Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

PERFORMANCE MODELING AND ANALYSIS FOR A LARGE HETEROGENEOUS DISTRIBUTED SYSTEM: UCLA-SEASnet

Joseph Betser
Alberto Avritzer
Jack W. Carlyle
Walter J. Karplus

September 1988
CSD-880073

Performance Modeling and Analysis for a Large Heterogeneous Distributed System: UCLA-SEASnet*†

Joseph Betser, Alberto Avritzer‡
Jack W. Carlyle, Walter J. Karplus
UCLA Computer Science Department
Los Angeles, CA 90024
betser@cs.ucla.edu / (213) 206-3584

August 25, 1988

Abstract

We address the performance analysis of Locus[1] [PopeWalk85] family - distributed and heterogeneous networks rendering parallel PC-Interface[2] service. We first present the SEASnet environment - a trend setting educational computing paradigm. We derive the workload characteristics within SEASnet, and define three classes of users: *Interactive, High-Interactive, and Communication-Intensive.* We present a series of performance measurements we conducted with the various user classes. We then construct a queueing network model that describes SEASnet. Emphasis is made on simplicity and functionality of the model. We show the model to be valid under various load patterns. We suggest ways to synthesize such systems from user requirements, as well as ways to balance the load and achieve better user service and resource utilization.

---

[*] UCLA-SEASnet - UCLA School of Engineering and Applied Sciences network

[†] This work was sponsored in part by the UCLA-IBM Joint Study D850915

[‡] Sponsored in part by CNPq-Brazil grant 200366/86

[1] Locus is the predecessor to the forthcoming IBM AIX/370

[2] PC-Interface (PCI) is a trade mark of Locus Computing Corporation (LCC)

1 Introduction

Locus is the predecessor of the newly announced IBM AIX/370. It is a Unix compatible distributed operating system that supports a large variety of architectures. Some of the mainframe architectures supported are 4300, 9370 and the 3090. A set of such backbone servers is called a "Transparent Computing Facility" cluster (TCF cluster). Personal workstations such as high-end PS/2 and PC/RT are supported as well, and they can be used to access a TCF backbone cluster.

The UCLA School of Engineering and Applied Sciences Network (SEASnet) is the primary computing facility of the school [Stenstro87]. SEASnet also serves as a pilot/demonstration center, as it is the most experienced production environment for pre-AIX/370. SEASnet demonstrates a high degree of heterogeneity, as it consists of two 4361s and a 4381 as backbone servers (as well as a number of DEC VAX/750s, SUNs, FPS, and others as shown in Figure 1). On the client side, SEASnet users use primarily IBM PC/ATs running DOS to access the system (ascii terminals and PC/RTs are also available).

In this work we shall concentrate on the PC to backbone-cluster connection. This connection is accomplished by an operating system bridge called PC-Interface (PCI). PCI allows a PC running DOS to transparently access an AIX or Unix file system through the so-called network drive (Drive E:). Many such sessions run in parallel allowing many users to work concurrently on their network file systems.

We will show an analysis of SEASnet that gives us modeling power to describe such systems and further synthesize new configurations.

In section 2 we present the user load that SEASnet is subjected to under daily operation. These load patterns have led us to the definition of the user classes for this work. We shall outline the procedures we used to help us characterize SEASnet and its user load.

In section 3 we describe the procedures that were used to measure and characterize the components of SEASnet, such that a system model could be derived.

Section 4 describes the queueing network model that we constructed, as a result of measuring and analyzing SEASnet under a variety of load conditions. Emphasis is made on simplicity and effectiveness of the model.

Section 5 presents validation results for the model. We show model predictions as well as measured results, such that the fit can be observed.

In section 6 we present a brief review of an important application. We

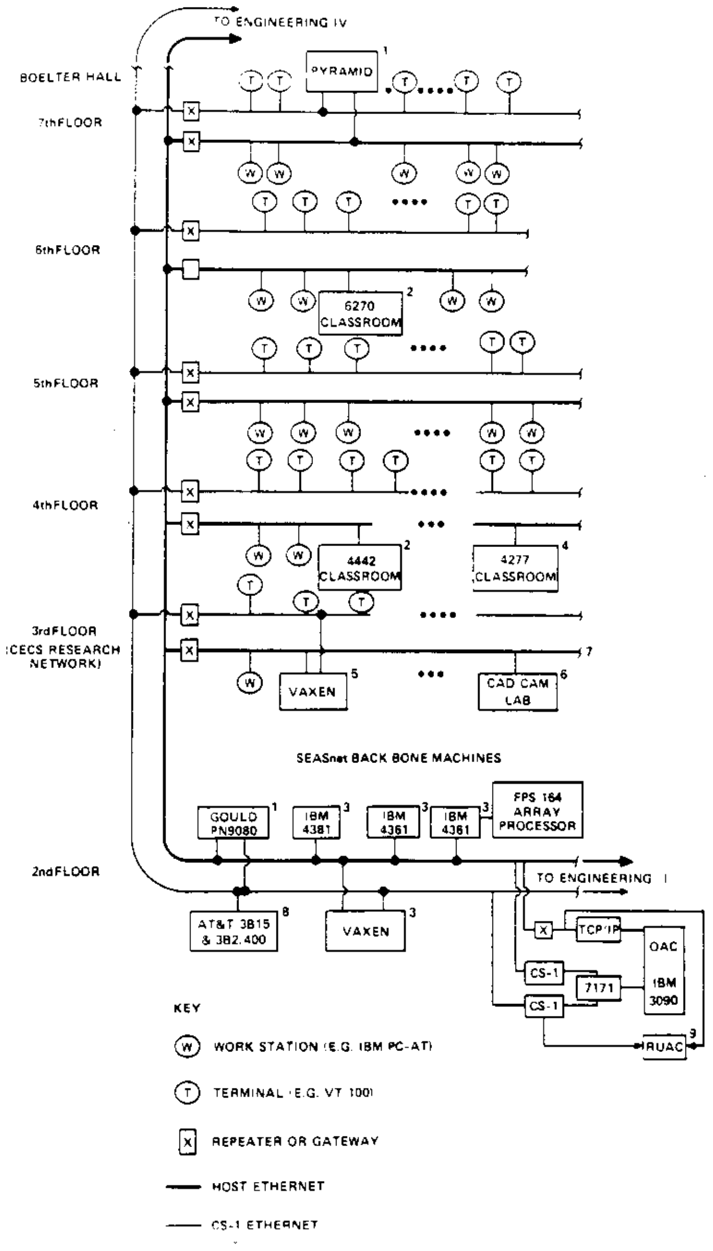


Figure 1: SEASnet - General Layout within the School of Engineering

present a configuration synthesis effort based on our model, and ultimately show the cost/performance curves that readily yield the configurations of choice.

Section 7 caps this paper with conclusions and suggested directions for further research.

2 SEASnet Workload Characterization

Our focus for this work is on the PCI networking option. PC-Interface allows any number of the 100+ PCs in the school to access any of the backbone machines. Figure 2 shows the part of SEASnet on which we focus in our analysis. The backbone servers provide extensive parallel file service for class instruction, as well as support for research work.

We have studied the usage patterns of the user community in SEASnet in two parallel approaches :

1. We surveyed and characterized the various engineering classes in terms of the computational load on SEASnet.
2. We measured the actual traffic on SEASnet during the quarter.

Based on our study of the SEASnet user behavior [Betser88], we introduced three classes of users in our characterization:

1. Interactive (Int) : Short interactive commands (list directory, copy short file over the network) interleaved with 60 seconds think time. This corresponds to mostly localized PC-DOS work, with occasional file service from the backbone network.
2. High Interactive (HI) : Short interactive commands in succession (as in Int), but only 1-2 seconds think time. This corresponds to higher *interactive* network use than most users exercise.
3. Communication Intensive (CI) : Massive continuous file transfer from backbone servers to PCs. This is the highest traffic load that a single PCI session can exercise on the backbone network. Several such sessions can saturate a backbone server. Only few users present such load to the system, and for very short periods of time.

We surveyed several of the instructors who were using SEASnet to teach their courses. We have found that most classes required service corresponding to a combination of classes Int and HI, with a few classes requiring

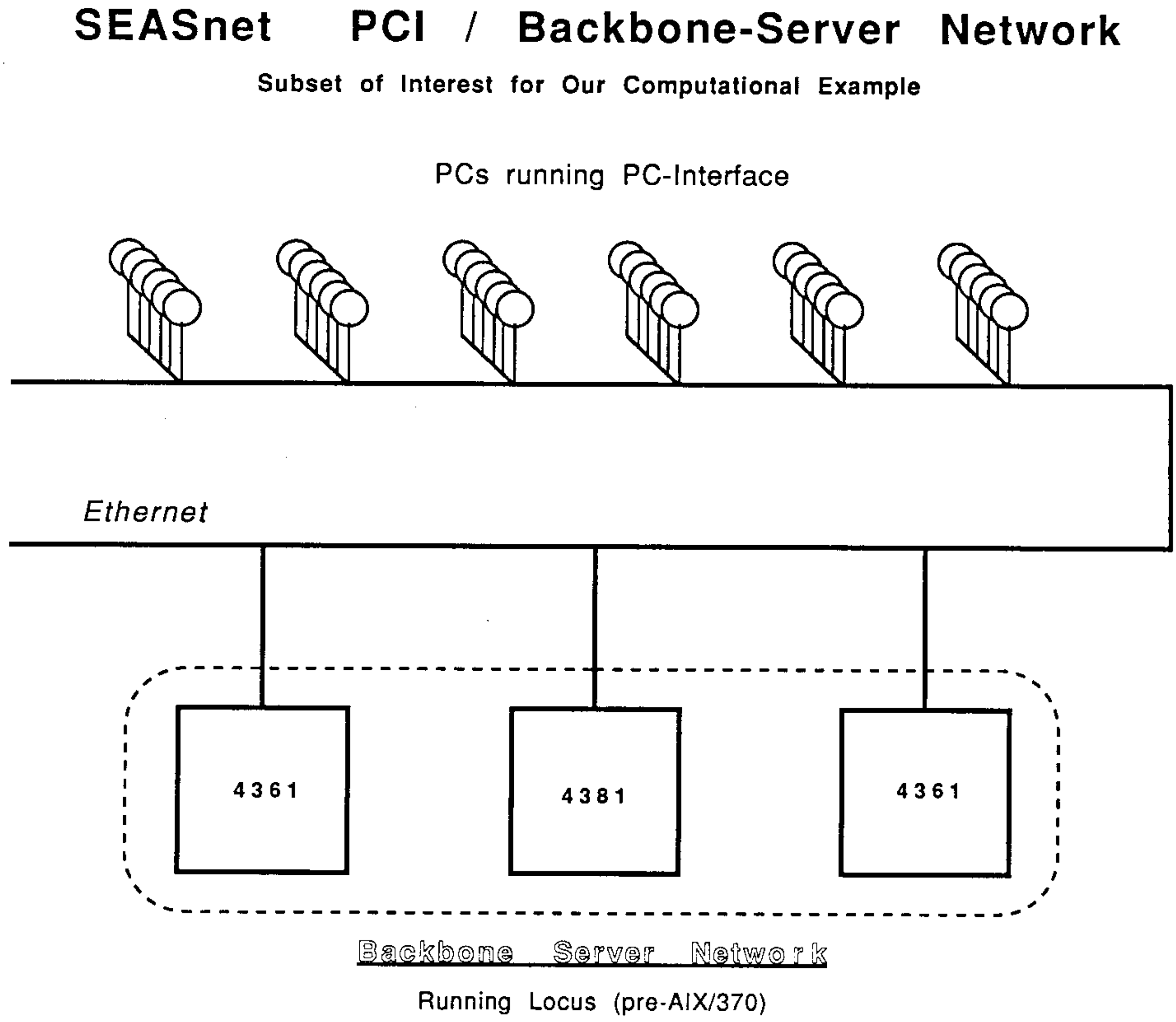


Figure 2: SEASnet - PCI / Backbone-Server Network

somewhere between HI and CI. A detailed report of this study is presented in [Betser88]. We shall mention here that measured traffic on SEASnet shows similar trends, with apparent greater user dependence on local PC resources, such that actual traffic is less than expected by the faculty in charge (20-30 Kbyte/sec).

The three user classes defined above will be used throughout the paper to describe the various measurements, modeling, analysis, and synthesis efforts presented.

3 Performance Measurements

Extensive measurements were carried out on SEASnet in order to parameterize its resources and construct a queueing network model. In addition to traffic measurements that were mentioned in the previous section, many other measurements were made on SEASnet, in a controlled environment.

Our goal was to construct a simple, yet accurate model for SEASnet. Hence we were interested in the most significant resource contention points within SEASnet. Thus we have conducted measurements from the PC classrooms, as well as directly on the backbone servers. Measurements were conducted during times that SEASnet was not used (school vacation, maintenance etc.), such that the entire system resources were available for study. PCI sessions were initiated, and scripts corresponding to the various user classes were run on the PCs. Measurement of system response time, local and global throughputs were made, as a function of the number of PCI sessions[3]. Such measurements were done for several different types of backbone servers. Additionally, measurements were made directly on the servers, such as I/O, cpu, and communication speeds. The details of measurements are reported elsewhere [Betser87, BeLaCaKa87, Betser88]. We present in Figure 3 some illustrative samples.

As a result of these measurements we have obtained I/O rates for the various backbone machines, as well as cpu timings and overheads for file and network service. Additionally, we have obtained PC communication bounds for the Ethernet cards, as well as maximum screen throughput for screen driven outputs. As an example, maximum PC-Ethernet communication rate was clocked at 25 KByte/sec[4]. This was done by using class CI jobs

---

[3] We note that the number of users is equal to the number of PCI sessions in our architecture

[4] Ethernet 3Com board

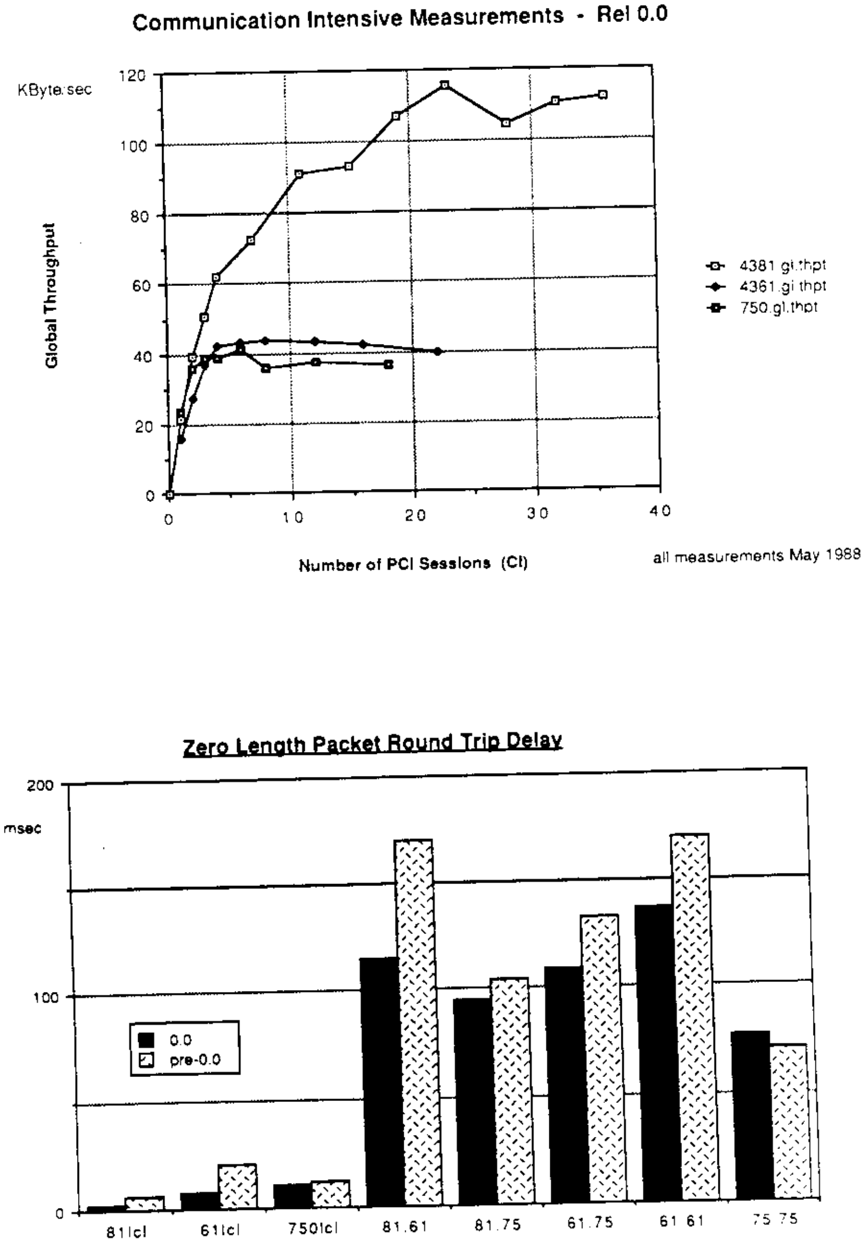


Figure 3: Illustrative Samples of Measurement Sets Conducted on SEASnet

for measurements. An important system response benchmark was the delay incurred in a 1 Mega-Byte file transfer. This is a crucial yardstick for a file service oriented system. We used this benchmark during experimentation with jobs of other classes, such as Int and HI. We also used this measure in the synthesis example of section 6.

The next logical step is the model construction, which is described in the following section.

4 The Performance Model

The performance figures for workloads with customers of different classes were derived through measurements, written specifications, and simulation. With the aid of many such experiments, under various configurations and loads combinations, we have iterated and identified the contention points for SEASnet. We then constructed the queueing network model shown in Figure 4. This model describes a Locus backbone server supporting a variety of PCI sessions.

We model the primary governing parameters of the Locus-PCI operation paradigm. It is important to note that we present the contention points as the CPU speed and I/O capacity of the backbone servers. We also present a bottleneck at the PC end, in the way of the Ethernet interface speed. We have observed that these are the primary resource contention modules within this architecture[5].

We describe now some of the modeling paradigms we incorporated into this *first order effect* model.

1. I/O device speed. The effective available throughput is available from observing backbone servers at a saturated situation.
2. CPU service. Implementation of Locus on different architecture results in machine-specific services. These services are related to layered operating system strategies. We lump these effects into equivalent machine instruction execution, the cpu service.
3. Buffered I/O. Ethernet packets are 1KBytes long, and I/O pages are 4Kbyte long. The I/O buffers are used to minimize I/O operations. Hence, 3/4 of the time Ethernet packets are provided directly from the buffer, without physical I/O operation.

---

[5] The Ethernet is an example of a non-bottlenecking resource. We have never gotten even close to full Ethernet throughput during our extensive experimentation

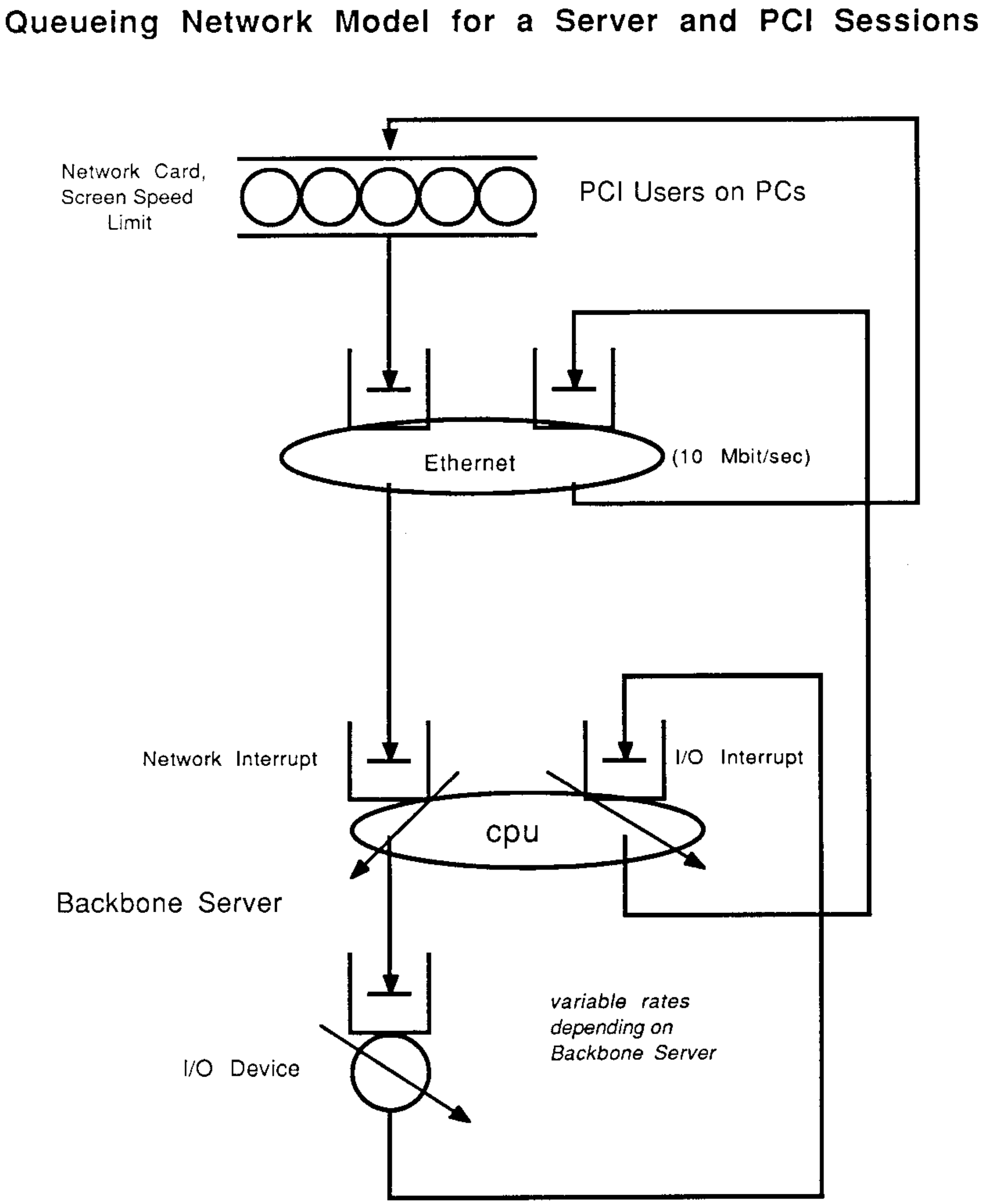


Figure 4: Queueing Network Model for a Backbone and PCI Sessions

4. Job Swapping. Jobs of class Int have 60 second think times. During that time the dos-server on the backbone can get swapped out of memory. This activity is modeled by increased I/O to accommodate the swap operation.

5. Variable Think Time. What sets apart jobs of class Int from jobs of class HI is the think time at the PC end. By modifying this parameter, we have been successful in modeling both classes.

6. Ethernet Card at PC. This contention point is a significant bottleneck for the CI jobs. The service rate for this part of the model was determined from measurements and incorporated into the queueing network.

This model was used to evaluate system performance by discrete event simulation techniques.

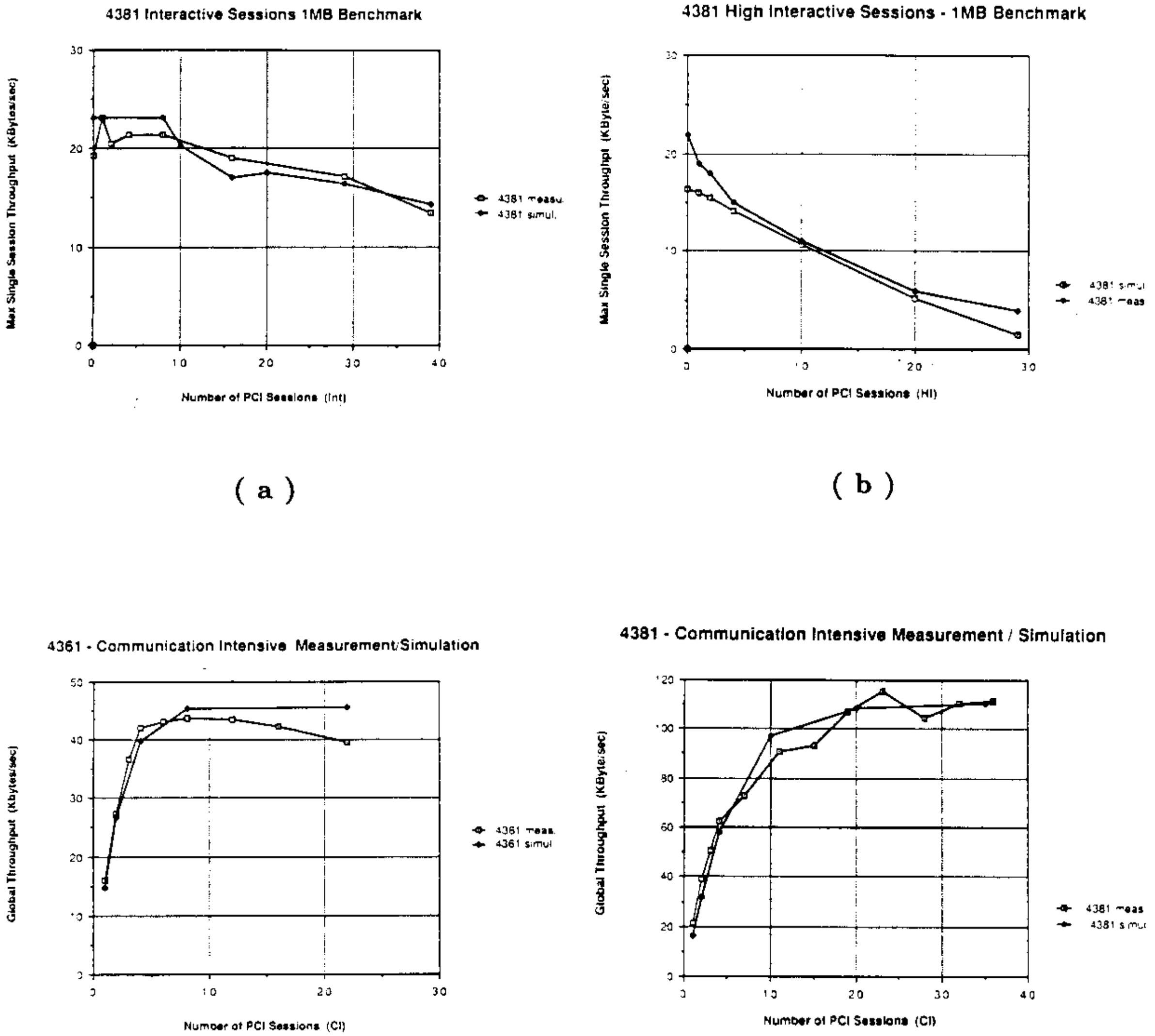
5 Performance Model Validation

The tuning and validation of of this model was a long iterative process, and is reported in detail in [Betser88]. We present in Figures 5 and 6 some recent comparisons between measured results and simulated results generated by the queueing network model.

We present results for all three classes, and for both 4381 and 4361 architectures. Class Int on 4381 is presented on plot 5(a). Class HI on 4381 is presented on plot 5(b). Plots 5(c) and 5(d) show CI users on 4361 and 4381 respectively. Plots 6(a) and 6(b) describe combinations of HI and CI classes on 4361 and 4381 respectively. We note that model predictions and measured system performance are in good agreement throughout. This gives us increased confidence in the model. We are now ready to embark on modeling of future configurations.

In fact, our model was used to simulate the configurations that we report in the application example of section 6. These extensive simulations were utilized to construct the delay space tables. These tables were in turn used as input to the optimization and synthesis algorithms we subsequently executed.

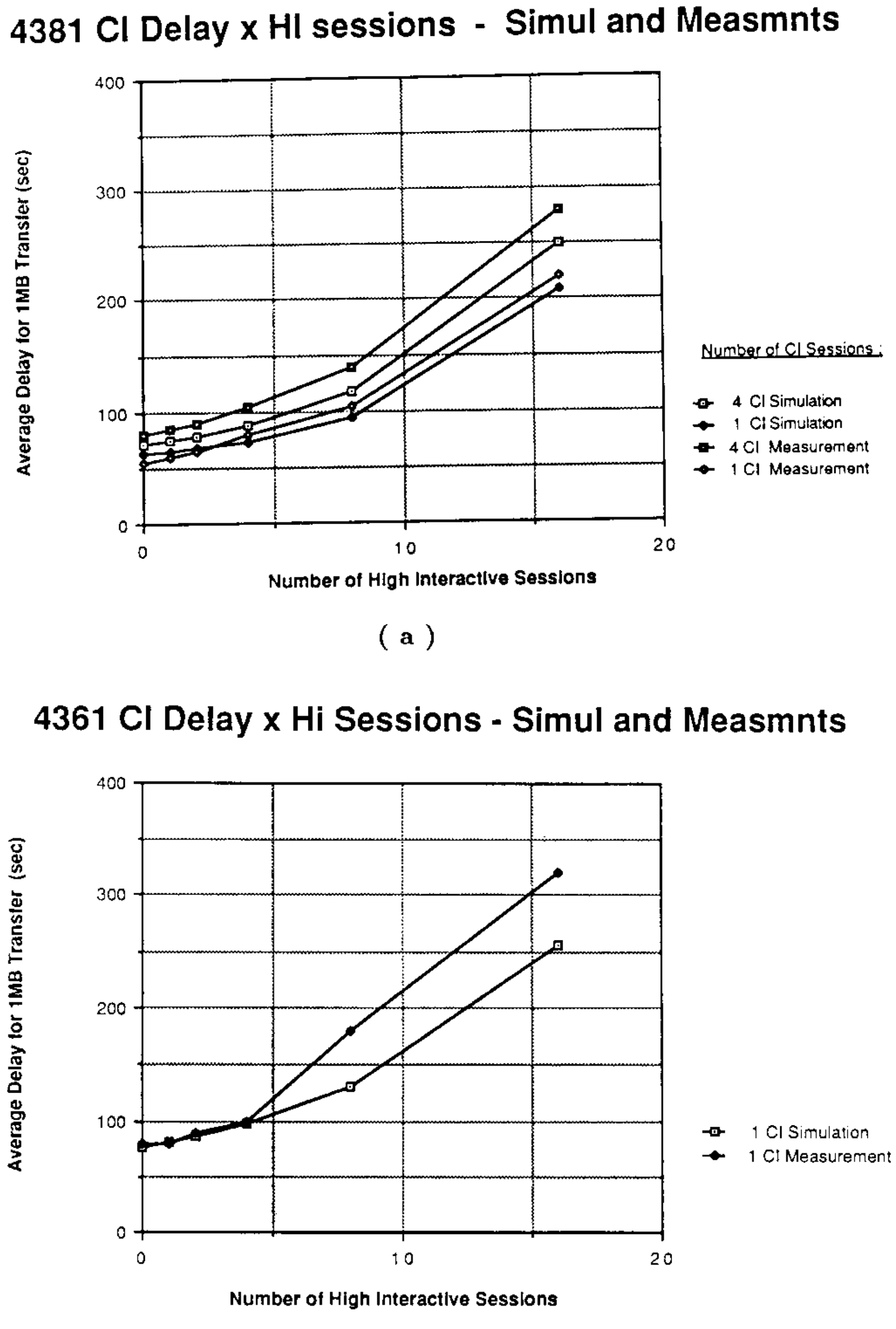
Having described the model construction and validation, we are now ready for a brief presentation of a recent application we completed. In the next section we present the results of the configuration synthesis algorithms we developed.



(a) (b)

(c) (d)

Figure 5: Comparison of Measurement and Simulation Results



(a)

(b)

Figure 6: Comparison of Measurement and Simulation - Multiple Classes

6 Model Application - Configuration Synthesis

The model we developed is of a general nature. To illustrate its application, we present the results of configuration synthesis algorithms we developed recently. Our algorithms used for input the parameter tables that were generated using the queueing network model we presented in this paper. The Synthesis algorithms are an extension of the Capacity and Flow Assignment (CFA) [Kleinro76], and Flow Deviation Method [FraGerKle73]. We used the session support capacity of the backbone servers as the major resource for allocation.

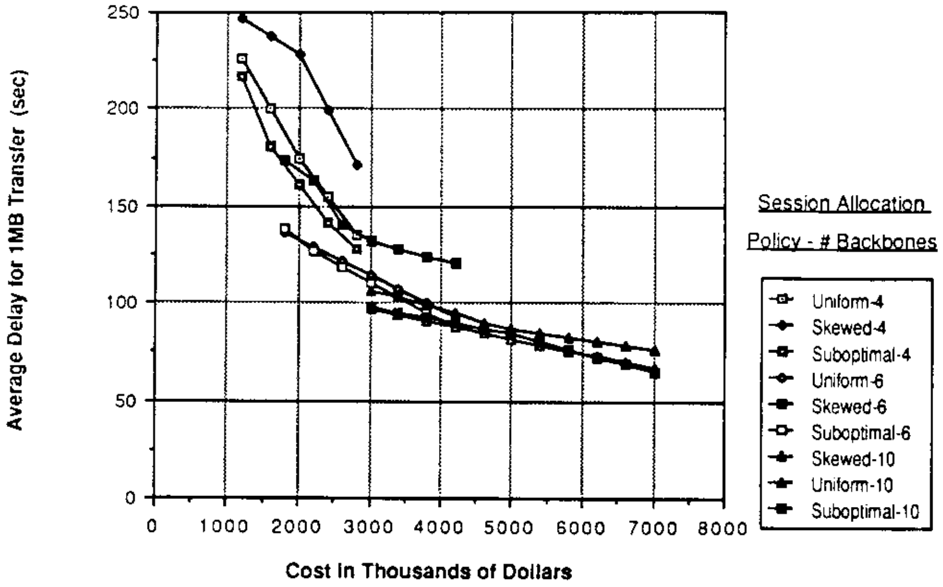
The problem we addressed consists of a given user load, distributed among the three session classes. Also given is the total number of backbone servers and their respective costs. Our algorithms devise the optimal session allocation for each configuration set. We can easily examine the cost-performance curves of these optimal allocations and apply our cost and performance constraints to arrive at the most suitable configuration to service the user demands.

Complete description of the algorithms can be found in [Betser88, BeAvCaKa88]. We present here the results of such an optimization for a specific example. We are given with a user load of (48Int,24HI,18CI - total 90 sessions). We examine topologies consisting of 4, 6, and 10 backbone servers. The backbone servers are of type 4361 and/or 4381. The costs are illustrative costs, based on vendor provided information.

We plot our final results in Figure 7. The optimization results are compared with two other session allocation policies - uniform allocation, and skewed allocation. Clearly, the optimization results in better cost-performance across the board. It should also be noted that there is a pronounced knee to the curves, which corresponds to the recommended working range for a good solution.

7 Conclusions and Future Research

In this work we considered the analysis of a heterogeneous distributed system on which many workstations, users, sessions, and backbone servers execute in parallel. We have derived a model which provides simple means to evaluate system performance through queueing network simulation. The model is based on system measurements and simulation, and is the result of a refined, iterative design process. By emphasizing the points of resource contention


Cost Performace with 4,6, & 10 Backbone Servers

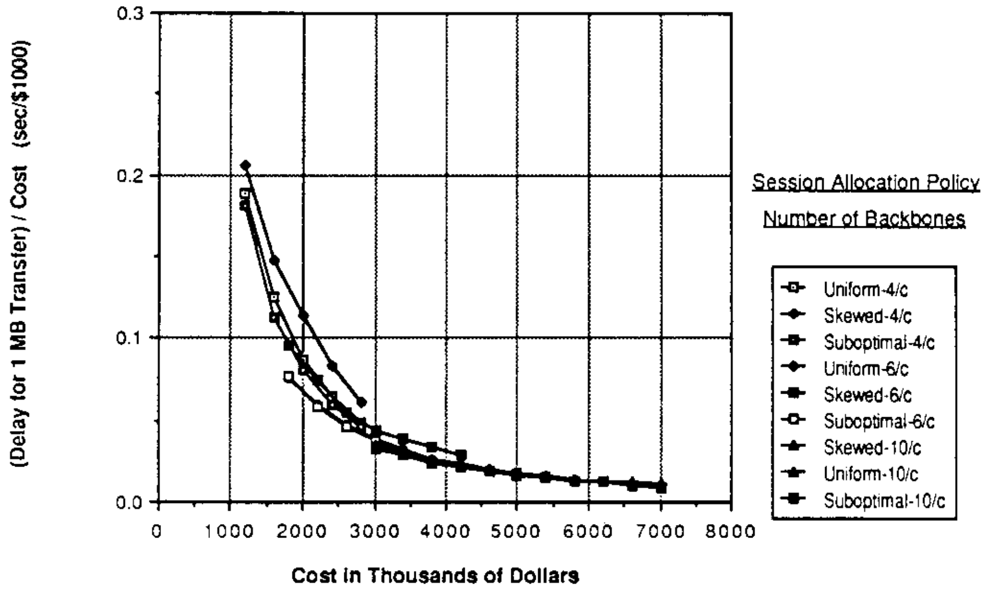

Relative Cost Performance with 4,6,10 Backbone Servers

Figure 7: Optimization Results for Configuration Synthesis

on our heterogeneous resource set, we have been able to achieve substantial reduction in complexity for our model. We have shown recent comparisons between results predicted by the model and figures measured directly on the system - good agreement was demonstrated. To demonstrate the generality of our model, we further presented an application which employs the model to obtain backbone configuration synthesis for distributed systems of these characteristics.

Numerous other extensions could be pursued using our results as a departure point. Firstly, additional effort could be invested to further refine this model, and include second order effects, to enhance the accuracy of the PCI-Backbone model. The model could be extended to include additional user classes, such as interactive users, and X-windows users with graphics capabilities. Issues of reliability and distributed file access could be incorporated. Clearly, additional complexity and computational costs will be involved for many of these model extensions. In the way of applications to the model, an important area to pursue is the area of load balancing. Resource utilization is important also after the system synthesis is complete. Available resources should be assigned to changing tasks in either static or dynamic policies [EagLazZah88, TantTowWol88].

A number of these issues are currently under investigation at UCLA.

References

[AvrGer88] A. Avritzer and M. Gerla "Load Balancing in a Distributed Transaction System", In preparation.

[BaChMuPa75] F. Baskett, K. Chandy, R. R. Muntz, and F. Palacios, " Open, Closed, and Mixed Networks of Queues with Different Classes of Customers", JACM Col 22, pp 248-260, April 1975.

[Betser84] J. Betser, "Performance Modeling and Enhancement within the Locus Distributed System", M.S. Thesis, UCLA Computer Science Department, 1984.

[BetGerPop84] J. Betser, M. Gerla, and G. J. Popek, "A Dual priority MVA Model for a Large Distributed System : LOCUS", Performance '84, Proceedings of the 10th International Conference on Computer Performance (ed. E. Gelenbe), Paris France 19-21 December 1984, pp 51-66.

[Betser87] J. Betser, "Performance Evaluation and Prediction for Large Heterogeneous Distributed Systems", Ph.D. Dissertation Prospectus, UCLA Computer Science Department, Los Angeles, CA 90024, 1987.

[BeLaCaKa87] J. Betser, Nick Lai, Jack W. Carlyle, Walter J. Karplus, "LOCUS and SEASnet - Performance Analysis", Technical Report, UCLA Computer Science, 1987.

[Betser88] J. Betser, "Performance Evaluation and Prediction for Large Heterogeneous Distributed Systems", Ph.D. Dissertation, UCLA Computer Science Department, Los Angeles, CA 90024, 1988.

[BeAvCaKa88] J. Betser, A. Avritzer, J. W. Carlyle, W. J. Karplus, "Configuration Synthesis for a Heterogeneous Backbone Cluster and a PC-Interface Network", UCLA Computer Science Dept, Los Angeles, CA 90024, 1988 (submitted).

[CheCarKar86] Shun X. Cheung, Jack W. Carlyle, and Walter J. Karplus, "Asynchronous Distributed Simulation of a Communication Network", in Proceedings of the Summer Computer Simulation Conference, Society for Computer Simulation, July, 1986.

[DeSMunLav84] E. de Souza e Silva, R. R. Muntz, and S. S. Lavenberg, "A Clustering Approximation Technique for Queueing Network Models with a Large Number of Chains", IEEE Transactions on Computers, Vol C-35, No 5, May 1986.

[DeSoGerl84] E. de Souza e Silva and M. Gerla, "Load Balancing in Distributed Systems with Multiple Classes and Site Constraints", Proceedings Performance '84, E. Gelenbe (ed), Paris 1984, North Holland, pp 17-33.

[EagLazZah88] D. Eager, E. Lazowska, and J. Zahorjan, "The Limited Performance Benefits of Migrating Active Processes for Load Sharing", Sigmetrics 88, May 1988.

[Fox66] B. Fox, "Discrete Optimization Via Marginal Analysis", Management Science, Nov 1966.

[FraGerKle73] L. Fratta, M. Gerla and L. Kleinrock, "The Flow Deviation Method - An Approach to Store-and-Forward Communication Network Design", Networks 3, 1973.

[Kleinrock76] L. Kleinrock, "Queueing Systems, Volume II: Computer Applications", Wiley-Interscience, New York, 1976.

[Lavenber83] S. S. Lavenberg (editor), "Computer Performance Modeling Handbook", Academic Press, 1983.

[LaZaGrSe84] Edward D. Lazowska, John Zahorjan, G. Scott Graham, and Kenneth C. Sevcik, "Quantitative System Performance - Computer System Analysis Using Queueing Networks Models", Prentice Hall, Englewood Cliffs, New Jersey 07632, 1984.

[PopeWalk85] G. J. Popek, B. J. Walker, "The Locus Distributed System Architecture", MIT Press, Cambridge Mass, 1985.

[Stenstro87] Michael K. Stenstrom, "SEASnet - A Network for Educational Computing", Proceedings 1987 ASEE (American Society for Engineering Education) Conference, Reno, Nevada, pp. 1540-1543, 1987.

[TantTows85] A. N. Tantawi and D. Towsley, "Optimal Static Load Balancing in Distributed Computer Systems, JACM, Vol 32, No 2 April 1985, pp 445-465.

[TanTowWol88] A. N. Tantawi, D. Towsley and J. Wolf, "Optimal Allocation of Multiple Class Resources in Computer Systems", Sigmetrics, May 1988, pp 253-260.

[WoodTrip86] C. M Woodside and S. K. Tripati, "Optimal Allocation of file Servers in a Local Network Environment", IEEE Trans. on Software Engineering, Vol 12, No 8, August 1986, pp 844-848.