	62.539	-0.34%
1	2	5	1.137	0.940	3.94%	1.137	0.941	3.91%	2.339	2.497	-3.16%
1	2	25	5.057	3.681	5.50%	5.055	3.684	5.48%	12.149	12.463	-1.26%
1	2	125	10.163	7.239	2.34%	10.162	7.272	2.31%	62.098	62.687	-0.47%
2	0.5	5	0.917	0.853	1.29%	0.918	0.853	1.29%	2.323	2.499	-3.52%
2	0.5	25	4.105	3.226	3.52%	4.106	3.228	3.51%	12.146	12.498	-1.41%
2	0.5	125	8.035	6.658	1.10%	8.036	6.683	1.08%	62.105	62.973	-0.69%
2	1	5	1.001	0.891	2.19%	1.001	0.891	2.19%	2.333	2.499	-3.32%
2	1	25	4.566	3.509	4.23%	4.566	3.514	4.21%	12.149	12.489	-1.36%
2	1	125	9.159	7.435	1.38%	9.159	7.445	1.37%	62.109	63.023	-0.73%
2	2	5	1.137	0.933	4.08%	1.137	0.934	4.06%	2.350	2.499	-2.99%
2	2	25	5.376	3.917	5.84%	5.376	3.918	5.83%	12.156	12.515	-1.43%
2	2	125	11.314	8.818	2.00%	11.312	8.745	2.05%	62.111	63.706	-1.28%

Table 4.9: Comparison of mean queue lengths (storage and retrieval requests) and average inventory level in the rack for $\lambda_S = \lambda_R = 1$ and 90% expected utilization (A: Analytical, S: Simulation)

			Utilization			Throughput			SCV
$C_{aS}^2 = C_{aR}^2$	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	0.5	5	0.700	0.812	-13.85%	0.777	0.903	13.88%	0.541
0.5	0.5	25	0.865	0.885	-2.27%	0.961	0.983	2.27%	0.664
0.5	0.5	125	0.896	0.897	-0.13%	0.995	0.997	0.16%	0.684
0.5	1	5	0.687	0.796	-13.73%	0.763	0.885	13.79%	0.695
0.5	1	25	0.861	0.883	-2.55%	0.956	0.981	2.58%	0.866
0.5	1	125	0.896	0.897	-0.17%	0.995	0.997	0.21%	0.900
0.5	2	5	0.666	0.769	-13.39%	0.740	0.856	13.57%	0.992
0.5	2	25	0.852	0.878	-2.92%	0.947	0.976	2.92%	1.265
0.5	2	125	0.895	0.896	-0.15%	0.994	0.996	0.19%	1.330
1	0.5	5	0.692	0.750	-7.77%	0.769	0.834	7.84%	0.589
1	0.5	25	0.861	0.870	-1.00%	0.957	0.967	1.05%	0.707
1	0.5	125	0.895	0.894	0.15%	0.995	0.994	-0.08%	0.722
1	1	5	0.680	0.738	-7.86%	0.756	0.820	7.82%	0.741
1	1	25	0.857	0.867	-1.14%	0.952	0.964	1.24%	0.909
1	1	125	0.895	0.894	0.10%	0.994	0.993	-0.14%	0.937
1	2	5	0.661	0.716	-7.72%	0.734	0.796	7.80%	1.034
1	2	25	0.850	0.862	-1.45%	0.944	0.958	1.46%	1.308
1	2	125	0.894	0.894	0.01%	0.994	0.993	-0.05%	1.367
2	0.5	5	0.677	0.667	1.54%	0.753	0.741	-1.51%	0.691
2	0.5	25	0.855	0.843	1.39%	0.950	0.936	-1.42%	0.799
2	0.5	125	0.894	0.889	0.60%	0.994	0.988	-0.56%	0.797
2	1	5	0.667	0.657	1.57%	0.741	0.730	-1.57%	0.839
2	1	25	0.851	0.840	1.30%	0.946	0.933	-1.36%	1.001
2	1	125	0.894	0.889	0.55%	0.993	0.988	-0.51%	1.012
2	2	5	0.650	0.641	1.47%	0.723	0.712	-1.45%	1.128
2	2	25	0.844	0.832	1.44%	0.938	0.925	-1.37%	1.400
2	2	125	0.893	0.888	0.57%	0.992	0.986	-0.62%	1.442

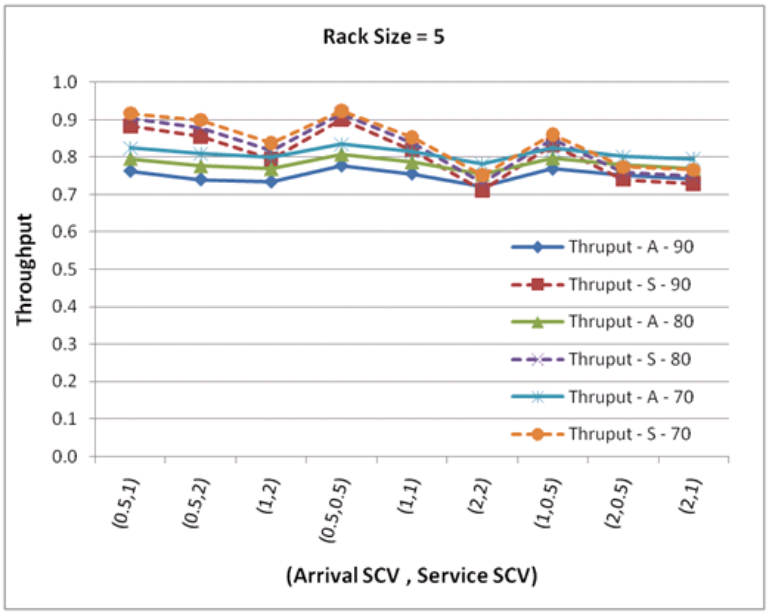
Table 4.10: Comparison of utilization and throughput of the shared server for $\lambda_S = \lambda_R = 1$ and 90% expected utilization (A: Analytical, S: Simulation)

Insights from the Balanced Case

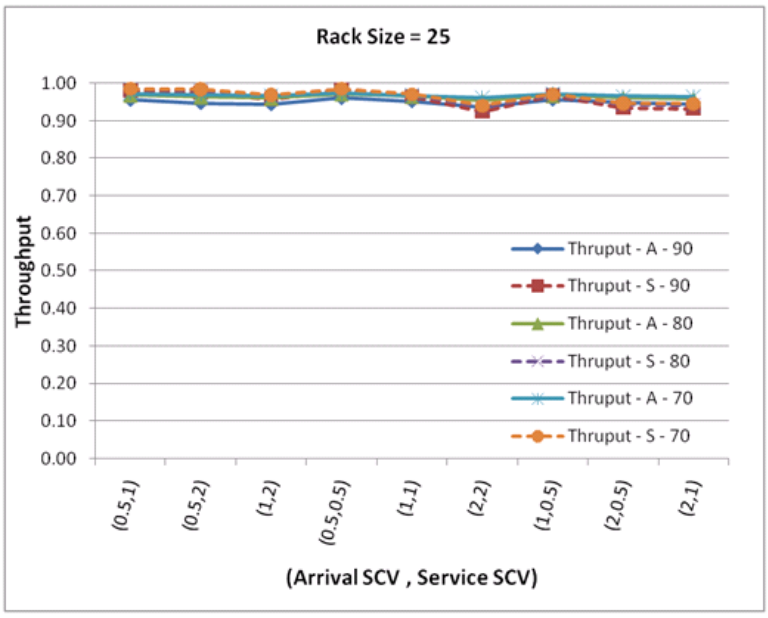
Tables 4.6, 4.8, and 4.10 indicate that the throughput of the shared-server system (defined as the departure rate of the retrieval requests) is an increasing function of the rack size and (trivially) limited by the arrival rate of storage and retrieval requests. As the rack size increases, the number of "lost" requests decreases resulting in an increase in the throughput. A key insight with respect to the throughput of the shared server model is that it is robust to changes in the variability of either the arrival process or the service process for large rack sizes. Figure 4.11 shows the system throughput for the nine variability settings for balanced systems when the rack size is 5, 25 and 125 respectively. We also clearly see that the analytical model tracks the simulation model accurately and that the gap between the two reduces considerably for large rack sizes.

Another observation made is that throughput of the system is higher at lower utilization i.e. system throughput is an increasing function of the service rate especially for small rack sizes. Again this can be attributed to the decrease in percentage of lost requests. From Figure 4.11, we can see that the system throughput at 70% expected utilization is higher than the system throughput at 90% expected utilization for a rack size of 5. As the rack size increases, the difference between the throughputs reduces and is then limited the arrival rate of the storage and retrieval requests. With respect to actual utilization of the shared server, we can draw similar conclusions from Table 4.6, 4.8, and 4.10. We note that the actual utilization is an increasing function of the rack size (number of kanbans) in the system. Figure 4.12 illustrates the actual utilization of the shared server at the three expected utilization levels, 70%, 80% and 90% respectively. We can see that for large rack sizes, the system reaches the expected utilization levels and becomes insensitive to changes in the variability of either the arrival or service process.

With respect to the average inventory level in the rack, we find that it is maintained at almost half the maximum storage size. This is because the arrival rates for the storage and retrieval requests are equal. From tables 4.5, 4.7, and 4.9, we can see that the maximum absolute percentage error for the average inventory in the rack is 3.95% and the average error is 1.31%.

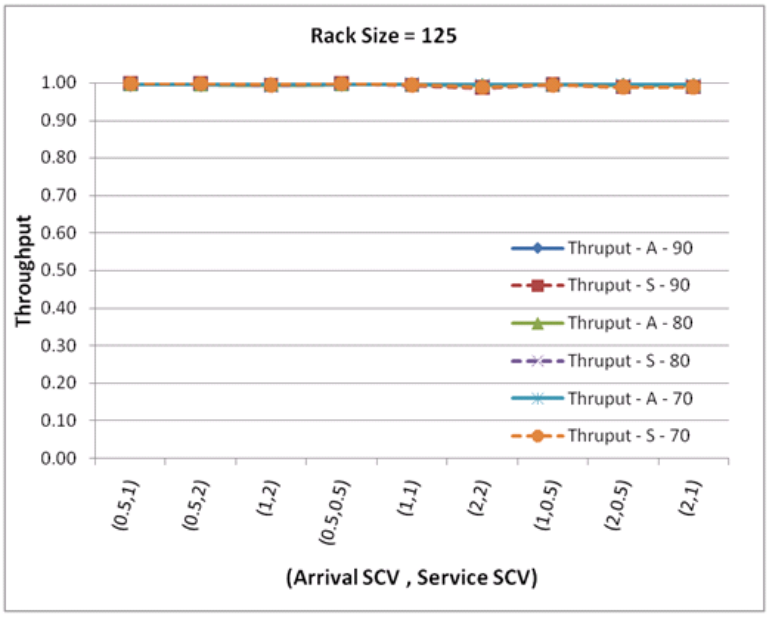


(a) Rack Size = 5



(b) Rack Size = 25

Figure 4.11: Retrieval throughput as a function of system variability in a balanced system

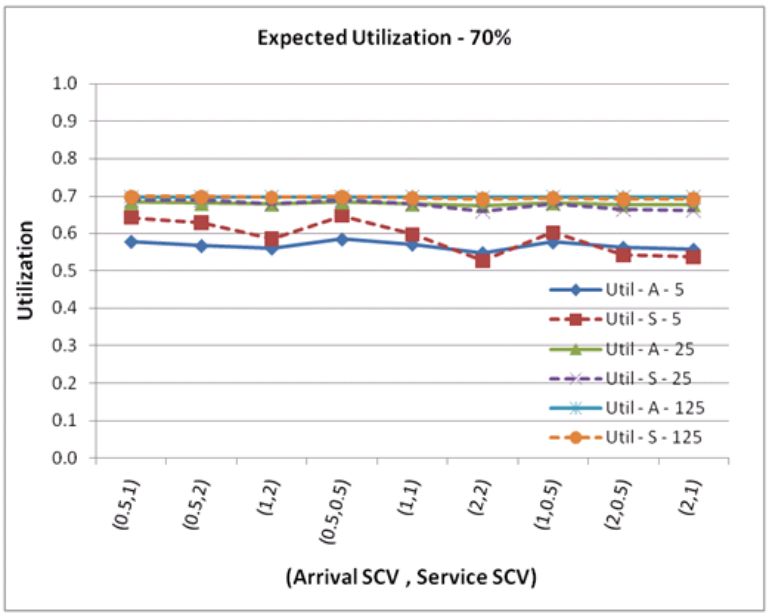


(c) Rack Size = 125

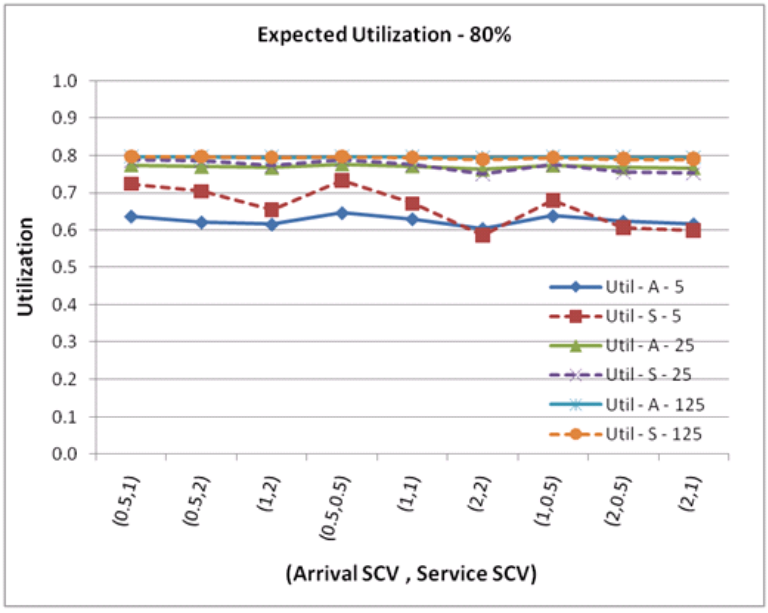
Figure 4.11: Retrieval throughput as a function of system variability in a balanced system (contd)

Tables 4.5, 4.7, and 4.9 also summarize the results of the mean queue length of the storage and retrieval requests. We see that the mean queue length behavior is almost identical for the queues because the rate and SCV of the arrival processes are same. Therefore, it is sufficient to analyze one of them. Figure 4.13 illustrates the mean queue length (storage requests) as a function of system variability for the rack sizes, 5, 25 and 125 respectively. Similar to the system throughput, the mean queue length is an increasing function of the rack size and utilization. The mean queue length is also quite robust to changes in the variability of the service and arrival processes. We also note that the proportional increase in mean queue length is less than the proportional increase in the rack size.

In the next section, we discuss the accuracy of the shared-server model for the unbalanced case and develop insights into its behavior.

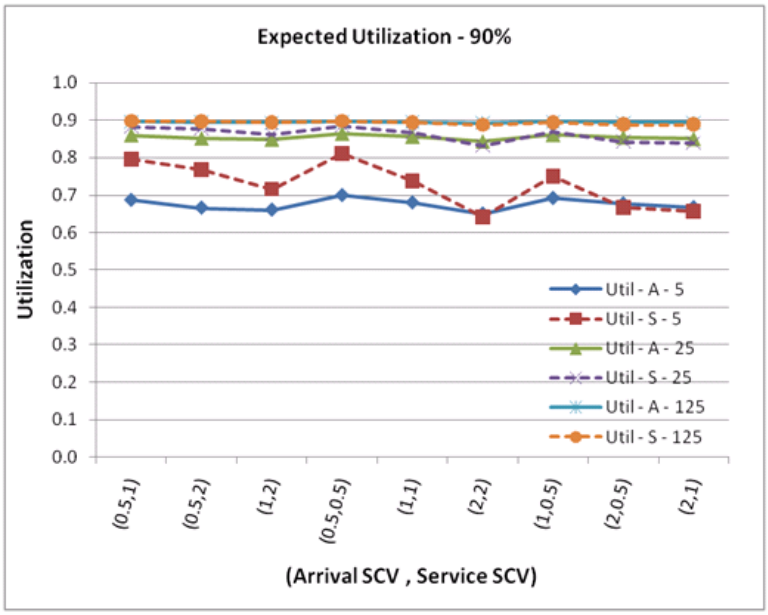


(a) Expected Utilization - 70 percent



(b) Expected Utilization - 80 percent

Figure 4.12: Utilization as a function of system variability in a balanced system

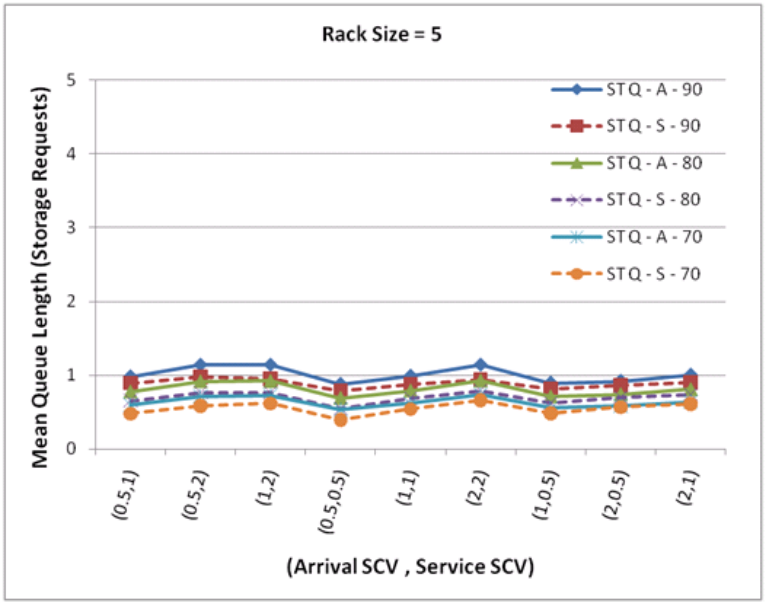


(c) Expected Utilization - 90 percent

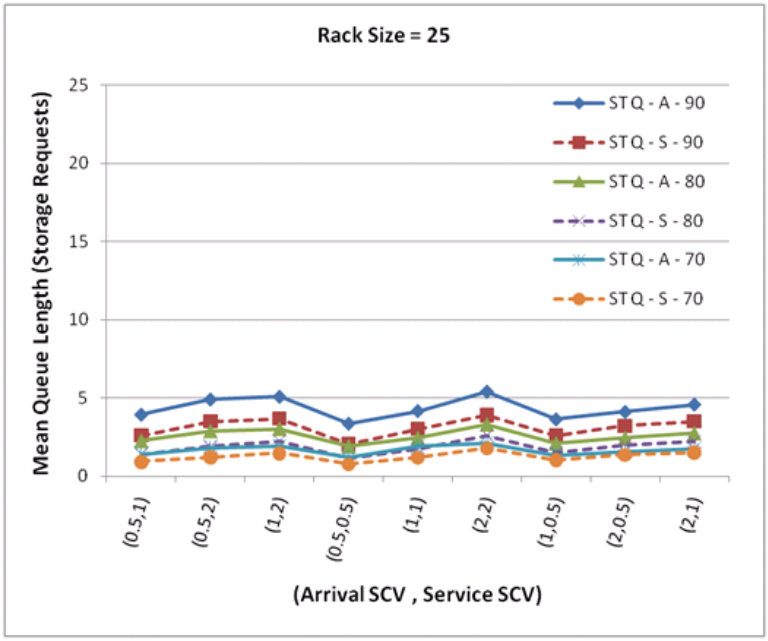
Figure 4.12: Utilization as a function of system variability in a balanced system (contd.)

Unbalanced Case

In the unbalanced case, the storage and retrieval requests have the same arrival rate ($\lambda_S = \lambda_R$) but different variability ($C_{aS}^2 \neq C_{aR}^2$). The estimates of mean queue length (storage and retrieval requests) and the average inventory in the rack at 70%, 80% and 90% expected utilization for the unbalanced case are given in Tables 4.11, 4.13, and 4.15 respectively. The estimates of throughput and utilization of the shared-server are reported in Tables 4.12, 4.14, and 4.16 for the three expected utilization levels. The results from Tables 4.11, 4.13, and 4.15 indicate that the maximum absolute error for the mean queue length of storage (retrieval) request is 6.46% (6.47%). The maximum absolute error for the average inventory in the rack is 8.06%. We note that the above maximum percentage errors are found at 90% expected utilization levels similar to the balanced case. In the case of throughput and actual utilization, the maximum absolute percentage error is 10.83% and 10.82% respectively at 90% utilization levels. As mentioned earlier, these errors are within acceptable ranges for performance evaluation models based on queueing approximations. Next, we develop insights into the behavior of the shared-server model for the unbalanced case.

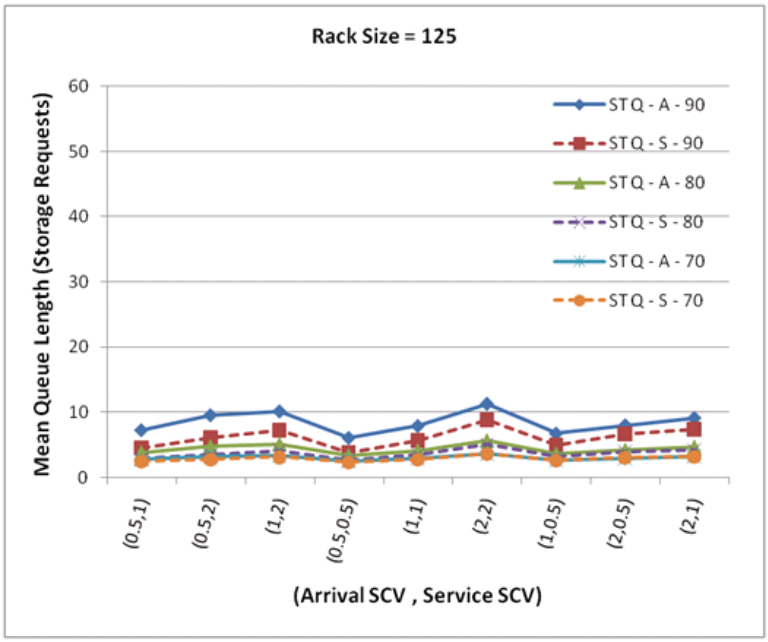


(a) Rack Size = 5



(b) Rack Size = 25

Figure 4.13: Mean queue length (storage requests) as a function of system variability in a balanced system



(c) Rack Size = 125

Figure 4.13: Mean queue length (storage requests) as a function of system variability in a balanced system (contd.)

Insights from the Unbalanced Case

Like the balanced case, Tables 4.12, 4.14, and 4.16 indicate that the throughput of the shared server system is an increasing function of the rack size and is (trivially) limited by the arrival rate of storage and retrieval requests. Tables 4.12, 4.14, and 4.16 indicate that the retrieval throughput of the shared-server system is an increasing function of the rack size but is limited by the arrival rate of storage and retrieval requests. We also note that the throughput of the shared server model is robust to the changes in the variability of either the arrival process or the service process for large rack sizes. Figure 4.14 illustrates the retrieval throughput as a function of the rack size for the three expected utilization levels for the case of the unbalanced system; the SCV of the service time is fixed at a high level of 2. We note that the throughput is insensitive to the differences in the variability of the storage or retrieval request arrival processes, and we can see similar effects to the changes in variability of the service time.

				Storage Queue			Retrieval Queue			Average Inventory		
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	1	0.5	5	0.531	0.442	1.79%	0.562	0.434	2.56%	2.458	2.642	-3.68%
0.5	2	0.5	5	0.542	0.496	0.93%	0.575	0.479	1.91%	2.541	2.702	-3.23%
1	0.5	0.5	5	0.544	0.434	2.19%	0.532	0.442	1.79%	2.382	2.357	0.51%
1	2	0.5	5	0.556	0.527	0.57%	0.576	0.513	1.26%	2.507	2.597	-1.81%
2	0.5	0.5	5	0.574	0.479	1.90%	0.543	0.497	0.91%	2.316	2.293	0.46%
2	1	0.5	5	0.576	0.513	1.26%	0.556	0.527	0.58%	2.357	2.404	-0.94%
0.5	1	1	5	0.599	0.515	1.67%	0.608	0.501	2.15%	2.462	2.653	-3.82%
0.5	2	1	5	0.606	0.564	0.84%	0.632	0.534	1.96%	2.541	2.720	-3.58%
1	0.5	1	5	0.608	0.502	2.12%	0.599	0.516	1.65%	2.390	2.347	0.86%
1	2	1	5	0.616	0.581	0.71%	0.633	0.561	1.44%	2.509	2.604	-1.90%
2	0.5	1	5	0.632	0.534	1.96%	0.606	0.565	0.83%	2.326	2.275	1.03%
2	1	1	5	0.633	0.561	1.43%	0.617	0.581	0.71%	2.365	2.395	-0.59%
0.5	1	2	5	0.714	0.609	2.11%	0.720	0.591	2.59%	2.470	2.653	-3.67%
0.5	2	2	5	0.717	0.648	1.38%	0.733	0.605	2.57%	2.543	2.733	-3.81%
1	0.5	2	5	0.720	0.591	2.59%	0.714	0.609	2.11%	2.404	2.345	1.18%
1	2	2	5	0.723	0.648	1.49%	0.733	0.620	2.26%	2.513	2.611	-1.96%
2	0.5	2	5	0.733	0.604	2.59%	0.717	0.649	1.36%	2.344	2.265	1.58%
2	1	2	5	0.733	0.621	2.24%	0.722	0.649	1.47%	2.380	2.390	-0.20%

(a) Rack size = 5 and utilization = 70%

Table 4.11: Comparison of mean queue length (storage and retrieval requests) and average inventory in the rack at 70% expected utilization in an unbalanced system (A: Analytical, S: Simulation)

				Storage Queue			Retrieval Queue			Average Inventory		
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	1	0.5	25	1.251	0.899	1.41%	1.312	0.923	1.55%	12.501	12.659	-0.63%
0.5	2	0.5	25	1.313	1.046	1.07%	1.484	1.131	1.41%	12.850	12.547	1.21%
1	0.5	0.5	25	1.312	0.925	1.55%	1.251	0.901	1.40%	12.136	12.389	-1.01%
1	2	0.5	25	1.398	1.173	0.90%	1.509	1.222	1.15%	12.671	12.522	0.59%
2	0.5	0.5	25	1.484	1.127	1.43%	1.313	1.046	1.07%	11.791	12.359	-2.27%
2	1	0.5	25	1.509	1.222	1.15%	1.398	1.177	0.88%	11.973	12.420	-1.79%
0.5	1	1	25	1.444	1.053	1.56%	1.501	1.070	1.72%	12.498	12.682	-0.74%
0.5	2	1	25	1.502	1.205	1.19%	1.666	1.279	1.55%	12.840	12.594	0.99%
1	0.5	1	25	1.501	1.072	1.72%	1.444	1.054	1.56%	12.141	12.359	-0.87%
1	2	1	25	1.583	1.325	1.03%	1.689	1.368	1.29%	12.665	12.552	0.45%
2	0.5	1	25	1.666	1.272	1.58%	1.502	1.203	1.20%	11.803	12.304	-2.00%
2	1	1	25	1.689	1.369	1.28%	1.583	1.328	1.02%	11.980	12.397	-1.67%
0.5	1	2	25	1.812	1.341	1.88%	1.864	1.350	2.06%	12.492	12.717	-0.90%
0.5	2	2	25	1.865	1.500	1.46%	2.016	1.542	1.89%	12.824	12.813	0.04%
1	0.5	2	25	1.864	1.351	2.05%	1.812	1.345	1.87%	12.150	12.292	-0.57%
1	2	2	25	1.938	1.612	1.30%	2.037	1.632	1.62%	12.657	12.703	-0.19%
2	0.5	2	25	2.016	1.540	1.90%	1.865	1.504	1.44%	11.823	12.136	-1.25%
2	1	2	25	2.038	1.628	1.64%	1.938	1.603	1.34%	11.991	12.359	-1.47%

(b) Rack size = 25 and utilization = 70%

Table 4.11: Comparison of mean queue length (storage and retrieval requests) and average inventory in the rack at 70% expected utilization in an unbalanced system (contd.)

				Storage Queue			Retrieval Queue			Average Inventory		
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	1	0.5	125	2.518	2.477	0.03%	2.524	2.435	0.07%	63.110	63.147	-0.03%
0.5	2	0.5	125	2.669	2.490	0.14%	2.692	2.867	-0.14%	64.637	60.715	3.14%
1	0.5	0.5	125	2.523	2.428	0.08%	2.517	2.491	0.02%	61.478	61.716	-0.19%
1	2	0.5	125	2.749	2.722	0.02%	2.764	2.988	-0.18%	63.832	60.845	2.39%
2	0.5	0.5	125	2.692	2.740	-0.04%	2.669	2.467	0.16%	59.971	63.454	-2.79%
2	1	0.5	125	2.765	2.834	-0.06%	2.749	2.628	0.10%	60.783	63.401	-2.09%
0.5	1	1	125	2.758	2.632	0.10%	2.764	2.592	0.14%	63.123	63.125	0.00%
0.5	2	1	125	2.910	2.661	0.20%	2.934	3.043	-0.09%	64.604	60.714	3.11%
1	0.5	1	125	2.764	2.589	0.14%	2.758	2.648	0.09%	61.504	61.732	-0.18%
1	2	1	125	2.990	2.899	0.07%	3.007	3.167	-0.13%	63.813	60.856	2.37%
2	0.5	1	125	2.933	2.920	0.01%	2.910	2.641	0.21%	59.997	63.449	-2.76%
2	1	1	125	3.006	3.013	-0.01%	2.989	2.805	0.15%	60.779	63.363	-2.07%
0.5	1	2	125	3.241	2.926	0.25%	3.247	2.896	0.28%	63.092	63.094	0.00%
0.5	2	2	125	3.389	2.949	0.35%	3.418	3.218	0.16%	64.559	60.882	2.94%
1	0.5	2	125	3.246	2.893	0.28%	3.241	2.940	0.24%	61.514	61.802	-0.23%
1	2	2	125	3.468	3.227	0.19%	3.491	3.314	0.14%	63.766	62.876	0.71%
2	0.5	2	125	3.419	3.166	0.20%	3.390	3.099	0.23%	60.047	62.134	-1.67%
2	1	2	125	3.491	3.293	0.16%	3.467	3.239	0.18%	60.822	62.467	-1.32%

(c) Rack size = 125 and utilization = 70%

Table 4.11: Comparison of mean queue length (storage and retrieval requests) and average inventory in the rack at 70% expected utilization in an unbalanced system (contd.)

				Utilization			Throughput			SCV
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	1	0.5	5	0.580	0.624	-7.02%	0.829	0.891	-7.01%	0.525
0.5	2	0.5	5	0.572	0.590	-3.03%	0.817	0.843	-2.99%	0.683
1	0.5	0.5	5	0.580	0.624	-7.02%	0.829	0.891	-7.01%	0.472
1	2	0.5	5	0.569	0.570	-0.25%	0.812	0.814	-0.24%	0.700
2	0.5	0.5	5	0.572	0.590	-3.03%	0.817	0.843	-3.06%	0.518
2	1	0.5	5	0.569	0.570	-0.23%	0.812	0.814	-0.24%	0.588
0.5	1	1	5	0.574	0.619	-7.25%	0.820	0.884	-7.24%	0.638
0.5	2	1	5	0.567	0.584	-3.00%	0.809	0.835	-3.05%	0.791
1	0.5	1	5	0.574	0.619	-7.25%	0.820	0.884	-7.24%	0.587
1	2	1	5	0.563	0.564	-0.14%	0.805	0.807	-0.23%	0.808
2	0.5	1	5	0.567	0.584	-2.98%	0.809	0.835	-3.03%	0.632
2	1	1	5	0.563	0.565	-0.30%	0.805	0.807	-0.30%	0.700
0.5	1	2	5	0.563	0.607	-7.22%	0.805	0.867	-7.23%	0.860
0.5	2	2	5	0.556	0.573	-2.90%	0.795	0.819	-2.94%	1.005
1	0.5	2	5	0.563	0.607	-7.20%	0.805	0.867	-7.22%	0.812
1	2	2	5	0.554	0.553	0.09%	0.791	0.792	-0.13%	1.022
2	0.5	2	5	0.557	0.573	-2.88%	0.795	0.819	-2.93%	0.854
2	1	2	5	0.554	0.555	-0.27%	0.791	0.792	-0.21%	0.919

(a) Rack size = 5 and utilization = 70%

				Utilization			Throughput			SCV
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	1	0.5	25	0.682	0.684	-0.31%	0.974	0.979	-0.44%	0.637
0.5	2	0.5	25	0.680	0.675	0.71%	0.971	0.964	0.71%	0.830
1	0.5	0.5	25	0.682	0.684	-0.31%	0.974	0.979	-0.44%	0.560
1	2	0.5	25	0.679	0.670	1.31%	0.970	0.958	1.24%	0.847
2	0.5	0.5	25	0.680	0.675	0.71%	0.971	0.965	0.62%	0.597
2	1	0.5	25	0.679	0.671	1.16%	0.970	0.959	1.14%	0.691
0.5	1	1	25	0.681	0.684	-0.47%	0.973	0.978	-0.50%	0.780
0.5	2	1	25	0.679	0.674	0.70%	0.970	0.964	0.54%	0.972
1	0.5	1	25	0.681	0.684	-0.47%	0.973	0.978	-0.50%	0.703
1	2	1	25	0.678	0.670	1.15%	0.968	0.958	1.07%	0.989
2	0.5	1	25	0.679	0.675	0.55%	0.970	0.964	0.54%	0.740
2	1	1	25	0.678	0.670	1.15%	0.968	0.958	1.07%	0.833
0.5	1	2	25	0.679	0.683	-0.64%	0.969	0.977	-0.73%	1.064
0.5	2	2	25	0.676	0.674	0.36%	0.966	0.963	0.30%	1.255
1	0.5	2	25	0.679	0.684	-0.79%	0.969	0.977	-0.73%	0.988
1	2	2	25	0.675	0.669	0.96%	0.965	0.955	1.03%	1.271
2	0.5	2	25	0.676	0.674	0.36%	0.966	0.963	0.40%	1.024
2	1	2	25	0.675	0.668	1.11%	0.965	0.955	1.03%	1.117

(b) Rack size = 25 and utilization = 70%

Table 4.12: Comparison of actual utilization and retrieval throughput at 70% expected utilization in an unbalanced system (A: Analytical, S: Simulation)

				Utilization			Throughput			SCV
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	1	0.5	125	0.697	0.696	0.17%	0.996	0.995	0.10%	0.650
0.5	2	0.5	125	0.697	0.695	0.27%	0.996	0.993	0.25%	0.839
1	0.5	0.5	125	0.697	0.696	0.17%	0.996	0.995	0.10%	0.573
1	2	0.5	125	0.697	0.694	0.39%	0.995	0.991	0.43%	0.856
2	0.5	0.5	125	0.697	0.695	0.27%	0.996	0.993	0.25%	0.607
2	1	0.5	125	0.697	0.694	0.39%	0.995	0.992	0.33%	0.702
0.5	1	1	125	0.697	0.696	0.16%	0.996	0.995	0.09%	0.798
0.5	2	1	125	0.697	0.695	0.26%	0.995	0.993	0.24%	0.987
1	0.5	1	125	0.697	0.696	0.16%	0.996	0.995	0.09%	0.721
1	2	1	125	0.697	0.693	0.52%	0.995	0.991	0.41%	1.005
2	0.5	1	125	0.697	0.695	0.26%	0.995	0.993	0.24%	0.755
2	1	1	125	0.697	0.694	0.37%	0.995	0.992	0.31%	0.848
0.5	1	2	125	0.697	0.697	-0.01%	0.996	0.995	0.05%	1.095
0.5	2	2	125	0.697	0.696	0.07%	0.995	0.993	0.20%	1.283
1	0.5	2	125	0.697	0.696	0.13%	0.996	0.995	0.05%	1.017
1	2	2	125	0.696	0.694	0.35%	0.995	0.991	0.38%	1.299
2	0.5	2	125	0.697	0.695	0.22%	0.995	0.992	0.30%	1.052
2	1	2	125	0.696	0.694	0.35%	0.995	0.991	0.38%	1.146

(c) Rack size = 125 and utilization = 70%

Table 4.12: Comparison of actual utilization and retrieval throughput at 70% expected utilization in an unbalanced system (contd.)

				Storage Queue			Retrieval Queue			Average Inventory		
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	1	0.5	5	0.691	0.604	1.73%	0.707	0.587	2.40%	2.395	2.674	-5.59%
0.5	2	0.5	5	0.697	0.663	0.68%	0.740	0.627	2.25%	2.474	2.757	-5.66%
1	0.5	0.5	5	0.707	0.587	2.39%	0.691	0.604	1.74%	2.321	2.325	-0.07%
1	2	0.5	5	0.712	0.675	0.73%	0.739	0.654	1.70%	2.441	2.623	-3.64%
2	0.5	0.5	5	0.739	0.627	2.25%	0.697	0.664	0.66%	2.259	2.238	0.41%
2	1	0.5	5	0.739	0.653	1.72%	0.712	0.675	0.74%	2.298	2.376	-1.55%
0.5	1	1	5	0.776	0.691	1.70%	0.789	0.668	2.41%	2.400	2.680	-5.59%
0.5	2	1	5	0.779	0.738	0.82%	0.812	0.689	2.46%	2.474	2.770	-5.92%
1	0.5	1	5	0.788	0.669	2.39%	0.777	0.691	1.71%	2.333	2.318	0.31%
1	2	1	5	0.789	0.732	1.15%	0.811	0.705	2.12%	2.444	2.629	-3.70%
2	0.5	1	5	0.812	0.689	2.46%	0.779	0.738	0.82%	2.275	2.226	0.98%
2	1	1	5	0.811	0.705	2.11%	0.790	0.733	1.13%	2.311	2.370	-1.17%
0.5	1	2	5	0.917	0.782	2.70%	0.924	0.759	3.30%	2.411	2.671	-5.20%
0.5	2	2	5	0.915	0.816	1.98%	0.935	0.759	3.51%	2.475	2.770	-5.89%
1	0.5	2	5	0.924	0.758	3.32%	0.917	0.783	2.68%	2.354	2.325	0.57%
1	2	2	5	0.920	0.800	2.39%	0.932	0.765	3.35%	2.449	2.625	-3.51%
2	0.5	2	5	0.935	0.758	3.53%	0.915	0.816	1.98%	2.302	2.232	1.39%
2	1	2	5	0.932	0.764	3.37%	0.920	0.799	2.41%	2.333	2.369	-0.72%

(a) Rack size = 5 and utilization = 80%

Table 4.13: Comparison of mean queue length (storage and retrieval requests) and average inventory in the rack at 80% expected utilization in an unbalanced system (A: Analytical, S: Simulation)

				Storage Queue			Retrieval Queue			Average Inventory		
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	1	0.5	25	1.939	1.313	2.50%	2.067	1.337	2.92%	12.434	12.710	-1.10%
0.5	2	0.5	25	2.005	1.567	1.75%	2.364	1.643	2.88%	12.835	12.720	0.46%
1	0.5	0.5	25	2.066	1.334	2.93%	1.939	1.315	2.50%	12.015	12.320	-1.22%
1	2	0.5	25	2.154	1.741	1.65%	2.386	1.785	2.41%	12.628	12.634	-0.02%
2	0.5	0.5	25	2.364	1.639	2.90%	2.006	1.568	1.75%	11.622	12.173	-2.21%
2	1	0.5	25	2.387	1.785	2.41%	2.156	1.746	1.64%	11.831	12.306	-1.90%
0.5	1	1	25	2.276	1.593	2.73%	2.395	1.600	3.18%	12.428	12.777	-1.40%
0.5	2	1	25	2.334	1.845	1.95%	2.669	1.889	3.12%	12.812	12.851	-0.16%
1	0.5	1	25	2.393	1.602	3.17%	2.276	1.595	2.73%	12.029	12.257	-0.91%
1	2	1	25	2.471	1.996	1.90%	2.689	2.025	2.65%	12.615	12.714	-0.40%
2	0.5	1	25	2.668	1.884	3.13%	2.333	1.843	1.96%	11.648	12.055	-1.63%
2	1	1	25	2.688	2.027	2.64%	2.471	1.999	1.89%	11.847	12.240	-1.57%
0.5	1	2	25	2.892	2.110	3.13%	2.994	2.100	3.58%	12.410	12.900	-1.96%
0.5	2	2	25	2.938	2.328	2.44%	3.233	2.315	3.67%	12.767	13.253	-1.94%
1	0.5	2	25	2.994	2.093	3.61%	2.891	2.106	3.14%	12.048	12.122	-0.29%
1	2	2	25	3.056	2.439	2.47%	3.251	2.438	3.25%	12.590	12.881	-1.16%
2	0.5	2	25	3.234	2.336	3.59%	2.938	2.333	2.42%	11.699	11.897	-0.79%
2	1	2	25	3.251	2.421	3.32%	3.056	2.422	2.54%	11.879	12.142	-1.05%

(b) Rack size = 25 and utilization = 80%

Table 4.13: Comparison of mean queue length (storage and retrieval requests) and average inventory in the rack at 80% expected utilization in an unbalanced system (contd.)

				Storage Queue			Retrieval Queue			Average Inventory		
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	1	0.5	125	3.467	2.965	0.40%	3.580	2.942	0.51%	63.085	63.133	-0.04%
0.5	2	0.5	125	3.648	3.151	0.40%	3.986	3.561	0.34%	64.785	60.815	3.18%
1	0.5	0.5	125	3.579	2.935	0.51%	3.465	2.979	0.39%	61.311	61.709	-0.32%
1	2	0.5	125	3.849	3.496	0.28%	4.073	3.780	0.23%	63.904	60.927	2.38%
2	0.5	0.5	125	3.987	3.445	0.43%	3.650	3.128	0.42%	59.626	63.420	-3.03%
2	1	0.5	125	4.073	3.627	0.36%	3.850	3.398	0.36%	60.514	63.329	-2.25%
0.5	1	1	125	3.951	3.273	0.54%	4.063	3.256	0.65%	63.085	63.100	-0.01%
0.5	2	1	125	4.131	3.485	0.52%	4.465	3.888	0.46%	64.736	60.787	3.16%
1	0.5	1	125	4.062	3.251	0.65%	3.952	3.288	0.53%	61.319	61.764	-0.36%
1	2	1	125	4.327	3.824	0.40%	4.550	4.108	0.35%	63.880	60.979	2.32%
2	0.5	1	125	4.463	3.774	0.55%	4.130	3.461	0.54%	59.670	63.394	-2.98%
2	1	1	125	4.550	3.955	0.48%	4.327	3.724	0.48%	60.534	63.315	-2.23%
0.5	1	2	125	4.906	3.854	0.84%	5.012	3.838	0.94%	63.046	63.162	-0.09%
0.5	2	2	125	5.081	4.291	0.63%	5.411	4.254	0.93%	64.674	63.255	1.14%
1	0.5	2	125	5.013	3.829	0.95%	4.905	3.865	0.83%	61.337	61.805	-0.37%
1	2	2	125	5.271	4.428	0.67%	5.493	4.545	0.76%	63.833	62.224	1.29%
2	0.5	2	125	5.412	4.312	0.88%	5.081	4.290	0.63%	59.741	61.473	-1.39%
2	1	2	125	5.495	4.460	0.83%	5.272	4.456	0.65%	60.572	61.636	-0.85%

(c) Rack size = 125 and utilization = 80%

Table 4.13: Comparison of mean queue length (storage and retrieval requests) and average inventory in the rack at 80% expected utilization in an unbalanced system (contd.)

				Utilization			Throughput			SCV
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	1	0.5	5	0.642	0.706	-9.01%	0.803	0.882	-8.94%	0.554
0.5	2	0.5	5	0.634	0.665	-4.65%	0.793	0.831	-4.65%	0.672
1	0.5	0.5	5	0.642	0.706	-9.01%	0.803	0.882	-8.94%	0.513
1	2	0.5	5	0.631	0.641	-1.64%	0.788	0.802	-1.71%	0.683
2	0.5	0.5	5	0.634	0.665	-4.65%	0.793	0.831	-4.65%	0.543
2	1	0.5	5	0.631	0.642	-1.78%	0.788	0.802	-1.71%	0.596
0.5	1	1	5	0.633	0.696	-8.99%	0.792	0.870	-9.02%	0.688
0.5	2	1	5	0.626	0.656	-4.59%	0.782	0.820	-4.56%	0.801
1	0.5	1	5	0.633	0.696	-8.99%	0.792	0.870	-9.02%	0.648
1	2	1	5	0.623	0.633	-1.64%	0.778	0.791	-1.62%	0.813
2	0.5	1	5	0.626	0.656	-4.59%	0.782	0.820	-4.63%	0.678
2	1	1	5	0.623	0.633	-1.64%	0.778	0.792	-1.70%	0.729
0.5	1	2	5	0.618	0.678	-8.85%	0.773	0.848	-8.92%	0.949
0.5	2	2	5	0.612	0.640	-4.44%	0.765	0.799	-4.36%	1.055
1	0.5	2	5	0.618	0.678	-8.85%	0.772	0.848	-8.93%	0.914
1	2	2	5	0.609	0.618	-1.49%	0.761	0.773	-1.60%	1.067
2	0.5	2	5	0.612	0.640	-4.44%	0.765	0.800	-4.44%	0.941
2	1	2	5	0.609	0.618	-1.49%	0.761	0.773	-1.60%	0.989

(a) Rack size = 5 and utilization = 80%

				Utilization			Throughput			SCV
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	1	0.5	25	0.776	0.782	-0.82%	0.970	0.978	-0.82%	0.672
0.5	2	0.5	25	0.773	0.771	0.25%	0.966	0.963	0.28%	0.810
1	0.5	0.5	25	0.776	0.782	-0.82%	0.970	0.978	-0.82%	0.614
1	2	0.5	25	0.772	0.765	0.86%	0.965	0.957	0.79%	0.820
2	0.5	0.5	25	0.773	0.771	0.25%	0.966	0.964	0.18%	0.634
2	1	0.5	25	0.772	0.765	0.86%	0.965	0.957	0.79%	0.701
0.5	1	1	25	0.773	0.781	-0.97%	0.967	0.977	-1.00%	0.847
0.5	2	1	25	0.771	0.769	0.22%	0.963	0.963	0.10%	0.984
1	0.5	1	25	0.773	0.781	-0.97%	0.967	0.977	-1.00%	0.789
1	2	1	25	0.770	0.764	0.72%	0.962	0.955	0.70%	0.993
2	0.5	1	25	0.771	0.770	0.09%	0.963	0.963	0.00%	0.807
2	1	1	25	0.770	0.764	0.72%	0.962	0.955	0.71%	0.874
0.5	1	2	25	0.769	0.780	-1.38%	0.962	0.974	-1.25%	1.191
0.5	2	2	25	0.767	0.767	-0.05%	0.958	0.959	-0.05%	1.327
1	0.5	2	25	0.769	0.779	-1.26%	0.962	0.974	-1.25%	1.133
1	2	2	25	0.765	0.761	0.58%	0.957	0.951	0.66%	1.336
2	0.5	2	25	0.767	0.768	-0.18%	0.958	0.960	-0.15%	1.152
2	1	2	25	0.765	0.760	0.71%	0.957	0.951	0.66%	1.219

(b) Rack size = 25 and utilization = 80%

Table 4.14: Comparison of actual utilization and retrieval throughput at 80% expected utilization in an unbalanced system (A: Analytical, S: Simulation)

				Utilization			Throughput			SCV
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	1	0.5	125	0.797	0.796	0.08%	0.996	0.995	0.08%	0.686
0.5	2	0.5	125	0.796	0.794	0.28%	0.995	0.993	0.23%	0.818
1	0.5	0.5	125	0.797	0.796	0.08%	0.996	0.995	0.08%	0.629
1	2	0.5	125	0.796	0.793	0.38%	0.995	0.991	0.40%	0.827
2	0.5	0.5	125	0.796	0.794	0.28%	0.995	0.993	0.23%	0.646
2	1	0.5	125	0.796	0.793	0.38%	0.995	0.992	0.30%	0.712
0.5	1	1	125	0.796	0.796	0.05%	0.996	0.995	0.05%	0.868
0.5	2	1	125	0.796	0.794	0.25%	0.995	0.993	0.20%	1.000
1	0.5	1	125	0.796	0.796	0.05%	0.996	0.995	0.05%	0.811
1	2	1	125	0.796	0.792	0.48%	0.995	0.991	0.38%	1.008
2	0.5	1	125	0.796	0.794	0.25%	0.995	0.993	0.20%	0.828
2	1	1	125	0.796	0.793	0.35%	0.995	0.992	0.28%	0.893
0.5	1	2	125	0.796	0.796	0.00%	0.995	0.995	0.00%	1.231
0.5	2	2	125	0.796	0.794	0.20%	0.995	0.992	0.25%	1.363
1	0.5	2	125	0.796	0.796	0.00%	0.995	0.995	0.00%	1.174
1	2	2	125	0.795	0.793	0.30%	0.994	0.991	0.32%	1.371
2	0.5	2	125	0.796	0.794	0.20%	0.995	0.992	0.25%	1.191
2	1	2	125	0.795	0.793	0.30%	0.994	0.991	0.32%	1.257

(c) Rack size = 125 and utilization = 80%

Table 4.14: Comparison of actual utilization and retrieval throughput at 80% expected utilization in an unbalanced system (contd.)

				Storage Queue			Retrieval Queue			Average Inventory		
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	1	0.5	5	0.876	0.819	1.14%	0.895	0.787	2.15%	2.338	2.705	-7.33%
0.5	2	0.5	5	0.876	0.874	0.03%	0.926	0.811	2.30%	2.411	2.814	-8.06%
1	0.5	0.5	5	0.894	0.788	2.13%	0.876	0.819	1.15%	2.271	2.293	-0.44%
1	2	0.5	5	0.890	0.849	0.83%	0.923	0.817	2.11%	2.381	2.649	-5.36%
2	0.5	0.5	5	0.926	0.811	2.29%	0.876	0.874	0.03%	2.214	2.182	0.65%
2	1	0.5	5	0.922	0.818	2.09%	0.891	0.849	0.83%	2.251	2.350	-1.98%
0.5	1	1	5	0.976	0.903	1.46%	0.990	0.870	2.39%	2.346	2.701	-7.09%
0.5	2	1	5	0.973	0.939	0.68%	1.010	0.869	2.82%	2.411	2.813	-8.04%
1	0.5	1	5	0.990	0.870	2.39%	0.976	0.903	1.47%	2.288	2.297	-0.19%
1	2	1	5	0.983	0.904	1.57%	1.007	0.867	2.79%	2.385	2.651	-5.33%
2	0.5	1	5	1.010	0.868	2.83%	0.973	0.939	0.68%	2.237	2.183	1.07%
2	1	1	5	1.006	0.867	2.79%	0.983	0.904	1.58%	2.269	2.349	-1.60%
0.5	1	2	5	1.133	0.974	3.18%	1.140	0.947	3.86%	2.361	2.683	-6.45%
0.5	2	2	5	1.127	0.995	2.64%	1.147	0.922	4.50%	2.414	2.796	-7.65%
1	0.5	2	5	1.140	0.948	3.84%	1.133	0.975	3.16%	2.313	2.316	-0.07%
1	2	2	5	1.131	0.961	3.39%	1.143	0.918	4.51%	2.392	2.645	-5.06%
2	0.5	2	5	1.147	0.925	4.45%	1.127	0.998	2.58%	2.271	2.203	1.35%
2	1	2	5	1.144	0.918	4.52%	1.131	0.960	3.41%	2.297	2.353	-1.12%

(a) Rack size = 5 and utilization = 90%

Table 4.15: Comparison of mean queue length (storage and retrieval requests) and average inventory in the rack at 90% expected utilization in an unbalanced system (A: Analytical, S: Simulation)

				Storage Queue			Retrieval Queue			Average Inventory		
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	1	0.5	25	3.376	2.364	4.05%	3.632	2.341	5.17%	12.386	12.970	-2.34%
0.5	2	0.5	25	3.403	2.812	2.36%	4.111	2.783	5.31%	12.844	13.307	-1.85%
1	0.5	0.5	25	3.631	2.340	5.16%	3.378	2.365	4.05%	11.891	12.050	-0.64%
1	2	0.5	25	3.653	2.979	2.69%	4.108	2.964	4.58%	12.601	12.950	-1.40%
2	0.5	0.5	25	4.109	2.785	5.30%	3.404	2.814	2.36%	11.439	11.619	-0.72%
2	1	0.5	25	4.106	2.961	4.58%	3.654	2.973	2.72%	11.686	12.005	-1.28%
0.5	1	1	25	3.936	2.881	4.22%	4.160	2.841	5.28%	12.363	13.081	-2.87%
0.5	2	1	25	3.950	3.243	2.83%	4.578	3.181	5.59%	12.778	13.481	-2.81%
1	0.5	1	25	4.160	2.842	5.27%	3.936	2.879	4.23%	11.920	11.923	-0.01%
1	2	1	25	4.168	3.340	3.31%	4.572	3.309	5.05%	12.559	13.028	-1.88%
2	0.5	1	25	4.577	3.184	5.57%	3.950	3.246	2.82%	11.512	11.445	0.27%
2	1	1	25	4.572	3.306	5.06%	4.168	3.333	3.34%	11.733	11.952	-0.87%
0.5	1	2	25	4.882	3.654	4.91%	5.059	3.615	5.78%	12.329	13.174	-3.38%
0.5	2	2	25	4.881	3.863	4.07%	5.392	3.775	6.47%	12.675	13.706	-4.13%
1	0.5	2	25	5.061	3.612	5.79%	4.880	3.651	4.92%	11.967	11.832	0.54%
1	2	2	25	5.052	3.858	4.78%	5.385	3.814	6.29%	12.497	13.138	-2.57%
2	0.5	2	25	5.394	3.779	6.46%	4.879	3.868	4.04%	11.626	11.274	1.41%
2	1	2	25	5.387	3.809	6.31%	5.050	3.864	4.74%	11.807	11.770	0.15%

(b) Rack size = 25 and utilization = 90%

Table 4.15: Comparison of mean queue length (storage and retrieval requests) and average inventory in the rack at 90% expected utilization in an unbalanced system (contd.)

				Storage Queue			Retrieval Queue			Average Inventory		
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	1	0.5	125	6.223	4.408	1.45%	6.670	4.408	1.81%	63.183	63.037	0.12%
0.5	2	0.5	125	6.440	5.202	0.99%	7.748	5.574	1.74%	65.196	61.132	3.25%
1	0.5	0.5	125	6.670	4.422	1.80%	6.226	4.456	1.42%	61.034	61.679	-0.52%
1	2	0.5	125	6.984	5.778	0.96%	7.844	6.030	1.45%	64.133	61.200	2.35%
2	0.5	0.5	125	7.747	5.460	1.83%	6.441	5.160	1.02%	59.025	63.145	-3.30%
2	1	0.5	125	7.842	5.887	1.56%	6.984	5.666	1.05%	60.082	63.121	-2.43%
0.5	1	1	125	7.403	5.187	1.77%	7.834	5.159	2.14%	63.148	63.095	0.04%
0.5	2	1	125	7.611	5.973	1.31%	8.878	6.283	2.08%	65.120	61.348	3.02%
1	0.5	1	125	7.834	5.195	2.11%	7.404	5.217	1.75%	61.066	61.726	-0.53%
1	2	1	125	8.135	6.535	1.28%	8.971	6.731	1.79%	64.095	61.409	2.15%
2	0.5	1	125	8.878	6.199	2.14%	7.611	5.986	1.30%	59.100	62.907	-3.05%
2	1	1	125	8.971	6.640	1.86%	8.135	6.443	1.35%	60.117	63.125	-2.41%
0.5	1	2	125	9.669	6.714	2.36%	10.070	6.697	2.70%	63.068	63.293	-0.18%
0.5	2	2	125	9.861	7.628	1.79%	11.055	7.723	2.67%	64.964	62.617	1.88%
1	0.5	2	125	10.070	6.733	2.67%	9.667	6.782	2.31%	61.132	61.491	-0.29%
1	2	2	125	10.349	7.984	1.89%	11.140	7.961	2.54%	64.006	63.360	0.52%
2	0.5	2	125	11.055	7.572	2.79%	9.859	7.667	1.75%	59.246	60.253	-0.81%
2	1	2	125	11.141	8.103	2.43%	10.347	8.092	1.80%	60.202	61.479	-1.02%

(c) Rack size = 125 and utilization = 90%

Table 4.15: Comparison of mean queue length (storage and retrieval requests) and average inventory in the rack at 90% expected utilization in an unbalanced system (contd.)

				Utilization			Throughput			SCV
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	1	0.5	5	0.696	0.780	-10.83%	0.773	0.867	-10.82%	0.581
0.5	2	0.5	5	0.688	0.732	-6.07%	0.764	0.814	-6.12%	0.666
1	0.5	0.5	5	0.696	0.780	-10.83%	0.773	0.867	-10.82%	0.550
1	2	0.5	5	0.684	0.706	-3.10%	0.760	0.785	-3.15%	0.673
2	0.5	0.5	5	0.688	0.733	-6.19%	0.764	0.814	-6.18%	0.569
2	1	0.5	5	0.684	0.707	-3.24%	0.760	0.785	-3.15%	0.608
0.5	1	1	5	0.683	0.766	-10.80%	0.759	0.851	-10.79%	0.732
0.5	2	1	5	0.676	0.719	-5.92%	0.752	0.799	-5.99%	0.814
1	0.5	1	5	0.683	0.765	-10.68%	0.759	0.850	-10.72%	0.703
1	2	1	5	0.673	0.695	-3.11%	0.748	0.772	-3.11%	0.821
2	0.5	1	5	0.676	0.720	-6.06%	0.752	0.800	-6.06%	0.722
2	1	1	5	0.673	0.695	-3.11%	0.748	0.772	-3.11%	0.759
0.5	1	2	5	0.663	0.742	-10.61%	0.737	0.824	-10.60%	1.027
0.5	2	2	5	0.658	0.698	-5.76%	0.731	0.777	-5.95%	1.102
1	0.5	2	5	0.663	0.742	-10.61%	0.737	0.824	-10.60%	0.999
1	2	2	5	0.655	0.677	-3.19%	0.728	0.752	-3.15%	1.109
2	0.5	2	5	0.658	0.699	-5.89%	0.731	0.777	-5.95%	1.018
2	1	2	5	0.655	0.676	-3.06%	0.728	0.752	-3.16%	1.053

(a) Rack size = 5 and utilization = 90%

				Utilization			Throughput			SCV
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	1	0.5	25	0.863	0.878	-1.70%	0.959	0.976	-1.70%	0.704
0.5	2	0.5	25	0.860	0.863	-0.41%	0.955	0.960	-0.49%	0.787
1	0.5	0.5	25	0.863	0.878	-1.70%	0.959	0.976	-1.70%	0.666
1	2	0.5	25	0.858	0.856	0.22%	0.953	0.952	0.18%	0.791
2	0.5	0.5	25	0.860	0.864	-0.52%	0.955	0.961	-0.58%	0.673
2	1	0.5	25	0.858	0.857	0.11%	0.953	0.952	0.09%	0.714
0.5	1	1	25	0.859	0.875	-1.85%	0.954	0.973	-1.91%	0.906
0.5	2	1	25	0.856	0.861	-0.64%	0.951	0.957	-0.67%	0.990
1	0.5	1	25	0.859	0.875	-1.85%	0.954	0.973	-1.91%	0.869
1	2	1	25	0.854	0.853	0.11%	0.949	0.949	0.00%	0.993
2	0.5	1	25	0.856	0.861	-0.64%	0.951	0.958	-0.77%	0.875
2	1	1	25	0.854	0.854	-0.01%	0.949	0.949	0.00%	0.916
0.5	1	2	25	0.851	0.870	-2.20%	0.946	0.966	-2.14%	1.305
0.5	2	2	25	0.848	0.855	-0.82%	0.942	0.950	-0.79%	1.389
1	0.5	2	25	0.851	0.870	-2.20%	0.946	0.966	-2.14%	1.267
1	2	2	25	0.847	0.847	-0.04%	0.941	0.941	0.00%	1.392
2	0.5	2	25	0.848	0.855	-0.82%	0.942	0.950	-0.79%	1.273
2	1	2	25	0.847	0.846	0.08%	0.941	0.941	0.00%	1.314

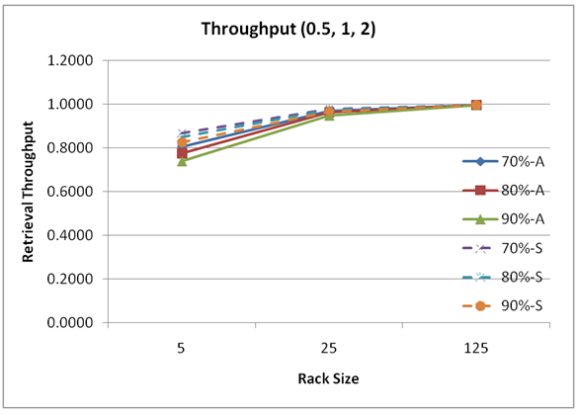
(b) Rack size = 25 and utilization = 90%

Table 4.16: Comparison of actual utilization and retrieval throughput at 90% expected utilization in an unbalanced system (A: Analytical, S: Simulation)

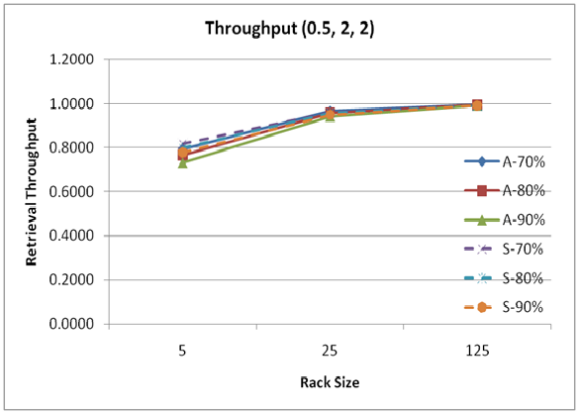
				Utilization			Throughput			SCV
C_{aS}^2	C_{aR}^2	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	1	0.5	125	0.896	0.895	0.07%	0.995	0.995	0.01%	0.719
0.5	2	0.5	125	0.895	0.893	0.24%	0.995	0.993	0.15%	0.790
1	0.5	0.5	125	0.896	0.895	0.07%	0.995	0.995	0.01%	0.687
1	2	0.5	125	0.895	0.891	0.43%	0.994	0.991	0.31%	0.792
2	0.5	0.5	125	0.895	0.893	0.24%	0.995	0.993	0.15%	0.691
2	1	0.5	125	0.895	0.892	0.31%	0.994	0.991	0.31%	0.726
0.5	1	1	125	0.895	0.895	0.02%	0.995	0.995	-0.03%	0.935
0.5	2	1	125	0.895	0.893	0.19%	0.994	0.992	0.21%	1.005
1	0.5	1	125	0.895	0.895	0.02%	0.995	0.995	-0.03%	0.902
1	2	1	125	0.894	0.891	0.38%	0.994	0.991	0.27%	1.007
2	0.5	1	125	0.895	0.893	0.19%	0.994	0.993	0.11%	0.907
2	1	1	125	0.894	0.892	0.27%	0.994	0.991	0.27%	0.942
0.5	1	2	125	0.894	0.895	-0.07%	0.994	0.995	-0.12%	1.364
0.5	2	2	125	0.894	0.893	0.10%	0.993	0.992	0.11%	1.435
1	0.5	2	125	0.894	0.895	-0.07%	0.994	0.995	-0.12%	1.332
1	2	2	125	0.894	0.890	0.40%	0.993	0.990	0.28%	1.437
2	0.5	2	125	0.894	0.893	0.10%	0.993	0.992	0.11%	1.336
2	1	2	125	0.894	0.893	0.07%	0.993	0.991	0.18%	1.371

(c) Rack size = 125 and utilization = 90%

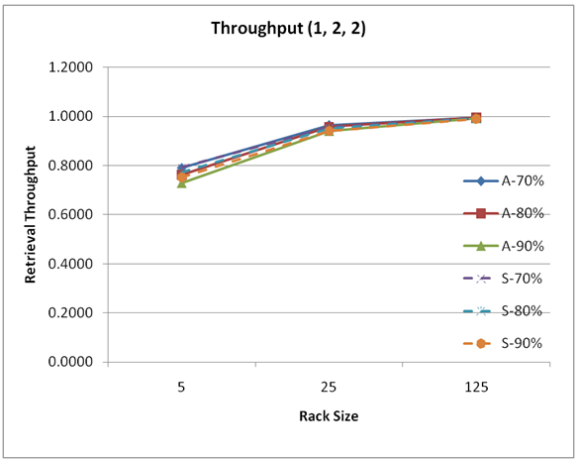
Table 4.16: Comparison of actual utilization and retrieval throughput at 90% expected utilization in an unbalanced system (contd.)



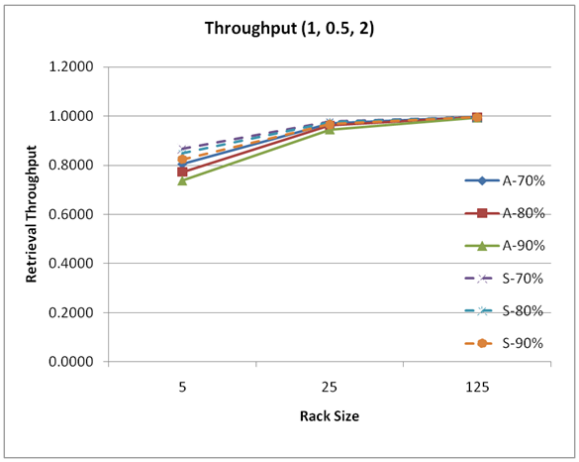
(a) Arrival SCV: Storage = 0.5, Retrieval = 1



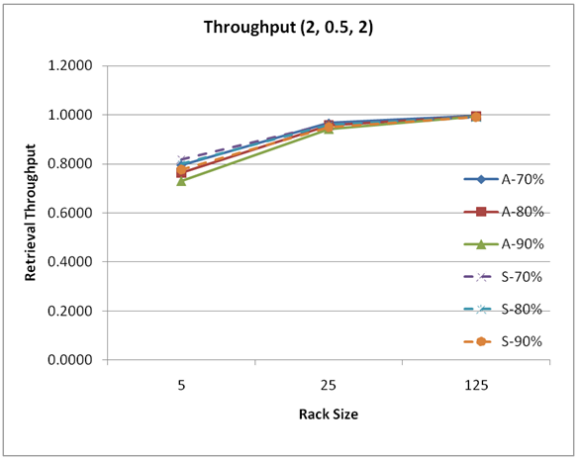
(b) Arrival SCV: Storage = 0.5, Retrieval = 2



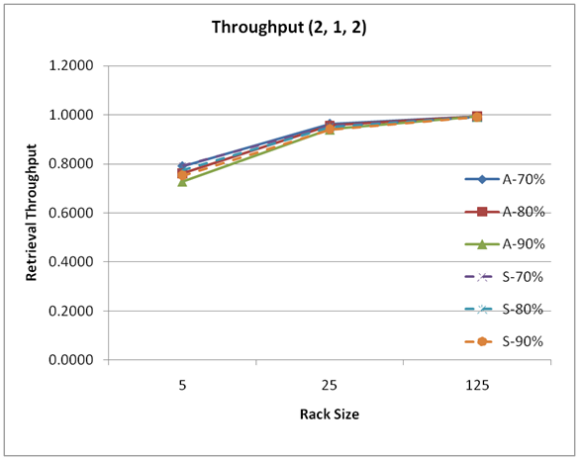
(c) Arrival SCV: Storage =1, Retrieval = 2



(d) Arrival SCV: Storage = 1, Retrieval = 0.5



(e) Arrival SCV: Storage = 2, Retrieval = 0.5



(f) Arrival SCV: Storage = 2, Retrieval = 1

Figure 4.14: Retrieval throughput from a unbalanced shared-server system at 90% expected utilization

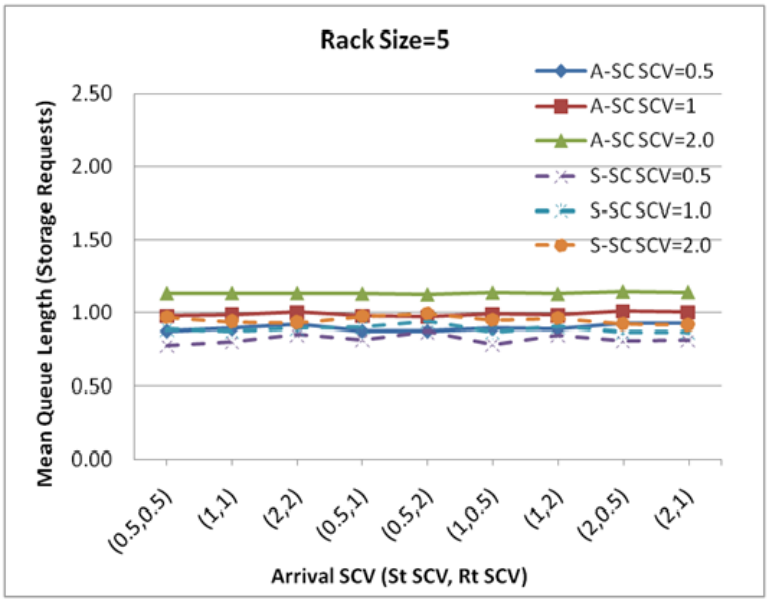
Figures 4.15 and 4.16 illustrate the expected queue length of the storage requests and retrieval requests respectively, at 90% expected utilization level for the three rack sizes. The queue length is represented as a function of variability of the arrival processes for various service time SCVs. We also include results of the balanced system, so that we can understand the effect of difference in arrival SCVs of the storage and retrieval requests. We note that for small rack sizes, the difference in variability seems to have less effect on both the storage and retrieval queue lengths compared to large rack sizes. Also, the mean queue length of the storage requests is greater than that of the retrieval requests when the SCV of the storage request arrival process is greater than the SCV of the retrieval request arrival process and vice-versa. This should be expected as mean queue length is usually an increasing function of arrival process variability (Whitt, 1983).

Figure 4.17 illustrates the average number of items in the rack at 90% expected utilization for the three rack sizes. As in the case of queue length analysis, we include the results of the balanced system as well. We see that average number in the rack is robust to the changes in arrival variability in a balanced system, but is not so in the unbalanced system. When the arrival SCV of storage request is greater than the arrival SCV of the retrieval request, the average number of items in the rack is less than when the arrival SCVs are equal. In addition, as the difference between the SCVs increases, so does the difference in average number of items in the racks. This difference is marked when the rack sizes are large.

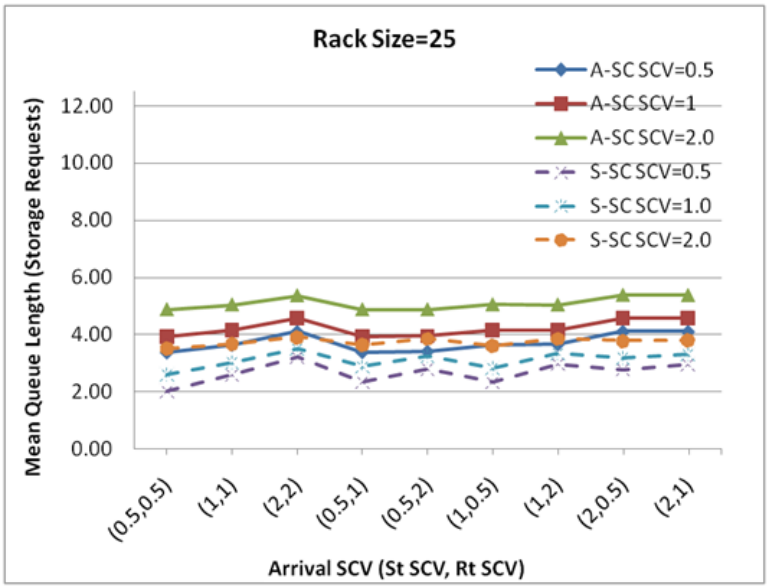
In the following section, we verify the accuracy of the SCV of the departure process of the retrieval requests from the shared-server system.

4.3.8 Departure Process from the Retrieval Processing Station

In the previous sections, we developed two-moment approximations for the retrieval throughput, and queue length performance measures of the shared-server. For the shared-server to become a part of a larger network of warehouse operations, it is imperative that we analyze and verify the departure process of the retrieval requests from the shared-server. The retrieval requests departing from the Retrieval Processing (RP) station form the arrival stream for subsequent replenishment operations. In this section, we will verify that

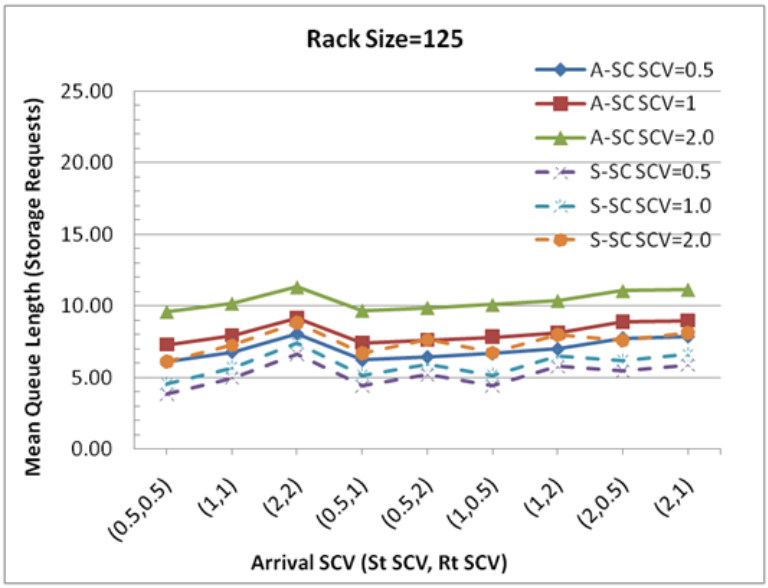


(a) Rack Size = 5



(b) Rack Size = 25

Figure 4.15: Mean queue length of storage requests in an unbalanced shared-server system at 90% expected utilization



(c) Rack Size = 125

Figure 4.15: Mean queue length of storage requests in an unbalanced shared-server system at 90% expected utilization (contd.)

the parameters describing the departure process, especially the SCV, sufficiently represents the arrival process at a downstream queue.

The retrieval requests leave the system after the completion of the retrieval operation at the Retrieval Processing (RP) station . We model the processing station as a $GI/G/1$ queue operating on a FCFS discipline. The arrival process to the RP come from the Retrieval Synchronization Station (J_R), and the service process at RP is suitably modified to represent the single command cycles as described in the earlier section.

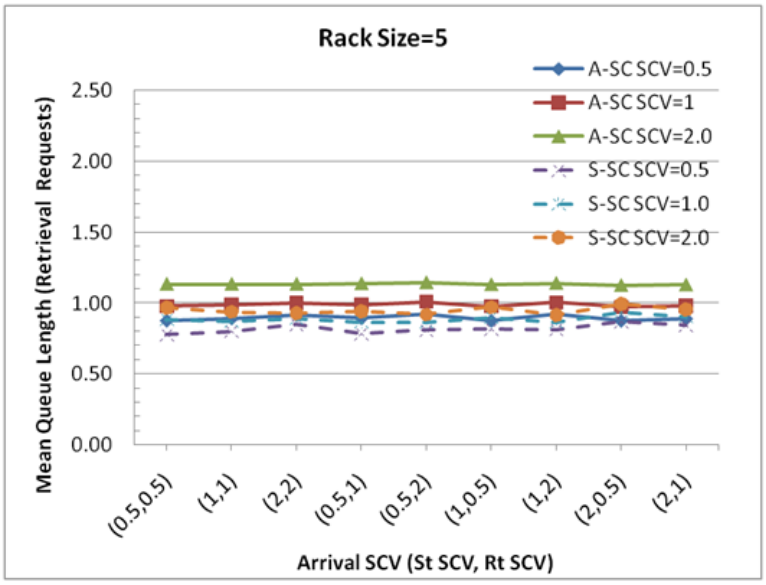
Using the principle of flow conservation, the departure rate from the RP is the effective arrival rate into RP.

$$\lambda_{dR} = \lambda_{aR,j} \tag{4.21}$$

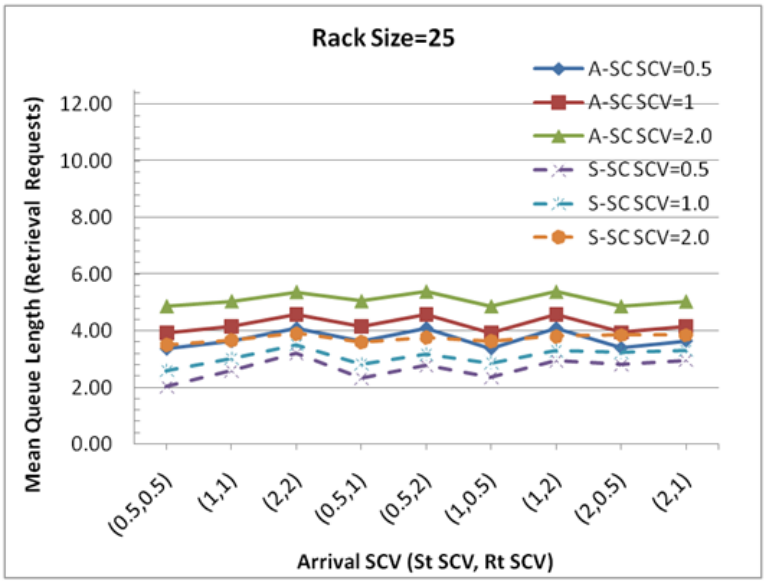
The SCV of departure process of the retrieval requests from the processing station is given by (Whitt, 1983),

$$C_{dR}^2 = \rho_{mSC}^2 C_{aR,j}^2 + (1 - \rho_{mSC}^2) C_{mSC}^2 \tag{4.22}$$

Where $C_{aR,j}^2$ is the SCV of the arrival process into the station, C_{mSC}^2 is the SCV of the

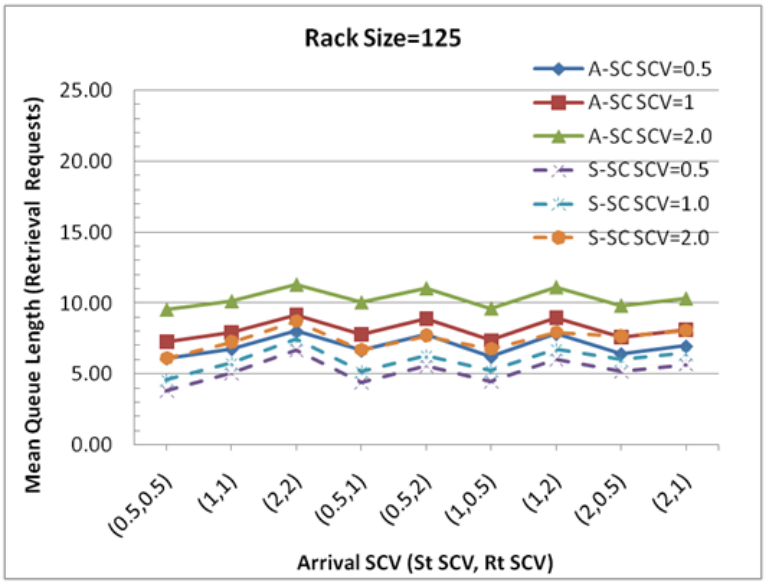


(a) Rack Size = 5



(b) Rack Size = 25

Figure 4.16: Mean queue length of retrieval requests in an unbalanced shared-server system at 90% expected utilization



(c) Rack Size = 125

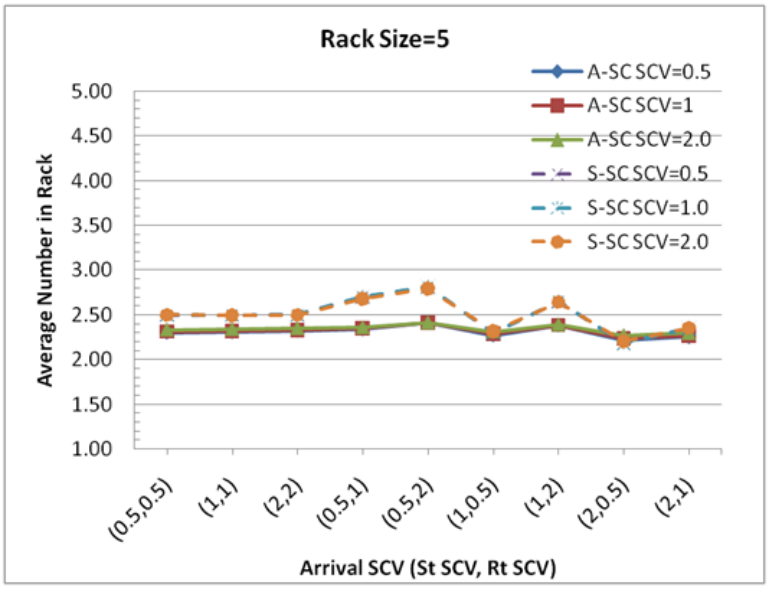
Figure 4.16: Mean queue length of retrieval requests in an unbalanced shared-server system at 90% expected utilization (contd.)

modified service process at the RP, and ρ_{mSC} is the shared-server utilization for the retrieval operation. In this section, we would like to verify that the variability parameter (C^2_{dR}) characterizes the departure process well enough.

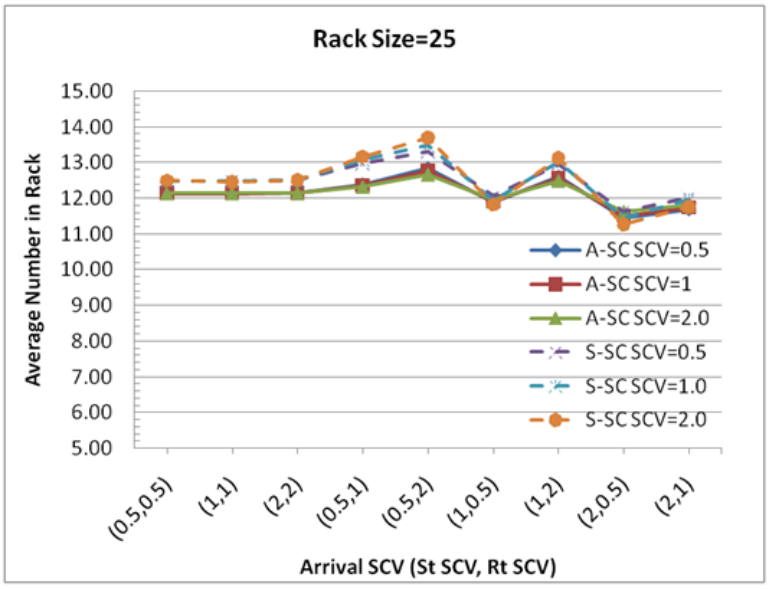
Typically, in determining the variability parameter, the departure process is approximated by a renewal process comprised of a sequence of i.i.d inter-departure times such that the variability of the departure process is the variability of the approximate renewal process. Many times the departure process is not renewal and the inter-departure times are not independent. Whitt (1982) described two approximate methods of determining the variability of the departure process from a queue; asymptotic method and stationary interval method. The stationary interval method ignores the dependency between the successive inter-departure times and assumes that the departure process variability is the SCV of the inter-departure times.

In the asymptotic method, the departure process variability is defined as the limit of the normalized variance of partial sums, given by

$$C^2_{dR} = \lim \frac{Var(S_N)}{N(E[X]^2)} \tag{4.23}$$

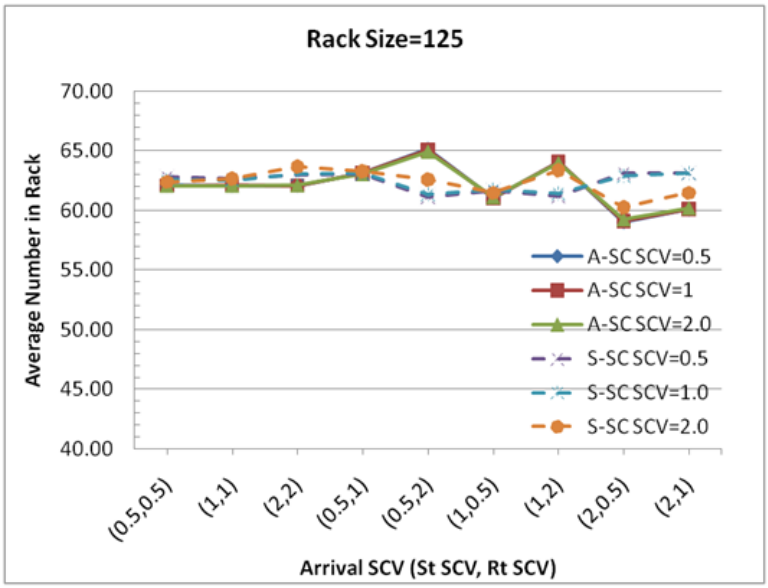


(a) Rack Size = 5



(b) Rack Size = 25

Figure 4.17: Average inventory in rack in an unbalanced shared-server system at 90% expected utilization



(c) Rack Size = 125

Figure 4.17: Average inventory in rack in an unbalanced shared-server system at 90% expected utilization (contd.)

Where $S_N = \sum X_i$ and $X_1, X_2, \ldots, X_N$ are successive inter-departure times in a simulation with run length N. The variance term includes the covariance term, and hence both the methods agree when the departure process is assumed to be renewal.

In this study, we verify the accuracy of the variability parameter by studying the accuracy of the performance measures of a downstream operation, such as a loading operation as shown in Figure 4.18. Table 4.17 summarizes the results for utilization and mean queue length at the loading station with a single server. The expected utilization for the shared-server and the loading server is set at 90%. The simulation statistics were collected for 500,000 entities and averaged for 10 replications. The experiments were conducted for the balanced case; the SCV of the service time distribution at the loading station was set equal to that of the shared-server.

As before, we use relative percentage error in the case of utilization and normalized percentage error in the case of queue length. From Table 4.17, we see that average (maximum) absolute percentage error in utilization of the loading operator is 3.24% (13.97%) and that in queue length is 3.65% (24.92%). We notice that the maximum percentage error occurs when the rack sizes are small (5). Numerical results for the shared-server model had

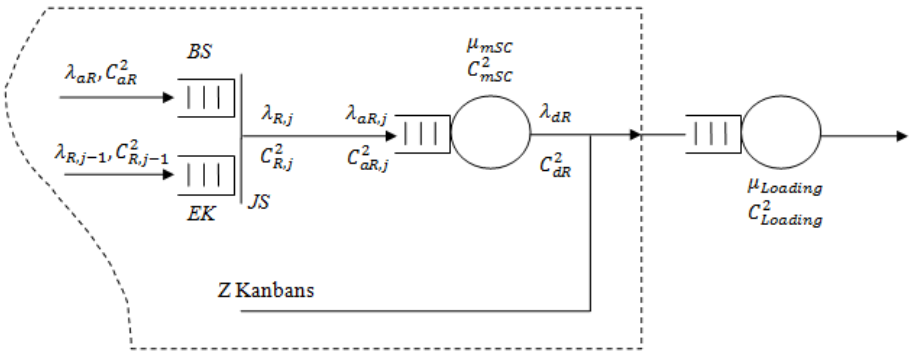


Figure 4.18: Shared-server model with a downstream loading operation

indicated that when the rack size is small the errors in the throughput rates are generally higher. This implies that the errors in the utilization and queue length for the downstream loading operation can be attributed to the error in departure rate (throughput) from the upstream shared-server model, rather than on the variability parameter of the departure process.

4.4 Summary

In this chapter, we presented a detailed description of the shared-server system and two solution approaches to derive the performance measures of the shared-server. The CTMC model can be used for reasonably sized systems under Markovian assumptions. The CQN model relaxes the Markovian assumptions to model general arrivals and general service times. Extensive experimentation confirms that the solution approach based on the parametric-decomposition method works well under a broad range of conditions. We also verified the accuracy of the departure process from the shared-server system, so that it can be used as building block to develop end-to-end performance models of the warehouse.

				Single Shared Server				Loading Operators (m=1)					
Shared Server		Loading		Throughput			SCV	Utilization			Queue Length		
Arrival SCV	Service SCV	Service SCV	Rack Size	A	S	%E	A	A	S	%E	A	S	%E
0.5	0.5	0.5	5	0.777	0.903	-13.88%	0.541	0.699	0.813	-13.97%	0.772	1.343	-11.42%
0.5	0.5	0.5	25	0.961	0.983	-2.27%	0.664	0.865	0.885	-2.28%	3.166	3.099	0.27%
0.5	0.5	0.5	125	0.995	0.997	-0.16%	0.684	0.896	0.897	-0.13%	4.512	3.628	0.71%
0.5	1	1	5	0.763	0.885	-13.79%	0.695	0.687	0.797	-13.85%	1.197	2.045	-16.96%
0.5	1	1	25	0.956	0.981	-2.58%	0.866	0.860	0.884	-2.67%	4.904	5.015	-0.44%
0.5	1	1	125	0.995	0.997	-0.21%	0.900	0.895	0.897	-0.19%	7.247	5.994	1.00%
0.5	2	2	5	0.740	0.856	-13.57%	0.992	0.666	0.769	-13.38%	1.892	3.138	-24.92%
0.5	2	2	25	0.947	0.976	-2.92%	1.265	0.853	0.878	-2.90%	7.943	8.460	-2.07%
0.5	2	2	125	0.994	0.996	-0.19%	1.330	0.895	0.894	0.09%	12.570	10.813	1.41%
1	0.5	0.5	5	0.769	0.834	-7.84%	0.589	0.692	0.751	-7.88%	0.787	1.076	-5.77%
1	0.5	0.5	25	0.957	0.967	-1.05%	0.707	0.861	0.871	-1.10%	3.181	3.501	-1.28%
1	0.5	0.5	125	0.995	0.994	0.08%	0.722	0.895	0.894	0.16%	4.633	4.858	-0.18%
1	1	1	5	0.756	0.820	-7.82%	0.741	0.680	0.738	-7.85%	1.190	1.568	-7.56%
1	1	1	25	0.952	0.964	-1.24%	0.909	0.857	0.868	-1.24%	4.878	5.097	-0.88%
1	1	1	125	0.994	0.993	0.14%	0.937	0.895	0.894	0.11%	7.365	7.255	0.09%
1	2	2	5	0.734	0.796	-7.80%	1.034	0.661	0.717	-7.84%	1.862	2.404	-10.84%
1	2	2	25	0.944	0.958	-1.46%	1.308	0.850	0.862	-1.44%	7.846	7.972	-0.50%
1	2	2	125	0.994	0.993	0.05%	1.367	0.894	0.895	-0.09%	12.640	12.115	0.42%
2	0.5	0.5	5	0.753	0.741	1.51%	0.691	0.677	0.666	1.70%	0.809	0.908	-1.97%
2	0.5	0.5	25	0.950	0.936	1.42%	0.799	0.855	0.842	1.51%	3.229	3.757	-2.11%
2	0.5	0.5	125	0.994	0.988	0.56%	0.797	0.894	0.889	0.61%	4.876	6.810	-1.55%
2	1	1	5	0.741	0.730	1.57%	0.839	0.667	0.657	1.57%	1.187	1.260	-1.46%
2	1	1	25	0.946	0.933	1.36%	1.001	0.851	0.840	1.31%	4.864	5.100	-0.94%
2	1	1	125	0.993	0.988	0.51%	1.012	0.894	0.890	0.45%	7.584	9.225	-1.31%
2	2	2	5	0.723	0.712	1.45%	1.128	0.650	0.641	1.47%	1.813	1.831	-0.36%
2	2	2	25	0.938	0.925	1.37%	1.400	0.844	0.833	1.32%	7.684	7.370	1.26%
2	2	2	125	0.992	0.986	0.62%	1.442	0.893	0.890	0.36%	12.770	13.874	-0.88%

Table 4.17: Verification of SCV of the departure process of the retrieval requests from the shared-server system at 90% expected utilization

Chapter 5

Shared-Server System: Multi-server case

In this chapter we extend the shared-server system to the multi-server case. Each of the m parallel servers represents an S/R machine operating in an aisle (Figure 5.1). The development and analysis of the shared-server system with multiple servers follows the steps developed for the single shared-server system in Chapter 4. We develop a similar closed queueing network model with the processing stations now characterized as multi-server stations. We assume that all the servers are identical. We make use of the approximations developed for GI/G/m queues by Whitt (1993). In the following sections, we describe the modifications made to the service times at the storage and retrieval processing stations, the modifications made to developing the linkage equations connecting the processing stations and synchronization stations, and the overall numerical procedure to solve the queueing network. We conclude the chapter by summarizing the results of the numerical experiments that verify the accuracy of the analytical model for the multi-server case.

5.1 Modifications to the Service Time

The shared-server system is represented by an equivalent queueing network model, and the storage and retrieval processing stations are represented by multi-server nodes. In the single server model, the service time at the storage processing station is modified to account for

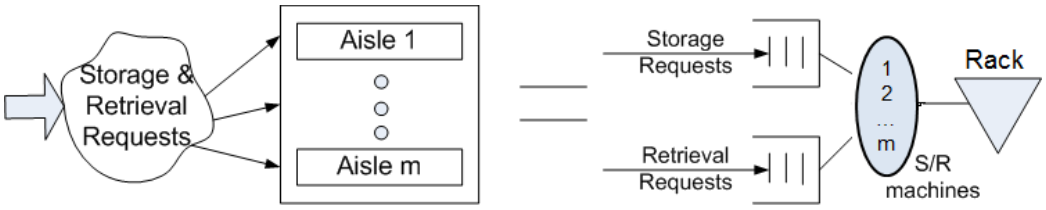


Figure 5.1: Multiple aisles (S/R machines) in the warehouse and shared-server system with multiple servers

the time spent on the retrieval operations and vice-versa. The modification was independent of the number of servers in the system, and is based on the storage/retrieval request arrival rates to the server. The service time modification described for single shared-server system (4.16 - 4.18) can be used in the case of the multi-server system as the total workload will be shared equally by the m servers.

The mean and the variance of the modified storage service time is reproduced here for convenience,

$$\begin{aligned} E[S'] &= E[S] + E[N_R] * E[R] \\ E[N_R] &= \tfrac{p_S}{1-p_S} \\ E[S] &= E[R] = \mu_{SC}^{-1} = \tau_{SC} \\ E[S'] &= \tau_{mSC} = \tfrac{\tau_{SC}}{1-p_S} \end{aligned} \tag{5.1}$$

$$\begin{aligned} Var[S'] &= Var[S] + Var[\textstyle\sum_{N_R} R] \\ Var[S'] &= Var[S] + Var[N_R] * (E[R])^2 + E[N_R] * Var[R] \\ Var[R] &= C_{SC}^2 * \tau_{SC}^2 \\ Var[N_R] &= \tfrac{p_S}{(1-p_S)^2} \\ C_{mSC}^2 &= \tfrac{Var[S']}{\tau_{mSC}^2} \end{aligned} \tag{5.2}$$

The number of customers in the closed loop part of the queueing network is the sum of the number of rack spaces/kanbans plus the number of servers in the system, since we assume that each of the servers can be active independently at the storage and retrieval processing stations at the same time.

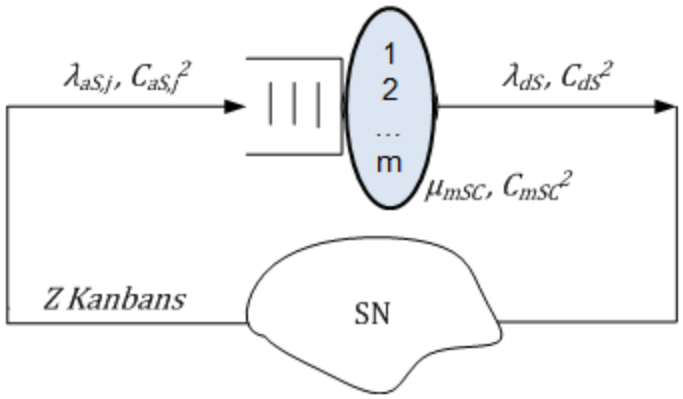


Figure 5.2: The multi-server storage processing station

5.2 Characterization of the Storage Processing Station

In line with the two moment approximation method, we assume that the arrival process $(\lambda_{aS,j}, C_{aS,j}^2)$ to the multi-server node is a renewal process conditioned on the event that the arrival process shuts off when all the customers are at the processing station. Together with the parameters describing the service process, the storage processing station can be described by the 6-tuple $(\lambda_{aS,j}, C_{aS,j}^2, \mu_{mSC}, C_{mSC}^2, m, Z)$. The parameters describing the service times are the modified single command service times, described in the previous section. The characterization step will be complete with the description of the departure process parameters and performance measures of interest.

By flow conservation principle, the mean of the inter-departure times of the storage requests is given by,

$$\lambda_{dS,j}^{-1} = \lambda_{aS,j}^{-1} \tag{5.3}$$

The SCV of the departure process of the storage requests is based on an approximation for a GI/G/m queue (Whitt, 1993). Let $\rho_S = {}^{\lambda_{aS,j} * \tau_{mSC}}/_{m}$ be the utilization of the server at the storage processing station. The SCV of the departure process from the processing station (C_{dS}^2) is then given as,

$$C_{dS}^2 = 1 + \rho_S^2(C_{aS,j}^2 - 1) + \frac{\rho_S^2}{\sqrt{m}}(C_{mSC}^2 - 1) \tag{5.4}$$

To obtain the queue length at the SP station, we first obtain the waiting time in queue

($W_{q,SP}$). The approximations proposed by Whitt (1993) can be used to obtain the waiting time at the processing station.

When $C^2_{aS,j} = C^2_{mSC} \geq 1$

$$W_q(\rho_S, C^2_{aS,j}, C^2_{mSC}, m) = \left(\frac{C^2_{aS,j} + C^2_{mSC}}{2}\right) W_q(M/M/m) \tag{5.5}$$

When $C^2_{aS,j} \neq C^2_{mSC}$

$$W_q(\rho_S, C^2_{aS,j}, C^2_{mSC}, m) = \phi_q(\rho, C^2_{aS,j}, C^2_{mSC}, m) \left(\frac{C^2_{aS,j} + C^2_{mSC}}{2}\right) W_q(M/M/m) \tag{5.6}$$

where

$$\phi(\rho, C^2_a, C^2_S, m) = \begin{cases} \left(\frac{4(C^2_a - C^2_S)}{4C^2_a - 3C^2_S}\right)\phi_1(m,\rho) + \left(\frac{C^2_S}{4C^2_a - 3C^2_S}\right)\psi((C^2_a + C^2_S)/2, m, \rho) & C^2_a \geq C^2_S \\ \left(\frac{C^2_a - C^2_S)}{2C^2_a + 2C^2_S}\right)\phi_3(m,\rho) + \left(\frac{C^2_S + 3C^2_a}{2C^2_a + 2C^2_S}\right)\psi((C^2_a + C^2_S)/2, m, \rho) & C^2_a \leq C^2_S \end{cases} \tag{5.7}$$

$$\psi(m, \rho, C^2) = \begin{cases} 1 & C^2 \geq 1 \\ \phi_4(m,\rho)^{2(1-C^2)} & 0 \leq C^2 < 1 \end{cases} \tag{5.8}$$

$$\gamma(m,p) = min\left\{0.24, (1-\rho)(m-1)\left(\frac{\sqrt{4+5m}-2}{16m\rho}\right)\right\} \tag{5.9}$$

$$\begin{aligned} \phi_1(m,\rho) &= 1 + \gamma(m,\rho) \\ \phi_2(m,\rho) &= 1 - 4\gamma(m,\rho) \\ \phi_3(m,\rho) &= \phi_2(m,\rho) e^{\left(\frac{-2(1-\rho)}{3\rho}\right)} \\ \phi_4(m,\rho) &= min\left\{1, \left(\frac{\phi_1(m,\rho) + \phi_2(m,\rho)}{2}\right)\right\} \end{aligned} \tag{5.10}$$

Then, using Little's law, the number of kanbans in queue at the storage processing station is given by

$$L_{q,SP} = \lambda_{aS,j} * W_{q,SP} \tag{5.11}$$

The reader should note that there are other approximations available for the waiting

Parameter	Levels (values)
Service Times	2 (corresponding to 80% and 90% utilization levels)
SCV of service time distribution	3 (0.5, 1 and 2)
SCV of inter-arrival time distribution	3 (0.5, 1 and 2) and ($C_S^2 = C_R^2$)
Rack size	3 (5, 25, and 125)
Number of servers	1 (3)

Table 5.1: Experimental design for the shared-server system: multi server case

time in system for the $GI/G/m$ queue such as the KLB approximation and also, correction factors to account for the multi-server in a closed system (Suri & Sahu, 2007). In this research, we found that a combination of the approximations developed by Whitt (1993) and Kamath et al. (1988) worked well for most of the test cases.

Algorithms 3.1 and 3.2 are modified to incorporate these changes and then used to solve the queueing network model. The accuracy of the model is tested by comparing the analytical results with the simulation estimates for the performance measures of interest. The models are tested for the configurations shown in Table 5.1.

The performance measures computed are the average number in queue for the storage and retrieval requests, average inventory in the rack, server utilization and the throughput of the retrieval requests from the system. The relative percentage error (RE) is used in the case of utilization and throughput, as before and normalized error (NE) is used to compare the results in the case of number in queue and inventory.

5.3 Accuracy of the Multi-Server Model

The input parameters to the multi-server model are the arrival parameters of the storage and retrieval requests, the rack size, the parameters of the single-command service time, and the number of servers. We study the sensitivity of the multi-server system to the variability in arrivals and service processes. We study the system only under balanced conditions i.e., the arrival rate and variability of the storage requests are the same as that of the retrieval requests. The estimates of the mean queue length (storage and retrieval requests) and average inventory in the rack at 80% and 90% expected utilization are given in Tables 5.2 and 5.4 respectively. The estimates of the throughput and utilization are reported in

Tables 5.3 and 5.5 for the two expected utilization levels. The results from Tables 5.2 and 5.4 indicate that the maximum absolute percentage error for the mean queue length of storage (retrieval) requests is 6.52% (6.52%). The maximum absolute percentage error for the average inventory in the rack is 10.01%. We note that the above errors are found at the 90% expected utilization levels. In the case of throughput and actual utilization, the maximum absolute percentage error is 18.42% and 18.37% at 90% utilization levels, respectively. While the accuracy of the analytical model for the multi-server case is not as good as the single-server case, the error percentages are still within the acceptable error range for a large number of cases examined.

5.4 Summary

In this chapter, we developed a multi server model of the shared-server system representing storage areas with multiple S/R machines or operators. We described the modifications made to the queueing network model of the single shared-server system and the solution procedure to solve the multi server case . Experiments conducted for the balanced configuration indicate that the solution approach works well in most of the cases. In the next chapter, we focus on the development of a queueing-inventory model of the order-picking system.

			Storage Queue			Retrieval Queue			Average Inventory		
$C_{aS}^2 = C_{aR}^2$	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	0.5	5	0.454	0.401	1.06%	0.454	0.401	1.07%	2.143	2.5	-7.15%
0.5	0.5	25	1.528	0.942	2.34%	1.528	0.945	2.33%	11.681	12.5	-3.28%
0.5	0.5	125	2.916	2.499	0.33%	2.917	2.460	0.37%	61.619	62.5	-0.70%
0.5	1	5	0.517	0.476	0.82%	0.518	0.476	0.83%	2.157	2.5	-6.87%
0.5	1	25	1.799	1.157	2.57%	1.801	1.160	2.56%	11.688	12.5	-3.25%
0.5	1	125	3.284	2.731	0.44%	3.285	2.693	0.47%	61.616	62.5	-0.71%
0.5	2	5	0.648	0.553	1.90%	0.649	0.553	1.91%	2.186	2.5	-6.28%
0.5	2	25	2.352	1.550	3.21%	2.352	1.550	3.21%	11.696	12.5	-3.22%
0.5	2	125	4.058	3.144	0.73%	4.058	3.149	0.73%	61.611	62.5	-0.71%
1	0.5	5	0.477	0.468	0.17%	0.477	0.468	0.18%	2.157	2.5	-6.85%
1	0.5	25	1.724	1.289	1.74%	1.724	1.288	1.74%	11.689	12.5	-3.24%
1	0.5	125	3.209	2.993	0.17%	3.209	3.051	0.13%	61.619	62.5	-0.71%
1	1	5	0.540	0.511	0.57%	0.540	0.511	0.58%	2.171	2.5	-6.58%
1	1	25	1.999	1.485	2.06%	2.001	1.484	2.07%	11.695	12.5	-3.22%
1	1	125	3.593	3.240	0.28%	3.594	3.302	0.23%	61.615	62.5	-0.71%
1	2	5	0.665	0.556	2.17%	0.665	0.556	2.18%	2.199	2.5	-6.03%
1	2	25	2.520	1.826	2.78%	2.520	1.823	2.79%	11.703	12.5	-3.19%
1	2	125	4.343	3.707	0.51%	4.343	3.751	0.47%	61.610	62.5	-0.71%
2	0.5	5	0.521	0.556	-0.69%	0.522	0.556	-0.68%	2.186	2.5	-6.29%
2	0.5	25	2.160	1.779	1.52%	2.161	1.776	1.54%	11.706	12.5	-3.18%
2	0.5	125	3.873	3.706	0.13%	3.873	3.676	0.16%	61.616	62.5	-0.71%
2	1	5	0.584	0.575	0.18%	0.585	0.575	0.19%	2.199	2.5	-6.02%
2	1	25	2.412	1.945	1.87%	2.411	1.941	1.88%	11.709	12.5	-3.16%
2	1	125	4.245	3.972	0.22%	4.244	3.951	0.23%	61.614	62.5	-0.71%
2	2	5	0.691	0.595	1.92%	0.691	0.595	1.92%	2.222	2.5	-5.56%
2	2	25	2.837	2.207	2.52%	2.837	2.201	2.54%	11.717	12.5	-3.13%
2	2	125	4.908	4.372	0.43%	4.908	4.542	0.29%	61.609	62.5	-0.71%

Table 5.2: Comparison of mean queue lengths (storage and retrieval requests) and average inventory level in the rack for $\lambda_S = \lambda_R = 1$ and 80% expected utilization (A: Analytical, S: Simulation)

			Utilization			Throughput			SCV
$C_{aS}^2 = C_{aR}^2$	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	0.5	5	0.619	0.737	-15.97%	0.774	0.922	-16.01%	0.546
0.5	0.5	25	0.773	0.788	-1.90%	0.966	0.985	-1.93%	0.677
0.5	0.5	125	0.797	0.797	-0.05%	0.996	0.997	-0.12%	0.698
0.5	1	5	0.615	0.731	-15.92%	0.768	0.913	-15.88%	0.623
0.5	1	25	0.771	0.787	-1.98%	0.964	0.984	-2.04%	0.777
0.5	1	125	0.797	0.797	-0.06%	0.996	0.997	-0.14%	0.803
0.5	2	5	0.605	0.721	-16.13%	0.756	0.901	-16.10%	0.773
0.5	2	25	0.768	0.787	-2.40%	0.960	0.983	-2.36%	0.975
0.5	2	125	0.796	0.798	-0.23%	0.995	0.997	-0.17%	1.012
1	0.5	5	0.614	0.686	-10.44%	0.768	0.857	-10.39%	0.609
1	0.5	25	0.770	0.776	-0.72%	0.963	0.971	-0.81%	0.753
1	0.5	125	0.796	0.795	0.15%	0.995	0.994	0.13%	0.772
1	1	5	0.610	0.680	-10.34%	0.762	0.850	-10.37%	0.685
1	1	25	0.769	0.775	-0.80%	0.961	0.970	-0.92%	0.853
1	1	125	0.796	0.794	0.26%	0.995	0.994	0.11%	0.876
1	2	5	0.601	0.673	-10.77%	0.751	0.841	-10.75%	0.833
1	2	25	0.766	0.775	-1.21%	0.957	0.968	-1.14%	1.050
1	2	125	0.796	0.796	-0.03%	0.995	0.994	0.08%	1.086
2	0.5	5	0.605	0.615	-1.64%	0.756	0.768	-1.56%	0.737
2	0.5	25	0.765	0.754	1.46%	0.956	0.943	1.37%	0.909
2	0.5	125	0.795	0.790	0.68%	0.994	0.988	0.61%	0.920
2	1	5	0.600	0.610	-1.57%	0.751	0.763	-1.61%	0.810
2	1	25	0.764	0.753	1.41%	0.954	0.942	1.36%	1.007
2	1	125	0.795	0.790	0.67%	0.994	0.988	0.60%	1.025
2	2	5	0.593	0.606	-2.19%	0.741	0.756	-2.05%	0.955
2	2	25	0.761	0.752	1.18%	0.951	0.939	1.29%	1.205
2	2	125	0.795	0.792	0.38%	0.994	0.989	0.47%	1.234

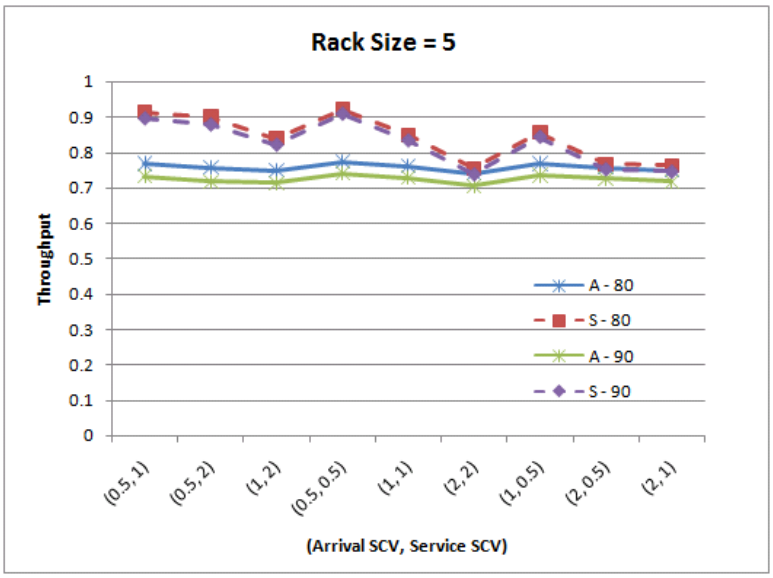
Table 5.3: Comparison of utilization and throughput of the multi - shared server for $\lambda_S = \lambda_R = 1$ and 80% expected utilization (A: Analytical, S: Simulation)

			Storage Queue			Retrieval Queue			Average Inventory		
$C_{aS}^2 = C_{aR}^2$	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A	S	%E
0.5	0.5	5	0.608	0.628	-0.40%	0.609	0.628	-0.39%	2.000	2.500	-10.01%
0.5	0.5	25	3.022	1.818	4.82%	3.024	1.821	4.81%	11.425	12.500	-4.30%
0.5	0.5	125	5.693	3.622	1.66%	5.694	3.592	1.68%	61.316	62.500	-0.95%
0.5	1	5	0.694	0.711	-0.35%	0.694	0.711	-0.34%	2.022	2.500	-9.56%
0.5	1	25	3.549	2.331	4.87%	3.550	2.332	4.87%	11.436	12.500	-4.26%
0.5	1	125	6.727	4.287	1.95%	6.727	4.294	1.95%	61.315	62.500	-0.95%
0.5	2	5	0.858	0.780	1.57%	0.859	0.781	1.55%	2.065	2.500	-8.70%
0.5	2	25	4.512	3.097	5.66%	4.511	3.098	5.65%	11.457	12.500	-4.17%
0.5	2	125	8.798	5.625	2.54%	8.797	5.647	2.52%	61.315	62.500	-0.95%
1	0.5	5	0.631	0.646	-0.31%	0.631	0.647	-0.32%	2.014	2.500	-9.72%
1	0.5	25	3.331	2.360	3.88%	3.332	2.357	3.90%	11.435	12.500	-4.26%
1	0.5	125	6.383	4.658	1.38%	6.384	4.828	1.24%	61.317	62.500	-0.95%
1	1	5	0.715	0.693	0.43%	0.715	0.694	0.42%	2.036	2.500	-9.29%
1	1	25	3.840	2.732	4.43%	3.841	2.732	4.43%	11.446	12.500	-4.22%
1	1	125	7.427	5.386	1.63%	7.427	5.439	1.59%	61.316	62.500	-0.95%
1	2	5	0.870	0.739	2.63%	0.871	0.738	2.65%	2.076	2.500	-8.48%
1	2	25	4.737	3.281	5.82%	4.736	3.282	5.81%	11.465	12.500	-4.14%
1	2	125	9.445	6.691	2.20%	9.444	6.728	2.17%	61.316	62.500	-0.95%
2	0.5	5	0.674	0.706	-0.65%	0.674	0.706	-0.64%	2.042	2.500	-9.16%
2	0.5	25	3.952	2.989	3.85%	3.953	2.985	3.87%	11.456	12.500	-4.18%
2	0.5	125	7.831	6.406	1.14%	7.832	6.265	1.25%	61.319	62.500	-0.94%
2	1	5	0.755	0.722	0.66%	0.756	0.722	0.67%	2.062	2.500	-8.76%
2	1	25	4.399	3.214	4.74%	4.401	3.211	4.76%	11.466	12.500	-4.13%
2	1	125	8.836	7.060	1.42%	8.836	7.003	1.47%	61.319	62.500	-0.94%
2	2	5	0.892	0.737	3.10%	0.893	0.737	3.11%	2.097	2.500	-8.06%
2	2	25	5.151	3.521	6.52%	5.149	3.518	6.52%	11.482	12.500	-4.07%
2	2	125	10.711	8.393	1.85%	10.709	8.352	1.89%	61.318	62.500	-0.95%

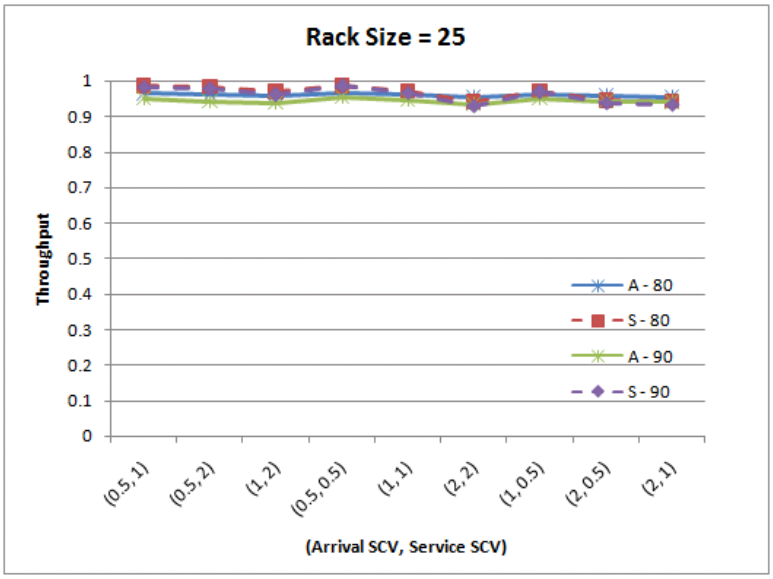
Table 5.4: Comparison of mean queue lengths (storage and retrieval requests) and average inventory level in the rack for $\lambda_S = \lambda_R = 1$ and 90% expected utilization (A: Analytical, S: Simulation)

			Utilization			Throughput			SCV
$C_{aS}^2 = C_{aR}^2$	C_{SC}^2	Rack Size	A	S	%E	A	S	%E	A
0.5	0.5	5	0.667	0.817	-18.37%	0.741	0.908	-18.42%	0.593
0.5	0.5	25	0.859	0.885	-2.95%	0.954	0.984	-3.03%	0.747
0.5	0.5	125	0.896	0.897	-0.16%	0.995	0.997	-0.19%	0.775
0.5	1	5	0.660	0.806	-18.18%	0.733	0.895	-18.15%	0.678
0.5	1	25	0.855	0.884	-3.27%	0.950	0.982	-3.28%	0.862
0.5	1	125	0.895	0.897	-0.19%	0.995	0.997	-0.23%	0.900
0.5	2	5	0.645	0.790	-18.35%	0.717	0.878	-18.37%	0.844
0.5	2	25	0.848	0.881	-3.79%	0.942	0.978	-3.65%	1.088
0.5	2	125	0.895	0.898	-0.38%	0.994	0.997	-0.30%	1.148
1	0.5	5	0.662	0.759	-12.78%	0.736	0.843	-12.76%	0.639
1	0.5	25	0.855	0.871	-1.79%	0.950	0.968	-1.82%	0.790
1	0.5	125	0.895	0.894	0.11%	0.995	0.994	0.05%	0.813
1	1	5	0.655	0.750	-12.68%	0.728	0.833	-12.68%	0.723
1	1	25	0.852	0.868	-1.89%	0.946	0.965	-1.96%	0.905
1	1	125	0.895	0.894	0.08%	0.994	0.993	0.11%	0.937
1	2	5	0.641	0.739	-13.21%	0.713	0.821	-13.21%	0.887
1	2	25	0.845	0.865	-2.36%	0.938	0.960	-2.22%	1.132
1	2	125	0.894	0.894	0.01%	0.993	0.993	0.04%	1.185
2	0.5	5	0.653	0.677	-3.57%	0.725	0.752	-3.54%	0.731
2	0.5	25	0.848	0.843	0.63%	0.943	0.937	0.58%	0.882
2	0.5	125	0.894	0.889	0.55%	0.993	0.987	0.62%	0.888
2	1	5	0.646	0.670	-3.57%	0.718	0.745	-3.66%	0.813
2	1	25	0.845	0.840	0.60%	0.939	0.934	0.56%	0.998
2	1	125	0.894	0.889	0.52%	0.993	0.988	0.48%	1.012
2	2	5	0.634	0.664	-4.46%	0.705	0.737	-4.42%	0.976
2	2	25	0.839	0.835	0.48%	0.932	0.929	0.40%	1.225
2	2	125	0.893	0.890	0.34%	0.992	0.987	0.51%	1.260

Table 5.5: Comparison of utilization and throughput of the shared server for $\lambda_S = \lambda_R = 1$ and 90% expected utilization (A: Analytical, S: Simulation)

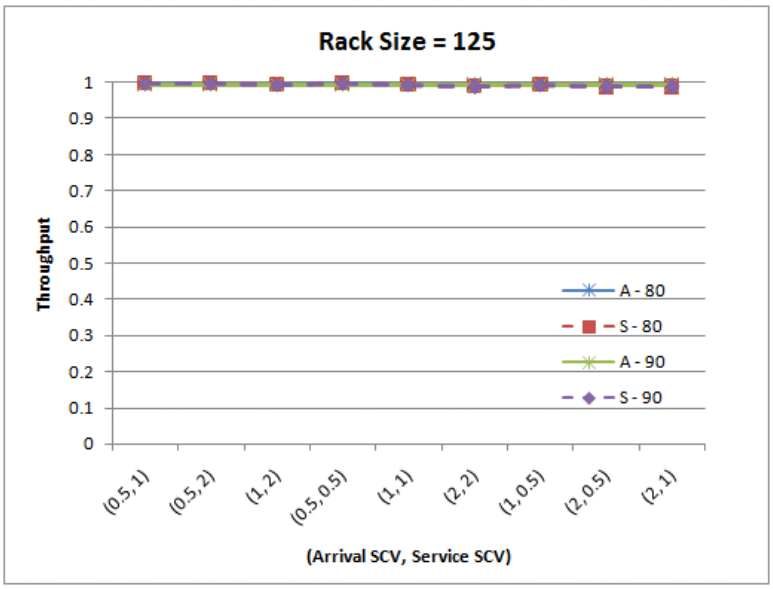


(a) Rack Size = 5



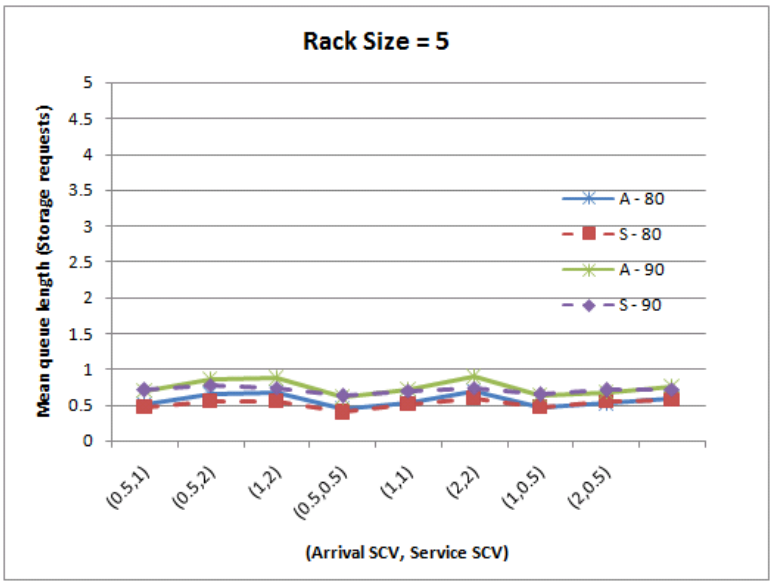
(b) Rack Size = 25

Figure 5.3: Retrieval throughput as a function of system variability in a balanced shared-server system with multiple servers

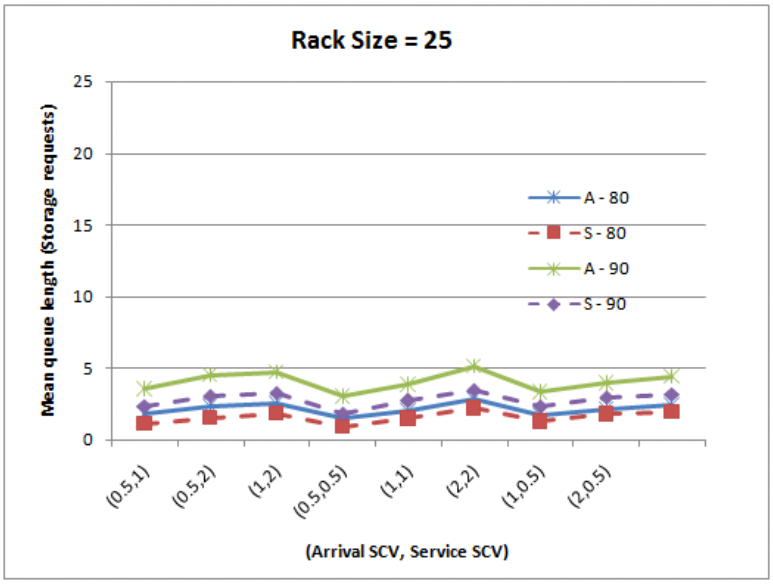


(c) Rack Size = 125

Figure 5.3: Retrieval throughput as a function of system variability in a balanced shared-server system with multiple servers (contd.)

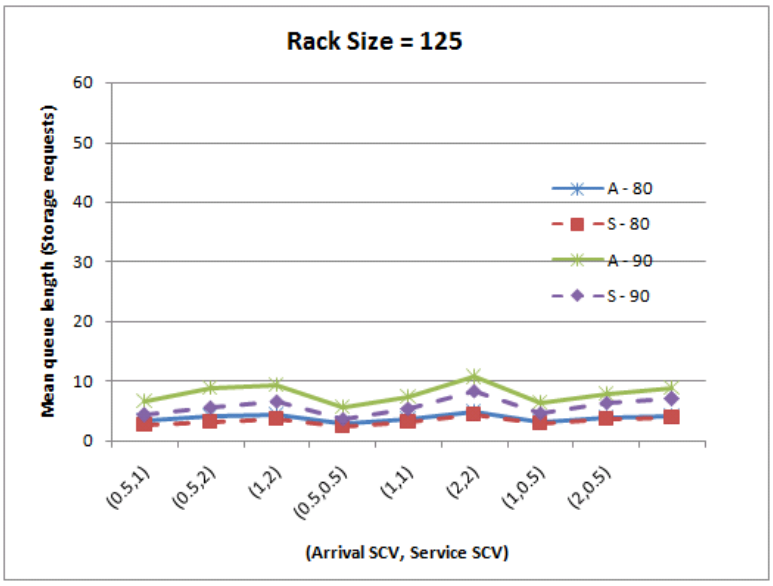


(a) Rack Size = 5



(b) Rack Size = 25

Figure 5.4: Mean queue length (storage requests) as a function of system variability in a balanced shared-server system with multiple servers



(c) Rack Size = 125

Figure 5.4: Mean queue length (storage requests) as a function of system variability in a balanced shared-server system with multiple servers (contd.)

Chapter 6

Order-Picking System

Unit-load is one that can be stored or moved as a single entity at one time, such as a pallet, container or tote, regardless of the number of individual items that make up the load (Tompkins et al., 2003). The unit load can range from a single part to a carton, to pallet of cases, to a container consisting of pallets moved by rails and ships. In the simplest form of a warehouse, the configuration of unit-load remains the same (pallet-in/pallet-out). Not all warehouses can be that simple, and picking in different composition is essential. Order-picking is the process of removing the items from storage to meet a specific customer demand and represents a basic function of the warehouse. Order-picking typically happens at the forward storage area in a warehouse.

The configuration of the unit-load is maintained between two moves/shipment points serviced by a material handling/movement device, but can differ between consecutive moves in and out of inventory as illustrated in the Figure 6.1 because of order-picking. In this chapter, we focus on the development of a queueing-inventory model that can handle such changes in the configuration of the unit-load.

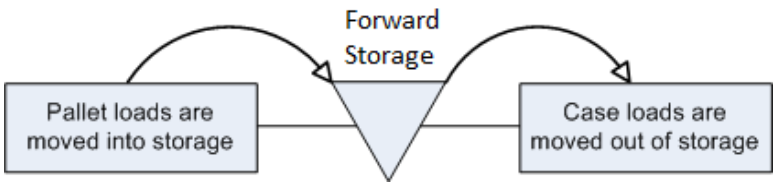


Figure 6.1: Changing unit-load configuration

6.1 Description of the Order-Picking Model

In this research, we assume that configuration of the items that is stored is larger than that is retrieved from the store. Let us assume that pallet loads are moved into the storage and that customer orders are retrieved in case units. Upon receiving a customer order, individual cases are picked from the forward store. When all the items from a particular pallet are picked, the pallet is replaced with another from the upstream reserve storage area. A queueing-inventory (QI) model of such a system is shown in Figure 6.2.

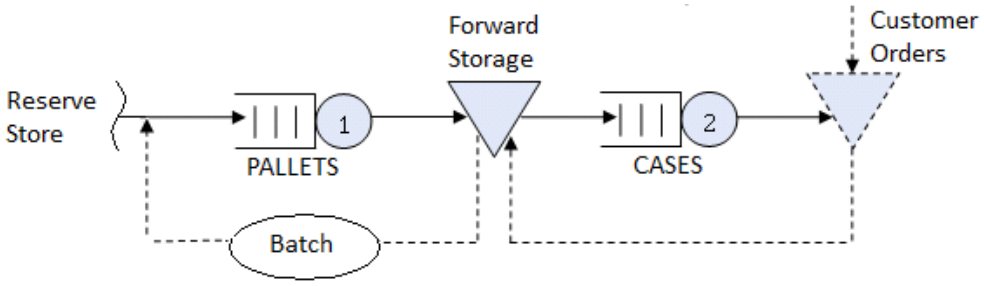


Figure 6.2: A queueing-inventory model that illustrates changing unit-load configuration

In a queueing-inventory model, a stage consists of a processing station and an output store. An arriving customer demand is satisfied from the inventory in the output store if available; else it is backordered. The customer demand immediately triggers an order to replenish the inventory. The replenishment order picks up a part from the output store of the previous stage if available and joins the queue to be processed. Each customer demand triggers a replenishment order at each stage of a multi-stage queueing-inventory model. The maximum planned inventory at each stage is called the base stock level.

The QI model for the order-picking system consists of two such stages in tandem with the following modifications. The stage 1 has a batching station in addition to the regular processing station, and the output store of stage 2 has a base stock level of zero. The output store of stage 1 represents the planned inventory at the rack which is the forward store and the processing stations represent the material movement in and out of the forward store.

The functioning of the unit-load system is as follows. The customer orders (in case quantities) are received at the dummy store of stage 2. The order picks up the required quantity from the forward store and immediately joins the processing queue, i.e., ready to

be retrieved from the store. If the items are not available, then the order is backordered at the stage 2. The customer order triggers a replenishment order at the forward store. The orders wait at the batching station to form pallet quantities. Once such orders are formed, the orders pickup pallets from the upstream stages and join the processing queue to be moved to the rack. In this study, the unit-load system operates under a stationary demand-pull or base stock policy with one-for-one replenishment policy.

We make the following assumptions about the order-picking system. We assume that there is an ample supply of pallet loads at stage 1. Both the stages are characterized by single servers with unit capacity, handling a single class of items. We note that the replenishment order for pallet loads at stage 1 consists of replenishment orders and backorders of cases. We assume that the demand arrival process and the service times at either of the processing stations follow a general distribution. Since the base stock level is set to zero, stage 2 can be analyzed as a simple $GI/G/1$ queue, with a modified arrival process from stage 2 that accounts for orders that find items at the forward store immediately and backorders. In section 6.2, we develop a model for stage 1: a single stage system with batch processing with unlimited supply of pallet loads.

6.2 Single Stage QI model with Batch Processing

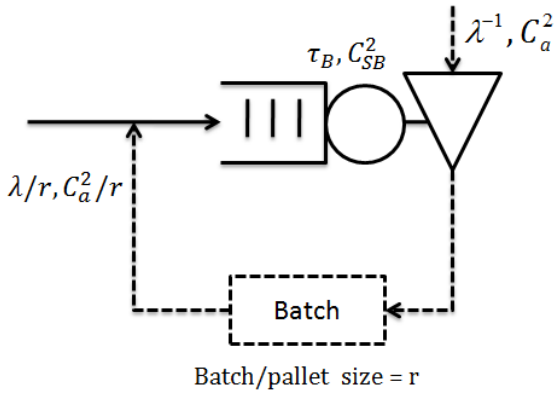


Figure 6.3: Single stage QI model with batch processing

In this section, we model the upstream stage of the order-picking model where the

replenishment orders for case loads are batched and processed as pallet orders (see Figure 6.3). Let each pallet load consist of r cases. The customer demand consumes an item/case from the material store and triggers a replenishment order or places a back-order if a case is not available. The order proceeds to the batching station where it waits until a batch of size of r is formed. Once a pallet load is formed, the pallet immediately joins the queue at the processing station to be processed, i.e. a pallet from the reserve store is ready to be retrieved. The pallets after being processed/retrieved are immediately split into individual items/cases at the forward store.

The following notation is used in this chapter.

λ^{-1}, C_a^2 - mean and SCV of the demand arrival process

τ_B, C_{SB}^2 - mean and SCV of the batch/pallet service process

S - size of the inventory store (in cases)

r - pallet size (in cases)

ρ - utilization of the server at the processing station

N - Number of orders in the system (in cases)

N_B - Number of pallets at the processing station

N_O - Number of orders at the batching station (in cases)

I_F- Inventory level at the forward store

B_F- Backorder level at the forward store

The input to the model is represented by a 6-tuple; the parameters describing the demand arrival process (λ^{-1}, C_a^2), the store size (S), the pallet size (r), the parameters describing the batch/pallet service (τ_B, C_{SB}^2). The performance measures of interest are the average inventory level at the rack $(E[I_F])$, and the average number of backorders in the system $(E[B_F])$. The distribution of number of orders in system can be computed using equations 6.1 - 6.3 from which the distribution of backorder and inventory level can be determined.

To obtain the distribution of number of orders in the system (N) , we need to know the number of orders at the batching station (N_O) and the number of batches/pallets at the processing station (N_B) .

$$P(N=n) = \begin{cases} P(N_O = n) * P(N_B = 0) & n < r \\ P(N_O = n - \lfloor n/r \rfloor * r) * P(N_B = \lfloor n/r \rfloor) & n \geq r \end{cases} \tag{6.1}$$

The distribution of number of orders at the batching station can be obtained as follows. Let us assume that the external arrival process is a renewal process; then the arrival at the batching station is also a renewal process with a rate (λ) and SCV (C_a^2). The maximum number of orders that can wait in the batching station is $(r-1)$ and the orders have an equal probability of $\frac{1}{r}$ (Segal & Whitt, 1989), which is exact for Poisson arrivals.

$$P(N_O = n) = \begin{cases} \frac{1}{r} & 0 \leq n < r \\ 0 & otherwise \end{cases} \tag{6.2}$$

6.2.1 Single-Server Processing Station

The processing station can be modeled either as a single-server station or a multi-server station. In this section, we model the processing station as a $GI/G/1$ queue where each customer (pallets) is a batch of r orders (cases).

The procedure to obtain the number of batches/pallets at the processing station is as follows. Let $E[N_B]$ be the average number of batches at the processing station. The arrival rate and SCV of the inter-arrival time of the batches into the processing station is λ/r and C_a^2/r, respectively (Bitran & Tirupati, 1989). $E[N_B]$ can be calculated using Kramer-Langenbach-Beltz approximation (Kramer & Langenbach-Belz, 1976). Then, the distribution of number of batches is given by Buzacott & Shantikumar (1993).

$$P(N_B = n) = \begin{cases} 1 - \rho & n = 0 \\ \rho(1-\sigma)\sigma^{n-1} & n > 0 \end{cases} \tag{6.3}$$

where

$$\sigma = \frac{E[N_B] - \rho}{E[N_B]}$$

From $P(N=n)$, the distribution of backorders and inventory can be easily derived for a single stage system as shown in Buzacott & Shantikumar (1993). The inventory level is

given by

$$P(I_F = k) = \begin{cases} P(N = S - k) & k = 1, 2, ..S \\ P(N \geq S) & k = 0 \end{cases} \tag{6.4}$$

Then the average inventory level at the rack is given by

$$E[I_F] = \sum_{k=1}^{S} kP(I_F = k) \tag{6.5}$$

The average number of backorders in the system is given by

$$E[B_F] = E[N] + E[I_F] - S \tag{6.6}$$

The accuracy of the models is tested by comparing the analytical results with simulation estimates. The steady state estimate of a performance measure is obtained by averaging over appropriate number of replications. The warm-up period is estimated using Welch's method (Welch, 1983) and is set at 50,000 entities. The statistics were collected for 200,000 entities and the performance measures were averaged over 10 replications.

Relative percentage error (RE) is used to measure the accuracy of the analytical model. When the magnitude of the performance measure is itself small, absolute error is considered better than RE. Robustness of the analytical model will be tested by varying the parameters of the inter-arrival time distribution for the replenishment orders, service time distribution and planned inventory levels at the rack. The performance measures will be examined under low (SCV=0.5), medium (SCV=1.0) and high (SCV=2.0) variability of inter-arrival and service times. An experimental design is provided in Table 6.1 for the single stage QI system with batching. The arrival rate for the customer order is fixed at one and utilization is set at 80% and 90% levels.

Accuracy of the Single-Server QI Model

The estimates of average inventory and average backorders at the rack at 80% and 90% utilization for the unit-load system are given in Tables 6.2, 6.3, 6.4, and 6.5. We report the

Parameters	Levels
Batch Size	2 and 4
Arrival Rate, Arrival SCV	1, {0.5, 1, 2}
Service SCV	{0.5, 1, 2}
Utilization (Batch processing)	80% and 90%
Forward Store Capacity	5, 10, and 15

Table 6.1: Experimental setup for single stage QI system with batching

		Average Inventory			Average Backorder		
Service SCV	BaseStock	A	S	%E	A	S	%E
1	5	1.498	1.473	-1.71%	3.139	3.191	1.64%
1	10	4.836	4.801	-0.74%	1.477	1.519	2.76%
1	15	9.053	9.006	-0.52%	0.694	0.724	0.030
0.5	5	1.880	1.805	-4.14%	1.011	1.013	0.17%
0.5	10	6.107	6.018	-1.47%	0.238	0.226	-0.012
0.5	15	10.924	10.843	-0.75%	0.054	0.051	-0.003
2	5	1.235	1.282	3.69%	7.935	7.821	-1.45%
2	10	3.698	3.866	4.35%	5.398	5.405	0.13%
2	15	6.971	7.202	3.21%	3.671	3.741	1.87%

Table 6.2: Average inventory and average backorder at 80% utilization and batch size of 2 (single-server processing station)

absolute error in cases where the performance measures are themselves small. The results indicate that the maximum absolute percentage error for the average inventory is 8.52% and for the average backorder is 2.92% for a batch size of 4. When the batch size is 2, the maximum absolute percentage errors are 7.21% and 4.61% for the average inventory and average backorder respectively. We note that these errors occur at 90% utilization. Figures 6.4 and 6.5 graphically compare the analytical and simulation estimates for some of the configurations.

		Average Inventory			Average Backorder		
Service SCV	BaseStock	A	S	%E	A	S	%E
1	5	0.798	0.776	-0.0221	9.952	10.159	2.04%
1	10	2.836	2.797	-1.39%	6.990	7.18	2.65%
1	15	5.755	5.694	-1.06%	4.908	5.077	3.32%
0.5	5	1.077	1.009	-0.0678	4.018	4.103	2.07%
0.5	10	4.071	3.954	-2.95%	2.012	2.048	1.76%
0.5	15	8.065	7.931	-1.69%	1.006	1.026	1.94%
2	5	0.634	0.673	0.0389	22.234	21.333	-4.22%
2	10	1.997	2.152	7.21%	18.597	17.811	-4.41%
2	15	3.9542	4.209	6.05%	15.554	14.869	-4.61%

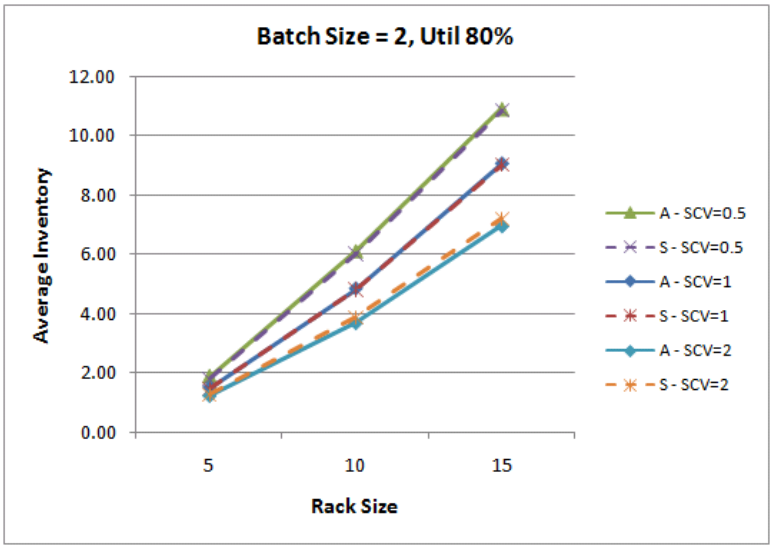
Table 6.3: Average inventory and average backorder at 90% utilization and batch size of 2 (single-server processing station)

		Average Inventory			Average Backorder		
Service SCV	BaseStock	A	S	%E	A	S	%E
1	5	0.761	0.674	-0.0873	7.700	7.679	-0.27%
1	10	2.931	2.865	-2.31%	4.870	4.87	0.00%
1	15	6.142	6.079	-1.04%	3.081	3.085	0.14%
0.5	5	0.801	0.668	-0.1325	3.668	3.683	0.40%
0.5	10	3.659	3.482	-5.09%	1.527	1.498	-1.92%
0.5	15	7.769	7.586	-2.41%	0.636	0.601	-0.035
2	5	0.734	0.677	-0.0573	15.910	15.73	-1.14%
2	10	2.402	2.508	4.23%	12.578	12.562	-0.12%
2	15	4.768	5.032	5.24%	9.944	10.086	1.41%

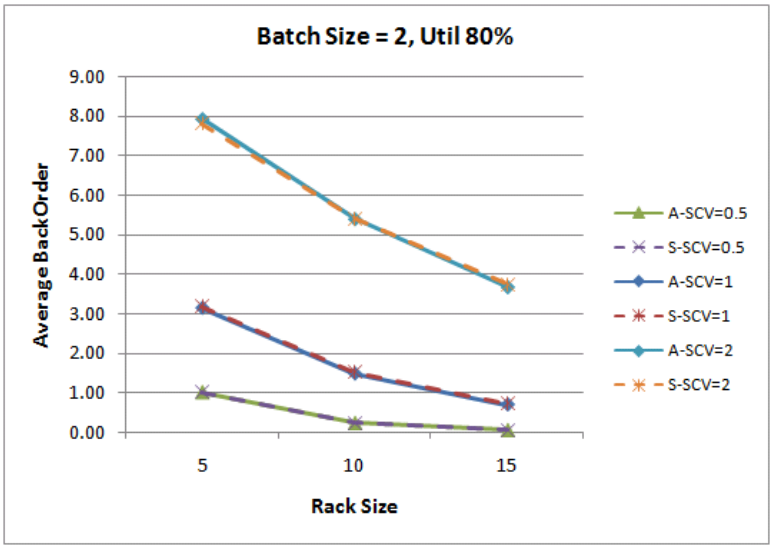
Table 6.4: Average inventory and average backorder at 80% utilization and batch size of 4 (single-server processing station)

		Average Inventory			Average Backorder		
Service SCV	BaseStock	A	S	%E	A	S	%E
1	5	0.385	0.334	-0.0513	19.8550	19.963	0.54%
1	10	1.574	1.518	-3.69%	16.044	16.147	0.64%
1	15	3.495	3.427	-1.98%	12.964	13.055	0.69%
0.5	5	0.414	0.334	-0.0796	9.641	9.803	1.65%
0.5	10	2.133	1.965	-8.52%	6.360	6.434	1.15%
0.5	15	4.699	4.74	0.87%	4.196	4.21	0.32%
2	5	0.369	0.332	-0.0366	40.362	41.147	1.91%
2	10	1.236	1.286	3.86%	36.230	37.1	2.34%
2	15	2.527	2.681	5.74%	32.521	33.496	2.91%

Table 6.5: Average inventory and average backorder at 90% utilization and batch size of 4 (single-server processing station)

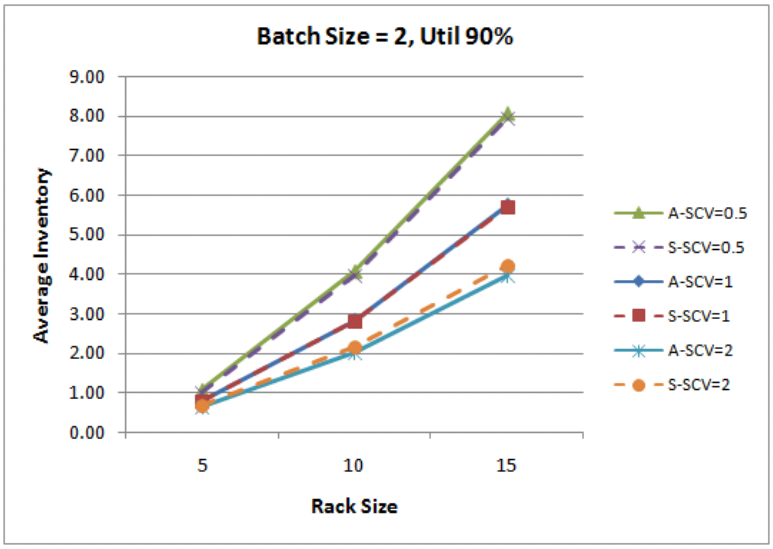


(a) Average inventory when batch size = 2

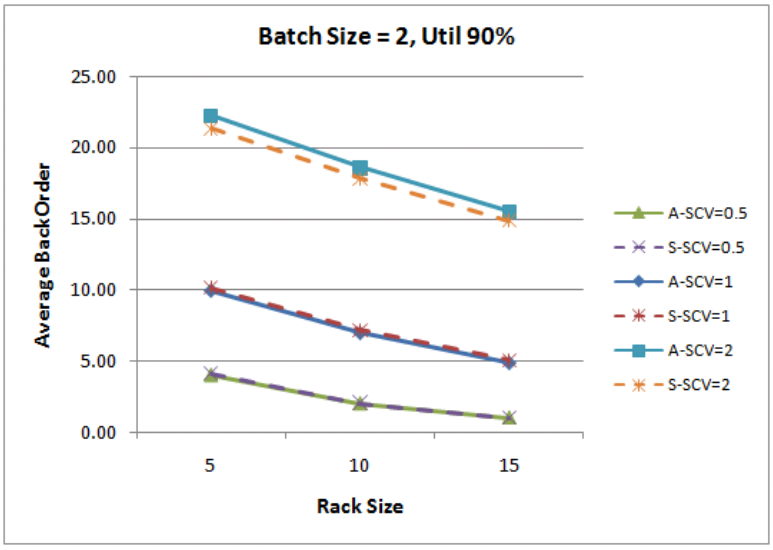


(b) Average backorder when batch size = 2

Figure 6.4: Average inventory level and average backorders at the rack at 80% utilization (single-server processing station)



(a) Average inventory when batch size = 2



(b) Average backorder when batch size = 2

Figure 6.5: Average inventory level and average backorders at the rack at 90% utilization (single-server processing station)

In all our experiments, the SCV of the inter-arrival times of the external demand is same as the variability of batch/pallet service times. Tables 6.2 - 6.5 indicate that average inventory at the rack decreases whereas the average number of backorders increases as the variability in the system increases. We note that the average inventory level decreases as the batch size increases for the same base stock level, because the replenishment orders spend more time in batching and processing than at the forward store. These conclusions hold independent of the utilization level of the server. We also note that the analytical model tracks the simulation model quite well and that the percentage errors are within acceptable limits. The errors are much lower than the errors reported by Sivaramakrishnan (1998) for a similar configuration of the single stage QI model with batch processing.

6.2.2 Multi-Server Processing Station

In this section, we model the processing station as a $GI/G/m$ queue where there are m independent and identical servers. The analysis of the system with a multiple server station follows the general procedure described in section 6.2.1. We use the procedure developed in Whitt (1993) to find the distribution of number in system in a multi-server queue. The distribution of number of batches/pallets at the processing station is then given by

$$P(N_B = k) = \begin{cases} P(Q = k - m) \mid P(Q > 0) & k \geq m + 1 \\ p(k) \mid P(Q = 0) & 0 \leq k \leq m \end{cases} \tag{6.7}$$

where Q is the queue length random variable. $p(k)$ is a truncated Poisson distribution with intensity α, which is found by matching the exact value of expected number of busy servers. The steps to find the distribution of number in system are described in detail in Whitt (1993) and the steps to find the average inventory level and average number of back-orders at the forward storage remains the same as before. For numerical verification, we set the number of servers at three and follow the experimental design presented in 6.1.

		Average Inventory			Average Backorder		
Service SCV	BaseStock	A	S	%E	A	S	%E
0.5	5	0.599	0.539	0.060	2.555	2.435	4.95%
0.5	10	3.805	3.680	3.41%	0.762	0.576	0.186
0.5	15	8.345	8.233	1.36%	0.302	0.129	0.173
1	5	0.646	0.604	0.042	4.504	4.512	-0.17%
1	10	3.283	3.239	1.37%	2.142	2.147	-0.24%
1	15	7.199	7.116	1.17%	1.058	1.024	3.29%
2	5	0.681	0.664	0.017	8.450	8.226	2.72%
2	10	2.925	2.957	-1.09%	5.694	5.520	3.15%
2	15	6.082	6.205	-1.99%	3.851	3.767	2.23%

Table 6.6: Average inventory and average backorder at 80% utilization and batch size of 2 (multi-server processing station)

Accuracy of the Multi-Server QI Model

The estimates of average inventory and average backorders at the rack at 80% and 90% utilization for the unit load system are given in Tables 6.6, 6.7, 6.8, and 6.9. We report the absolute error in cases where the performance measures are small (typically less than 1). The results from Tables 6.6 - 6.9 indicate that the maximum absolute percentage error for the average inventory is 10.30% and for the average backorder is 10.62% at a batch size of 4. When the batch size is 2, the maximum absolute percentage errors are 4.93% and 19.68% on the average inventory and average backorder respectively. We note that these errors occur at 90% utilization. Figures 6.6 and 6.7 graphically compare the analytical and simulation estimates for some of the configurations.

		Average Inventory			Average Backorder		
Service SCV	BaseStock	A	S	%E	A	S	%E
0.5	5	0.142	0.062	0.080	8.924	8.627	3.44%
0.5	10	1.027	0.865	0.162	4.808	4.429	8.56%
0.5	15	3.521	3.359	4.83%	2.303	1.924	19.68%
1	5	0.185	0.115	0.070	11.930	11.765	1.41%
1	10	1.120	0.998	0.122	7.865	7.648	2.84%
1	15	3.283	3.202	2.53%	5.028	4.853	3.61%
2	5	0.098	0.071	0.027	41.334	44.179	-6.44%
2	10	0.571	0.51	0.061	36.834	39.618	-7.03%
2	15	1.618	1.542	4.93%	32.908	35.65	-7.69%

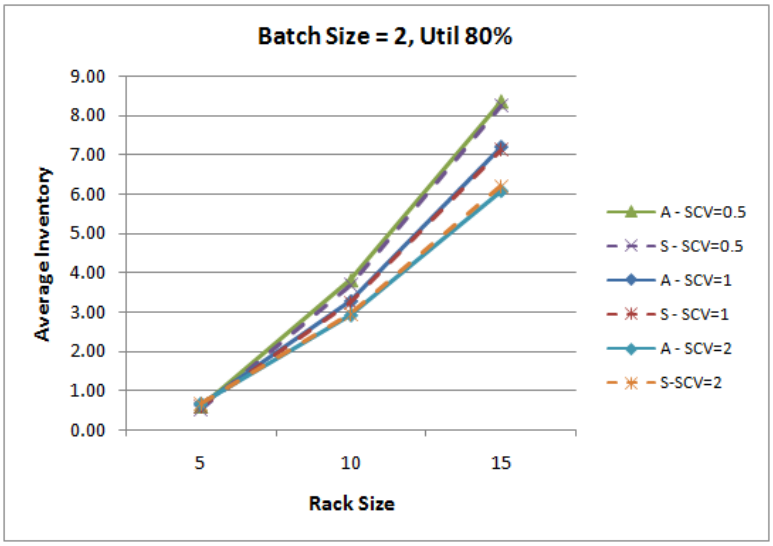
Table 6.7: Average inventory and average backorder at 90% utilization and batch size of 2 (multi-server processing station)

		Average Inventory			Average Backorder		
Service SCV	BaseStock	A	S	%E	A	S	%E
0.5	5	0.288	0.249	0.039	6.310	6.274	0.57%
0.5	10	2.214	2.142	3.35%	3.236	3.168	2.14%
0.5	15	5.730	5.558	3.10%	1.752	1.584	10.62%
1	5	0.312	0.286	0.026	11.798	11.999	-1.68%
1	10	1.803	1.764	2.22%	8.289	8.477	-2.22%
1	15	4.363	4.280	1.94%	5.849	5.993	-2.40%
2	5	0.329	0.314	0.015	22.835	22.994	-0.69%
2	10	1.577	1.540	2.42%	19.084	19.219	-0.70%
2	15	3.463	3.445	0.53%	15.971	16.125	-0.96%

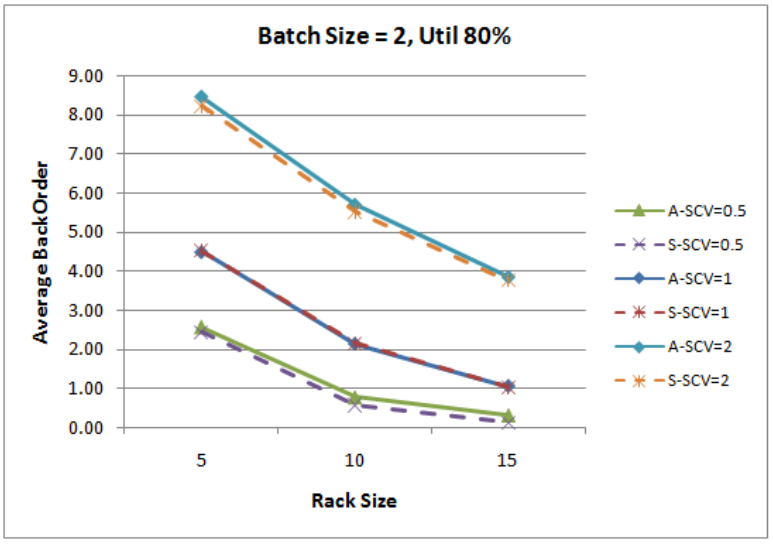
Table 6.8: Average inventory and average backorder at 80% utilization and batch size of 4 (multi-server processing station)

		Average Inventory			Average Backorder		
Service SCV	BaseStock	A	S	%E	A	S	%E
0.5	5	0.059	0.023	0.036	15.802	15.542	1.67%
0.5	10	0.484	0.388	0.096	11.227	10.907	2.94%
0.5	15	1.857	1.757	5.68%	7.600	7.275	4.46%
1	5	0.081	0.048	0.033	24.621	24.595	0.11%
1	10	0.535	0.464	0.071	20.076	20.011	0.32%
1	15	1.700	1.636	3.91%	16.242	16.183	0.36%
2	5	0.225	0.162	0.063	18.340	18.385	-0.24%
2	10	1.197	1.085	10.30%	14.312	14.308	0.03%
2	15	3.139	3.074	2.10%	11.254	11.296	-0.37%

Table 6.9: Average inventory and average backorder at 90% utilization and batch size of 4 (multi-server processing station)

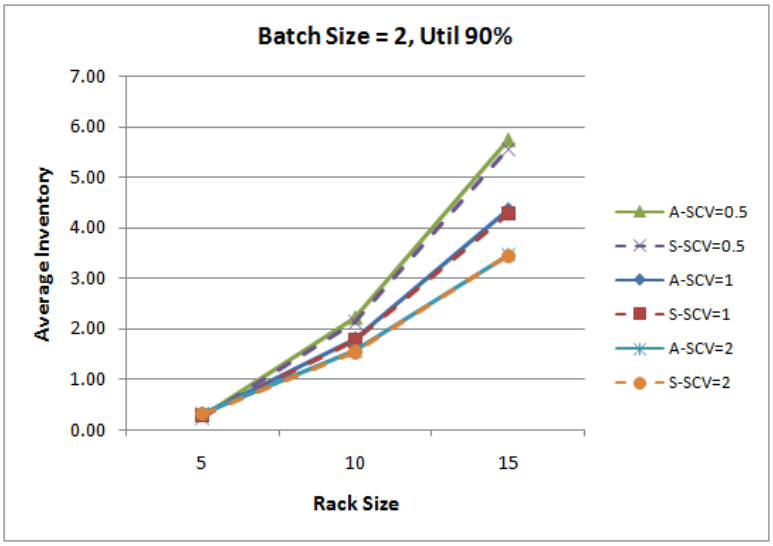


(a) Average inventory when batch size = 2

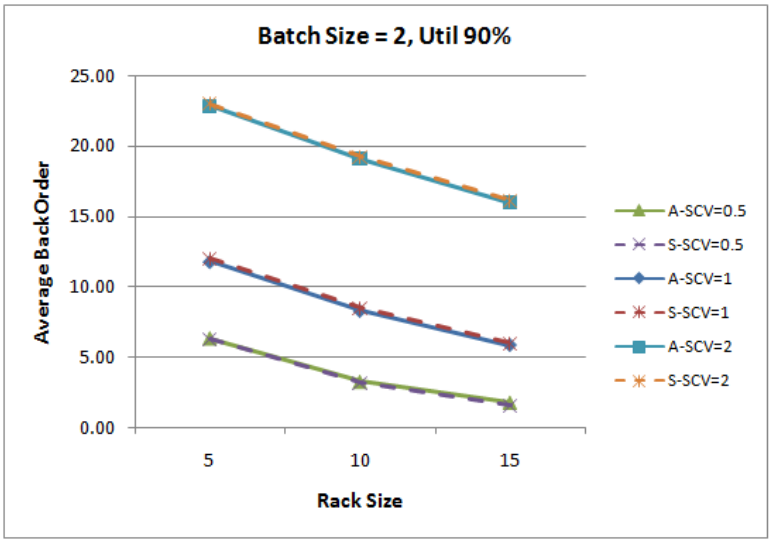


(b) Average backorder when batch size = 2

Figure 6.6: Average inventory level and average backorders at the rack at 80% utilization (multi-server processing station)



(a) Average inventory when batch size = 2



(b) Average backorder when batch size = 2

Figure 6.7: Average inventory level and average backorders at the rack at 90% utilization (multi-server processing station)

6.3 Summary

In this chapter, we discussed the order-picking system from the perspective of changing unit-load configuration. We developed a queueing-inventory model of a single stage system with a batching station, with unlimited supply of raw materials with both single and multi server processing nodes. In the next chapter, we develop an integrated model of the warehouse operations using the shared-server system and order-picking system as building blocks.

Chapter 7

Integrated Warehouse Model

In this chapter, we develop a comprehensive model of the warehouse, applying the models developed in the earlier chapters, namely, the shared-server system and the order-picking system. We briefly describe the warehouse system that is modeled, the representative queueing-inventory model and its assumptions, and describe the solution procedure. The results from the analytical model are then compared with the estimates from simulation experiments.

7.1 Warehouse Description

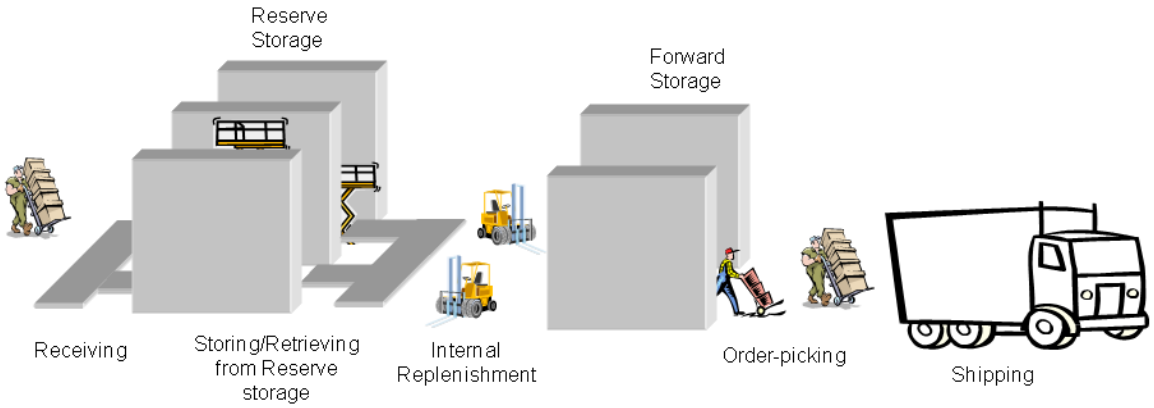


Figure 7.1: Iconic model of the warehouse

Let us assume that pallet loads are received into the warehouse and are staged at the I/O point of the reserve storage area. The S/R machines or operators transfer the pallet

load from the I/O point to the empty slots (if available) for storage. Customer demand is received into the warehouse in less-than-pallet-load quantities. The customer orders are picked from the forward storage area and shipped immediately. When orders consume an equivalent of a pallet load at the forward storage area, a replenishment pallet is transferred from the reserve storage. The pallet load is removed from the rack by the S/R machines or operators and staged at the I/O point of the reserve storage, which are then moved into the forward storage area. We assume that the workers/resources that replenish the forward storage are independent of either the order-pickers or those at the reserve storage. We also assume that the customer orders are satisfied as soon as they are loaded onto the truck/ or ready to be shipped at the outbound staging area. A representation of such a warehouse is shown in the Figure 7.1

7.2 Queueing-Network Description

A queueing-inventory network model of the warehouse system is shown in Figure 7.2. Stage 1 represents the shared-server system representing the material movement to/from the reserve store, stage 2 represents the internal replenishment of forward store from the reserve store, and stage 3 represents the picking operation from the forward store to meet the customer demand.

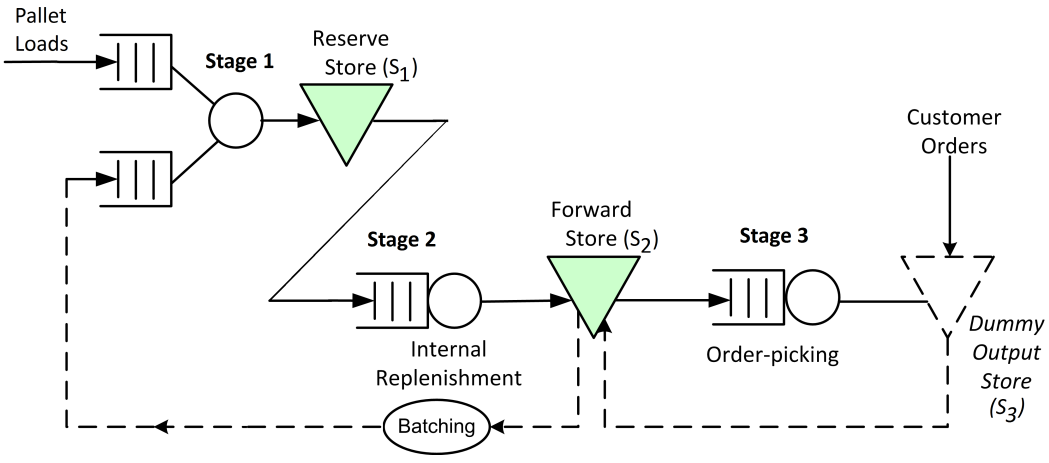


Figure 7.2: Queueing - Inventory model of the warehouse

We assume that the system operates under a stationary-demand pull system or a base-

stock policy. The base stock policy is represented by non-negative integers $S_i \geq 0$, $i = 1, 2, 3$. The quantity S_i represents the maximum planned inventory at each stage. We assume non-zero planned inventory at reserve store (S_1) and forward store (S_2) only. S_1 is specified in pallet loads and S_2 is specified in case loads. Customer demand occurs at the picking stage and it is for one case unit at a time. The customer order is received at the dummy output store ($S_3 = 0$) and it immediately signals a replenishment order at the output store of the upstream stage, the forward store. If an item is available at the forward store, it immediately joins the queue to be picked else it is backordered at stage 2.

At the internal replenishment stage, the consumption of an item/order at the forward store triggers a replenishment order to the reserve storage. The orders are batched at the batching station to make equivalent pallet load orders before placing the replenishment orders at the reserve storage. These orders are then treated as the retrieval requests for the shared-server system. It is important to note that the arrivals into the forward store are pallet loads and departures are in case loads. We assume that pallet loads are received into the warehouse independently of the customer demand and directly at the I/O station of the shared-server system. These pallet arrivals are then treated as the storage requests for the shared-server system.

Initially, we model the integrated system with single server stations and then extend to include multi-server stations. We assume general arrival times for the customer demand & pallets for storage, and general service times at all the processing/material movement stages. The inputs to the integrated model, the performance measures of interest and the solution procedure to analyze the integrated model are presented in the next section.

7.3 Analysis of the Integrated Model

The input parameters to the model are

λ_C, C_C^2 = Arrival rate and SCV of the inter-arrival times of customer demand for cases

λ_S, C_S^2 = Arrival rate and SCV of the inter-arrival times for the pallets to be stored

τ_i, C_i^2 = the mean and SCV of processing time at stage i, $i = 1, 2, 3$

S_1, S_2 = Planned inventory level at reserve and forward stores respectively

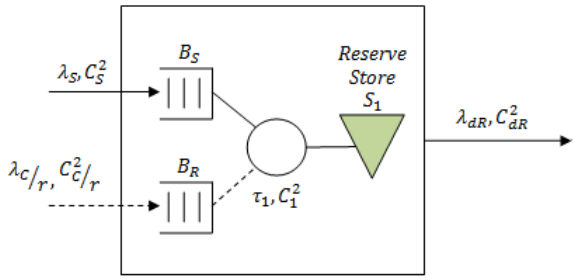


Figure 7.3: Input to and output from the Shared-server stage

r = Pallet size (number of cases per pallet)

m_i = Number of servers at stage i, $i = 1, 2, 3$

The performance measures of interest are the average inventory levels at the reserve and forward stores ($E[I_1]$ and $E[I_2]$), the average back-order level at the forward store ($E[B_2]$), and the average number of orders in the order-picking stage ($E[N_3]$).

We decompose the integrated model into individual stages and obtain the steady state performance measures using the solution methods developed in the earlier chapters.

7.3.1 Integrated Model : Single-Server Case

In the following analysis, we assume single server at all the stages.

Shared-server system: The inputs to the shared-server system are the parameters of the arrival process of the storage and retrieval requests, the capacities of the respective queues, the parameters of the single-command processing times and the size of the reserve storage area. The arrivals occur in pallet loads at the shared-server system. The arrival parameters for the retrieval requests are converted to equivalent pallet load quantities, since the customer demand occurs in case quantities. We set the arrival rate of storage requests equal to that of retrieval requests. Also, we set the capacities of the queues equal to that of the reserve storage area. The inputs and outputs of the shared-server system are illustrated in the Figure 7.3.

The departure process of the retrieval requests from the shared-server system is the arrival process into the Internal-Replenishment stage. The parameters of the departure process of the retrieval requests and the average inventory at the reserve storage ($E[I_1]$) (in

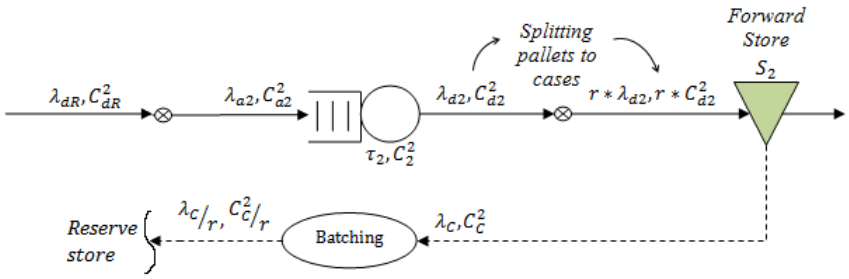


Figure 7.4: Input to and output from Internal-Replenishment stage

pallets) are obtained as explained in chapter 4.

Arrivals to the Internal-Replenishment stage: We note that in the shared-server system, we have assumed finite buffer capacities for the storage and retrieval requests to wait. Hence there is a possibility of loss of the requests, resulting in the departure rate from the shared-server system that is not the same as the equivalent pallet arrival rate. For modeling purposes, we artificially increase arrival rate of the storage and retrieval requests in the shared-server system such that the departure rate of the retrieval requests is same as the customer demand rate (in equivalent pallet quantities). Then, the departure rate of the retrieval requests is the arrival rate at the internal-replenishment stage. The SCV of the inter-departure times of the retrieval requests is the SCV of the inter-arrival times at the Internal-Replenishment stage.

$$\begin{aligned} \lambda_{a2} &= \lambda_{dR} \\ C_{a2}^2 &= C_{dR}^2 \end{aligned} \tag{7.1}$$

Analysis of the Internal-Replenishment stage: The pallets are immediately split into individual cases at the forward store. Once, the parameters of the internal arrival process are known, the performance measures ($E[I_2]$ and $E[B_2]$) can be obtained using the solution procedure given in equations (6.1 - 6.5) in cases. The parameters of the departure process is given by

$$\begin{aligned} \lambda_{d2} &= \lambda_{a2} \\ C_{d2}^2 &= (1-\rho_2^2)C_{a2}^2 + \rho_2^2 C_{S2}^2 \end{aligned} \tag{7.2}$$

The inputs and output parameters of the internal-replenishment stage is illustrated in Figure 7.4.

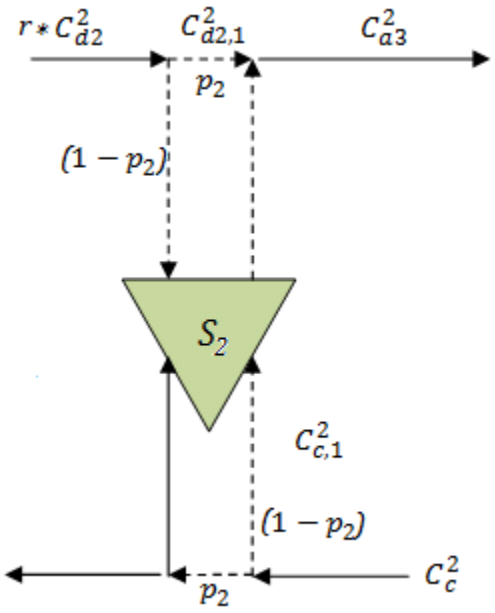


Figure 7.5: Superposition of upstream and downstream arrivals to the order-picking queue

Arrivals to the Order-Picking stage: The arrival process to the order-picking queue is the superposition of two processes; the orders that find an item in the forward store immediately and proceed directly to the picking queue and the upstream orders that satisfy the backorder at the forward store as illustrated in Figure 7.5.

The arrival rate to the order-picking queue is given by

$$\lambda_{a3} = \lambda_C (= r * \lambda_{d2}) \tag{7.3}$$

The SCV of the arrival process is calculated using the following equation.

$$\begin{aligned} C_{C,1}^2 &= (1-p_2)C_C^2 + p_2 \\ C_{d2,1}^2 &= p_2(r * C_{d2}^2) + (1-p_2) \\ C_{a,3}^2 &= (1-p_2)C_{C,1}^2 + p_2 C_{d2,1}^2 \end{aligned} \tag{7.4}$$

where p_2 is the probability of backorder at stage 2 (there is no item at the forward store when a customer demand arrives), is calculated using the following equation.

$$p_2 = 1 - \sum_{k=1}^{S_2} P(N_2 = S_2 - k) \tag{7.5}$$

where S_2 is the basestock level at the forward store and N_2 is the number of cases/orders in stage 2.

The departure process from the internal-replenishment stage is modified by the pallet size (r) because the pallets are converted into cases before added to the forward store.

Analysis of the Order-Picking stage: Now, we know all the parameters characterizing the arrival and service processes at the order-picking stage as illustrated in Figure 7.6. The performance measure ($E[N_3]$) can be obtained using the equations 7.6.

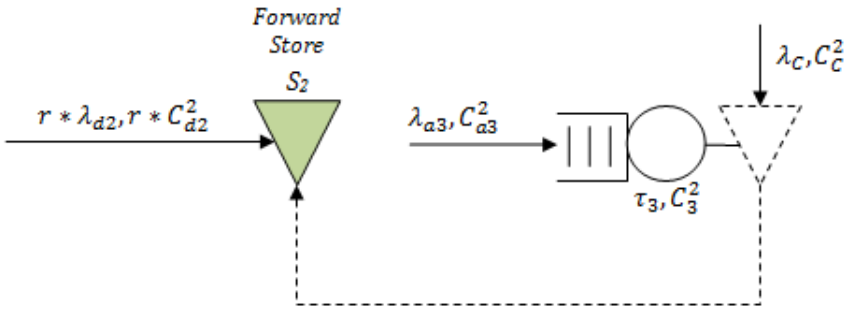


Figure 7.6: Input to and output from Order-Picking stage

$$\begin{aligned} W_{q3} &= g(\rho_3, C_{a3}^2, C_3^2) \left(\tfrac{C_{a3}^2 + C_3^2}{2}\right) \left(\tfrac{\rho_3}{1-\rho_3}\right) \tau_3 \\ g(\rho_3, C_{a3}^2, C_3^2) &= \begin{cases} exp\left(\tfrac{-2(1-\rho_3)(1-C_{a3}^2)^2}{3\rho_3(C_{a3}^2 + C_3^2)}\right) & C_a^2 < 1 \\ exp\left(\tfrac{-(1-\rho_3)(C_{a3}^2 - 1)}{\rho_3 + 4C_3^2}\right) & C_a^2 \geq 1 \end{cases} \\ E[N_3] &= W_{q3} * \lambda_{a3} \end{aligned} \tag{7.6}$$

This completes the solution procedure to solve the integrated model and compute the performance measures. The solution procedure is illustrated in algorithm 7.3.1.

Algorithm 7.3.1: SolveIntegratedModel()

comment: Solution Procedure to Solve the Integrated Model

Inputs:

Arrival parameters of storage requests (pallets) (λ_S, C_S^2) and
customer demand parameters (cases) (λ_C, C_C^2)

Size of reserve and forward storage areas, and pallet size (S_1, S_2, r)

Service parameters at the shared-server, internal-replenishment server
and order-picking servers (τ_i, C_i^2)

Number of servers at each stage (m_i, in the multi-server case)

comment: We set $\lambda_S = \frac{\lambda_C}{r}$ in the shared-server model.

Step1: Solve the shared-server system using Algorithm 4.3.2.

Step2: Compute the departure process parameters from the shared-server system (λ_{dR} and C_{dR}^2) and adjust for loss.

while $|\lambda_{dR} - \lambda_{c/r}| < \varepsilon$

Increase λ_S $(= \lambda_c/r)$

Solve the shared-server system with the new input parameters

end while

Step3: Set $\lambda_{a2} = \lambda_{dR}$ and $C_{a2}^2 = C_{dR}^2$, and solve the internal-replenishment stage.

Step4: Compute the parameters of the departure process from the internal-replenishment stage (λ_{d2}, C_{d2}^2).

Step5: Split the pallet loads into cases before adding to the forward store.

Step6: Obtain the parameters of the combined arrival process to the order-picking stage (λ_{a3}, C_{a3}^2) and solve the order-picking stage.

The experimental design used in the evaluation of our analytical procedure to solve the integrated model is summarized in Table 7.1. In all the cases, we assume the same parameter for the SCV of all arrival and service processes. Table 7.2 describes all the experiments in

Parameter	Levels / Parameters
Customer demand rate	1
Pallet Size and Pallet arrival rate	2 (0.5), 4 (0.25)
Utilization	0.8, 0.9
SCV of the demand IAT	0.5, 1, 2
SCV of the service times	0.5, 1, 2
Reserve Store Size	5, 25
Forward Store Size	10, 50 and 20, 100 for pallet size 2 and 4 resp.

Table 7.1: Experimental design to evaluate the integrated model

	Shared-Server					Internal-Replenishment		Order-picking			
	St Queue	Rt Queue	Rack	Pallet Arrival Rate	Service Time	Service Time	Forward Store	Service Time	Pallet Size	Demand Rate	SCV
C1	5	5	5	0.5	0.8	1.6	10	0.8	2	1	0.5, 1, 2
C2	5	5	5	0.5	0.9	1.8	10	0.9	2	1	0.5, 1, 2
C3	25	25	25	0.5	0.8	1.6	50	0.8	2	1	0.5, 1, 2
C4	25	25	25	0.5	0.9	1.8	50	0.9	2	1	0.5, 1, 2
C5	5	5	5	0.25	1.6	3.2	20	0.8	4	1	0.5, 1, 2
C6	5	5	5	0.25	1.8	3.6	20	0.9	4	1	0.5, 1, 2
C7	25	25	25	0.25	1.6	3.2	100	0.8	4	1	0.5, 1, 2
C8	25	25	25	0.25	1.8	3.6	100	0.9	4	1	0.5, 1, 2

Table 7.2: Complete set of experiments to evaluate the integrated model (single server case)

detail.

Accuracy of the Integrated Model: Single Server Case

The inputs to the integrated model are the arrival parameters of the customer demand and storage requests, service parameters for the shared-server, internal-replenishment and picking operations, inventory size at the reserve storage and forward storage area, and pallet size. The performance measures are the average inventory levels at the reserve and forward store, the average backorder level at the forward store, and the average number of customer orders in the order-picking stage. We also provide the queue length performance measures at the shared-server system. As mentioned in chapter 4 and 5, normalized percentage error is is calculated for the queue length and inventory performance measures at the shared-server. Relative percentage error is calculated for all other performance measures.

Tables 7.3 - 7.11 summarize the results of the analytical model and compare them against the simulation estimates. Not including the case C6, in the case of shared-server system, the maximum absolute error on the mean queue length of storage (retrieval) request is 15.50% (13.14%) and on the average inventory at the reserve store is 4.90%. In the internal-replenishment stage, the maximum absolute error in inventory level at the forward store is 12.32%, and average absolute error in backorder is 12.32%. In the order-picking stage, the average absolute error in mean number of orders is 2.34% and a maximum absolute error at 22.26%. We note that all these errors occur either at high utilization levels of 90% or when the storage size is large. One of the reasons for the high error percentages in the internal-replenishment and order-picking stages is that the shared-server system is an unbalanced system. Any error in the estimation of parameters of the departure process from the shared-server system will be amplified in the downstream stages. Also, we do additional modifications to account for the losses of storage and retrieval requests, which could be another source of inaccuracy.

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
C1	0.730	2.166	2.161	2.242	0.790	5.577	0.597	0.800	2.610
C2	0.790	2.811	2.800	2.184	0.887	3.660	3.295	0.900	5.894
C3	0.795	4.802	4.825	12.061	0.795	44.661	0.000	0.800	2.335
C4	0.891	7.980	7.917	11.920	0.891	39.400	0.088	0.900	4.916
C5	0.731	2.156	2.148	2.227	0.792	10.728	1.246	0.800	2.610
C6	0.791	2.804	2.787	2.170	0.889	6.904	7.026	0.900	6.835
C7	0.795	4.749	4.783	11.988	0.796	88.940	0.000	0.800	2.335
C8	0.891	7.900	7.804	11.823	0.892	78.350	0.175	0.900	4.916

Table 7.3: Analytical estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 0.5$ (single-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
C1	0.779	1.513	1.527	2.471	0.800	6.021	0.228	0.799	2.288
C2	0.871	2.036	2.143	2.413	0.900	3.981	2.001	0.900	4.821
C3	0.796	5.253	4.553	13.194	0.800	45.793	0.000	0.799	2.288
C4	0.896	6.243	5.606	13.144	0.900	41.985	0.006	0.900	4.821
C5	0.782	1.433	1.508	2.408	0.800	12.236	0.244	0.799	2.292
C6	0.875	1.934	2.164	2.307	0.899	8.372	2.654	0.900	4.827
C7	0.797	4.450	5.187	11.778	0.800	91.992	0.000	0.799	2.292
C8	0.896	5.396	6.123	11.789	0.899	85.718	0.000	0.900	4.827

Table 7.4: Simulation estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 0.5$ (single-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
C1	6.29%	-13.06%	-12.68%	4.58%	1.25%	7.37%	-0.369	-0.13%	-14.07%
C2	9.30%	-15.50%	-13.14%	4.58%	1.44%	8.06%	-64.67%	0.00%	-22.26%
C3	0.13%	1.80%	-1.09%	4.53%	0.63%	2.47%	0.0000	-0.13%	-2.05%
C4	0.56%	-6.95%	-9.24%	4.90%	1.00%	6.16%	-0.082	0.00%	-1.97%
C5	6.52%	-14.46%	-12.80%	3.62%	1.00%	12.32%	-1.002	-0.13%	-13.87%
C6	9.60%	-17.40%	-12.46%	2.74%	1.11%	17.53%	-164.73%	0.00%	-41.60%
C7	0.25%	-1.20%	1.62%	-0.84%	0.50%	3.32%	0.000	-0.13%	-1.88%
C8	0.56%	-10.02%	-6.72%	-0.14%	0.78%	8.60%	-0.1750	0.00%	-1.84%

Table 7.5: Error estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 0.5$ (single-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
C1	0.716	2.418	2.420	2.237	0.791	4.678	1.873	0.800	4.076
C2	0.772	3.082	3.076	2.188	0.883	2.938	7.040	0.900	9.862
C3	0.794	5.651	5.701	11.948	0.794	42.438	0.013	0.800	4.000
C4	0.887	9.667	9.588	11.786	0.887	35.180	0.662	0.900	9.016
C5	0.717	2.403	2.405	2.211	0.781	9.386	3.252	0.800	4.196
C6	0.773	3.071	3.062	2.165	0.872	6.102	11.889	0.900	11.223
C7	0.794	5.552	5.629	11.811	0.795	84.574	0.024	0.800	4.000
C8	0.887	9.532	9.409	11.615	0.889	69.689	1.312	0.900	9.031

Table 7.6: Analytical estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 1$ (single-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
C1	0.753	1.874	1.988	2.394	0.800	4.824	1.483	0.799	3.997
C2	0.835	2.389	2.585	2.336	0.899	2.835	6.997	0.900	9.054
C3	0.792	5.871	5.500	12.812	0.800	43.344	0.002	0.799	3.997
C4	0.890	7.548	7.430	12.624	0.899	36.238	0.399	0.900	9.054
C5	0.761	1.771	1.982	2.300	0.800	9.985	1.912	0.799	3.995
C6	0.844	2.299	2.673	2.184	0.900	6.003	10.334	0.899	8.914
C7	0.794	5.222	5.757	11.974	0.800	88.074	0.001	0.799	3.995
C8	0.892	6.900	7.557	11.881	0.900	75.978	0.308	0.899	8.914

Table 7.7: Simulation estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 1$ (single-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
C1	4.91%	-10.88%	-8.64%	3.14%	1.13%	3.03%	-26.30%	-0.13%	-1.98%
C2	7.54%	-13.86%	-9.82%	2.96%	1.78%	-3.63%	-0.61%	0.00%	-8.92%
C3	-0.25%	0.88%	-0.80%	3.46%	0.75%	2.09%	-0.011	-0.13%	-0.08%
C4	0.34%	-8.48%	-8.63%	3.35%	1.33%	2.92%	-0.263	0.00%	0.42%
C5	5.78%	-12.64%	-8.46%	1.78%	2.38%	6.00%	-1.34	-0.13%	-5.03%
C6	8.41%	-15.44%	-7.78%	0.38%	3.11%	-1.65%	-15.05%	-0.11%	-25.90%
C7	0.00%	-1.32%	0.51%	0.65%	0.63%	3.97%	-0.023	-0.13%	-0.13%
C8	0.56%	-10.53%	-7.41%	1.06%	1.22%	8.28%	-1.004	-0.11%	-1.31%

Table 7.8: Error estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 1$ (single-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
C1	0.693	2.759	2.791	2.234	0.797	3.712	5.440	0.800	6.569
C2	0.745	3.426	3.448	2.200	0.884	2.232	15.850	0.900	17.597
C3	0.790	7.185	7.303	11.762	0.791	38.542	0.261	0.800	7.005
C4	0.879	12.337	12.304	11.610	0.899	26.705	5.599	0.900	15.962
C5	0.696	2.739	2.785	2.194	0.789	7.422	9.895	0.800	7.214
C6	0.747	3.414	3.445	2.169	0.875	4.620	26.782	0.900	22.167
C7	0.791	7.007	7.193	11.524	0.792	76.802	0.491	0.800	7.010
C8	0.880	12.130	12.065	11.342	0.883	58.008	5.613	0.900	16.361

Table 7.9: Analytical estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 2$ (single-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
C1	0.704	2.239	2.391	2.366	0.799	3.906	5.400	0.799	7.222
C2	0.774	2.653	2.879	2.324	0.900	2.141	18.367	0.901	17.259
C3	0.782	6.988	6.974	12.417	0.798	38.861	0.266	0.800	7.214
C4	0.877	9.425	9.954	11.939	0.900	28.101	4.662	0.900	16.989
C5	0.718	2.135	2.418	2.257	0.799	8.135	7.929	0.800	7.170
C6	0.789	2.589	2.978	2.190	0.901	4.516	29.175	0.901	17.375
C7	0.786	6.593	6.804	12.225	0.798	80.352	0.268	0.799	7.188
C8	0.881	9.185	9.876	11.717	0.900	60.488	6.111	0.901	17.750

Table 7.10: Simulation estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 2$ (single-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
C1	1.56%	-10.40%	-8.00%	2.64%	0.25%	4.97%	-0.74%	-0.13%	9.04%
C2	3.75%	-15.46%	-11.38%	2.48%	1.78%	-4.25%	13.70%	0.11%	-1.96%
C3	-1.02%	-0.79%	-1.32%	2.62%	0.88%	0.82%	0.0050	0.00%	2.90%
C4	-0.23%	-11.65%	-9.40%	1.32%	0.11%	4.97%	-20.10%	0.00%	6.05%
C5	3.06%	-12.08%	-7.34%	1.26%	1.25%	8.76%	-24.80%	0.00%	-0.61%
C6	5.32%	-16.50%	-9.34%	0.42%	2.89%	-2.30%	8.20%	0.11%	-27.58%
C7	-0.64%	-1.66%	-1.56%	2.80%	0.75%	4.42%	-0.2230	-0.13%	2.48%
C8	0.11%	-11.78%	-8.76%	1.50%	1.89%	4.10%	8.15%	0.11%	7.83%

Table 7.11: Error estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 2$ (single-server)

	Shared-Server					Internal-Replenishment		Order-picking			
	St Queue	Rt Queue	Rack	Pallet Arrival Rate	Service Time	Service Time	Forward Store	Service Time	Pallet Size	Demand Rate	SCV
M1	5	5	5	0.5	2.4	4.8	10	2.4	2	1	0.5, 1, 2
M2	5	5	5	0.5	2.7	5.4	10	2.7	2	1	0.5, 1, 2
M3	25	25	25	0.5	2.4	4.8	50	2.4	2	1	0.5, 1, 2
M4	25	25	25	0.5	2.7	5.4	50	2.7	2	1	0.5, 1, 2
M5	5	5	5	0.25	4.8	9.6	20	4.8	4	1	0.5, 1, 2
M6	5	5	5	0.25	5.4	10.8	20	5.4	4	1	0.5, 1, 2
M7	25	25	25	0.25	4.8	9.6	100	4.8	4	1	0.5, 1, 2
M8	25	25	25	0.25	5.4	10.8	100	5.4	4	1	0.5, 1, 2

Table 7.12: Complete set of experiment to evaluate the integrated model (multi server)

7.3.2 Integrated Model : Multi-Server Case

In this section, we assume multiple servers at all the stages. In addition, we assume the same number of servers at all stages (shared-server, internal-replenishment, and order-picking) $m_i = 3$.

We decompose the integrated model into individual stages with multi servers. We analyze each stage using the multi-server models developed in the earlier chapters (chapter 5 for shared-server system and chapter 6 for the order-picking system) appropriately modifying the arrival process to each of the stages.

The complete set of experiments is illustrated in the Table 7.12. The service times at the processing stations are appropriately modified to set the utilization levels at 0.8 and 0.9.

Accuracy of the Integrated Model: Multi Server Case

In addition to the inputs of the single server model, the number of servers at each of the stage is specified which is set at three in all the stages.

Tables 7.13 - 7.21 summarize the results of the analytical model and compare them

against the simulation estimates. In the case of shared-server system, the maximum absolute error for the mean queue length of storage (retrieval) request is 23.42% (17.26%) and average inventory at the reserve store is 13.12%. In the internal-replenishment stage, the maximum absolute error in inventory level at the forward store is 16.82% and average absolute error in backorder is 23.46% with a maximum of 67.56%. In the order-picking stage, the maximum error in backorder is 17.58%. We note that all these errors occur either at high utilization levels of 90% or when the pallet size is large or both.

7.4 Summary

In this chapter, we demonstrated the applicability of the single and multi server models of shared-server and order-picking system as building blocks in the development of comprehensive model of warehouses. The results from the large number of experiments indicate that the accuracy of the solution procedure is acceptable in most scenarios though warranting further refinement in cases with high utilization and/or large pallet size. The queueing-inventory model of the warehouse thus provides a framework to analyze both capacity and congestion issues simultaneously in the context of warehouse performance evaluation. In the next chapter, we summarize the results and contribution of this research effort.

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
M1	0.698	2.192	2.185	1.885	0.790	3.579	1.247	0.800	3.912
M2	0.749	2.815	2.801	1.731	0.886	2.103	4.808	0.900	7.296
M3	0.793	4.973	5.001	11.474	0.794	41.983	0.006	0.800	3.558
M4	0.887	8.601	8.500	11.142	0.888	36.549	0.193	0.900	6.269
M5	0.699	2.184	2.174	1.871	0.771	7.310	2.029	0.800	3.878
M6	0.750	2.809	2.788	1.718	0.873	4.390	8.001	0.900	7.290
M7	0.794	4.923	4.967	11.399	0.794	83.584	0.012	0.800	3.558
M8	0.888	8.528	8.377	11.042	0.888	72.614	0.383	0.900	6.269

Table 7.13: Analytical estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 0.5$ (multi-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
M1	0.779	1.311	1.349	2.467	0.8	3.692	0.573	0.8	3.559
M2	0.872	1.829	1.957	2.387	0.9	2.156	3.123	0.9	6.205
M3	0.796	4.931	4.672	12.848	0.8	43.119	0	0.8	3.559
M4	0.895	5.924	5.63	12.873	0.9	39.047	0.014	0.9	6.205
M5	0.784	1.255	1.311	2.448	0.8	7.251	0.784	0.8	3.559
M6	0.878	1.746	1.95	2.322	0.899	4.304	4.775	0.9	6.205
M7	0.798	4.923	4.367	13.212	0.8	86.466	0	0.8	3.559
M8	0.897	5.739	5.321	13.037	0.899	79.532	0.003	0.9	6.205

Table 7.14: Simulation estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 0.5$ (multi-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
M1	10.40%	-17.62%	-16.72%	11.64%	1.25%	3.06%	-0.67	0.00%	-9.92%
M2	14.11%	-19.72%	-16.88%	13.12%	1.56%	2.46%	-53.95%	0.00%	-17.58%
M3	0.38%	-0.17%	-1.32%	5.50%	0.75%	2.63%	-0.0060	0.00%	0.03%
M4	0.89%	-10.71%	-11.48%	6.92%	1.33%	6.40%	-0.18	0.00%	-1.03%
M5	10.84%	-18.58%	-17.26%	11.54%	3.63%	-0.81%	-1.25	0.00%	-8.96%
M6	14.58%	-21.26%	-16.76%	12.08%	2.89%	-2.00%	-67.56%	0.00%	-17.49%
M7	0.50%	0.00%	-2.40%	7.25%	0.75%	3.33%	-0.0120	0.00%	0.03%
M8	1.00%	-11.16%	-12.22%	7.98%	1.22%	8.70%	-0.38	0.00%	-1.03%

Table 7.15: Error estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 0.5$ (multi-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
M1	0.688	2.418	2.403	1.895	0.787	3.342	2.288	0.800	4.964
M2	0.735	3.086	3.062	1.758	0.885	1.934	8.226	0.900	9.967
M3	0.792	5.773	5.804	11.342	0.793	40.560	0.021	0.800	4.987
M4	0.883	10.459	10.250	10.991	0.884	33.580	0.658	0.900	10.046
M5	0.689	2.404	2.382	1.870	0.776	6.662	4.044	0.800	4.964
M6	0.736	3.076	3.040	1.735	0.874	4.007	13.989	0.900	9.962
M7	0.792	5.669	5.723	11.196	0.793	80.826	0.041	0.800	4.987
M8	0.884	10.327	10.015	10.805	0.886	66.603	1.288	0.900	10.046

Table 7.16: Analytical estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 1$ (multi-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
M1	0.756	1.626	1.741	2.399	0.8	3.265	2.077	0.8	4.958
M2	0.84	2.143	2.351	2.326	0.901	1.777	8.353	0.9	10.079
M3	0.792	5.465	5.363	12.522	0.8	41.194	0.006	0.8	4.958
M4	0.891	7.178	7.11	12.554	0.901	33.93	0.506	0.9	10.079
M5	0.765	1.548	1.732	2.324	0.8	6.454	3.02	0.8	5.002
M6	0.85	2.078	2.43	2.194	0.9	3.564	12.939	0.901	10.095
M7	0.794	5.58	5.047	13.17	0.8	83.435	0.001	0.8	5.002
M8	0.894	7.125	6.897	12.829	0.9	71.016	0.391	0.901	10.095

Table 7.17: Simulation estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 1$ (multi-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
M1	8.99%	-15.84%	-13.24%	10.08%	1.63%	-2.36%	-10.16%	0.00%	-0.12%
M2	12.50%	-18.86%	-14.22%	11.36%	1.78%	-8.84%	1.52%	0.00%	1.11%
M3	0.00%	-1.23%	-1.76%	4.72%	0.88%	1.54%	-0.0150	0.00%	-0.58%
M4	0.90%	-13.12%	-12.56%	6.25%	1.89%	1.03%	-0.15	0.00%	0.33%
M5	9.93%	-17.12%	-13.00%	9.08%	3.00%	-3.22%	-33.91%	0.00%	0.76%
M6	13.41%	-19.96%	-12.20%	9.18%	2.89%	-12.43%	-8.12%	0.11%	1.32%
M7	0.25%	-0.36%	-2.70%	7.90%	0.88%	3.13%	-0.0400	0.00%	0.30%
M8	1.12%	-12.81%	-12.47%	8.10%	1.56%	6.21%	-0.90	0.11%	0.49%

Table 7.18: Error estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 1$ (multi-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
M1	0.669	2.787	2.780	1.927	0.788	3.084	4.879	0.800	6.753
M2	0.711	3.483	3.470	1.817	0.886	1.777	15.631	0.900	15.278
M3	0.788	7.303	7.394	11.136	0.790	37.953	0.205	0.800	7.525
M4	0.874	13.644	13.318	10.824	0.886	28.611	3.469	0.900	16.452
M5	0.672	2.771	2.758	1.887	0.773	6.278	8.203	0.800	6.773
M6	0.713	3.474	3.452	1.784	0.877	3.641	26.541	0.900	15.207
M7	0.789	7.115	7.255	10.872	0.791	75.647	0.380	0.800	7.528
M8	0.876	13.443	12.944	10.527	0.879	58.786	4.670	0.900	16.561

Table 7.19: Analytical estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 2$ (multi-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
M1	0.713	1.934	2.098	2.367	0.8	2.943	5.568	0.8	7.72
M2	0.785	2.379	2.615	2.326	0.902	1.537	19.285	0.9	17.68
M3	0.782	6.361	6.814	11.961	0.802	37.539	0.333	0.8	7.756
M4	0.878	9.074	9.462	12.082	0.9	27.142	4.905	0.9	17.547
M5	0.726	1.839	2.1	2.284	0.801	5.879	8.79	0.8	7.647
M6	0.799	2.303	2.707	2.192	0.901	3.117	31.218	0.899	17.408
M7	0.786	6.034	6.291	12.263	0.801	77.41	0.307	0.8	7.694
M8	0.882	8.647	9.387	11.797	0.903	57.438	6.143	0.9	17.689

Table 7.20: Simulation estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 2$ (multi-server)

	Shared-Server				Internal-Replenishment			Order-picking	
Case	UTIL	STQ	RTQ	INV	UTIL	INV	BO	UTIL	ORDERS
M1	6.17%	-17.06%	-13.64%	8.80%	1.50%	-4.79%	12.37%	0.00%	12.53%
M2	9.43%	-22.08%	-17.10%	10.18%	1.77%	-15.61%	18.95%	0.00%	13.59%
M3	-0.77%	-3.77%	-2.32%	3.30%	1.50%	-1.10%	0.1280	0.00%	2.98%
M4	0.46%	-18.28%	-15.42%	5.03%	1.56%	-5.41%	29.28%	0.00%	6.24%
M5	7.44%	-18.64%	-13.16%	7.94%	3.50%	-6.79%	6.68%	0.00%	11.43%
M6	10.76%	-23.42%	-14.90%	8.16%	2.66%	-16.81%	14.98%	-0.11%	12.64%
M7	-0.38%	-4.32%	-3.86%	5.56%	1.25%	2.28%	-0.0730	0.00%	2.16%
M8	0.68%	-19.18%	-14.23%	5.08%	2.66%	-2.35%	23.98%	0.00%	6.38%

Table 7.21: Error estimates of the performance measures when $C_S^2 = C_C^2 = C_i^2 = 2$ (multi-server)

Chapter 8

Summary, Conclusions and Future Research

In this chapter, we summarize the research conducted in this dissertation effort, followed by the contributions made to the areas of queueing-inventory models and warehouse performance evaluation. The final section of this chapter summarizes some directions for future research.

8.1 Research Summary

The main research goal for this dissertation was the development of analytical performance evaluation models for warehouses that can address queueing and inventory issues simultaneously. To this end, two important configurations commonly found in warehouses were studied in this research; a shared-server system and an order-picking system. These two building blocks were then used in the development of an end-to-end warehouse model.

In chapter 4, we developed analytical models of the shared-server system for the single server case. Initially we developed a CTMC based model of the shared-server under Markovian assumptions. To address general arrival processes and general storage/retrieval times, an approximate queueing network model of the shared-server was developed. The approximate analytical model was developed for a shared-server operating in a single command mode. In chapter 5, the shared-server system was extended to model the multi-server case

to better represent multiple S/R machines serving the storage area. Several configurations were tested by comparing results of the analytical model with simulation estimates and the results indicated that the approximation method performs well for a wide range of parameter values. The SCV of the departure process of the retrieval requests was analyzed, since this becomes the input process to the downstream operations in the warehouse.

In the single server case, 90% of analytical results had less than 5% absolute relative percentage error and 96% were within 10% error. In the multi-server case for the shared-server, 82% of analytical results had less than 5% error and 91% less than 10% error. These results indicate that the approximate model of the shared-server system using the queueing network approach performs very well. On close observation, we note that higher percentage errors occur in the estimation of the throughput and utilization measures.

In chapter 6, we developed a queueing-inventory model of an order-picking system and developed an analytical solution procedure to solve a single-stage queueing-inventory model with a batching station that forms a key component of the order-picking system. The single server model is then extended to include multiple servers. The models developed address general arrival times and service times for retrieval requests. Several configurations were tested for both single and multi server cases and the results indicated that the approximation method performs very well in a majority of the cases examined. All performance measures (average inventory and average backorders) had absolute relative percentage errors less than 10% in the single server case. In the multi server case, 94% of the analytical results had absolute percentage error less than 10%.

In chapter 7, a warehouse configuration is defined that includes a receiving process into storage and retrieval from the reserve storage area, replenishment from the reserve to the forward storage area and order-picking from the forward storage area. The shared-server system and order-picking system were then used to develop a queueing-inventory model of the warehouse. Numerical experiments showed that the analytical model performed reasonably well in both single and multi server cases. In the integrated model, 80% of the analytical results had less than 10% relative percentage error in the single server case and 61% of the results in the multi server case. One of the reasons for higher errors is that any error in the estimation of throughput and SCV of a departure process in an upstream stage

will affect the accuracy of performance prediction of the downstream stages. We also note that these errors were more pronounced in backorder measures than in inventory related measures.

8.2 Research Contributions

The primary contribution of this dissertation is the development of analytical performance evaluation models that model the impact of inventory decisions (planned inventory levels at the forward and reserve store) together with material handling capacity issues in a single queueing based framework for warehouse systems. Doing so, provides us with a means to study the combined effect of inventory decisions and material handling capacity decisions in a warehouse. The various contributions are summarized below.

- The shared-server system is the most important component of a warehouse system. Modeling the shared-server is key to the development of analytical models of a complete warehouse system. Though other researchers such as Lee (1997) and Bozer & Cho (2005) have developed analytical models of AS/RS, our approach explicitly models the inventory store size within the same framework and is a first. We modeled the shared-server using a CTMC, which gives us an exact method for solving the shared-server system under Markovian assumptions. Perhaps, the most significant contribution is the development of the queueing network model of the general shared-server system. By comprehensively analyzing the shared-server model under balanced and unbalanced conditions, we were able to develop insights into the behavior of the system.

- Extending the single shared-server to include multiple servers enhanced the applicability of the analytical models to realistic situations such as storage areas with multiple S/R machines or operators.

- This research effort also addressed the changing nature of the product configuration between storage operations in a warehouse. By modeling the order-picking operation as a queueing-inventory model with a batching station, we developed a valuable ex-

tension to the class of such models that allows for changing product configuration in and out of inventory stores.

- We demonstrated the applicability of these models as key building blocks in the development of integrated end-to-end warehouse models, thereby enabling the study of two important decisions in the warehouse, namely, resource capacity and storage size simultaneously.

8.3 Future Directions

Significant part of the research effort was focused on the development of the shared-server system. Though the approximate model worked well in most of the test cases, further investigation is warranted. The shared-server system is a first model of its kind and the following provides some ideas for future research.

8.3.1 Research related to the shared-server system

One of the major issues with modeling of the shared-server is the stability issue. Detailed investigation of the stability issues in the shared-server system could be a subject of future research. See Appendix (A.2) for more information about steady state behavior of the shared-server system.

The CTMC model of the shared-server under single-command service operation was numerically solved to obtain the performance measures. A closed form solution to the CTMC model could be a subject of future research.

The accuracy of the analytical model for the shared-server for general arrival and service time distributions greatly depends on the accuracy of the synchronization station model. Any improvement in the accuracy of the performance estimates of the synchronization station will improve the accuracy of the shared-server system; and hence, this could be a subject of future research.

The analytical model of the shared-server system is based on a closed-network model with number of kanbans representing the size of the reserve storage area. In addition, the synchronization stations have a capacity limit on the number of requests waiting for the

kanbans. Because of these fixed queue capacities, some of the service requests are lost. Though this may not be significant in a standalone system, it could lead to inaccuracies in modeling the downstream operations when the shared-server is used as a part of a larger model. Modeling the "lost" arrivals aspect of the shared-server system could be a topic for future research.

The shared-server system was modeled under single-command cycle assumptions. It will be interesting and useful to study the shared-server in a dual-command mode.

8.3.2 Research related to warehouse system

In this dissertation, we did not model multiple classes of customers. Extending these models to such configurations could be a subject for further investigation. We have assumed unit order quantity for customer demand in this research. Developing models that can handle bulk demand could be a subject of future research.

Rapid performance evaluation tools based on queueing network models are available for manufacturing systems analysis. Development of such a rapid performance evaluation tool for warehouse analysis and design is now a real possibility as the models that we have developed are able to explicitly capture the size of inventory stores - a key decision in warehouse designs.

Bibliography

Abdelkrim, B., Zaki, S., & Noureddine, G. (2003). Performance analysis for multi-aisle automated storage/retrieval systems using visual Petri net developer. In *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation.* Piscataway, USA.

Arantes, J. C., Jing, G. G., & Houshmand, A. (1998). Using simulation to evaluate the robustness of a conveyor network's queueing model when conveyor's speed change. In R. J. Graves, L. F. McGinnis, D. J. Medeiros, R. E. Ward, & M. R. Wilhelm (Eds.), *Progress in Material Handling Research: 1998.*

Ashayeri, J., & Goetschalckx, M. (1988). Analysis and design of order picking systems. In *9th Int. Conf. on Automation in Warehousing* (p. 125-135). Brussels, Belgium.

Bastani, A. (1988). Analytical solution of closed-loop conveyor systems with discrete and deterministic material flow. *European Journal of Operational Research*, *35*(2), 187-192.

Berg, J. P. Van den. (1999). A literature survey on planning and control of warehousing systems. *IIE Transactions*, *31*, 751-762.

Berg, J. P. Van den, & Gademann, A. J. R. M. (2000). Simulation study of an automated Storage/retrieval system. *International Journal of Production Research*, *38*(6), 1339-1356.

Berg, J. P. Van den, & Zijm, W. H. M. (1999). Models for warehouse management: Classification and examples. *International Journal of Production Economics*, *59*, 519-528.

Bitran, G., & Tirupati, D. (1989). Approximations for multi-class departure processes. *Queueing Systems: Theory and Applications*, *38*, 205-212.

Bodner, D., Govindaraj, T., Karathur, K. N., Zerangue, N. F., & McGinnis, L. F. (2002). A process model and support tool for warehouse design. In *2002 NSF Design, Service and Manufacturing Grantees and Research Conference.* Arlington, VA.

Bozer, Y. A., & Cho, M. (2005). Throughput performance of automated storage/retrieval systems under stochastic demand. *IIE Transactions*, *37*(4), 367-378.

Bozer, Y. A., & Hsieh, Y. (2005). Throughput performance analysis and machine layout for discrete-space closed-loop conveyors. *IIE Transactions*, *37*(1), 77-89.

Bozer, Y. A., Quiroz, M. A., & Sharp, G. P. (1988). An evaluation of alternative control strategies and design issues for automated order accumulation and sortation systems. *Material Flow*, *4*(4), 265-282.

Bozer, Y. A., & White, J. A. (1990). Design and Performance Models for End-of-Aisle Order Picking Systems. *Management Science*, *36*(7), 852.

Bozer, Y. A., & White, J. A. (1996). A generalized design and performance analysis model for end-of-aisle order-picking systems. *IIE Transactions*, *28*(4), 271.

Buzacott, A. J., & Shantikumar, G. J. (1993). *Stochastic Models of Manufacturing Systems.* Englewood Cliffs, NJ: Prentice Hall.

Cormier, G., & Gunn, E. A. (1992). A Review of Warehouse Models. *European Journal of Operational Research*, *58*(1), 3.

Dong, M., & Chen, F. F. (2005). Performance modeling and analysis of integrated logistics chains: An analytic framework. *European Journal of Operational Research*, *162*(1), 83-98.

Eldemir, F. (2003). *Analytical concepting for Integrated Material Handling Systems.* Ph. d. dissertation, Rensselaer Polytechnic Institute.

Frazelle, E. H. (2001). *World-Class Warehousing and Material Handling* (1st ed.). New York: McGraw-Hill.

Fukunari, M. (2003). *Analytical foundations for Autonomous Vehicle Storage and Retrieval Systems using load transfer station based dwell point strategies.* Ph. d. dissertation, Rensselaer Polytechnic Institute.

Goetschalckx, M., McGinnis, L. F., Bodner, D., Govindaraj, T., Sharp, G. P., & Huang, K. (2002). A systematic design procedure for small parts warehousing systems using modular drawer and bin shelving systems. In *2002 International Material Handling Research Conference.* Portland, ME.

Gray, A. E., Karmarkar, U. S., & Seidmann, A. (1992). Design and operation of order consolidation warehouse: Models and application. *European Journal of Operational Research*, *58*(1), 14-36.

Hariga, M., & Jackson, P. L. (1996). The warehouse scheduling problem: formulation and algorithms. *IIE Transactions*, *28*(2), 115.

Heipcke, S. (2000). *Applications of optimization with Xpress-MP.* United Kingdom: Dash Optimization.

Huang, S., Batta, R., & Nagi, R. (2003). *An integrated model for space determination and site selection of distribution centers.* University of Buffalo, Buffalo, NY.

Hur, S., Lee, Y. H., Lim, S. Y., & Lee, M. H. (2004). A performance estimation model for AS/RS by M/G/1 queuing system. *Computers & Industrial Engineering*, *46*(2).

Hur, S., & Nam, J. (2006). Performance analysis of automatic storage/retrieal systems by stochastic modeling. *International Journal of Production Research*, *44*(8), 1613-1626.

Johnson, E. M., & Meller, R. D. (2002). Performance analysis of a split-case sortation systems. *Manufacturing and Service Operations Management*, *4*(4), 258-274.

Johnson, M. E. (1998). The impact of sorting strategies on automated sortation system performance. *IIE Transactions*, *30*(1), 67.

Johnson, M. E., & Brandeau, M. L. (1996). Stochastic modeling for automated material handling system design and control. *Transportation Science*, *30*(4).

Kamath, M. (1989). *Analytical Performance Models for Automatic Assembly Systems.* Ph. d. dissertation, University of Wisconsin.

Kamath, M., Suri, R., & Sanders, J. L. (1988). Analytical performance models for closed loop flexible assembly systems. *The International Journal of Flexible Manufacturing Systems*, *1*, 51-84.

Kelton, W. D., Sadowski, R. P., & Sadowski, D. A. (2002). *Simulation with Arena.* New York: McGraw Hill.

Kramer, W., & Langenbach-Belz, M. (1976). Approximate Formulae for Delay in the Queueing System GI/G/1. In *Proceedings of the 8th International Teletraffic Congress.* Melbourne, Australia.

Krishnamurthy, A. (2002). *Analytical performance models for material control strategies in manufacturing systems.* Ph. d. dissertation, University of Wisconsin.

Krishnamurthy, A., & Suri, R. (2006). Performance analysis of single stage kanban controlled production systems using parametric decomposition. *Queueing Systems*, *54*, 141-162.

Kuehn, P. J. (1979). Approximate analysis of general queueing networks by decomposition. *IEEE Transactions on Communications*, *27*, 113-126.

Lee, H. F. (1997). Performance analysis for automated storage and retrieval systems. *IIE Transactions*, *29*(1), 15-28.

Lee, Y. H., Tanchoco, J. M. A., & Chun, S. J. (1999). Performance estimation models for AS/RS with unequal sized cells. *International Journal of Production Research*, *37*(18).

Lee, Y. J., & Zipkin, P. H. (1992). Tandem Queues with Planned Inventories. *Operations Research*, *40*(5), 936-947.

Linn, R. J., & Wysk, R. (1984). A simulation model for evaluating control algorithms of an automated storage/retrieval system. *1984 Winter Simulation Conference Proceedings (Cat. No. 84CH2098-2).*

Little, J. D. C. (1961). A Proof for the Queuing Formula: L = λW. *Operations Research*, *9*(3), 383-387.

Liu, L., Liu, X., & Yao, D. D. (2004). Analysis and optimization of a multistage inventory-queue system. *Management Science*, *50*(3).

Manzini, R., Gamberi, M., & Regattieri, A. (2005). Design and control of a flexible order-picking system (FOPS): A new integrated approach to the implementation of an expert system. *Journal of Manufacturing Technology Management*, *16*(1), 18-35.

McGinnis, L. F., Goetschalckx, M., Sharp, G. P., Bodner, D., & Govindaraj, T. (2000). Rethinking Warehouse Design Research. In *2000 International Material Handling Research Colloquium.* York, PA.

Park, B. C. (1999). Optimal dwell point policies for automated storage/retrieval systems with dedicated stoarage. *IIE Transactions*, *31*(10), 1011.

Petersen, C. G. I., & Aase, G. (2004). A comparison of picking, storage and routing policies in manual order picking. *International Journal of Production Economics*, *92*(1), 11-19.

Rao, A. K., & Rao, M. R. (1998). Solution procedures for sizing of warehouses. *European Journal of Operational Research*, *108*, 16-25.

Roll, Y., & Rosenblatt, M. J. (1983). Random vs grouped storage policy and their effect on warehouse capacity. *Material Flow*, *1*, 199-205.

Roll, Y., Rosenblatt, M. J., & Kadosh, D. (1989). Determining the size of a warehouse container. *International Journal of Production Research*, *27*(10), 1693-1704.

Rosenblatt, M. J., & Roll, Y. (1988). Warehouse Capacity in a stochastic environment. *International Journal of Production Research*, *26*(12), 1847-1851.

Rouwenhorst, B., Reuter, B., Stockrahm, V., Houtum, G. J. van, Mantel, R. J., & Zijm, W. H. M. (2000). Warehouse design and control: framework and literature review. *European Journal of Operational Research*, *122*(3), 515-533.

Russell, M. L., & Meller, R. D. (2003). Cost and throughput model of manual and automated order fulfillment systems. *IIE Transactions*, *35*, 589-603.

Sarker, B. R., & Babu, S. P. (1995). Travel time models in automated storage/retrieval systems: a critical review. *International Journal of Production Economics*, *40*(2).

Schefczyk, M. (1990). *Warehouse performance analysis: Techniques and applications* (Tech. Rep. No. MHRC-TD-90-11). Georgia Institute of Technology.

Schwarz, L. B., Graves, S. C., & Hausman, W. H. (1978). Scheduling policies for automatic warehousing systems: simulation results. *AIIE Transactions*, *10*, 260-270.

Segal, M., & Whitt, W. (1989). A Queueing Network Analyzer for Manufacturing. In *Proceedings of the 12th International Teletraffic Congress* (p. 1146-1152).

Sivaramakrishnan, S. (1998). *Analytical Models for the Performance Evaluation of Production-Inventory Systems.* Ph. d. dissertation, Oklahoma State University.

Sivaramakrishnan, S., & Kamath, M. (1997). Analytical models of multi-stage make-to-stock systems. In *6th Industrial Engineering Research Conference.* Miami, FL.

Srivathsan, S. (2005). *Analytical Performance Modeling of Supply Chain Networks* [Master's Thesis].

Sung, C. S., & Han, Y. H. (1992). Determination of Automated Storage/Retrieval System Size. *Engineering Optimization*, *19*, 269-286.

Suri, R., & Sahu, S. (2007). Performance Evaluation of Production Networks. In *Proceedings of the 14th Inustrial Engineering Research Conference.* Nashville, TN.

Suri, R., Sanders, J. L., & Kamath, M. (1993). Performance Evaluation of Production Networks. In S. C. Graves (Ed.), *Handbooks in OR & MS* (Vol. 4, p. 189-286).

Svoronos, A., & Zipkin, P. H. (1991). Evaluation of One-for-One Replenishment Policies for Multiechelon Inventory Systems. *Management Science*, *37*(1), 68-83.

Tompkins, J. A., White, J. A., Bozer, Y. A., & Tonchoco, J. M. A. (2003). *Facilities Planning* (Third ed.). New Jersey: John Wiley & Sons.

Welch, P. D. (1983). *The Statistical Analysis of Simulation Results* (S. S. Lavenberg, Ed.). New York, NY: Academic Press.

Whitt, W. (1982). Approximating a Point Process by a Renewal Process, I: Two Basic Methods. *Operations Research*, *30*(1), 125-147.

Whitt, W. (1983). The Queueing Network Analyzer. *The Bell System Technical Journal*, *62*(9), 2779-2815.

Whitt, W. (1993). Approximations for the GI/G/m queue. *Productions and Operations Management*, *2*, 114-161.

Whitt, W. (1994). Towards Better Multi-Class Parametric Decomposition Approximations for Open Queueing Networks. *Annals of Operations Research*, *48*, 221-248.

Yoon, C. S., & Sharp, G. P. (1995). Example application of the cognitive design procedure for an order pick system: Case Study. *European Journal of Operational Research*, *87*, 223-246.

Yoon, C. S., & Sharp, G. P. (1996). A structured procedure for analysis and design of order pick systems. *IIE Transactions*, *28*(5), 379-389.

Zipkin, P. H. (1995). Performance Analysis of a Multi-item Production-Inventory System Under Alternative Policies. *Management Science*, *41*(4), 690-703.

Appendix A

Appendix

A.1 Stationary Equations - Shared-Server System

The stationary equations are defined for the shared-server system for the single server case. The queue capacity is set independent of the rack size. Both the storage and retrieval buffers can reach the maximum at the same time. Arrivals to the queue are lost when the queues are full. Also, the arriving requests do not have information about the mode of the server.

When $m = 0, i = 0, j = 0, 0 \leq k \leq Z$

$$\begin{array}{cc} \mu_{SC} P_{R,0,0,k+1} = (\lambda_S + \lambda_R) P_{0,0,0,k} & k = 0 \\ \mu_{SC} P_{R,0,0,k+1} + \mu_{SC} P_{S,0,0,k-1} = (\lambda_S + \lambda_R) P_{0,0,0,k} & 0 < k < Z \\ \mu_{SC} P_{S,0,0,k-1} = (\lambda_S + \lambda_R) P_{0,0,0,k} & k = Z \end{array}$$

When $m = 0, 0 < i < B_S, j = 0, k = Z$

$$\mu_{SC} P_{S,i,0,k-1} + \lambda_S P_{0,i-1,0,Z} = (\lambda_S + \lambda_R) P_{0,i,0,k} \tag{A.1}$$

When $m = 0, i = B_S, j = 0, k = Z$

$$\mu_{SC} P_{S,i,0,k-1} + \lambda_S P_{0,i-1,0,Z} = \lambda_R P_{0,i,0,k} \tag{A.2}$$

when there is an item to be retrieved from the rack, server mode can not be "0".
When $m = 0, i = 0, 0 < j < B_R, k = 0$

$$\mu_{SC} P_{R,0,j,k+1} + \lambda_R P_{0,0,j-1,0} = (\lambda_S + \lambda_R) P_{0,0,j,k} \tag{A.3}$$

When $m = 0, i = 0, j = B_R, k = 0$

$$\mu_{SC} P_{R,0,j,k+1} + \lambda_R P_{0,0,j-1,0} = \lambda_S P_{0,0,j,k} \tag{A.4}$$

When there is a space to put an item in the rack or there is an item that can be retrieved, the server mode can not be "0".

When $m = S, i = 0, j = 0, 0 \leq k < Z$

$$\mu_{SC} P_{R,i+1,j,k+1} + \lambda_S P_{0,i,j,k} = (\lambda_S + \lambda_R + \mu_{SC}) P_{m,i,j,k} \tag{A.5}$$

When $m = S, i = 0, 0 < j < B_R, k = 0$

$$\mu_{SC} P_{R,i+1,j,k+1} + \lambda_S P_{0,i,j,k} + \lambda_R P_{S,i,j-1,k} = (\lambda_S + \lambda_R + \mu_{SC}) P_{m,i,j,k} \tag{A.6}$$

When $m = S, i = 0, 0 < j < B_R, 0 < k < Z$

$$p_s \mu_{SC} P_{R,i+1,j,k+1} + p_s \mu_{SC} P_{S,i+1,j,k+1} + \lambda_R P_{S,i,j-1,k} = (\lambda_S + \lambda_R + \mu_{SC}) P_{m,i,j,k} \tag{A.7}$$

When $m = S, i = 0, j = B_R, k = 0$

$$\mu_{SC1} P_{R,i+1,j,k+1} + \lambda_S P_{0,i,j,k} + \lambda_R P_{S,i,j-1,k} = (\lambda_S + \mu_{SC}) P_{m,i,j,k} \tag{A.8}$$

When $m = S, i = 0, j = B_R, 0 < k < Z$

$$p_s \mu_{SC} P_{R,i+1,j,k+1} + p_s \mu_{SC} P_{S,i+1,j,k+1} + \lambda_R P_{S,i,j-1,k} = (\lambda_S + \mu_{SC}) P_{m,i,j,k} \tag{A.9}$$

When $m = S, 0 < i < B_S, j = 0, 0 \leq k < Z$

$$\mu_{SC} P_{R,i+1,j,k+1} + p_s \mu_{SC} P_{S,i+1,j,k+1} + \lambda_S P_{S,i-1,j,k} = (\lambda_S + \lambda_R + \mu_{SC}) P_{m,i,j,k} \tag{A.10}$$

When $m = S, 0 < i < B_S, 0 < j < B_R, k = 0$

$$\mu_{SC} P_{R,i+1,j,k+1} + \lambda_S P_{S,i-1,j,k} + \lambda_R P_{S,i,j-1,k} = (\lambda_S + \lambda_R + \mu_{SC}) P_{m,i,j,k} \tag{A.11}$$

When $m = S, 0 < i < B_S, 0 < j < B_R, 0 < k < Z$

$$\mu_{SC} P_{R,i+1,j,k+1} + \mu_{SC} P_{S,i+1,j,k-1} + \lambda_S P_{S,i-1,j,k} + \lambda_R P_{S,i,j-1,k} = (\lambda_S + \lambda_R + \mu_{SC}) P_{m,i,j,k} \tag{A.12}$$

When $m = S, 0 < i < B_S, j = B_R, k = 0$

$$\mu_{SC} P_{R,i+1,j,k+1} + \lambda_S P_{S,i-1,j,k} + \lambda_R P_{S,i,j-1,k} = (\lambda_S + \mu_{SC}) P_{m,i,j,k} \tag{A.13}$$

When $m = S, 0 < i < B_S, j = B_R, 0 < k < Z$

$$\mu_{SC} P_{R,i+1,j,k+1} + \mu_{SC} P_{S,i+1,j,k-1} + \lambda_S P_{S,i-1,j,k} + \lambda_R P_{S,i,j-1,k} = (\lambda_S + \mu_{SC}) P_{m,i,j,k} \quad \text{(A.14)}$$

When $m = S, i = B_S, j = 0, 0 \leq k < Z$

$$\lambda_S P_{S,i-1,j,k} = (\lambda_R + \mu_{SC}) P_{m,i,j,k} \quad \text{(A.15)}$$

When $m = S, i = B_S, 0 < j < B_R, 0 \leq k < Z$

$$\lambda_S P_{S,i-1,j,k} + \lambda_R P_{S,i,j-1,k} = (\lambda_R + \mu_{SC}) P_{m,i,j,k} \quad \text{(A.16)}$$

When $m = S, i = B_S, j = B_R, 0 \leq k < Z$

$$\lambda_S P_{S,i-1,j,k} + \lambda_R P_{S,i,j-1,k} = (\mu_{SC}) P_{m,i,j,k} \quad \text{(A.17)}$$

When $m = R, i = 0, j = 0, 0 < k < Z$

$$(\lambda_S + \lambda_R + \mu_{SC}) P_{m,i,j,k} = \lambda_R P_{0,i,j,k} + P_r \mu_{SC} P_{S,i,j+1,k-1} + P_r \mu_{SC} P_{R,i,j+1,k+1} \quad \text{(A.18)}$$

When $m = R, i = 0, j = 0, k = Z$

$$(\lambda_S + \lambda_R + \mu_{SC}) P_{m,i,j,k} = \lambda_R P_{0,i,j,k} + \mu_{SC} P_{S,i,j+1,k-1} \quad \text{(A.19)}$$

When $m = R, i = 0, 0 < j < B_R, 0 < k < Z$

$$(\lambda_S + \lambda_R + \mu_{SC}) P_{m,i,j,k} = \lambda_R P_{R,i,j-1,k} + P_r \mu_{SC} P_{S,i,j+1,k-1} + P_r \mu_{SC} P_{R,i,j+1,k+1} \quad \text{(A.20)}$$

When $m = R, i = 0, 0 < j < B_R, k = Z$

$$(\lambda_S + \lambda_R + \mu_{SC}) P_{m,i,j,k} = \lambda_R P_{R,i,j-1,k} + \mu_{SC} P_{S,i,j+1,k-1} \quad \text{(A.21)}$$

When $m = R, i = 0, j = B_R, 0 < k \leq Z$

$$(\lambda_S + \mu_{SC}) P_{m,i,j,k} = \lambda_R P_{R,i,j-1,k} \quad \text{(A.22)}$$

When $m = R, 0 < i < B_S, j = 0, 0 < k < Z$

$$(\lambda_S + \lambda_R + \mu_{SC}) P_{m,i,j,k} = \lambda_S P_{R,i-1,j,k} + p_r \mu_{SC} P_{S,i,j+1,k-1} + p_r \mu_{SC} P_{R,i,j+1,k+1} \quad \text{(A.23)}$$

When $m = R, 0 < i < B_S, j = 0, k = Z$

$$(\lambda_S + \lambda_R + \mu_{SC})P_{m,i,j,k} = \lambda_S P_{R,i-1,j,k} + \lambda_R P_{0,i,j,k} + p_r \mu_{SC} P_{S,i,j+1,k-1} \tag{A.24}$$

When $m = R, 0 < i < B_S, 0 < j < B_R, 0 < k < Z$

$$(\lambda_S + \lambda_R + \mu_{SC})P_{m,i,j,k} = \lambda_S P_{R,i-1,j,k} + \lambda_R P_{R,i,j-1,k} + P_r \mu_{SC} P_{S,i,j+1,k-1} + P_r \mu_{SC} P_{R,i,j+1,k+1} \tag{A.25}$$

When $m = R, 0 < i < B_S, 0 < j < B_R, k = Z$

$$(\lambda_S + \lambda_R + \mu_{SC})P_{m,i,j,k} = \lambda_S P_{R,i-1,j,Z} + \lambda_R P_{R,i,j-1,Z} + P_r \mu_{SC} P_{S,i,j+1,Z-1} \tag{A.26}$$

When $m = R, 0 < i < B_S, j = B_R, 0 < k \leq Z$

$$(\lambda_S + \lambda_R + \mu_{SC})P_{m,i,j,k} = \lambda_S P_{R,i-1,j,k} + \lambda_R P_{R,i-1,j,k} \tag{A.27}$$

When $m = R, i = B_S, j = 0, 0 < k < Z$

$$(\lambda_R + \mu_{SC})P_{m,i,j,k} = \lambda_S P_{R,i-1,j,k} + P_r \mu_{SC} P_{S,i,j+1,k-1} + P_r \mu_{SC} P_{R,i,j+1,k+1} \tag{A.28}$$

When $m = R, i = B_S, j = 0, k = Z$

$$(\lambda_R + \mu_{SC})P_{m,i,j,k} = \lambda_S P_{R,i-1,j,k} + \lambda_R P_{0,i,j,k} + \mu_{SC} P_{S,i,j+1,k-1} \tag{A.29}$$

When $m = R, i = B_S, 0 < j < B_R, 0 < k < Z$

$$(\lambda_R + \mu_{SC})P_{m,i,j,k} = \lambda_S P_{R,i-1,j,k} + \lambda_R P_{R,i,j-1,k} + P_r \mu_{SC} P_{S,i,j+1,k-1} + P_r \mu_{SC} P_{R,i,j+1,k+1} \tag{A.30}$$

When $m = R, i = B_S, 0 < j < B_R, k = Z$

$$(\lambda_R + \mu_{SC})P_{m,i,j,k} = \lambda_S P_{R,i-1,j,k} + \lambda_R P_{R,i,j-1,k} + \mu_{SC} P_{S,i,j+1,k-1} \tag{A.31}$$

When $m = R, i = B_S, j = B_R, 0 < k \leq Z$

$$(\mu_{SC})P_{m,i,j,k} = \lambda_S P_{R,i-1,j,k} + \lambda_R P_{R,i,j-1,k} \tag{A.32}$$

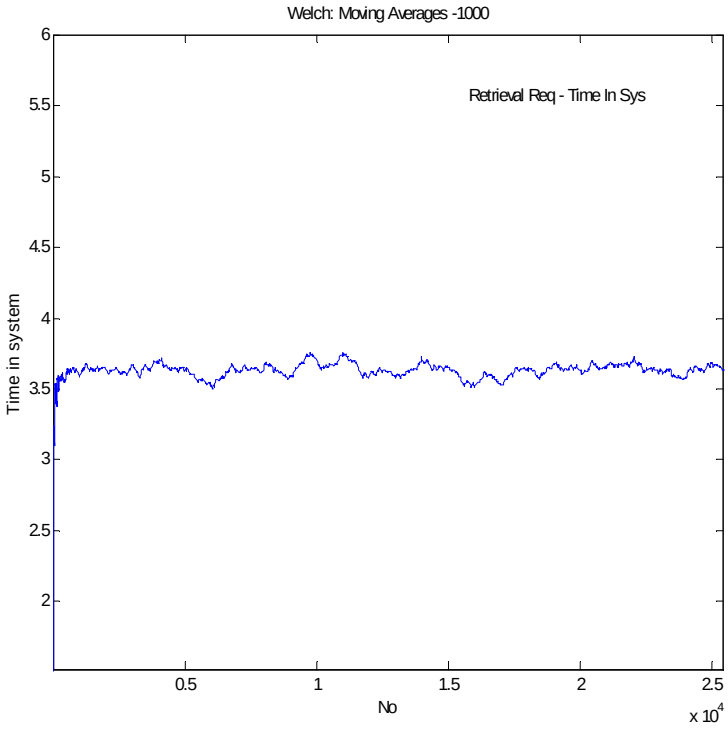
A.2 Simulation Study of the Shared-Server System

In this dissertation, the accuracy of the analytical results was determined by comparing them with simulation estimates. The performance measures obtained from the analytical models were steady state values. Hence, the simulation estimates must also represent steady state values. While performing steady state simulation experiments, a warm-up period has to be determined to remove any initialization bias, and a sufficient run-length has to be provided so that rare events occur a reasonable number of times.

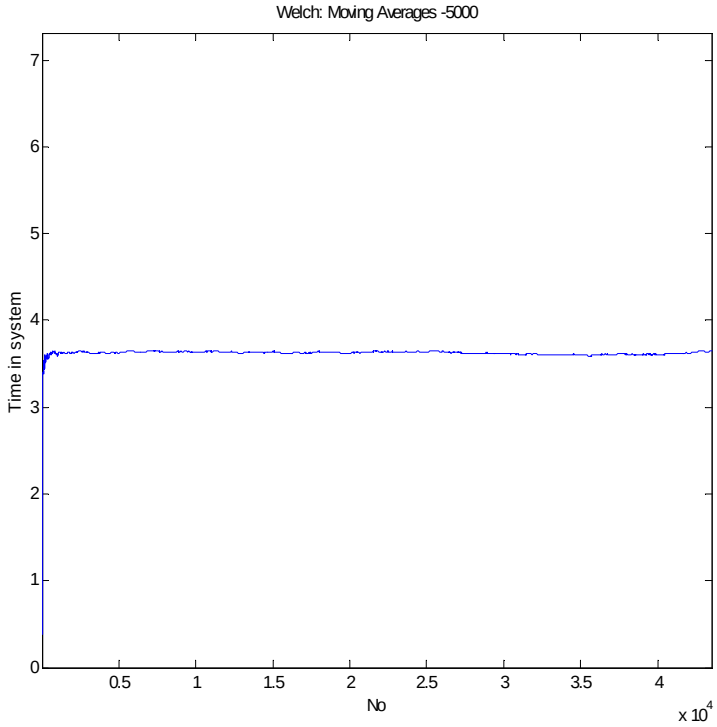
In the shared-server system, the factors that affect the warm-up period and run length were the parameters of the arrival (storage and retrieval requests) and service processes. Since, we cannot estimate the warm-up period and run length for every test configuration, we chose a system that has high variability in the arrival and service processes. In general, higher the variability longer will be the time to reach steady state, and longer would be the run length to get good estimates. Hence, we chose a system with high variability for the arrival and service parameters; hyper-exponential distribution (SCV = 2) for inter-arrival times and service times and a high utilization level of 90%, to set the warm-up and run length for all our experiments. Another important parameter in the shared-server system is the size of the inventory store. We set the rack size at 5 in the first case and 100 in the second case. We initialize the model with 50% of the maximum planned inventory level and used Welch's method Welch (1983) to determine the warm-up period.

The shared-server system posed significant difficulty in modeling because of the underlying issues with stability. The capacity limits on the rack, and the limits on the storage and retrieval request queues provide the necessary control on the operation of the shared-server. In addition, we assumed equal arrival rates for the storage and retrieval requests in this research. Figure A.1 illustrates the batch means for 10 replications of the time in system for the retrieval requests, when the size of the inventory store is 5. We see that the system exhibits steady state after the completion of few retrieval requests.

Figure A.2 illustrate the batch means for 10 replications of the time in system for the retrieval requests when the rack size is 100. A moving average window of 10,000 was used. The simulation statistics were collected for 1,000,000 entities.

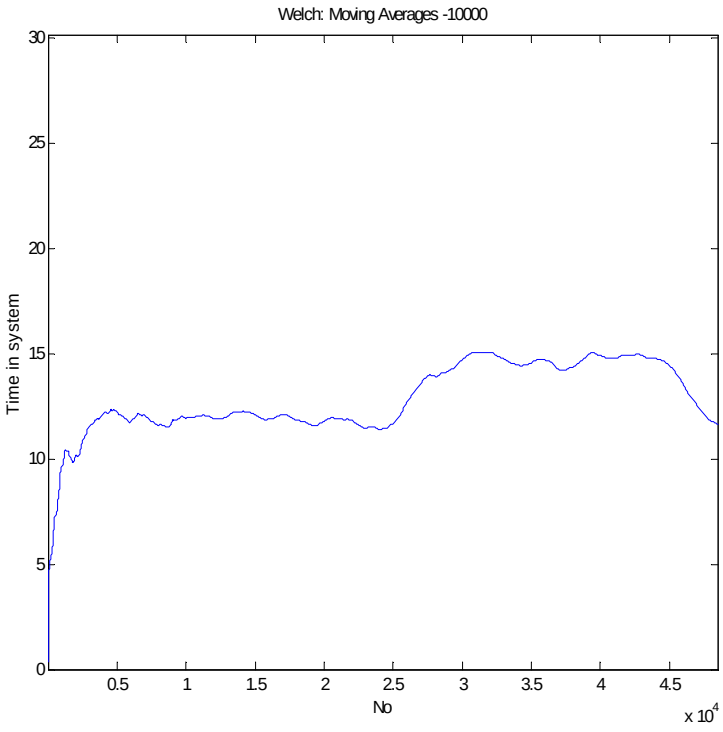


(a) Window length for moving average = 1000

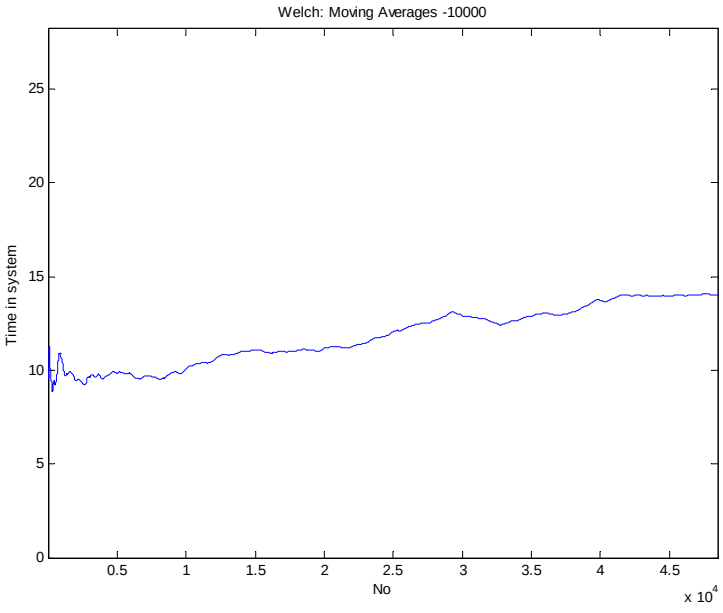


(b) Window length for moving average = 5000

Figure A.1: Plot of batch means of time in system for retrieval requests

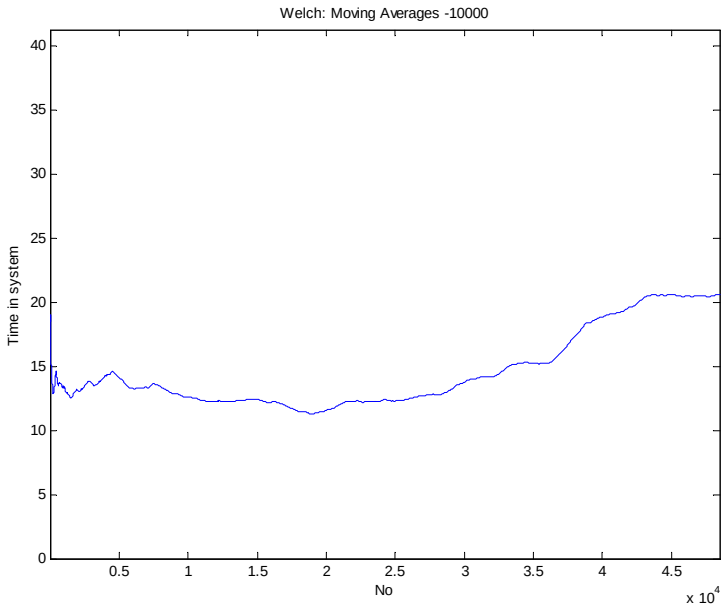


(a) Plot of batch means for the requests 1 - 50000

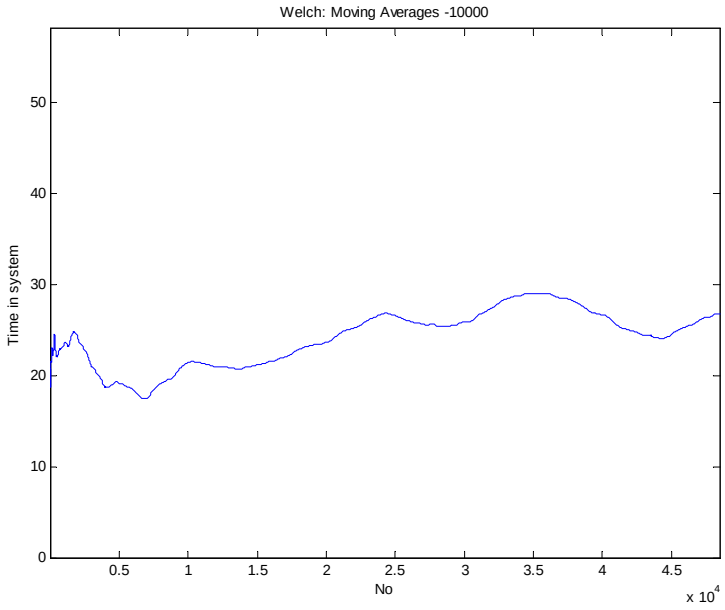


(b) Plot of batch means for the requests 50001 - 100000

Figure A.2: Plot of batch means of time in system for retrieval requests

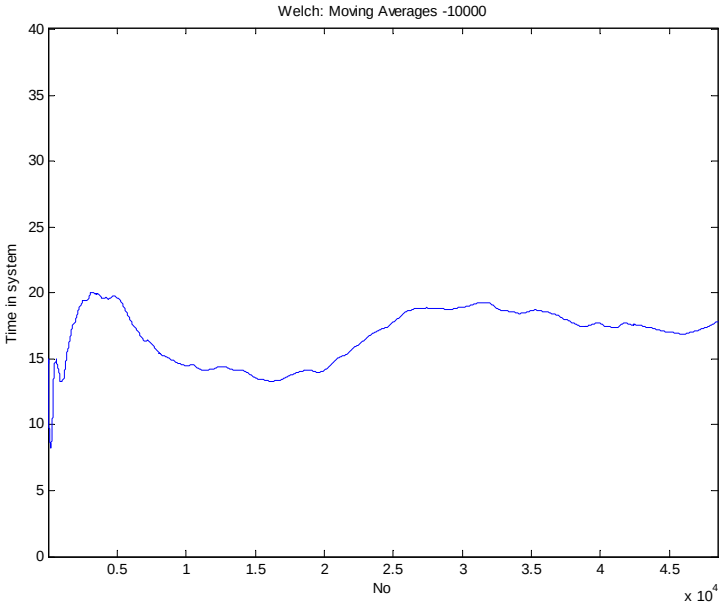


(c) Plot of batch means for the requests 100001 - 150000

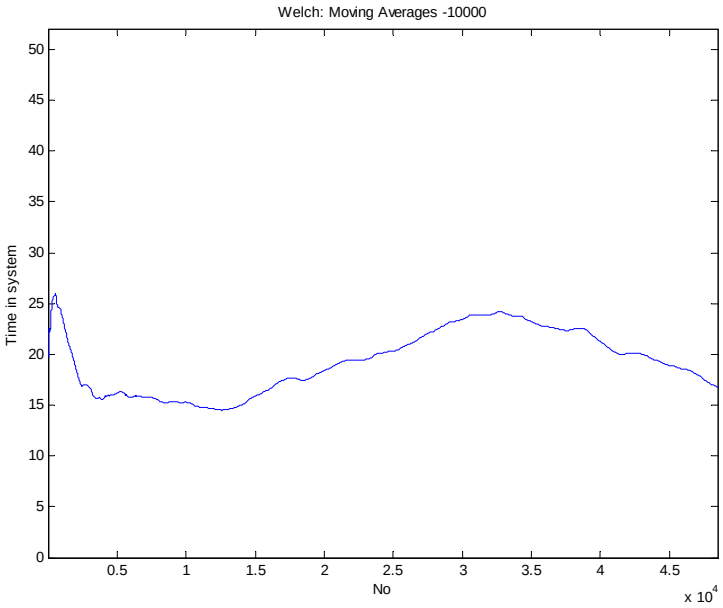


(d) Plot of batch means for the requests 150001 - 200000

Figure A.2: Plot of batch means of time in system for retrieval requests (contd.)

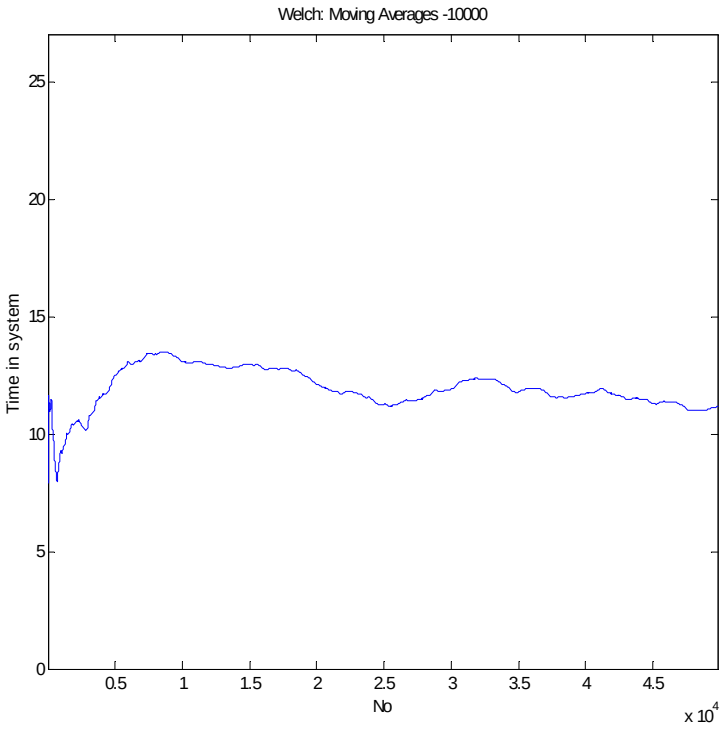


(e) Plot of batch means for the requests 450001 - 500000



(f) Plot of batch means for the requests 550001 - 600000

Figure A.2: Plot of batch means of time in system for retrieval requests (contd.)



(g) Plot of batch means for the requests 600001 - 650000

Figure A.2: Plot of batch means of time in system for retrieval requests (contd.)

We notice that the simulation estimates for the time in system starts to exhibit steady state behaviour after a very long time. Upon further testing, we estimated the warm-up period at 400,000 time units, and collected statistics for 600,000 entities.

Another decision that was required to conduct the simulation experiments was the random number seeds. Our preliminary experiements showed that the random number seed had a significant impact on the steady state behavior of the shared-server system. Arena simulation software provides 10 random number streams. Each of the arrival processes and service process were tested with different combinations of random streams before deciding on the warm-up period, run length and replications.

VITA

Karthik Ayodhiramanujan

Candidate for the Degree of

Doctor of Philosophy

Dissertation: INTEGRATED ANALYTICAL PERFORMANCE EVALUATION MODELS OF WAREHOUSES

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Chennai, Tamilnadu, India on May 31, 1977, the son of Sri. S. Ayodhiramanujan and Smt. A. Ganapriya Ayodhiramanujan.

Education:

Received the B. S. degree from S. V. National Institute of Technology, Surat, Gujarat, India, 1998, in Mechanical Engineering; received the M. S. degree from Oklahoma State University, Stillwater, Oklahoma, USA, 2001, in Mechanical Engineering ; completed the requirements for the degree of Doctor of Philosophy with a major in Industrial Engineering and Management, Oklahoma State University in July, 2009.

Experience:

Graduate Engineer Trainee, Kirloskar Oil Engines Ltd., India from August 1998 to June 1999; Graduate Research Assistant, School of Mechanical and Aerospace Engineering, Oklahoma State University from January 2000 to August 2001; Graduate Research Associate, School of Industrial Engineering and Management, Oklahoma State University from January 2002 to December 2006; Graduate Research Associate, Department of Biosystems Engineering, Oklahoma State University from January 2007 to January 2008; Associate with TransSolutions LLC., Dallas, TX from February 2008 to present.

Professional Membership: Alpha Pi Mu