ANALYSIS AND SIMULATION OF A HOMOGENEOUS COMPUTER NETWORK

by

Jacques Labetoulle*
Eric Manning and Richard Peebles**

Research Report CS-77-08

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

March 1977

*Institut de Recherche d'Informatique et d'Automatique
Rocquencourt, France

**Department of Computer Science
and
Computer Communications Networks Group
University of Waterloo
Waterloo, Ontario, Canada

ABSTRACT

A network of minicomputers to support transaction processing against a distributed data base is proposed. This paper gives a brief overview of the design based on a loop communicaions subnetwork and then describes analytic and simulation models that have been used to predict performance.

Service demands in this network are assumed to have the following characteristics:

- transaction: users at on-line terminals enter messages that invoke short computation, a few accesses to the data base and a response message that is then sent back to the terminal.
- locality of reference: transactions entering at one node in the network can almost always be serviced locally, the fraction requiring remote service is expected to be in the 10% to 30% range.

The network has been modelled analytically as a network of queues. The predictions of this model have been compared to those of a large simulation of the network in an attempt to validate that simulation model. Agreement is good for a two host network over a broad range of network loading. The simulation program has been used to study a large number of problems where we varied: transaction

characteristics, remote traffic fraction, host configuration and communications subnetwork speed. One such experiment is described here in detail.

KEYWORDS

distributed data bases

distributed processing

local balance

loops

message switched operating systems

multi-processor architectures

performance prediction

queueing networks

transaction processing

Jacksonian models

I. INTRODUCTION

1.1 Motivation

In this paper we examine the design of a computer network intended to support transaction processing for geographically distributed organizations. We believe that for many large organization the data base will exhibit _geographic locality of reference_; the data base can be partitioned into components such that most of the queries homing on a given component of the data base originate in a particular geographic region. There are many examples associated with business and industry -- credit and inventory records for example. At the same time there is a need to operate the collection of components as a single data base, to provide for occasional transactions which cross regional boundaries, and for managerial queries which span the entire data base.

Geographic locality of reference is only one of the reasons for creating logically unified but physically distributed data bases. If a data base contains information supplied by several agencies, each may insist as a matter of policy that 'its' data be held in 'its' hardware located on 'its' premises, quite apart from technical efficiencies which may accrue.

In the applications quoted, most queries take the form of transactions -- short messages evoking rapid responses plus side effects such as updating of the data base. For example when a customer buys gas with a credit card a transaction processing system would update the customers balance due and the gas stations inventory record. We conclude that a broad system of applications will involve transaction processing on distributed data bases.

It is important to note that we have examined only simple data base models to date - essentially direct access files. Support of relational and network data base models is currently under study.

1.2 Hardware Architecture

The general-purpose, heterogeneous network as exemplified by ARPANET [20] or CYCLADES [18] provides one possible vehicle for the support of a distributed data base. However, the advantage of heterogeneity (multiple CPU architectures) carries the penalty of complex protocols needed to overcome incompatibilities. One would prefer a single architecture for all host CPU's if data is the resource to be shared.

Second, transaction processing often does not require large, expensive CPUs. Modest CPU power suffices to read short requests, perform the simple calculations associated with data base probes, and format short replies.

Our simulations indicate that a minicomputer is more than adequate for many applications if enough disc store and a high data rate to disc are provided. We have therefore proposed a network of identical minicomputers (called MININET).

The cost of such networks could be relatively low; probably under $500,000 for five nodes (hardware cost). It is therefore plausible to assume that they could be marketed and managed as turnkey or package systems. Software would be tailored to the user's application prior to installation and would be modified off-line. It should not be subject to the continual abuses of program development; rather, it is software tailored with the aid of a very high level language. Hence we assume that we are able to specify host hardware, host software and the communications subnetwork, as a single, integrated system. A major goal of our research has been to investigate how far these freedoms can be exploited to yield simple, elegant structures.

1.3 Summary

The proposed architecture seems highly plausible but to gain confidence in the design it is desireable to obtain estimates of its performance under various loads. In Section 2 we give a brief overview of the design, sufficient to understand the modelling requirements. We turn to the problem of behaviour prediction in Section 3 (analysis) and Section 4 (simulation). Comparisons are provided in Section 5. The principal objective of the analysis was to validate the simulation model. The latter can then be used with greater confidence in exploring the behaviour of the network. Validation has been achieved.

2. DESIGN SUMMARY

This section describes the major results of our study of network structures for transaction processing on distributed data bases. Full details are available elsewhere [13,17]; only enough detail is provided here to motivate the analytic and simulation models proposed in Sections 3 and 4.

2.1 Communications Subnetwork

The communications subnetwork (or subnet) must cater to bursty traffic and have a low cost per port. Packet-switching is one possibility, although the price of an ARPANET IMP [2] is similar to the cost of our host. We have therefore selected a loop based on the Newhall-Farmer protocol [15] as subnet (see Appendix 2). It is well-suited to bursty traffic and the port cost is a few thousand dollars.[1]

2.2 Host Operating system

ARPANET has imposed virtual circuits (the link/socket construct) on a message-with-address or MWA subnet, because of the need to support traditional host operating systems. We have taken the opposite approach and have tried to use MWA switching throughout. (Transaction traffic is well-suited to the properties of MWA switching [21].) Consequently, the host operating system is MWA - a so-called Message-Switched Operating System. It is transaction-driven; nothing happens until a transaction arrives. The transaction generates a directed graph of message flows which are processed to yield a response message and side-effects. All message passing is managed by the communications nucleus, as described below.

Finally, each host runs a copy of the same operating system, and the network is a turn-key package so that users are not required nor permitted to write applications programs. This implies that the network environment is "friendly"; a close degree of logical coupling between hosts is possible since the behaviour of other hosts is predictable. This in turn can be exploited to yield further efficiency.

2.3 Primitives

There are exactly two primitive objects. The primitive data object is the <u>segment</u> and the primitive control object is the <u>task</u>. Program work areas, messages, and individual records of files are all examples of segments. A task is a collection of segments; the entities which transform messages are examples of tasks; files are also examples of tasks. (This approach replaces the file-concept 'open' with the task-concept 'active', allowing any number of tasks to access a file simultaneously.)

2.4 Communications Nucleus

The communications nucleus consists of the loop subnet, the switch task of the hosts' message-switched operating system, and a processor, interposed between the host and loop port, which is called the Communications Device or CD. (See Figure 2.1.) The function of the nucleus is to move message segments from a sending task to any other task, in the same or any other host. The sending task uses exactly the same protocol for both inter-host and intra-host message passing, thus providing a basis for rendering the physical distribution of the data base transparent to users. The nucleus uses a simple, fast technique due to S. Wecker [22] for intra-host transfers; a pointer is copied into one of the receiving task's segmentation registers. Inter-host message transfer involves the CDs and the loop, and is of course more elaborate. (Further details are provided in [13].)

2.5 Virtual Networb Address Spaces (RNAS and VNAS)

Tasks run in private virtual spaces, and virtual addresses spanning the network are provided to allow naming of remote tasks; hence the concept of Virtual Network Address Spaces (VNAS). These addresses are in fact interpreted as belonging to the virtual space of a remote switch task. Thus the Switch at each host controls over-the-network addressing and prevents misuse of the Real Network Address

Space (RNAS) capability. The virtual addressing also makes intra-host message passing efficient, as noted above. (Further details are provided in [13].)

The RNAS spans primary store, secondary storage device control & data registers, and I/O device registers of all hosts of a network. Application tasks have no direct access to the RNAS for obvious reasons; its capabilities are used by the switches and CDs to effect message transfer.

2.6 Inter-Host Message Segment Transfer

The sending switch task requests "permission to transfer" of the receiving switch using a VNAS address. The receiving switch returns an RNAS address pointing to an empty segment nominated to receive the message. The sending CD then fetches the message segment from its host's storage, breaks it into loop messages and transm ts them. (Each loop message is prefixed by the loop's necessary protocol, followed by retries if busy.) The receiving CD reassembles the segment and stores it in the nominated area; note that host CPUs do not participate in the message transfer after the initial exchange between switches. The CD, therefore, serves to interface between the linear segmented structure of host tasks and the MWA structure of the loop subnet. (Further details are provided in [13].)

2.7 Secondary Storage

Secondary storage is managed by a separate CPU called the Data Host or DH. The DH runs a message switched operating system identical to its associated host. Multiple DHs per host are feasible and the DH speaks to the loop on the same level of protocol as the host. This implies that transactions between a host and a remote DH can proceed without reference to the remote host. DH tasks include the files, whose data segments are records and whose procedure segments are the access method. The disc itself is treated as a linear segmented memory for uniformity and simplified storage management. (Further details are provided in [17].)

2.8 Simple File System

File access techniques have been investigated and a simple direct access file system has been designed. (Further details are provided in [17].)

2.9 Status

Efforts to implement a two-host prototype are under way and the following has been accomplished. Digital Equipment Corporation PDP-11 Model 45 minicomputers have been selected and acquired for the hosts, as well as Model 20s for the initial Communications Device/Data Host implementations. A Newhall loop has been acquired and is undergoing testing as the communications subnetwork. Loop ports suitable for interfacing to the Communications Devices have been designed. Currently the system is operating with an un-switched subnetwork; the Communications Devices are directly connected. A protocol for transferring message segments from one host to another via the loop has been designed and implemented as a Communications Device program in QPL-11, the Queen's University translator for PL-11 [14]. Finally, the Switch task has been designed and coded in the "C" language and inter-host message transfers have been done.

This paper discusses the problem of predicting the performance of the networking structure described above; both queueing-theoretic and simulation techniques have been applied successfully. Our queueing-theoretic analysis is described in the next section.

3. QUEUEING THEORETIC ANALYSIS

3.1 A Queueing Network Model

Our objective was to determine the effect on response times and queue sizes of loop speed, loop message length (blocksize), transaction arrival rates and message length distributions. One finds the usual exponentially shaped curves for response times and queue lengths; we wished especially to locate the "knees" of these curves. (By choosing system parameters so that the normal operating region lies well below the knee, one ensures that small fluctuations in load do not cause large fluctuations in performance.)

The first step was to model the network -- hosts and communications subnetwork -- as a network of queues. Figure 3.1 shows such a model for a two-host, single-loop network. Transactions enter from the terminals and queue for the host CPU (server labelled FM). Transactions to be processed locally then enter the disc queue (server labelled DH) and those requiring service at a remote host enter the loop queue. A transaction may require several disc accesses as indicated by the flow from server DH back to the DH queue. Also, each disc access spawns a "post-processing" request which returns to the FM queue. When all disc processing is complete, responses flow out of the FM server and back to the terminals (directly, or via the loop if processing occurred remotely). The other symbols of Figure 3.1 are explained in Sections 3.3 and 3.4. Computation of the transaction probabilities, pi, is illustrated in Section 5.1.

This model corresponds to reality in the following way. All of the programs -- the command processors, terminal handlers and message switch--residing in the host CPU are represented by a single server called the File Machine or FM. The rationale for this is that as each transaction arrives, the FM decides which host has the necessary component of the distributed data base, and routes the transaction accordingly.[2] The Data Host or DH is represented as a separate server because the program that drives the disc does not reside in the host CPU but rather in a mini dedicated to data access, as described in section 2.7. Finally, the loop is represented as a pair of servers; the service times are variable and interdependent as is described section 3.5.

3.2 Previous Work

Queueing-theoretic analysis of loop systems has been performed by Hayes and Sherman [6], Yuen et al. [24], Kaye et al. [8], Konheim and Meister [11] and by Cooper and Murray [4]. None of these papers considered a network of computers -- subscriber computers plus communications sub-network -- they deal only with loop communications sub-networks.

Hayes and Sherman [6] studied a loop structure operated according to the Pierce protocol, which requires radically different treatment from the Newhall-Farmer protocol discussed here. Hayes and Sherman calculated the average delay due to queueing at ports; this was done by computing the mean duration of loop busy and idle periods. Two approximations were used, corresponding roughly to light and heavy traffic conditions.

Yuen et al. [24] considered a loop with the Newhall-Farmer Protocol, again without reference to attached host computers. They obtained light-traffic approximations for the mean and variance of the time required for the 'permission-to-send' character to traverse the loop.

Kaye et al. [8] have performed extensive analysis of Newhall-Farmer loops, but only in the terminal-to-computer case where one particular port (the computer) participates in every conversation. Also, Richardson [19] has done a comparative analysis of the response-time performance of Pierce and Newhall-Farmer protocols, together with a variant of the Pierce protocol, again for the terminal-to-computer type of traffic. The Konheim and Meister studies focussed on terminal-to-computer communications, and they examined a modified Pierce protocol. Their work determined average queue lengths and average waiting times in both a simple loop system and a priority loop.

Finally, Cooper and Murray [4] have studied loop service systems where each centre on the loop has an arbitrary service time distribution and Poisson arrival traffic. They have constructed two models: in one the server moves to a centre and stays there until the queue is emptied; in the other a "gate" is put up at the end of the queue when the server arrives; only the customers in front of the gate are served before the server moves to the next centre on the loop. Neither model can be directly applied to our problem.

3.3 Global and Local Balance

We seek closed-form algebraic expressions for the means and variances of queue lengths and response times associated with the queueing network of Figure 3.1. Baskett, Chandy, Muntz, and Palacios-Gomez [1] have given general solutions to the problem, drawing on earlier work by Whittle [23] and Jackson [7]. The model used here is Jacksonian; that is we assume Poisson arrivals and exponential servers. The motivation for this is discussed further in Section 3.7. A brief summary of the major ideas (derivation of equations 3.1 and 3.2) is given in Appendix for readers who may not be familiar with research in queueing theory.

A state of this system is defined by an N-tuple (n1,n2,...,nN) where ni is the number of customers (messages our application) in the ith queue. We seek the state probabilities P(n1,n2,...,nN). These can be found by solving the global balance equation for the stationary state but it is simpler to use local balance if possible. It is known that solutions to the local balance equations will satisfy global balance. Although sufficient conditions for the existence of solutions to the local balance equations are not known in general they can be demonstrated to exist for a broad spectrum of models, see Baskett et al [1]. In particular they can be solved for a Jacksonian model.

3.4 Local Balance Solution

From the appendix (eq. A1.5) we know that

$$P(n1,\dots,nN) = C \prod_{j=1}^{N} \left(\frac{a_j}{\mu_j}\right)^{n_j} \qquad \dots .3.1$$

where j is the service rate at centre j, aj is the arrival rate at j and C is the normalizing constant. We wish now to compute the aj and C.

For the model of Figure 3.1, Equations A1.6 become

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} \lambda \\ \lambda \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & p_1 & 0 & 0 & p_3 \\ 0 & 0 & 0 & p_1 & p_3 & 0 \\ p_2 & 0 & (1-p_1) & 0 & 0 & 0 \\ 0 & p_2 & 0 & (1-p_1) & 0 & 0 \\ (1-p_2)p_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & (1-p_2)p_4 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix}$$

which yields

$$\left.\begin{aligned} a_1 = a_2 &= \frac{\lambda}{(1-p_2)\,(1-p_3p_4)} \\[2ex] a_3 = a_4 &= \frac{\lambda}{p_1(1-p_2)\,(1-p_3p_4)} \\[2ex] a_5 = a_6 &= \frac{\lambda p_4}{(1-p_3p_4)} \end{aligned}\right\} \quad \ldots\ldots\ldots(3.2)$$

substituting Equations 3.2 into Equation 3.1, we obtain

$$P(n_1 \ldots\ldots, n_6) = C \quad \frac{\lambda^M}{(1-p_2)^{n_1 + n_2 + n_3 + n_4} (1-p_3 p_4)^M} \quad X$$

$$\left(\frac{1}{\mu_1}\right)^{n_1} \left(\frac{1}{\mu_2}\right)^{n_2} \left(\frac{p_2}{p_1 \mu_3}\right)^{n_3} \quad X$$

$$\left(\frac{p_2}{p_1 p_4}\right)^{n_4} \left(\frac{p_4}{\mu_5}\right)^{n_5} \left(\frac{p_4}{\mu_6}\right)^{n_6} \qquad \ldots\ldots.(3.3)$$

Where M = n1+n2+n3+n4+n5+n6

The constant C is fixed by requiring

$$\sum_{\Omega} P(n_1 ,\ldots., n_6) = 1$$

which yields

$$C = \prod_{j=1}^{6} (1-\rho j) \qquad \ldots\ldots.(3.4)$$

where

$$\rho j = \frac{a_j}{\mu_j} \qquad \ldots\ldots.(3.5)$$

3.5 Loop Model

These equations can now be applied to construct a model of our loop network. As will be seen the model is quite simple and is accurate only under the following assumptions:

- Poisson arrival statistics for transactions entering the system
- Two host network with loop subnet utilizations less

than 0.6.

Values for the parameters of Equation 3.3 can be fixed by assumption, or by experimentation with the hardware and software -- except for the effective loop service parameters $\mu 5$ and $\mu 6$. (The File Machine's estimated service rate provides values for $\mu 1$ and $\mu 2$; the Data Host provides $\mu 3$ and $\mu 4$, and the assumed traffic provides λ. The fraction of transactions "going remote" provides p2; the number of disc accesses per transaction provides p1; and the number of response messages per query message provides p3. Examples are given in Section 5.). We now discuss a simple model which relates $\mu 5$ and $\mu 6$ to the known quantities Line Service Rate and Line Arrival Rate.

A model of Newhall-Farmer loop behaviour for <u>two ports</u> is constructed as follows. (Limitations of this model are discussed in sections 3.7 and 5.2.)

Fix attention on $\mu 5$. If the "other" port is idle (n6=0) then

$$\mu_5 = \mu_L$$

where is the Line Service Rate. If the other port is busy (n6 0) then the line capacity is divided evenly between the two ports and

$$\mu_5 = \mu_L/2$$

Let q be the probability that the 'other' queue is empty

Then

$$\mu_5 = q * \mu_L + (1-q) * \frac{\mu_L}{2} = \frac{\mu_L}{2}(1+q)$$

For a single exponential-Poisson process

$$P(\text{server idle}) = 1-\rho$$

Here, the server is the loop and

$$P(\text{loop idle}) = P(\text{port1 idle}) * P(\text{port2 idle}) = q^2$$

$$q^2 = 1 - \frac{2\lambda_\ell}{\mu_L}$$

where λ_ℓ is the message arrival rate at the loop and is determined by the arrival rate from the terminals and the fraction of traffic that goes remote. We <u>assume</u> that this relation is valid here, and obtain

$$\mu_5 = \frac{\mu_L}{2} \; 1 + \sqrt{1-\frac{2\lambda_\ell}{\mu_L}}$$

This has the desirable properties that

$$\lim_{\rho\to 0} (\mu_5) = \mu_L$$

$$\lim_{\rho\to} (\mu_5) = \frac{\mu_L}{2}$$

3.6 Performance Measures

The performance measures of prime interest here are response time and mean queue lengths. From these we can determine the maximum traffic rate which will produce a response time within acceptable limits and the amount of buffer storage required. The mean queue lengths are given by (equation A1.10);

$$\bar{n}_j = E(n_j) = \frac{\rho_j}{1-\rho_j} \qquad \text{.......(3.7)}$$

If W_j is the time spent by a customer at centre j, then Little's formula ($L = \lambda W$) gives

$$E(W_j) = \left(\frac{\rho_j}{1-\rho_j}\right) * \frac{1}{a_j} \qquad \text{.......(3.8)}$$

(where a_j is the arrival rate at centre S_j).

These expressions measure quantities of interest, and the values obtained from the local balance solution can be compared directly with values obtained by simulation.

3.7 Limitations

Two of the standard simplifying assumptions required for the above derivation may be considered serious. First, the arrival rate of new transactions is independent of the number currently in the system. In reality, most transaction terminals will not accept a new transaction until a response to the previous transaction has been received. Thus the arrival rate tends to decrease as network loading and response time increase. The model therefore errs on the conservative side. That is, queues _within_ the system will never grow as large as this analytic model allows.

Second, the loop model developed above is naive. It ignores some of the message transfer protocol (separate allocation request, response and data transfer messages), see Appendix 2. A more inaccurate approximation is that the mutual dependency of loop port behaviour is ignored, i.e., the assumption that $(1-\rho) = q^2$. Nevertheless, comparisons with a more detailed simulation model show good agreement for remote traffic fractions less than 0.6.

It is tempting to apply the extended models of Baskett et al [1] using state dependant arrival rates and customers of different classes. The former leads to extremely messy algebra and offers little additional insight for the effort. Customer classes can be handled only for service centre types that are not suitable for modelling the loop (e.g., processor shared); the loop is fundamentally a FIFO server. Hence we have opted to apply the simpler Jacksonian model.

4. A SIMULATION MODEL

4.1 Structure

The queueing-theoretic model contains a number of fairly severe assumptions as described above. To assess the validity of these simplifications, a more realistic model was constructed. It consists of the queueing network of Figure 3.1, plus the following refinements.

a) The loop

The partitioning of long message segments into several short loop messages was modelled, and the allocation request and response loop-messages were included. The pass-control structure of the loop was explicitly modelled, together with the loop protocol for setting up and tearing down virtual circuits. Hence the simple model of the loop described in Section 3.5 was replaced by a more realistic one.

b) Terminals

State-dependent arrivals were modelled, by inhibiting further requests from each terminal until the previous reply had been computed and transmitted. Moreover, the terminal model included "typing time" and "think time", unlike the simple Poisson process used in the queueing-theoretic model.

c) Hosts

Post-processing of the records retrieved by a data base probe was explicitly modelled as follows. Each customer (message) served by the Data Host generated two new messages; one was sent to the File Machine for post-processing, and the other returned to the DH queue to initiate the next disc access. Analytic techniques are not able to handle such "splitting" of customers upon leaving a service centre. Finally, numbers of hosts greater than two were permitted -- many of the examples studied modelled ten-host networks.

4.2 Solution

The refinements described above are not readily amenable to queueing-theoretic solution; it was therefore necessary to resort to computer simulation to obtain numeric results. The model was implemented as a SIMSCRIPT II.5 program which was run on a Honeywell

6050 computer. The implementation was a standard event-driven simulation and so is not described here; details are available in Peebles et al. [16].

5. COMPARISONS OF ANALYTIC AND SIMULATION RESULTS

5.1 Values of Analytic Parameters bij

In section 3 we have expressed the branching probabilities bij of Equation A1.6 in terms of four elementary transition probabilities p1,...,p4 as shown in Figure 3.1. In order to obtain numeric results for comparison with simulation we must assign values to these elementary transition probabilities. We begin by assuming symmetric traffic, with a common arrival rate of λ new transactions per second arriving at each host from its terminals, of which a fraction r require service at other hosts of the network. (Note that the value of r directly reflects the degree to which the database exhibits geographic locality of reference.)

Probability p1 is evaluated as follows. Each message served by the Data Host generates two new messages. One enters the Data Host queue to cause the next disc access, and the other is sent to the File Machine for post-processing. We model this situation by lumping all post-processing needed by a transaction into a single step. We assume that the number of disc accesses per transaction is uniformly distributed between 10 and 20. Therefore, (approximately) every fifteenth

Data Host output goes to the File Machine and

$$p_1 = 1/15$$

Probability p2 is estimated by the fraction of File Machine output which enters the Data Host queue. File Machine output per second comprises

(a) $(1-r)\lambda$ new transactions entering this host and requiring Data Host service at this host,

(b) $r\lambda$ new transactions entered at remote hosts but requiring Data Host service at this host,

(c) $(1-r)\lambda$ transactions previously entered and served at this host and requiring File Machine post-processing,

(d) $r\lambda$ transactions previously entered at remote hosts but served at this host and requiring File Machine post-processing before being returned to terminals at the remote hosts,

(e) $r\lambda$ new transactions entering this host but requiring remote Data Host service.

The last three items do not enter the Data Host queue at this host so that

$$p_2 = \frac{(1-r)\lambda + r\lambda}{(1-r)\lambda + r\lambda + (1-r)\lambda + r\lambda + r\lambda} = \frac{1}{2+r}$$

Probability p3 is estimated by the fraction of messages received from the loop which enter the File Machine queue. The message flow per second from the loop into this host comprises r messages which are requests for processing at this host, plus r responses from remote hosts to transactions which entered this host but required remote processing. The latter do not enter the File Machine queue (they are dispatched directly to terminals) so that

$$p_3 = 1/2$$

Probability p4 is estimated by the fraction of File Machine output which does not go to the Data Host and does go to the loop. From our discussion of p2, the traffic rate leaving the File Machine and not going to the Data Host queue comprises items c), d) and e) defined above. Of these, only the last two enter the loop queue so that

$$p_4 = \frac{r\lambda + r\lambda}{(1-r)\lambda + r\lambda + r\lambda} = \frac{2r}{1+r}$$

5.2 Comparisons

Several network structures of the type

discussed in this paper have been studied by both analytic and simulation techniques. Our objectives were to <u>calibrate the simulation program</u>, then to investigate the agreement between the two techniques, and finally to explore the properties of a "real" network. The results of three of these studies are reported here.

The first study modelled a two-host network, with 125 low-speed (110 baud) terminals and one disc per host. Transaction requests from each terminal had a mean length of 60 bytes[3] and a mean inter-arrival time of 90 seconds. Each transaction required 10 to 20 disc accesses (uniformly distributed) with a mean access time of 30 msec., and a response message with mean length of 60 bytes was returned to the terminal. These numbers are taken "out of the air" but are all defensible as representative of a plausible application. Each transaction required 5 msec. of initial processing time plus 2 msec. of post-processing time per disc access (not included in the analytic model). These processing times are also arbitrary assumptions; they cannot be properly justified until implementations are complete. We claim only that they are within an order of magnitude of values that will be found in a real system. (For example, we know of one measured banking application where each transaction requires an average of 40,000 instruction executions. At 2 micro seconds per instruction this is 80 milliseconds compared to our total of (5 + 15*2) milliseconds.) The loop ran at 5,000 baud and had a port buffer size of 128 bytes.

The parameter varied was r, the fraction of requests requiring remote service. The performance measures of interest were disc and loop utilization and loop mean queue length. (CPU utilization proved to be essentially zero and hence of little interest.) The analytic model predicts that disc utilization is independent of r, as is shown by substituting the values of p1,...,p4 from Section 5.1 into Equation 3.5) to obtain

$$\rho_{Disc} = \frac{15\lambda}{\mu_{Disc}}$$

For this experiment[4]

$$\lambda = \frac{125}{94.4} = 1.323 \text{ msg/sec.}$$

$$\frac{1}{\mu_{Disc}} = 30 * 10^{-3} \text{ sec.}$$

$$\text{so } \rho_{Disc} = 0.595$$

The simulation yielded values of ρ_{Disc} ranging from 0.59 to 0.57 for r ranging from 0.0 to 0.7.

Loop behaviour can be predicted analytically by combining Equations 3.2 and 3.6 with the p-values to obtain

$$\rho_p = 2r\lambda * \left[\frac{\mu_L}{2} \left(1 + \sqrt{1 - \frac{2\lambda\ell}{\mu_L}} \right) \right]^{-1}$$

where ρ_p refers to the utilization of a loop <u>port</u> and μ_L is the service rate parameter for the loop's transmission <u>line</u>. μ_L is derived as follows:

line speed = 5 * 10 bits/sec.

message length = 60 message bytes

+ 100 bytes of overhead

= 160 bytes x 8 bits/byte

= 1280 bits

$$\therefore \quad \mu_L = \frac{1}{256} \quad sec^{-1}$$

The overhead consists of 40 bytes for an allocation request message and 40 bytes for the response (Switch-to-Switch protocol) plus 20 bytes on the user's data message; see Appendix 2. The analytic model ignores the loop port limitation of 128 bytes.

For the line utilization we have

$$\rho_L = (2r) \times 2 \times \frac{\lambda}{\mu_L}$$

$$= \frac{4r\lambda}{\mu_L}$$

(for an N host network this would be $\rho_L = \frac{2Nr\lambda}{\mu_L}$).

Finally, we can invert Equation 3.7 to obtain

$$\rho_p = \frac{\bar{n}_p}{1+\bar{n}_p}$$

needed only because the simulation computes ρ_L and $\bar{n}_\rho$.

The predicted response time for remote transactions can be computed as follows. Each transaction goes through

- its local CPU: 5 msec.

- the loop queue:

$$E(W_p) = \frac{n_p}{\lambda_\ell} = (\bar{n}_p+1)\,\frac{1}{\mu_p}$$

(with the exponential assumption)

- the remote CPU: 5 msec.

- 15 times through the disc server

$$15 * E(W_D) = 15 * \left((\bar{n}_D + 1)\,\frac{1}{\mu_D}\right)$$

- back through the loop queue.

That is

$$\bar{t}_r = 0.01 + 2E(W_p) + 15E(W_D) \qquad \ldots\ldots.(5.6)$$

Analytic results obtained from these equations are compared with simulation results in Table 5.1, which shows close agreement for values of r up to 0.55. Figure 5.1 displays $\bar{t}_r$ as a function of r as predicted by the two models. In the simulation model the arrival rate of traffic from the terminals is approximated in the following way. It is known that output messages are 60 characters long. It takes 4.36 seconds to type these on a 110 baud terminal. The user then has an exponentially distributed "think" period with mean 90 seconds (a parameter of the run). For our comparisons the rate was thus taken to be (1/94.4) second in the analytic model calculations. The close agreement in the predicted response time is remarkable, since the analytic model ignores the interdependency of loop port behaviour, the detailed loop protocol and the dependency of $\bar{t}_r$ on system load. The queueing effects at these utilizations are small but not negligible - at 0.5 the expected service time is double that at 0. In any case the models agree quite well over the anticipated operating range of the network ($r < 0.5$).

The second experiment considered 10 hosts with 25 terminals each; all other parameters were as before. Disc utilizations were found to be 0.118 (simulation) and 0.125 (analytic), again within 5 per cent. It was necessary to extend the model of loop behaviour; for N hosts equation 3.6 generalizes to

$$\mu_p = \mu_L \sum_{r=0}^{N-1} \binom{N-1}{r} \frac{q^{N-r-1}(1-q)^r}{(r+1)}$$

where

$$q = \left(1 - \frac{2Nr\lambda}{\mu_L}\right)^{1/N}$$

Table 5.2 shows the analytic and simulation values for $\bar{n}_p$ and L; the <u>results diverge seriously</u> for $r > 0.4$. Here the interdependence of loop port queue lengths shows up at much lower loop utilizations. Thus it is unwise to attempt to predict the behaviour of several-host networks with high values of r using agreement in the predicted response time is remarkable, since the analytic model ignores the interdependency of loop port behaviour, the detailed loop protocol and the dependency of $\bar{t}_r$ on system load. The queueing effects at these utilizations are small but not negligible - at 0.5 the expected service time is double that at 0. In any case the models agree quite well over the anticipated operating range of the network ($r < 0.5$).

The second experiment considered 10 hosts with 25 terminals each; all other parameters were as before. Disc utilizations were found to be 0.118 (simulation) and 0.125 (analytic), again within 5 per cent. It was necessary to extend the model of loop behaviour; for N hosts equation 3.6 generalizes to

$$\mu_p = \mu_L \sum_{r=0}^{N-1} \binom{N-1}{r} \frac{q^{N-r-1}(1-q)^r}{(r+1)}$$

where

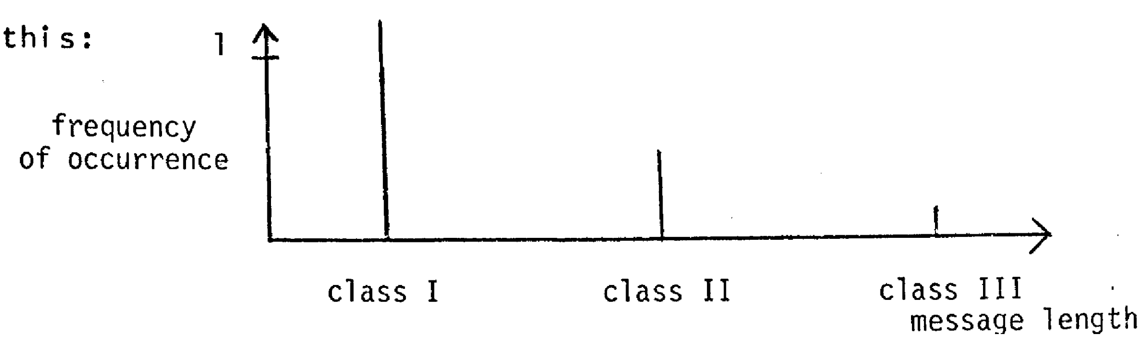
$$q = \left(1 - \frac{2Nr\lambda}{\mu_L}\right)^{1/N}$$

Table 5.2 shows the analytic and simulation values for $\bar{n}_p$ and L; the <u>results diverge seriously</u> for $r > 0.4$. Here the interdependence of loop port queue lengths shows up at much lower loop utilizations. Thus it is unwise to attempt to predict the behaviour of several-host networks with high values of r using our current analytic model.

5.3 Simulations of "realistic networks"

The third study investigated the feasibility of "real" networks of the type studied here, by choosing plausible parameter values for a "real" network based on current technology. Five hosts, each having 30 terminals and five disc drives were modelled. The disc mean service time was set at 80 msec. to reflect current hardware, and transactions arrived at each terminal at 10-second intervals. Loop speeds of 20 and 50 Kbaud were assumed; these speeds are readily obtainable in practice. Table 5.3 shows mean lengths of disc queues, loop queues and response times for several values of r. We note that for the 50 Kbaud loop, response times depend solely on disc speed for all values of r.

In a real commercial application one would expect message traffic to be approximated more closely by fixed length messages due to standard formats for transactions. However, the existence of many transaction classes will mean there are several fixed lengths, The message length distribution then will look like this:



The simulation model allows us to define several such message classes. Figure 5.2 illustrates the results of one experiment. There were two message classes used:

class I - transaction updates

input: 100 bytes (constant)

reply: 20 bytes (constant)

disc accesses: 10-20 (uniform)

terminal - 110 baud

think time - 60 sec.

class II - simple managerial queries

input: 100 bytes (constant)

reply: 500 to 5000 bytes (uniform)

disc accesses: 50-100 (uniform)

terminal - 1200 baud

think time - 300 sec.

The loop blocksize was 128 bytes, disc access time 25 msec. (exponential). The upper pair of curves shows remote response times for a 5 Kbits/sec. loop and the lower pair for a 10Kbits/sec. loop. The dashed line is for the managerial queries and the solid line is for the clerical transaction messages. The response times converged slightly as the loop saturated but the differences are determined principally by the fact that managerial queries require many more disc accesses.

This graph illustrates how the simulation model is now being applied. It is clear that for the traffic presented a 5Kbits/sec. loop is not adequate. The principle of geographic locality of reference leads us to expect values of r less than 0.3. But this is precisely where the "knee" of the response time curve is and if the system loading was subject to peaking, response times would deteriorate rapidly. The 10 Kbits/sec. loop, however, provides good response times all the way to r = 0.7 and would be stable under load fluctuations around r = 0.3.

It is not to be concluded that the ideal loop speed is 10 Kbits/sec. The choice depends upon the application. But this experiment and others like it have convinced us that required loop speeds will be well within the capabilities of current technology. We conclude that the bottleneck is more likely to be the disc accessing time. This is the case for a "network in a room" architecture, where the network hosts are centralized and high speed loops are easily installed. If the communications subnetwork were a "long haul loop" or a packet switching system very high speed communications is not so easily achieved. A recent study (10) indicates that the ARPANET can provide at most 10 Kbpds to 20Kpbs data rates to user programs based on 50 Kbps lines. We conclude that disc accessing time will not always be the limiting system parameter.

6. CONCLUSIONS AND FURTHER WORK

6.1 Conclusions

An entire network of computers, consisting of two hosts and a loop communications subnetwork, has been modelled as a queueing network and closed-form analytic expressions for its important performance measures have been obtained. To our knowledge, this represents an advance since previous work in the modelling of computer networks has dealt with communications subnetworks only. The domain of validity of the analytic solutions has been explored by comparison with simulations; it has been shown to be remarkably large in view of the rather gross assumptions which were made. In particular the agreement was good for low values of r -- corresponding to high geographic locality of reference. Thus we are led to have faith in the models over the expected operating range of the network.

The assumptions of the analytic model (open Jacksonian network, two hosts, loop port independence) clearly limit its general applicability. Its great value to us has been validation of the simulation model. (Prior to this we had uneasy feelings about the correctness of some 1500 lines of Simscript code).

The assumptions of the simulation model are far less severe. For example, arbitrary service time distributions are allowed and the loop is modeled in detail. Those that most strongly affect the generality of our study are:

- the assumption of separate servers (File Machine, Data Host and Loop Port)

the assumption of a loop subnetwork.

The former implies the existence of processors dedicated to the activities of transaction decoding, data base accessing and data transmission, respectively. This is true of our trial implementation, but implementation of all of these functions on a single processor would lead to interference effects and require a completely different queueing network model. The assumption of a loop subnetwork is also true of our initial implementation work. If, however, a packet-switched subnetwork were used, further modeling would be required. The authors are not aware of any "black-box" models of packet-switching networks. Current models require a knowledge of the network topology and routing policies in order to predict delays.

Having debugged the simulation by comparison with analysis, we were able to simulate a realistic network. Good response times for realistic traffic loads were predicted assuming the use of inexpensive, readily-available hardware components. Our belief that networks of minis can be effectively applied to transaction processing has been strengthened. In the third experiment response times for transactions requiring remote service were no more than two seconds longer than locally processed transactions when r was less than 0.6. The suitability of a loop using the Newhall-Farmer protocol as the communications subnetwork was confirmed. (These loops were originally designed for terminal-to-computer communication. The small loop port buffers (on the order of 32 bytes or less) suitable for this type of communications are intolerable for our application. The "best" buffer size is a function of data traffic characteristics, but we believe that sizes in the range of 128 to 256 bytes will be broadly useful and easily implemented.)

6.2 Further Work

We will attempt to extend the domain of validity of the analytic model by the use of two techniques. First, attempts will be made to incorporate the feedback effect into terminal arrival rates, by state-dependent arrival techniques [3] or by the use of a parallel server model. Second, a more detailed model of the loop is being constructed, based on the work of Cooper and Murray on cyclic servers.

Simulation studies will continue, to identify optimal network structures for commercial and library applications of transaction processing. Finally, the validity of the queueing network model can only be established by comparing its behaviour with that of an actual network. The construction of a prototype two-host network using the facilities of the Waterloo Computer Networks Laboratory is in progress.

6.3 Acknowledgements

The contributions of Mr. Edward Averill, Mr. David Buck, Mr. Carl Condon, Mr. Marek Irland, Professor Johnny Wong and Professor Erol Gelenbe are gratefully acknowledged, as is the support of the National Research Council of Canada under Negotiated Grant D-52.

Finally, the support of the University of Waterloo Mathematics Faculty Computing Facility is greatly appreciated.

APPENDIX 1

In this appendix the results of queueing theory that have been applied in section 3 are briefly summarized so that readers who are not familiar with this theory can understand the paper as whole. The derivation of the system state equations is explained in an "intuitive" manner. A complete treatment of this material is available in the book by Kleinrock [9, pp 147-160] and extension to more general network models can be found in the paper by Baskett et al [1].

To model a network of queues we begin by defining a system state as:

$$An = (n_1 , n_2 ,\dots, n_N) \qquad \dots\dots \quad A1.1$$

where n_i is the number of customers (messages, in our application) awaiting or receiving service at the ith service centre, and N is the number of service centres. The system changes states whenever a customer arrives, leaves, or moves from one service centre to another. We assume that the set of states A is countable and that the state transition probabilities for small time intervals δt are given by:

$$P(An \text{ at}(t + \delta t) \,/\, Ai \text{ at } t) = \alpha_{in}\, \delta t + 0(\delta t) \qquad \dots\dots\text{A1.2}$$

where $0(\delta t)$ has the property that

$$\lim_{\delta t \to 0} \frac{0(\delta t)}{\delta t} = 0$$

These transition probabilities are time independent and are independent of the past history of system states - the assumptions of a Markov process. A network in which all service times are exponentially distributed and all external arrivals to the network are Poisson streams, is an example of one that has these properties.

Let $Pn(t)$ be the probability of finding the system in state A_n at time t. Then we may write $Pn(t + \delta t)$ in terms of $Pn(t)$ as follows:

$$Pn\,(t+\delta t) = Pn(t) * \text{ prob. no transition occurs}$$

$$+ \sum_i Pi(t) * \text{ (prob. transition from Ai to An)}$$

(The sum over i is over all possible system states.)

i.e., $$Pn(t + \delta t) = Pn(t) * \prod_{i \neq n} (1-(\alpha_{ni}\ \delta t + 0\ (\delta t)))$$

$$+ \sum_{i} (\alpha_{in}\ \delta t + 0\ (\delta t))\ P_i(t)$$

$$\therefore \quad P_n(t+\delta t) = P_n(t) * [1- \sum_{i \neq n} \alpha_{ni}\ \delta t + 0(\delta t)]$$

$$+ \sum_{i} (\alpha_{in}\ \delta t + 0(\delta t))\ P_i(t)$$

Re-arranging terms we get:

$$\frac{Pn(t+\delta t) - Pn(t)}{\delta t} = - \sum_{i \neq n} \alpha_{ni}\ P_n(t) + \sum_{i} \alpha_{in}\ Pi(t) + \frac{0(\delta t)}{\delta t}$$

and taking $\lim_{\delta t \to 0}$ we obtain

$$\frac{d}{dt}\ P_n(t) = - \sum_{i \neq n} \alpha_{ni}\ P_n(t) + \sum_{i} \alpha_{in}\ P_i(t)$$

Next we assume that the system is stable (no server has a higher customer arrival rate than customer service rate) and than a steady state is achieved where $\frac{d}{dt} P_n(t)=0$ for all n. This gives

$$\sum_{i \neq n} \alpha_{ni}\ P_n = \sum_{i} \alpha_{in}\ P_i \qquad \text{.......A1.3}$$

Equation A1.3 is called the Global Balance Equation. If we interpret terms such as α_{ni} and $\alpha_{in} P_i$ as probabilistic "flows" out of and into state An, it simply states that the total flow out of state An equals the total flow in. Notice that the flows are implictly summed over all of the service centres. The explicit sums are over all states in the state space.

The Local Balance Equations are obtained by _assuming_ that the flow into An _due to arrivals at the ith service centre_ Sj, can be equated to the flow out of An due to departures from Sj. Thus, for $n_j > 0$, equation A1.3 becomes:

$$\mu_j \; P(n_1,\ldots,n_j,\ldots,n_N) = a_j \; P(n_1,\ldots,n_j-1,\ldots,n_N), \quad \forall_j \ldots A1.4$$

where μj is the service rate parameter for Sj (usually specified in the network definition) and is therefore, the departure rate if $n_j > 0$. Variable j is the arrival rate of customers to Sj (usually determined from the external arrival rates and the transition probabilities).

It is known that solutions to the local balance equations, if they exist, are also solutions to the global balance equation. For "Jacksonian" networks (Poisson arrivals, exponential servers, state-independent transition probabilities) the local balance equations can be solved. They are introduced because they are usually much easier to solve than the global balance equation. Sets of difference equations such as A1.4 have solutions of the form

$$P(n_1,\ldots,n_N) = C \prod_{j=1}^{N} \left(\frac{a_j}{\mu_j}\right)^{n_j} \qquad \text{.......A1.5}$$

as was shown by Gordon and Newell [5]. Equation A1.5 gives the state probabilities for a network of servers with exponential service time distributions when all externally arriving traffic is Poisson distributed. It remains to determine aj and C.

The arrival rate at Sj comprises the rate of arrival to Sj from outside the system, λj, plus the sum of arrival rates from all other Si. The arrival rate from Si is equal to the departure rate from Si weighted by a branching probability bij.[5] The departure rate from Si equals the arrival rate into Si unless the queue is unstable. Hence

$$a_j = \lambda_j + \sum_{i=1}^{N} b_{ij}\ a_i \qquad \text{.......A1.6}$$

which can be solved to determine the aj up to a multiplicative constant absorbed into C.

The constant C in Equation A1.5 is fixed by insisting that

$$\sum_i P_i = 1 \qquad \text{(summed over all i in the state space).}$$

Applying this to equation A1.5 we get

$$C = \prod_{j=1}^{N} (1-\rho_j) \qquad ....A1.7$$

where $\rho j = a_j / \mu_j$ and ρ_j has the usual interpretation as the traffic intensity).

To obtain mean queue lengths lengths from the local balance solution we re-write Equation A1.5 using A1.7 and the ρ-notation to yield

$$P(n_1,\ldots,n_N) = \prod_{j=1}^{N} (1-\rho_j) * \prod_{j=1}^{N} \rho_j^{n_j} \qquad ....A1.8$$

Hence

$$P(n_k = \alpha) = \prod_{j=1}^{N} (1-\rho_j) * \sum_{\substack{An\varepsilon A \\ n_k=\alpha}} \rho_1^{n_1}\ldots\rho_k^{\alpha}\ldots\rho_N^{n_N}$$

$$= (1-\rho_k) \quad \rho_k^{\alpha} \prod_{j\neq k} (1-\rho_j) * \prod_{j\neq k} \sum_{nj=0}^{\infty} \rho_j^{n_j}$$

$$= (1-\rho_k) \quad \rho_k^{\alpha} \prod_{j\neq k} (1-\rho_j) * \prod_{j\neq k} \frac{1}{(1-\rho_j)}, \text{ for } \rho_j < 1$$

$$= (1-\rho_k) \, \rho_k^{\alpha} \qquad ....A1.9$$

Thus each service centre behaves like an independent M/M/1 queueing system, with the only interactions being defined through the a_j .

We can derive the expected length of each queue to be:

$$E(n_k) = \sum_{\alpha=0}^{\infty} \alpha(1-\rho_k)\, \rho_k^{\alpha}$$

$$= \frac{\rho_k}{1-\rho_k} \qquad \text{.......A1.10}$$

APPENDIX 2

A brief summary of the operation of Newhall Loop operation is provided here. Correctly modelling the loop is the most difficult part of the analysis of the network.

A Newhall Loop consists of a single high speed digital line closed on itself. Repeater-ports distributed around the loop are used to regenerate signals and to allow subscribers to send and receive data. Only one port is allowed to send at a given time; "permission to send" is a control signal that is passed from port to port around the loop. A port can transmit a message of any length up to an upper bound imposed by the length of its buffer register (taken to be 128 bytes in the paper), and then must relinquish control to the next port downstream. If a port has nothing to send it simply passes control immediately upon receipt. There is a 1-bit time delay at each port (not modelled).

Each subscriber attached to the loop is referenced by the id. number of its port. Messages on the loop are prefixed by three fields: a WHOTO id., a WHOFROM id. and OPCODE. The op.codes are one of: RYB (are you busy), NBY (not busy), TBM (transfer) and END. When port A wishes to send a loop message to port B the following sequence occurs:

port A	port B
------	------
RYB?	
	NBY
TBM (msg text)	
END	

Each of these four events requires the sender to have the pass control bit. At any given time a port can be busy communicating with at most one other port. If the sender of an RYB regains the pass control bit without receiving an NBY response he assumes the receiver was busy and retries (up to 5 times). The message is then moved to the back of the loop queue for a further re-try. If this, in turn, fails an error recovery routine (applications logic) is invoked.

The host-host protocol that we have adopted requires each block (loop buffer sized unit) of a message to be sent as a separate call like that illustrated above. Further, the host-host protocol (cf. section 2.6) requires that each user message be preceded by two other messages (a request for core allocation to hold the user message and an acknowledgement). The analytic models described ignore all of this, and simply add the corresponding number of overhead bits to the original message length and view it as a single transmission. The simulation model covers all details.

FOOTNOTES

[1]The availability of public packet switched networks will alter the economic arguments here (e.g., Canada's Datapac and Infoswitch services, the Telenet service in the USA, and others that are now planned)., p. 5.

[2]It is assumed here that the transaction input identifies the data location. Another alternative requires local Data Host service, for directory searches, prior to data file accesses. The general problem of locating data will be discussed in future reports., p. 12.

[3]All random variables were exponentially distributed, unless otherwise stated., p.29.

[4](See p. 29 for explanation of 94 sec. as interarrival time.), p. 31.

[5]Note: bij is a parameter that describes customer "routing" through the network of queues. In a network of M/M/1 queues $\alpha_{ij} = b_{ij}\,\mu_j$., p.47.

REFERENCES

1. Baskett, F. Chandy, K. M., Muntz, R. R. and Palacios, F. G., "Open, Closed and Mixed Networks of Queues with Different Classes of Customers", J. A. C. M., Vol. 22, No. 2 pp. 248-260, April 1975.

2. Bolt, Beranek & Newman, Cambridge, Mass., U.S.A., Personal Communication.

3. Buzen, J.P., "Computational Algorithms for Closed Queueing Networks with Exponential Servers," C.A.C.M., Vol. 16, No. 9, September 1973.

4. Cooper, R. B. and G. Murray, "Queues Served in Cyclic Order," Bell System Technical Journal, March 1968.

5. Gordon, W. J. and Newell, G. F., "Closed Queueing Systems with Exponential Servers," Operations Research, Vol. 15, pp. 254-265, 1967.

6. Hayes, J. F., Sherman, D. N., "Traffic Analysis of a Ring Switched Data Transmission System," Bell System Technical Journal, Vol. 50, 9, Nov. 1971, pp. 2947-2978.

7. Jackson, James R., "Jobshop-Like Queueing Systems," Management Science, Vol. 10, No. 1, October, 1963, p. 131.

8. Kaye, A. R., "Analysis of a Distributed Control Loop for Data Transmission," Symposium on Computer-Communications Networks and Teletraffic, Polytechnic Institute of Brooklyn, New York, April 1972, pp. 47-58.

9. Kleinrock, L., Queueing Systems, Vol 1 : Theory, John Wiley & Sons, 1975.

10. Kleinrock, L., Naylor, W. E. and Opderbeck, H., "A Study of Line Overhead in the ARPANET", CACM, Vol. 19, No.1, January 1976, pp3.

11. Konheim, A. G. and Meister, D., "Service in a Loop System," J. A. C. M., Vol. 21, No. 2, April 1974.

12. Little, J. D. C., "A Proof of the Queueing Formula L= W,"Operations Research, Vol. p., pp. 383-387, 1961.

13. Manning, Eric and Peebles R. W., "A Homogeneous Network for Data Sharing - Communications," CCNG Report E-12, March 1974.

14. Moffat, D. E., "QPL-14: A Programming Language for the PDP-14 Computer", M.Sci., Queen's University, Kingston, Ontario, January 1973.

15. Newhall, E. E. and Farmer W. D., "An Experimental Distributed Switching System to Handle Bursty Computer Traffic," Proc. A.C.M. Conf. Data Communications, Pine Mountain, Georgia, U.S.A., October 1969.

16. Peebles, R.W., Buck, D.L., and Manning, Eric, "Simulation Studies of a Homogeneous Computer Network for Data Sharing," CCNG Report E-19, March 1974.

17. Peebles, R. W. and Manning Eric G., "A Homogeneous Network for Data Sharing: Software Architecture and Data Management," CCNG Report E-16, April 1974.

18. Pouzin, L., "Presentation and Major Design Aspects of the Cyclades Computer Network," DATACOMM 73, ACM/IEEE 3rd Data Communications Symposium, St. Petersburg, Florida, Nov. 1973, pp. 80-87.

19. Richardson, T. G., "The Effect of Protocol on the Response Time of Loop Structures for Data Communications," CRC Report No. 1261, Ministere des Communications, Ottawa, Canada, August 1974.

20. Roberts, L. G. 4Wessler, BB. D., "Computer Network Development to Achieve Resource Sharing," AFIPS, SJCC Vol. 36, May 1970, pp. 543-549.

21. Walden, D.C., "Host-to-Host Protocols", in Network Sytems and Software Infotech state of the art report number 24, 1975, pp. 287-316.

22. Wecker, S., "A Building Block approach to multifunction Multiple Processor Operating Systems," Proc. of AIAA Computer Network Conference, Alabama, April 1973.

23. Whittle, P., "Equilibrium Distributions for an Open Migration Process", J. Appl. Probability, Vol. 5, pp. 567-571.

24. Yuen, M. L. T., Black, B. A., Newhall, E. E., Venetsanopoulos, A. N., "Traffic Flow in a Distributed Loop Switching System," Proc. Brooklyn Symp. Comp-Comm. and Teletraffic, Polytechnic Press, New York, 1972, pp. 29-46.

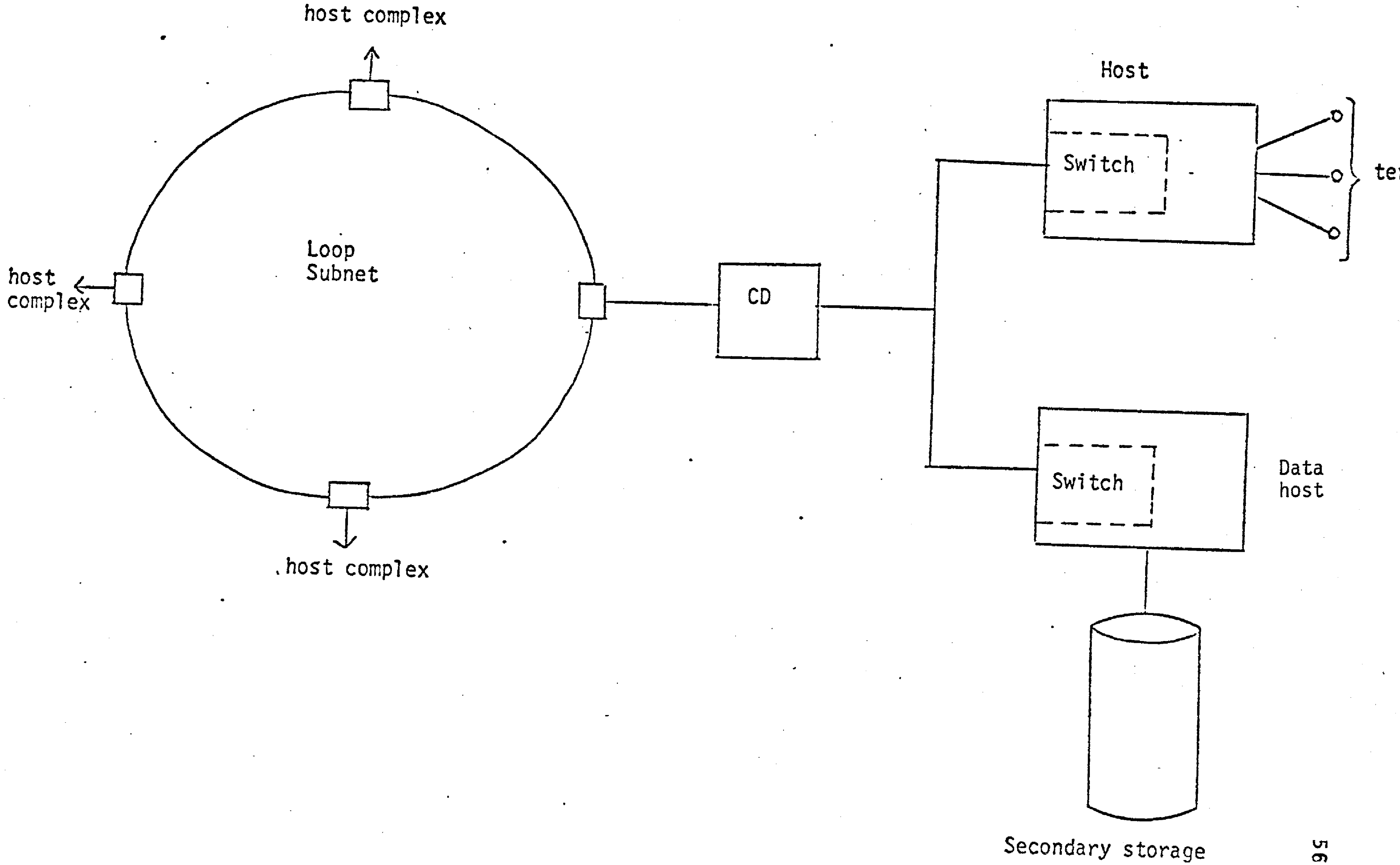


Figure 2.1 - Structure of Homogeneous Network

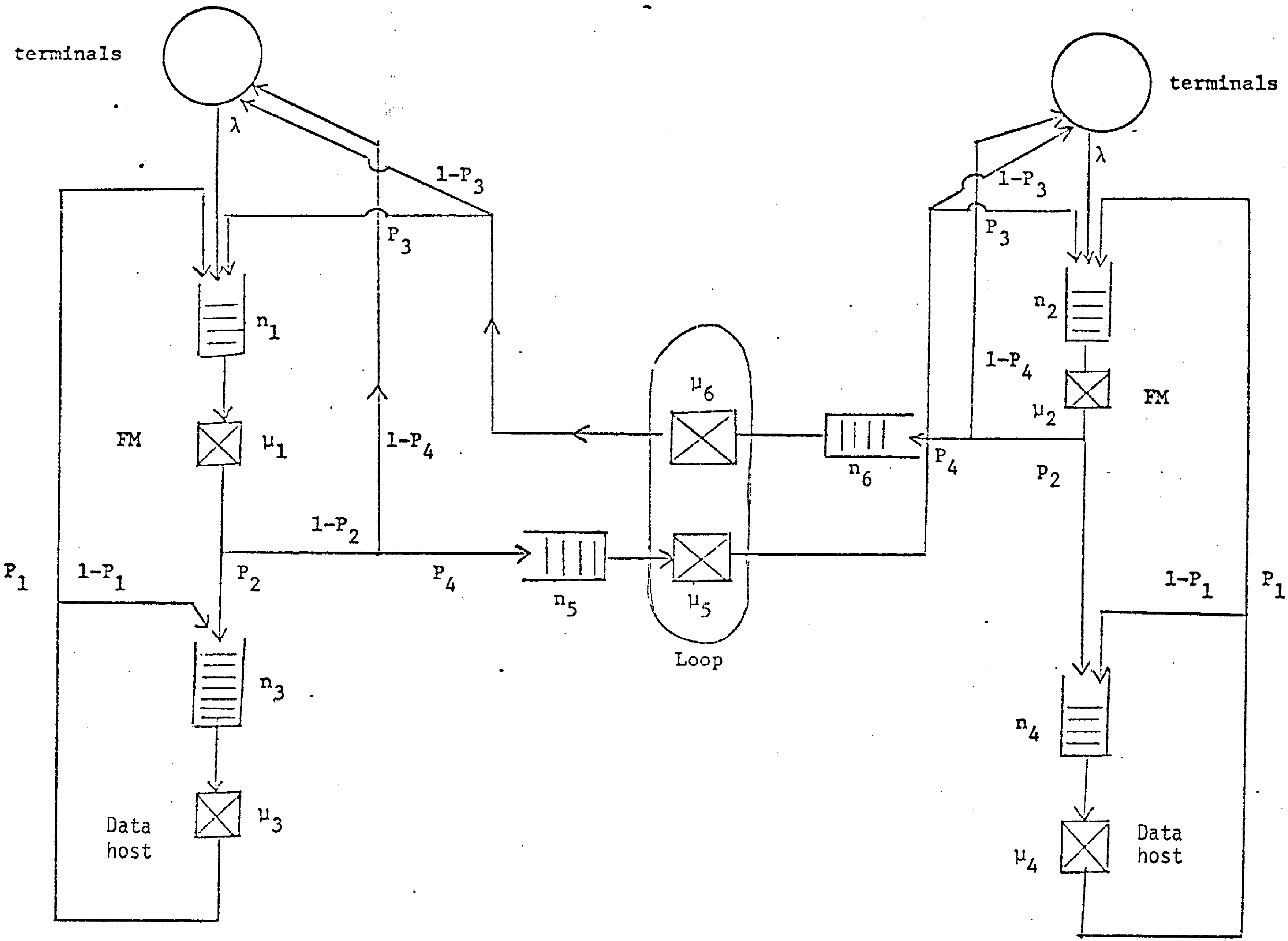


Figure 3.1 - A Queuing Network Model

fraction of remote traffic		0.1	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7
Analytic Model	ρ_L	0.14	0.28	0.35	0.42	0.49	0.56	0.62	0.69	0.76	0.83	0.90	0.97
	ρ_p	0.07	0.15	0.19	0.23	0.27	0.32	0.38	0.43	0.50	0.57	0.65	0.77
	$\bar{n}_p$	0.08	0.17	0.23	0.30	0.38	0.48	0.60	0.76	0.98	1.31	1.89	3.40
	$\bar{t}_r$	1.69	1.77	1.82	1.87	1.94	2.02	2.13	2.27	2.47	2.77	3.32	4.79
Simulation Model	ρ_L	0.15	0.29	0.34	0.40	0.48	0.56	0.63	0.67	0.75	0.82	0.90	0.92
	ρ_p*	0.07	0.16	0.19	0.24	0.29	0.35	0.41	0.45	0.55	0.63	0.75	0.76
	$\bar{n}_p$	0.08	0.19	0.24	0.32	0.41	0.54	0.70	0.81	1.23	1.73	3.03	3.54
	$\bar{t}_r$	1.64	1.75	1.74	1.87	1.92	2.02	2.06	2.23	2.67	3.15	4.42	4.86
	σ_t^2	0.78	0.81	0.77	0.91	0.97	0.97	1.30	1.34	2.22	3.81	8.86	8.45

*This ρ_p is calculated from the simulation $\bar{n}_p$ using $\rho_p = \dfrac{\bar{n}_p}{1+\bar{n}_p}$

Two host
125 terminals/host
1 disc/host
Loop Speed 5kbps
Transaction rate ≃ 1/92 sec.

Table 5.1: Comparison of Analytic and Simulation results for a two host network

Remote traffic		0.2	0.3	0.4	0.5
Analytic model	ρ_L	0.28	0.42	0.56	0.71
	$\bar{n}_p$	0.03	0.05	0.08	0.1
Simulation model	ρ_L	0.29	0.41	0.56	0.70
	$\bar{n}_p$	0.04	0.06	0.12	0.21

Table 5.2: Comparison for a Ten-host network

Loop Speed	remote traffic	$\bar{n}_D$	ρ_L	$\bar{n}_p$	Avg. remote response time (sec)	Local response time (sec)
20Kb	0.1	0.75	0.12	0.03	2.26	2.09
	0.3	0.76	0.36	0.11	2.32	2.09
	0.5	0.75	0.60	0.25	2.36	2.13
	0.7	0.71	0.88	1.33	3.09	2.03
	0.9	0.47	0.998	8.33	8.43	1.78
50Kb	0.1	0.77	0.05	0.01	2.29	2.12
	0.3	0.75	0.15	0.03	2.17	2.06
	0.5	0.77	0.24	0.06	2.21	2.15
	0.7	0.77	0.33	0.09	2.20	2.16
	0.9	0.75	0.43	0.14	2.19	2.05

Table 5.3: Behaviour of a Five Host network

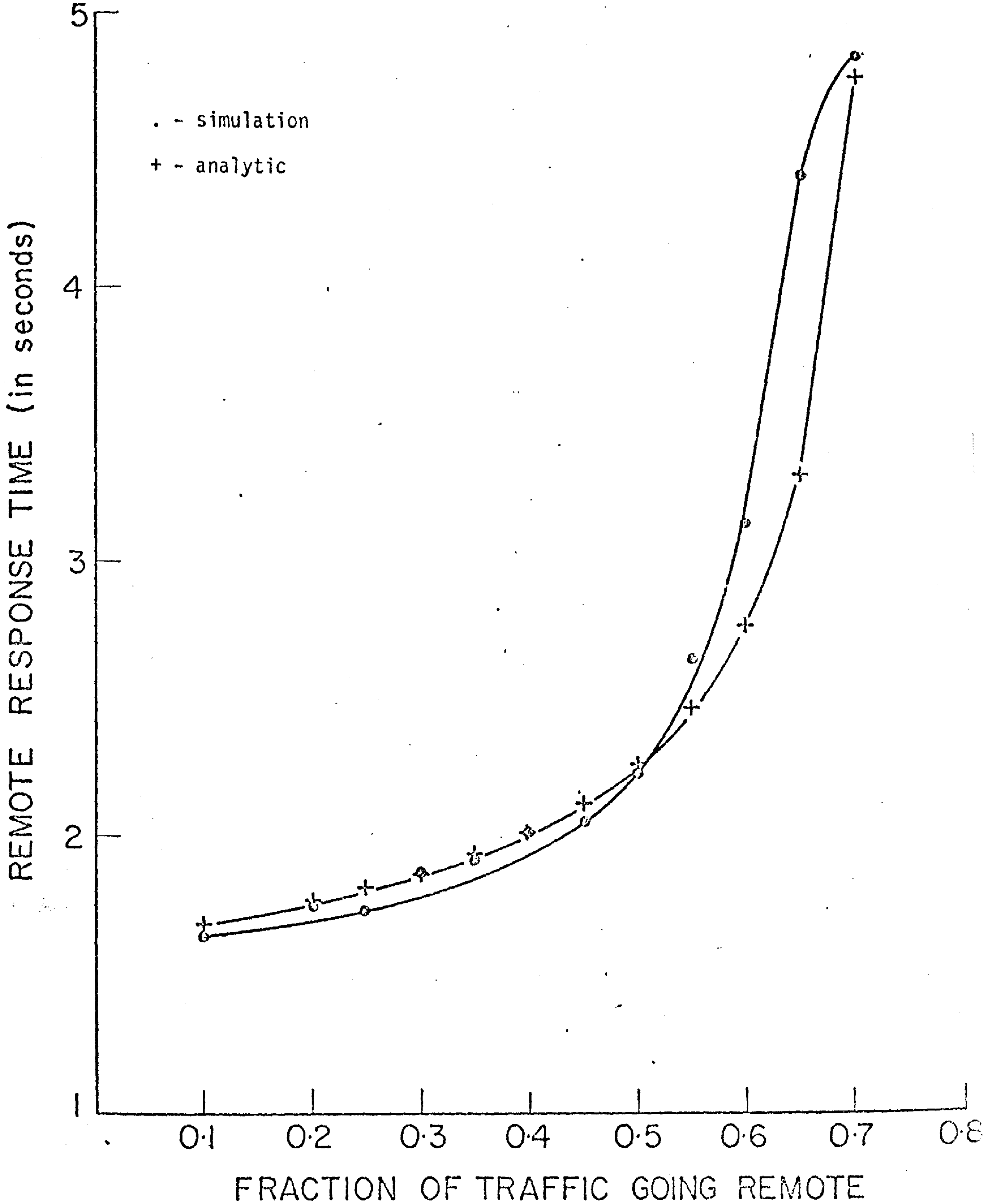


Figure 5.1: Remote response time versus remote traffic

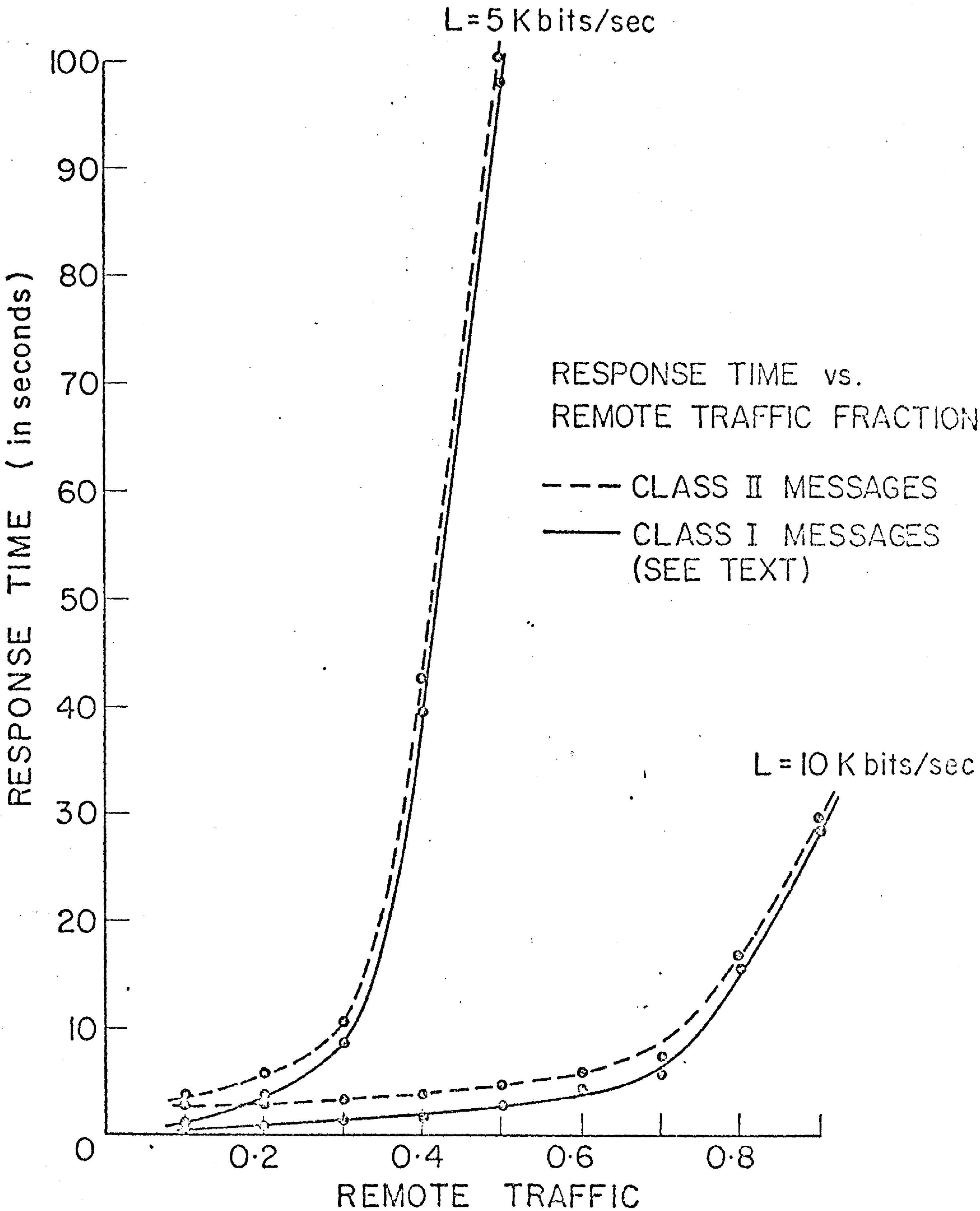


Figure 5.2: Mixed message types in a 10 host network