Proceedings of the 1987 Winter Simulation Conference
A. Thesen, H. Grant, W. David Kelton (eds.)

MODELING AND ANALYSIS OF TWO-ECHELON RESOURCE ALLOCATION STRATEGY IN PACKET SWITCHES

S.S. Liu
GTE Laboratories Inc., TRL, MS-48,
Waltham, MA 02254

J. Wang
Prime Computer Inc.,
Prime Park, Natick, MA 01760

A. Hsu
CONTEL Business Network,
130 Steamboat Rd., Great Neck, NY 11024

ABSTRACT

In this paper, we propose and analyze a two-echelon resource allocation strategy for packet switches. The strategy improves system responsiveness, reduces intrasystem communication, and is adaptive in handling traffic fluctuations. Two types of packet switch resource are considered here: packet buffers and logical channel numbers. The way of allocating these resources within a packet switch can directly impact the switch's throughput and call-processing capability. Performance of the strategy has been analyzed by a decomposition algorithm along with discrete event simulations. Numerical examples are given to demonstrate the uses of the algorithm and simulation to configure two-echelon allocation systems.

1. INTRODUCTION

A common problem in designing telecommunication systems is determining an effective way of allocating system resources, like buffers or channels, in order to attain desired system performance. Due to the reusability of telecommunication resources, the problem is different from classic inventory problems in which perishable or repairable items are considered. In addition, most telecommunication systems need two-echelon allocation strategies in order to maximize resource utilization under fluctuating demands and, at the same time, to provide fast services to delay-sensitive traffic.

In this paper, we propose a two-echelon resource allocation strategy for packet switches. The strategy is adaptive to traffic fluctuations and can be implemented in a distributed processing environment. Two types of packet switch resource are of concern: packet buffers and logical channel numbers. In packet switches, the availability of buffers and logical channel numbers determines the switching throughput and call set-up rate, respectively. Performance of the strategy has been analyzed by a decomposition algorithm along with discrete event simulations. Numerical examples are given to demonstrate the uses of the algorithm and simulation to configure two-echelon allocation systems.

Sherbrooke (1968) proposed the well-known METRIC Model for determining optimal stock levels for repairable items in a two-echelon setting. The allocation problem in this study differs from the METRIC and other similar models (Demmy and Presutti 1981, Graves 1982, and Nahamias 1981) in two aspects: (1) The current problem requires that resource items be returned to the central pool, i.e., the top echelon, after usage. In the METRIC model, however, a repairable item can possibly be fixed locally and returned for use at the original base station. (2) In our study, interaction between the two echelons is governed by a batch request/grant policy as opposed to the one-for-one policy, i.e., the (S–1, S) policy, found in the METRIC and other models. The problem of buffer management in packet switches has been considered by M. Irlant (1978) and J. Kaufman (1981); however, their studies were restricted to single-echelon policies, and are not applicable to the current problem.

This paper is organized as follows: In Section 2 we present the two-echelon resource allocation strategy. In Section 3 a heuristic decomposition algorithm for analyzing the strategy is described. In Section 4 a discrete event simulator is presented. In Section 5 three numerical examples are presented and discussed. Conclusions are given in Section 6.

2. A TWO-ECHELON RESOURCE ALLOCATION STRATEGY

In this section, a two-level resource allocation strategy is proposed for packet switches. The design of the strategy is aimed at improving system responsiveness, reducing internal communication, and meanwhile maintaining some degree of adaptiveness to traffic fluctuations. Without losing any generality, we use buffer allocations in packet switches as an example to illustrate the strategy.

Figure 1 shows a typical system configuration to which the strategy applies. The system consists of three components: (1) a central pool of buffers — the top echelon, (2) a group of line cards (LC) — the lower echelon, and (3) a manager in charge of allocating buffers from the central pool to the LCs. In packet switches, each line card (LC) is responsible for the packet receiving and transmission over a group of transmission lines.

Each LC has a local pool of buffers in order to aviod any delay in the packet receiving process. Upon the arrival of each packet, the servicing LC assisgns a buffer to the packet and reduces the local inventory by one. When the local inventory drops down to a reorder point, the LC asks the manager for a package of M buffers. Here, M is also the number of buffers allocated to each LC at system start-up

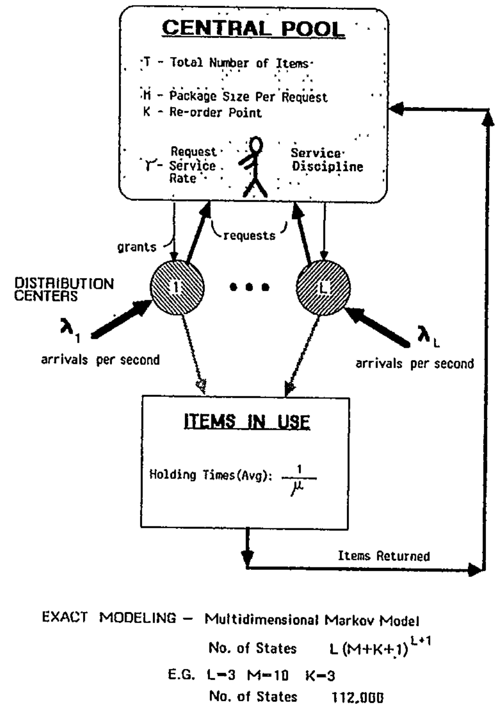


Figure 1: Two-Echelon Resource Allocation System

time. Based on the inventory level at the central pool, the manager responds to each request either with a full grant or with a partial grant of less than M buffers. In the case of a partial grant, if the local inventory is still below the reorder point, a subsequent request is issued immediately. Buffers released by packets are returned directly to the central pool. With the strategy defined as above, a decomposition algorithm is developed to analyze its performance in terms of packet blocking probability under the following assumptions: Poisson arrivals at each LC, an exponential buffer holding time distribution, an exponential request service time distribution, and the service-in-random-order discipline for the manager.

According to the above descriptions, the strategy can be fully specified by the following parameters:

T: Total number of buffers in the system

L: Number of local centers (or line cards)

M: Maximum package size of each grant

K: Reorder point at each LC

λ: Poisson arrival rates for each local center.

γ: Pool manager's exponential service rate.

$1/\mu$: Mean holding time for each item in service

These parameters also serve as inputs to the decomposition algorithm to be described in the next section.

3. DECOMPOSITION ALGORITHM

The buffer allocation strategy described in Section 2 can be precisely modeled as a multi-dimensional Markov model with the number of states proportional to $T(M+K+1)^L$. Because of the large number of states involved, it is sometimes impossible to perform direct analysis on the model for realistic systems. To avoid the problem of state space explosion, we propose a new function-level decomposition technique for modeling the strategy. The new technique is unique in that it decomposes the original system into several functionally related subsystems. By modeling each subsystem individually and defining mathematical bindings among these subsystems, we are able to iteratively solve for the steady-state solution of the original system. In contrast to other decomposition techniques that perform state space decomposition in original models, the new technique is easier to implement because it does not require an exact model to begin with.

In the study of the two-echelon resource allocation problem, the original system is divided into three subsystems: (a) A local system consisting of several one-dimensional local center models, (b) An S/N system consisting of a two-dimensional (S,N) model with S and N representing the number of items in service and in the central pool, respectively, and (c) A request system with a request service model characterizing the interactions between request and grant processes. To make the decomposition mathematically tractable, we adopt the service-in-random-order discipline for the request service model.

Figure 2 shows graphically how the original system in Figure 1 is decomposed into these subsystems. Each local center (LC) in the system is modeled as a one-dimensional Markov model with $(M+K+1)$ states, as shown in Figure 3. Each state represents the number of available items in the LC. Transitions occurring with a rate of λ denote the arrivals of demands at a LC; those with rate γ's are due to the receipt of grants from the pool manager. Only states corresponding to an inventory less than or equal to K items have the γ transitions. The derivation of γ is explained later. Each LC serves its own arrival stream and is assumed to be independent of other LC's. The probability of being in state 0 in a LC model is essentially the blocking probability observed at the LC.

DECOMPOSITION TECHNIQUE

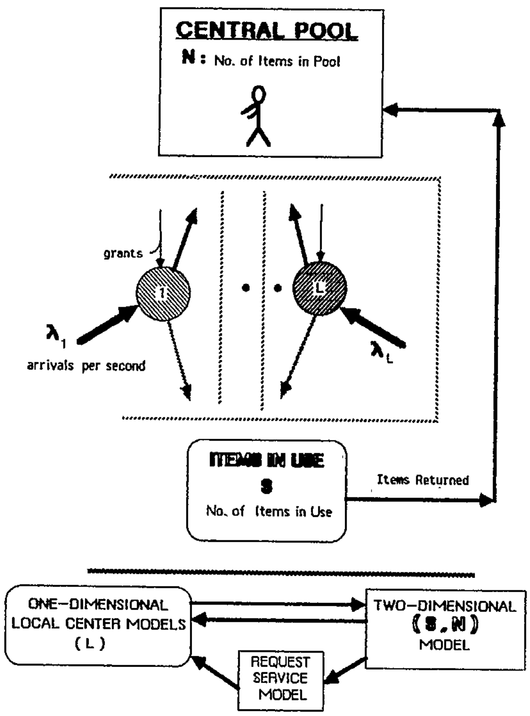


Figure 2: Decomposition of Two-Echelon System

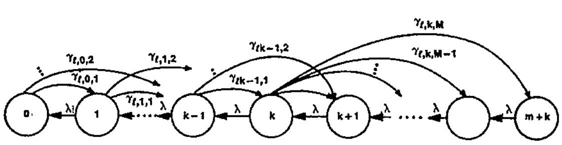


Figure 3: A Local Center Model

The (S,N) model partially characterizes the item distributions in the system. Here S represents the number of items in service, whereas N stands for number of items in the central pool. Let £ be the total number of items in all LC's, then S+N+£ is equal to T, the total number of items in the system. A typical (S,N) model is shown in Figure 6(b) in which each state can make three possible transitions to its adjacent states. See Figure 4. Path 1 occurs at a rate of $S \cdot \mu$, corresponding to a service completion. Path 2 takes place with a rate g(£) when a grant is offered to a requesting LC. Path 3 happens at a rate a(£) when an item is put in service. Both a(£) and g(£) are functions of the quantity £. For a given £, a(£) and g(£) represent the weighted averages of grant receipt and demand arrival rates, respectively.

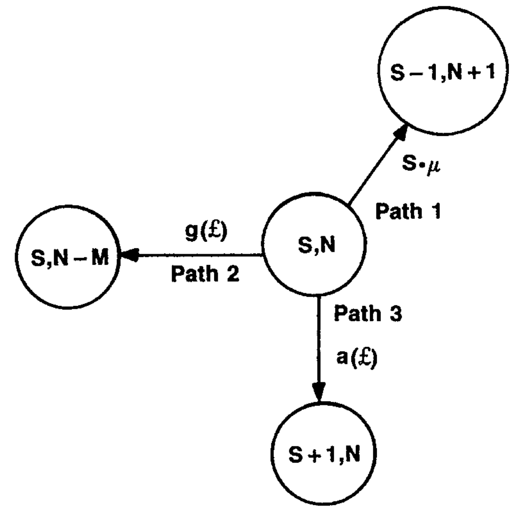


Figure 4: A (S,N) State with Three Transitions

The decomposition algorithm can be represented by the flow diagram in Figure 5. Since it is a closed loop system, the algorithm may begin at any of the decomposed parts. We choose to start at the local center models by assuming a steady-state solution for each of the local center models. With the assumed local center solutions, a dynamic programming (DP) model is employed to calculate four essential quantities:

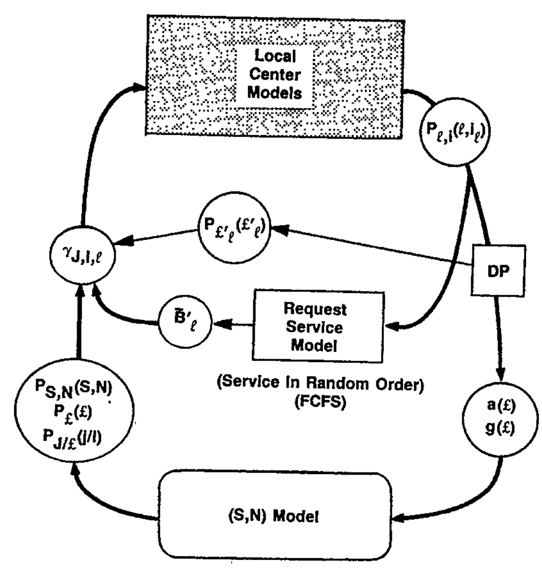


Figure 5: Decomposition Algorithm

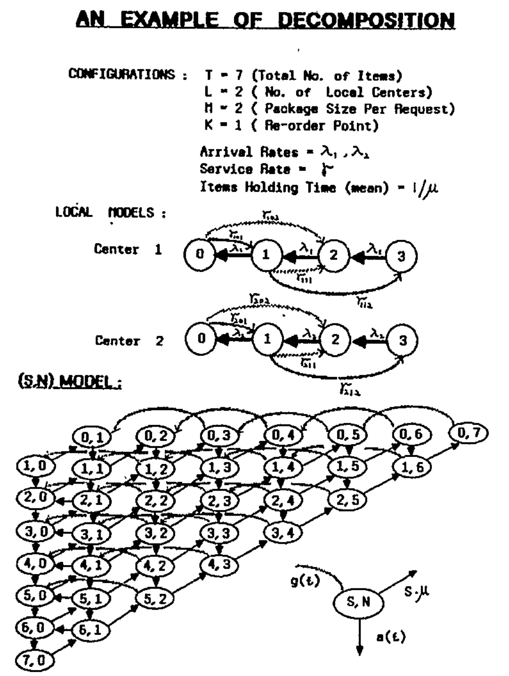


Figure 6: Construction of (S,N) and Local Models

$P_0(\pounds)$: the probability that at least one LC is making a request for a given £

$P_{\pounds'_\ell}(\pounds'_\ell)$: the probability that there are $\pounds'_\ell$ items in all LC's excluding the ℓth LC

a(£): a transition rate in the (S,N) model denoting the average rate that items enter in-use status when there are £ items in all LC's

g(£): a transition rate in the (S,N) model specifying the average rate that a grant will be issued by the pool manager when there are £ items in all LC's.

For each of the possible £ values, the DP module solves a combinatorial problem of allocating £ items among all LC's. It keeps track of all possible allocation patterns, and uses the patterns along with the state probabilities from the LC models to calculate the above quantities. The a(£)'s and g(£)'s completely define the (S,N) model, which can then be solved for the (S,N) state probability, $P_{S,N}$ (S,N). With the (S,N) state probabilities, we can derive the probability that there are £ items in all local centers, $P_\pounds(\pounds)$, given by:

$$P_\pounds(\pounds) = \sum_{S=0}^{T-\pounds} P_{S,N}(S,\ T-\pounds-S) \qquad (1)$$

and subsequently the probability that there is a grant of size J given that there are £ items in local centers, $P_{J/\pounds}$ (J/£), using:

$$P_{J/\pounds}(j/\pounds) = \begin{cases} \dfrac{P_{S,N}(T-\pounds-j)}{P_\pounds(\pounds)} \cdot P_0(\pounds) & ,j=0,\ 1,\ldots,M-1 \\ \dfrac{\sum_{n=m}^{T-\pounds} P_{S,N}(T-\pounds-n,n)}{P_\pounds(\pounds)} \cdot P_0(\pounds) & ,j=M \end{cases} \qquad (2)$$

The above probability along with the probability $P_{\pounds'_\ell}(\pounds'_\ell)$ are needed to calculate the probability that there is a grant of size J given that there are I items in the ℓth local center, P_{J/I_ℓ} (j/i). This is given by:

$$P_{J/I_\ell}(j/i) = \sum_{\pounds'_\ell=0}^{(L-1)(M+K)} P_{J/\pounds}(j/\pounds=\pounds'_\ell+I_\ell)P_{\pounds'_\ell}(\pounds'_\ell), \qquad (3)$$

where $P_{\pounds'_\ell}(\pounds'_\ell)$ is calculated from the DP model.

Given that requests are serviced by the SIRO discipline, we can express the request grant rate of package size J when there are I items in the ℓth local center, ($\gamma_{J,I,\ell}$), as a function of P_{J/I_ℓ} (j/i), manager's service rate (γ), and the average request queue length excluding ℓth local center, ($\bar{B}'_\ell$). The request grant rate is then given by:

$$\gamma_{J,I,\ell} = \frac{\gamma}{1+\bar{B}'_\ell} \cdot P_{J/I_\ell}(j/i) \qquad (4)$$

The $\bar{B}'_\ell$ in (4) is calculated by the equation:

$$\bar{B}'_\ell = \bar{B} - P_B(\ell), \qquad (5)$$

where

$$\bar{B} = \sum_{\ell=1}^{L} P_B(\ell) \qquad (6)$$

and $P_B(\ell)$, the probability that the ℓth local center is in the request queue, can be calculated using:

$$P_B(\ell) = \sum_{i_\ell=0}^{K} P_{\ell,i}(\ell,i_\ell). \qquad (7)$$

In (7), $P_{\ell,i}(\ell,i_\ell)$ is the steady-state probability of the ℓth local center with i items. Summing this probability to the reorder threshold K gives the probability that the ℓth local is in a request/grant situation.

The γ's calculated in (4) redefine all the LC models for the next iteration. Each subsequent iteration starts with solving the new local center models, and then follows the sequence of steps specified in Figure 5. The above process is carried out iteratively, and will be stopped under two conditions: (1) The current total blocking probability converges to the previous value within the specified tolerance. (2) The number of iterations reaches a specified count. Figure 6 illustrates by an example the construction of the LC and (S,N) models for a simple system. It is clear that the complexity of the decomposition algorithm is proportional to the number of states in the (S,N) model, which can be determined by:

$$[L(M+K)+1]\ [T+1-\frac{L(M+K)}{2}]. \qquad (8)$$

For a system with $T=200$, $L=8$, $M=10$, and $K=3$, the (S,N) model contains 15,645 states as opposed to 2.9×10^{11} states in an exact model. The decomposition technique presented here has greatly alleviated the state space problem and provided an efficient means for estimating the performance of two-echelon resource allocation strategy.

4. SIMULATIONS

The assumption about the pool manager's SIRO (service-in-random-order) service discipline is not quite realistic. In most service systems, the first-come, first-served scheme (FCFS) is used to schedule services. To investigate the FCFS case, a simulation model is developed to estimate the blocking probability. The simulation model is based on the same assumptions made in the decomposition model except that the service discipline of the pool manager is the FCFS scheme. The results of the simulation are also compared to the results obtained from the decomposition model to reveal the accuracy of the decomposition model toward exact solutions. The comparisons show that in most cases the decomposition results are within 15% of the simulation results, as will be shown in the next section.

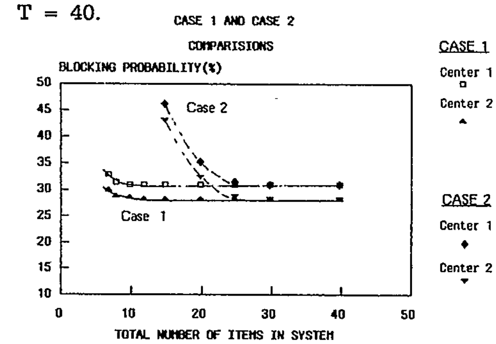
5. NUMERICAL EXAMPLES

In this section, we discuss three numerical examples.

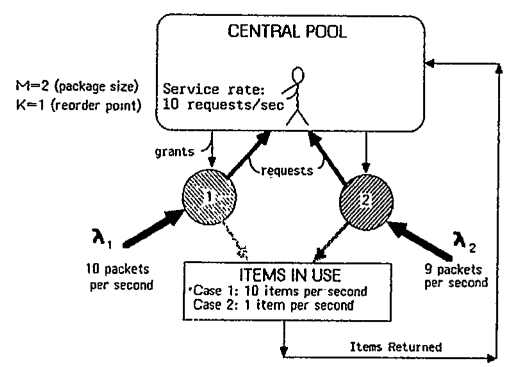
Example 1. Effects of different holding rates, μ

In this example, we compare the blocking probability experienced by the local centers for two different values of average holding time. In both cases, the total number of items in the system ranges from 7 to 40 items. There are two local centers with a maximum request grant size of two items and the reorder point at one item. The arrival rates at local centers 1 and 2 are 10 requests/sec and 9 requests/sec, respectively. The buffer manager's service rate is 10 requests/sec. The average holding times are 0.1 second and 1 second for case 1 and case 2, respectively. Figure 7(a) shows the configuration of both cases, and Figure 7(b) gives a graphical illustration of the results obtained from both cases. These results have been checked by three-dimensional exact models created for the SIRO discipline.

As expected, for each case, the local center with the higher arrival rate experienced higher blocking probabilities. However, a more interesting fact is that the blocking experienced by local center 1 and 2 of case 2 leveled off to the same value as that of the respective centers of case 1 as the total number of items in the system is increased to approximately 25 items and beyond. Since case 2 features a longer holding time, its blocking probability drops more significantly than those in case 1 as the T increases. This implies that, to some degree, a large T can make up the performance degradation due to long holding times. As is shown in Figure 7(b), when T is greater than 25, no performance gain can be realized by adding more items to the system. The flat region in Figure 7(b) suggests that another parameter, γ, is the limiting factor in determining the blocking probability. Indeed, if we change γ to 10^4 per second, the blocking probabilities in both cases drop nearly to zero at $T = 40$.



(b)



(a)

Figures 7(a) and (b): Example 1

Example 2. Single-Echelon versus Two-Echelon

In this example, we compare a single-echelon fixed allocation (FA) strategy to the two-echelon strategy under study. Assume there are 42 buffers to be shared among three line cards (LC), and each LC is receiving an incoming packet stream at the rate of 70.64 packets per second.

In the FA strategy, the 42 buffers are divided evenly among the three LC's, as shown in Figure 8. As a result, each LC gets 13 buffers, and the blocking probability at each LC is 0.03 according to the Erlang B formula.

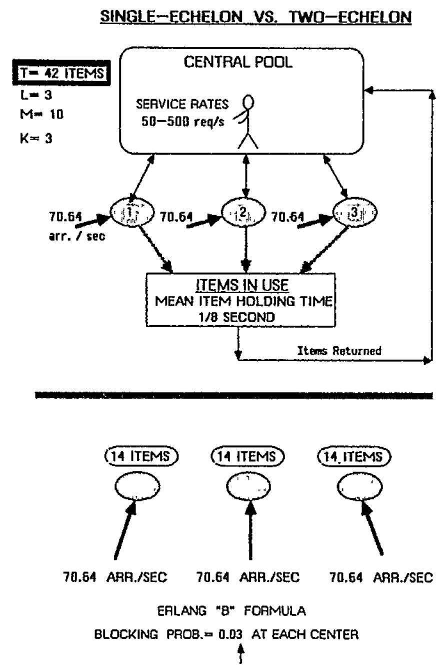


Figure 8: Example 2 — Single-Echelon vs Two-Echelon

For the two-echelon strategy, four additional parameters need to be defined. The full grant size (M) and the reorder point (K) are assumed to be ten buffers and three buffers, respectively. The mean buffer holding time is one eighth of a second, and the buffer manager's service rate ranges from 50 requests per second to 500 requests per second. Given the above parameters, we can calculate the blocking probabilities by the decomposition algorithm. In Figure 9 we plot the blocking probabilities as a function of manager service rates. It can be seen from Figure 9 that the two-echelon strategy can do better than the FA strategy when the buffer manager is fast enough in servicing requests. In this example, the break-even point is around 240 requests per second. Also shown in Figure 9 are the results from the simulator introduced in Section 4. It can be seen that there is about a 15% difference between the simulation and decomposition results. These differences can be attributed to the different queueing disciplines of the pool manager and the heuristic nature of the decomposition model.

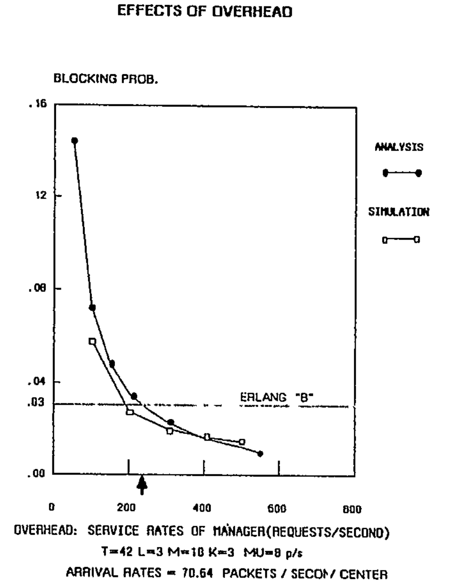


Figure 9: Example 2 — Numerical and Simulation Results

Example 3. A Design Example

In our final example, we try to find the optimal value of K, the reorder threshold, for a given system. We will consider a system with a total of 60 buffers and 3 local centers. Each local center has an arrival rate of 70.64 packets/sec. In addition, the system has a maximum request grant size of 10 buffers, a buffer holding rate of 8 packets/sec, and a reorder point ranging from 1 to 9 buffers. We considered two different manager's service rates: 20 requests/sec and 50 requests/sec. The results of the example are given in Figure 10. For service rate at 20 requests/sec, the optimal K is 9 buffers; at 50 requests/sec, the optimal K is 6 buffers.

This example provides the following insights: (1) The optimal K of a system with a slow manager is always greater than that of the same system with a fast manager. This is true because when the manager is slow, it makes sense to reorder earlier, i.e., using a larger K , to reduce the chance of out-of-stock situations. (2) Intuitively, a larger K means a larger local inventory, and thus a smaller blocking probability. However, this is not always true. The curve for $\gamma = 50$

requests/sec in Figure 10 illustrates the point by showing an upward bending for K's greater than 6 in the decomposition model, and greater than 8 in the simulation model. In this example, the upward bending can be attributed to either long holding times or an insufficient amount of buffers in the system. By increasing μ, the service rate, from 8 to 12 packets per second, the upward bending phenomenon is indeed rectified; see Figure 11.

Simulation results for both curves are also plotted in Figure 10 for the purpose of comparison. Basically the curves have the same shapes, with the simulation curves lying slightly below the decomposition curves. In simulation, the optimal K for the γ = 50 requests/sec case is 8 rather than 6 from the decomposition model.

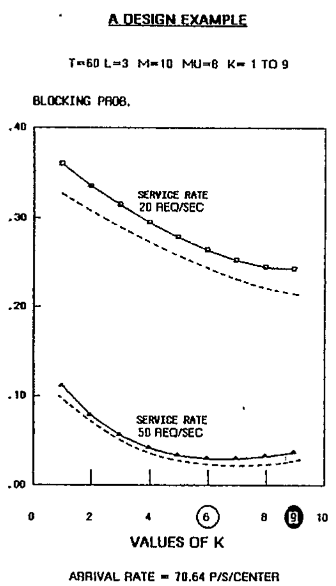


Figure 10: Example 3

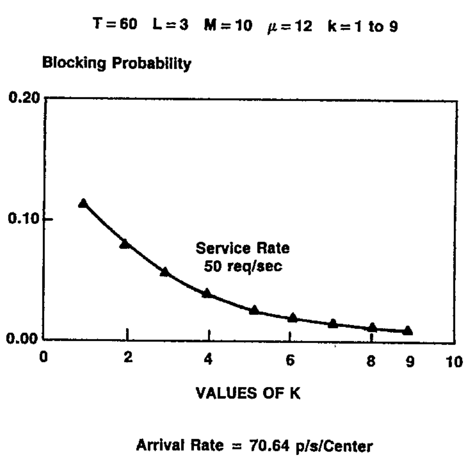


Figure 11: Blocking Probabilities Obtained with Increased μ

6. CONCLUSIONS

This paper proposed a two-echelon resource allocation strategy for packet switching systems. A function-level decomposition technique has been developed to analyze the performance of the strategy. Compared to the simulation of realistic systems, the decomposition method does provide good estimations on the system behavior as shown in the numerical examples. The two-echelon allocation strategy reduces the internal communication in packet switches, and is adaptive in handling traffic fluctuations. The adaptableness can be controlled dynamically by adjusting parameters M and K according to the load at each local center. Although the ultimate adaptability can be achieved only by single level complete sharing (CS) strategies, the CS strategy is impractical for packet switch applications due to the fast demand rate observed. Another commonly used strategy is the fixed allocation strategy (FA), whereby the buffers are divided evenly among the local centers. Compared to the FA strategy, the two-echelon strategy assures better performance when the pool manager's service rate exceeds a threshold value which can be determined by the analysis.

REFERENCES

Demmy, W.S. and Presutti, V.J. (1981). Multi-Echelon Inventory Theory in the Air Force Logistics Command. *TIMS Studies in the Management Sciences*, Vol. 16, North Holland, 279-297.

Graves, J.C. (1982). A Multi-Echelon Inventory Model for a Low Demand Repairable Item. *Sloan School of Management, M.I.T.*, Cambridge, MA, WP No. 1299.

Irland, M.I. (1978). Buffer Management in a Packet Switch. *IEEE Transactions on Communications*, Vol. COM-26, No. 3, 328-337.

Kaufman, J.S. (1981). Blocking in a Shared Resource Environment. *IEEE Transactions or Communications*, Vol. COM-29, No. 10, 1474-1481.

Nahamias, S. (1981). Managing Repairable Stem Inventory Systems: A Review. *TIMS Studies in the Management Sciences*, Vol. 16, North Holland, 253-277.

Sherbrooke, C.C. (1968). METRIC: A Multi-Echelon Technique for Recoverable Stem Control. *Operations Research*, Vol. 16, No. 1, 122-141.

AUTHORS' BIOGRAPHIES

STEVEN S. LIU is a Principal MTS at GTE Labs. He received a B.S. in Electrical Engineering from Cheng-Kung University (Taiwan) in 1969, an M.S.E.E. and a Ph.D. degree from Georgia

Institute of Technology in 1974 and 1979, respectively. His current interest is in next-generation packet switching system architectures.

JESSE WANG received a B.S. degree in Industrial Engineering from Polytechnical Institute of New York in 1983, and is currently enrolled in a master's degree program in Computer Science at Boston University. From 1984 to 1987 he was with GTE Laboratories as a software engineer. Since May 1987, he has been with Prime Computer Corp. as a senior system support analyst responsible for software development to support system benchmarking tasks.

ARTHUR HSU is a senior system analyst at CONTEL Business Network Corp., specializing in developing AI-based network design tools. Arthur received his B.S. and M.S. degrees in Civil Engineering from Massachusetts Institute of Technology in 1982 and 1984, respectively.