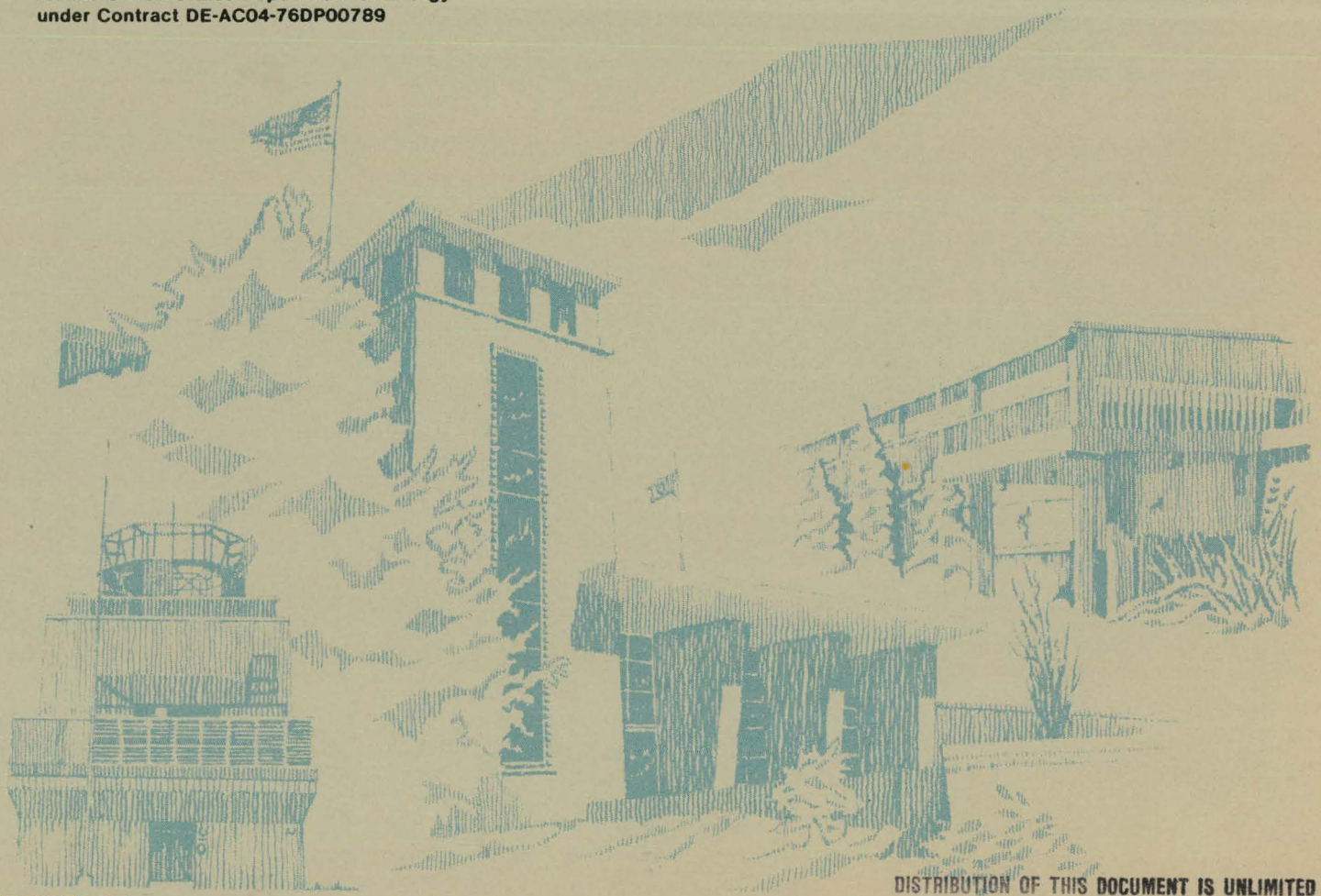
SANDIA REPORT SAND82-0176 • Unlimited Release
Printed March 1982

The Allocation of Computer Ports Within a Terminal Switching Network: An Application of Queuing Theory to Gandalf Port Contenders

Michael O. Vahle

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789



SF2900Q(8-81)

DISCLAIMER

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

NTIS price codes
Printed copy: A02
Microfiche copy: A01

PAGES 1 to 2

WERE INTENTIONALLY

LEFT BLANK

SAND82-0176
Unlimited Release
Printed March 1982

SAND--82-0176

DE82 013633

THE ALLOCATION OF COMPUTER PORTS WITHIN A TERMINAL SWITCHING NETWORK: AN APPLICATION OF QUEUING THEORY TO GANDALF PORT CONTENDERS

Michael O. Vahle
Division 2648
Sandia National Laboratories
Albuquerque, NM 87185

ABSTRACT

Queuing theory is applied to the problem of assigning computer ports within a terminal switching network to maximize the likelihood of instant connect. A brief background of the network is included to focus the statement of the problem.

Intentionally left blank

Table of Contents

Intentionally left blank

Introduction

The number of interactive computer terminals at Sandia National Laboratories is quickly approaching 2,000. The growth is expected to continue into the foreseeable future. These terminals must compete for a limited number of computer ports. Therefore a form of port contention was provided. Additionally, it is common for one terminal to need access to a variety of computer resources. To meet these requirements, a terminal switching network [1] is being installed. However, no technology can support an unlimited number of computer ports. This paper will describe an attempt to optimally assign the available network resources among the connected computers.

Environment

A Gandalf PACX IV is the backbone switch in the network. The PACX is an intelligent time division multiplexor which provides port contention. The PACX can support 512 terminals and 256 ports at asynchronous data rates up to 9.6 kbps or synchronous data rates up to 19.2 kbps. The 256 ports can be subdivided into groups (called classes) of service. For instance, ports 1-10 could be attached to a VAX and called Class 1, while ports 11-20 could be attached to a UNIVAC 1100/82 and be called Class 2. For simplicity, it is easiest to visualize a connection as a logical pairing of one terminal to a distinct port. The PACX can support 256 simultaneous connections.

A connection is formed at the request of a user (at a terminal). The user, through a standard procedure, can request the switch to provide a service by specifying the requested class. The switch checks to see if a port in the requested class is available. If a free port in the appropriate class exists, the switch makes the connection. If no free port is available and the terminal is permitted access to this class, the switch queues the request (if the user agrees) until a port becomes available.

To connect multiple PACXs together in a network requires that ports from one PACX be physically connected to the terminal side of a second PACX and a transit class of service be allocated on the first. A user attached to the first PACX needing a service on the second PACX must first request the transit class (i.e., port-terminal hardware connection) between PACXs and then, when that service is granted, specify the desired class on the second PACX.

Each time a PACX takes any action such as connecting, queuing, or disconnecting a terminal, it reports the action over a special output circuit. These reports are intercepted by a supervisory computer. The data is decoded and posted in a data base that profiles the entire system. The data can be used to calculate hold times, connect arrival rates, and other important system statistics, besides providing a detailed history of all terminals and ports.

The Problem

Ideally, one would like to use the statistics gathered during the operation of the network to reconfigure the network to provide the best possible service for the users. However, practical considerations such as existing wire paths, cost of multiplexing, location of equipment, and need for continuous service preclude any drastic global reconfigurations even if a theoretical optimal solution could be obtained. Some obvious reassignment of terminals is possible to reduce interswitch communications needs. However, the optimal assignment of ports to a class on a switch by switch basis is a realistic and realizable goal which would achieve immediate dividends. These dividends include:

1. Minimizing the number of computer channels to provide a determined level of service.

2. Reducing the required number of interconnects between PACXs.

The scheme developed to find the optimal PACX configuration for a specified level of service is the subject of the remainder of this paper. Specifically, the meaning of optimal is discussed in terms of queuing theory along with the algorithm to determine the optimal configuration.

Queuing Theory

The algorithm to allocate ports to classes attempts to maximize the probability that a terminal immediately gets connected when it requests a service. A secondary goal is to minimize the wait time if a terminal is placed into a queue. The calculation of connect probability and expected wait time is based on classical queuing theory (i.e., birth-and-death processes). The following summarizes the pertinent results of the detailed discussion of queuing theory given in Hiller and Lieberman [2].

In the context of queuing theory, birth refers to an arrival and death refers to the departure of a served customer. The following standard symbols are used in the formulas to be described later:

s = Number of servers in the system.

P_n = Probability that exactly n customers are in the queuing system.

λ_n = Mean arrival rate of new customers given that n are already in the system.

μ_n = Mean service rate when n customers are in the system.

W_q = Expected waiting time in queue.

L_q = Expected length of the queue.

The results that follow are based on these assumptions:

1. Only one birth or death can occur at a time.

2. The probability distributions for the time remaining until the next death (birth) is exponential with mean $\mu_n(\lambda_n)$.

3. The system is not in a transient state $\left(\frac{dP_n(t)}{dt} = 0\right)$.

With these assumptions satisfied, the following relations are obtained by balancing the expected number of occurrences that force entry into a state with those that force an exit.

$$P_1 = \frac{\lambda_0}{\mu_1} P_0$$

$$P_2 = \frac{\lambda_1}{\mu_2} P_1 + \frac{1}{\mu_2} (\mu_1 P_1 - \lambda_0 P_0) = \frac{\lambda_1}{\mu_2} P_1$$

$$\vdots$$

$$P_n = \frac{\lambda_{n-1}}{\mu_n} P_{n-1} + \frac{1}{\mu_n} (\mu_{n-1} P_{n-1} - \lambda_{n-2} P_{n-2}) = \frac{\lambda_{n-1}}{\mu_n} P_{n-1}$$

By defining

$$C_n = (\lambda_{n-1}/\mu_n)(\lambda_{n-2}/\mu_{n-1}) \cdots (\lambda_0/\mu_1), \text{ for } n = 1, 2, \ldots$$

we can use the above relationships to obtain

$$P_n = C_n P_0 \quad \text{for } n = 1, 2, \ldots$$

Substituting this expression into the equation

$$\sum_{n=0}^{\infty} P_n = 1$$

gives the following important result

$$P_0 = 1/\left(1 + \sum_{n=1}^{\infty} C_n\right) \quad .$$

The probability that no wait is incurred on an arrival is the probability that there are fewer than s customers in the system i.e.

$$\text{Prob}\{\text{no wait}\} = \sum_{n=0}^{s-1} P_n \quad .$$

Furthermore, if n > s then there are n - s people in the queue, therefore the expected length of the queue is

$$L_q = \sum_{n=s}^{\infty} (n - s) P_n \quad .$$

Two specific cases are used in the allocation algorithm. Both cases define λ_n and μ_n in terms of two constants λ and μ. These cases will both reach a steady state condition (i.e. nontransient state) if $\frac{\lambda}{s\mu} < 1$. To simplify the following discussion we define $\rho = \frac{\lambda}{s\mu}$ and demand that $\rho < 1$.

The first case we consider is specified by ($s \geq 1$)

$$\lambda_n = \lambda \text{ for } n = 0, 1, 2, \ldots$$
$$\mu_n = \begin{cases} n\mu & n = 1, 2, \ldots, s \\ s\mu & n = s, s+1, \ldots \end{cases} \quad .$$

In this case, we have multiple servers servicing an infinite number of customers who arrive at intervals independent of how busy the system is. We obtain by substituting into the general case above

$$C_n = \begin{cases} \frac{1}{n!} (\lambda/\mu)^n & n = 1, 2, \ldots, s \\ \frac{1}{s! s^{n-s}} (\lambda/\mu)^n & n = s, s+1, \ldots \end{cases}$$

$$P_0 = 1 \Big/ \left\{ \sum_{n=0}^{s-1} \left(\frac{\lambda}{\mu}\right)^n \frac{1}{n!} + \left(\frac{\lambda}{\mu}\right)^s \frac{1}{s!} \frac{s\mu}{s\mu - \lambda} \right\}$$

$$P_n = \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n P_0 \quad 0 \leq n \leq s$$

$$L_q = P_0 \frac{\rho}{s!(1-\rho)^2} \left(\frac{\lambda}{\mu}\right)^s$$

For this case, the expected wait in a queue,

$$W_q = \frac{1}{\lambda} L_q$$

follows from a result of Little [2]. Finally the general results with P_n as above give the probability of no wait as

$$\text{Prob\{no wait\}} = \sum_{n=0}^{s-1} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n P_0 \quad .$$

The other case is one where the population instead of being infinite is restricted to a finite size M. For this case we have

$$\lambda_n = \begin{cases} (M - n)\lambda & n = 0, 1, 2, \ldots, M \\ 0 & n \geq M \end{cases}$$

$$\mu_n = \begin{cases} n\mu & n = 1, 2, \ldots, s \\ s\mu & n = s, s+1, \ldots \end{cases} .$$

For this model we obtain from the general formulas

$$C_n = \frac{M!}{(M-n)!\, n!} \left(\frac{\lambda}{\mu}\right)^n \qquad 0 \leq n \leq s$$

$$P_0 = 1 \Big/ \left\{ \sum_{n=0}^{s-1} \frac{M!}{(M-n)!\, n!} \left(\frac{\lambda}{\mu}\right)^n + \sum_{n=s}^{M} \frac{M!}{(M-n)!\, s!\, s^{n-s}} \left(\frac{\lambda}{\mu}\right)^n \right\}$$

and $$P_n = P_0 \frac{M!}{(M-n)!\, n!} \left(\frac{\lambda}{\mu}\right)^n \qquad 0 \leq n \leq s \ .$$

Little's theorem for this case takes the form

$$W_q = L_q / \bar{\lambda} \text{ where}$$

$$\bar{\lambda} = \sum_{n=0}^{\infty} \lambda_n P_n \ .$$

These equations then allow one to calculate W_q and Prob{no wait} from the equations listed in the general result section as was done for the infinite population case.

The Algorithm

The mathematics overviewed in the last section is implemented in a program called MODEL. From a summary of the statistics gathered from actual operation of the terminal switching network, MODEL calculates an arrival rate λ and a service rate μ for each represented service. Using these parameters and the above formulas, MODEL generates for each service a table of values containing the Prob(no wait) and the corresponding expected wait for all number of

ports s from s_{MIN} to s_{MAX}. s_{MIN} is the smallest value of s that makes $\rho < 1$, hence guaranteeing stability. S_{MAX} is the first value of s that makes Prob {no wait} $\geq$ .99.

The operator specifies whether the infinite or finite population model is used during the calculation. In the case that the finite population model is chosen, the actual number of terminals that requested the service determines the population size M used in the formulas.

Once the table of probabilities and expected waits is calculated, MODEL selects the set of s (the number of ports for each service) that provides each service the highest Prob {no wait}. This selection is constrained by the requirement that the total number of ports must be less than or equal to the number of ports available on the switch. If the sum of s_{MAX} for each service is within the constraint, then MODEL would allocate the maximum number of ports to each service. However, if the sum is outside the constraint, MODEL calculates the difference D between Prob(no wait) for s_{MAX}, and Prob {no wait} for s_{MAX} -1 for each service. After locating the service for which the difference is minimal, MODEL redefines s_{MAX} for the identified service as s_{MAX} -1 and then rechecks the constraint. If two or more services have the same difference D, then MODEL reassigns s_{MAX} for the service that has the smallest expected wait, W_q, for s_{MAX} -1. MODEL will never eliminate a service by reassigning an s_{MAX} for a service where $s_{MAX} = s_{MIN}$. The procedure is iterated until the sum of original or reassigned s_{MAX} falls within the constraint or until no further reduction is possible.

To estimate the effect of assuming that λ and μ are constants as opposed to functions of time, MODEL also repeats the allocation based on statistics that characterize the busiest 30-minute period for each service. The recommendations based on the busiest 30 minutes provide a very conservative allocation.

For the infinite population case, MODEL calculates λ for a service by dividing the number of requests by the amount of time in the reporting period and μ by dividing the sum of the length of all connects by the number of successful connects. For the finite model case, the calculation for λ includes a division by the population M. For the finite population case, MODEL can recalculate an M (and therefore λ) that attempts to account for the fact that a terminal can't request service B while connected to service A. The reapportionment of M can be on the basis of percentage of time spent on a service or by the percentage of connects to a service. For example, if a terminal spent one-half of all its connect time on a particular service, it would contribute a one-half to the population M using that service. Likewise, in the other case, if one-third of all connects a terminal made are for a particular service, it would contribute a one-third to the population M for that service.

The following figures illustrate the output that MODEL produces. The results were calculated based on a finite population with no reapportionment. Both figures include the results for the busiest 30 minutes so that the effects of assuming that λ and μ are constants, can be estimated.

Figure 1 includes the Prob {no wait} and expected wait, W_q, as a function of the number of servers (i.e. ports) for the NOS time-sharing system. The parameters at the top of Figure 1 include the switch and class that represent NOS along with the statistics that characterize NOS. Note that average connect time is μ while the arrival rates are λ for the total period and busiest 30 minutes. All times are listed in minutes. The terminal total is the size, M, of the population that used NOS. The asteriks indicate that the calculated value was greater than 100.

The second figure shows the recommended configuration for a PACX with 256 ports available for assignment. The recommendation is based on statistics gathered over an eight-day period.

PACX=0 Class=2 Total Terminals=141. Total Connects=1520.
Avg Connect Time=27.8
Arrival Rates=0.00277 0.00331

Average			Busiest 30 mins.		
Ports	Prob of conn	Expected wait	Ports	Prob of conn	Expected wait
4.	0.00000	********	4.	0.00000	********
5.	0.00000	********	5.	0.00000	********
6.	0.00000	********	6.	0.00000	********
7.	0.00001	********	7.	0.00000	********
8.	0.00242	********	8.	0.00000	********
9.	0.05066	50.13781	9.	0.00044	********
10.	0.22799	21.72557	10.	0.01342	62.54070
11.	0.46411	9.39725	11.	0.09835	32.30371
12.	0.66245	4.25728	12.	0.28368	15.40882
13.	0.80060	1.99000	13.	0.49481	7.36197
14.	0.88823	0.93824	14.	0.67076	3.60610
15.	0.94034	0.43853	15.	0.79733	1.79493
16.	0.96965	0.20079	16.	0.88111	0.89486
17.	0.98529	0.08934	17.	0.93332	0.44144
18.	0.99321	0.03843	18.	0.96422	0.21354
			19.	0.98163	0.10063
			20.	0.99098	0.04599

Figure 1. Prob {no wait} and the expected wait W_q (in minutes) as calculated for the NOS time-share system.

Recommended Configuration for PACXO with 256 Ports

		Ports needed for Average load	Ports needed for Busiest 30 mins.
Class	2	18	20
Class	3	1	2
Class	4	1	4
Class	5	1	3
Class	6	1	2
Class	15	3	5
Class	16	9	11
Class	17	11	14
Class	21	3	5
Class	24	1	1
Class	25	1	1
Class	30	7	13
Class	33	5	9
Class	34	12	13
Class	35	13	17
Class	36	1	1
Class	43	2	3
Class	44	8	8
Class	46	1	2
Class	47	1	2
Class	62	4	8
Class	66	3	4
Class	75	4	7
Class	76	1	2
Class	77	1	2
Class	111	3	5
Class	113	2	6
Class	115	2	5
Class	116	1	2
Class	117	1	4
Class	160	3	5
Class	170	1	1
Class	176	1	3
Class	177	1	3
Total Ports		128	193

Figure 2. Recommended Configuration

Conclusion

By using data gathered during the actual operation of a terminal switching network, predictions of the expected performance as a function of the number of ports can be made. By adjusting the calculation of critical parameters a spectrum of predictions can be generated. These predictions can be used to optimize the likelihood of immediate connect and offer guidance in the allocation of computer ports in the switching network. Furthermore the predictions can be used as input to simulation programs [3] that can further refine the predictions.

MOV:cmg:6414A:12/08/82X

Reference

1. M. O. Vahle, L. F. Tolendino, and M. D. Thompson, "The SNLA Terminal Switching Network," Proceedings of the 24th Midwest Synposium in Circuits and Systems, University of New Mexico, Albuquerque, NM, pp 659-662 (June 1981).

2. F. S. Hillier and G. J. Lieberman, Operations Research, Chapter 9 (Holden-Day Inc., 1974).

3. L. F. Tolendino, "Simulating the Operation of a Port Contender with SIM4: Applying Monte Carlo Techniques to the Gandalf PACX IV" SAND82-0162, Sandia National Laboratories, Albuquerque, NM.

DISTRIBUTION:

2600	L. E. Hollingsworth
2610	D. C. Jones
2612	J. A. Cooper
2612-1	R. A. Trudo
2614	A. R. Iacoletti
2620	R. J. Detry
2630	E. K. Montoya
2635	P. A. Lemke
2636	W. F. Mason
2640	J. L. Tischhauser
2644	D. M. Darsey
2645	L. D. Bertholf
2646	M. R. Scott
2648	D. H. Schroeder
2648	M. D. Thompson
2648	L. F. Tolendino
2648	M. O. Vahle (5)
3141	L. J. Erickson (5)
3151	W. L. Garner (3)
	DOE/TIC
3154-3	C. Dalin (25)
	for DOE/TIC Unlimited Release

DO NOT MICROFILM

Org.	Bldg.	Name	Rec'd by	Org.	Bldg.	Name	Rec'd by

Sandia National Laboratories