University of Waterloo

RESEARCH IN QUEUEING MODELS FOR COMPUTER COMMUNICATIONS

by

J. W. Wong

Department of Computer Science
and
Computer Communications Networks Group
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

CCNG E-Report E-81

Computer Communications Networks Group

ISSN 0384-5702

RESEARCH IN QUEUEING MODELS FOR COMPUTER COMMUNICATIONS

by

J. W. Wong

Department of Computer Science
and
Computer Communications Networks Group
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

CCNG E-Report E-81

This work was supported by the Department of Communications of Canada.

TABLE OF CONTENTS

INTRODUCTORY REMARKS

In the last decade, we have seen the development of a number of computer communication networks. Typical examples are the ARPA (Advanced Research Projects Agency) network in the U.S.A. [1], the CYCLADES network in France [2], and the DATAPAC network in Canada [3]. These networks, as characterized by Kleinrock [1], consist of a communication network which provides the message service and a user-resource network where terminal users interact with remote computers via a communication network. A typical configuration of such networks is shown in Figure 1.

This report deals with the application of queueing models to computer communication networks. It consists of four parts. Part 1 is a state of the art survey of modelling and analysis of computer communication networks. Part 2 involves the performance analysis of flow control techniques and message routing algorithms in communication networks. Emphasis is placed on networks that are organized as a multi-level hierarchy. An example of a two-level hierarchical network is shown in Figure 2. Models for user-resource networks are discussed in part 3. The response time to local and remote users is analysed. Overhead due to host-to-host protocol is taken into consideration. Part 4 contains a couple of miscellaneous items related to the project and the concluding remarks of this study.

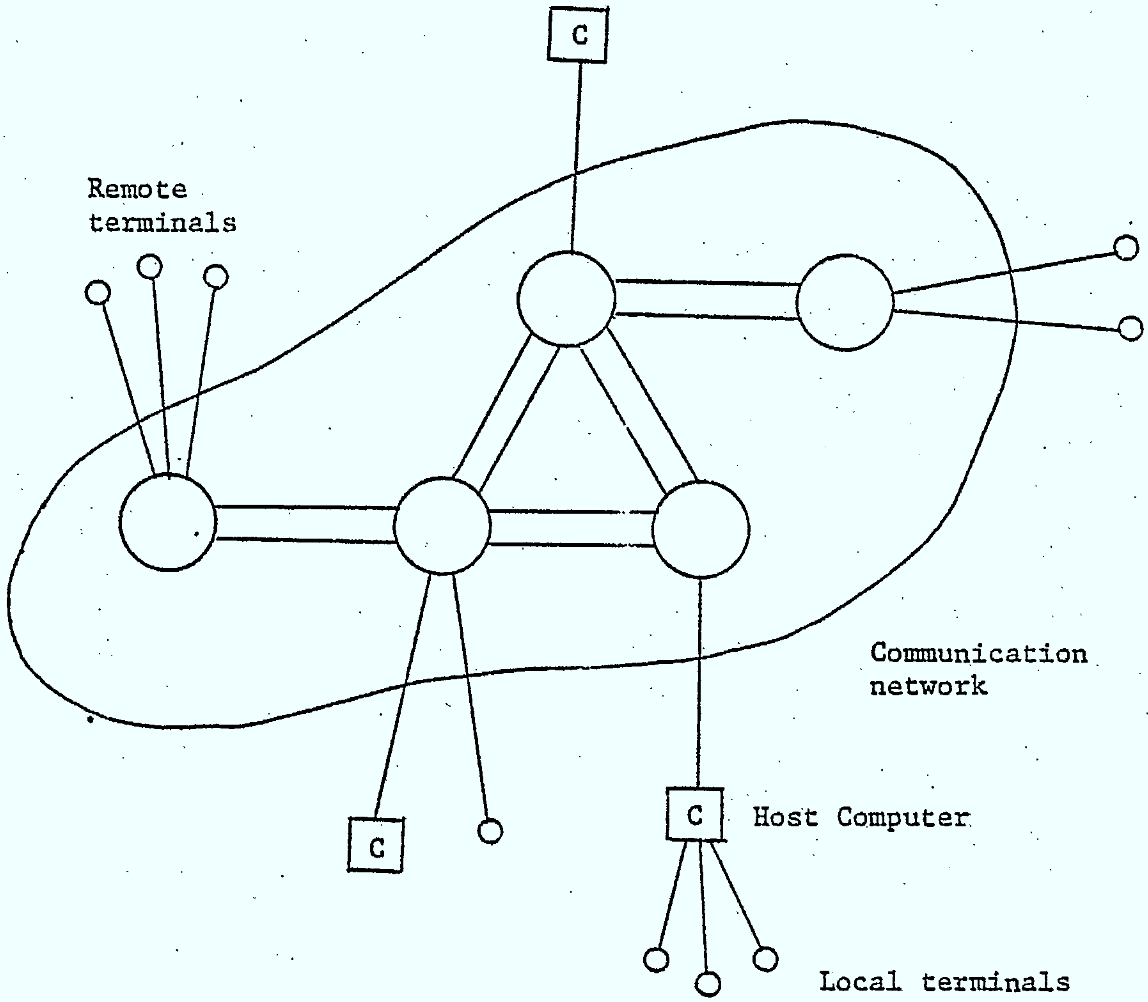


Figure 1. Computer Communication Network

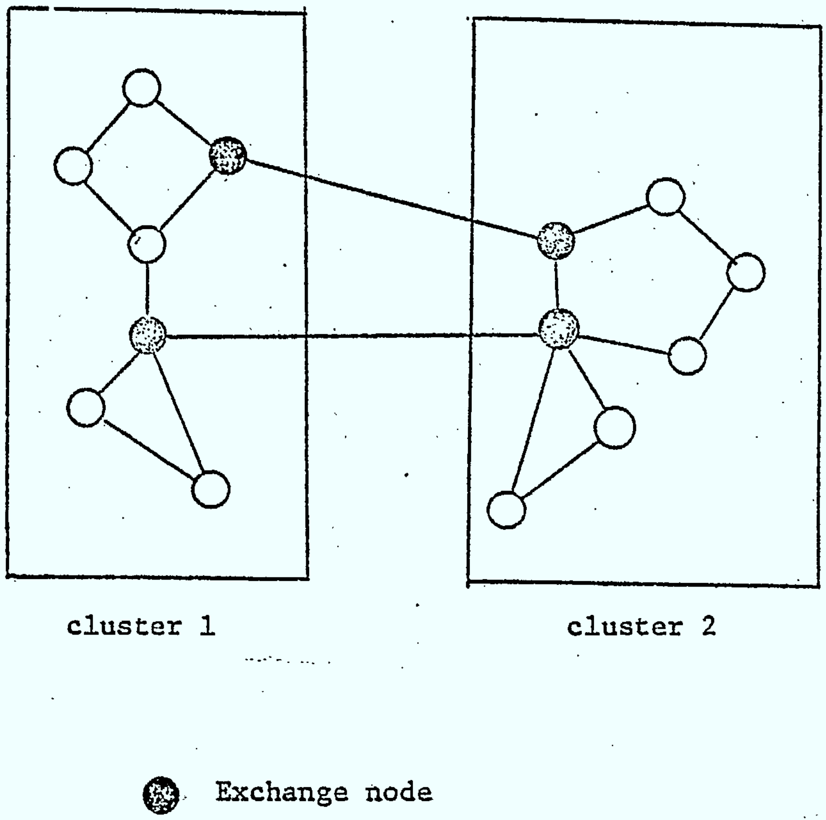


Figure 2. Hierarchical Network

PART 1 -- STATE OF THE ART SURVEY

A discussion on the application of queueing models to computer communication systems can be found in the reference: Kobayashi, H. and Konheim, A. "Queueing Models for Computer Communication Systems," IEEE Transactions on Communications, Vol. COM-25, Number 1, Jan 1977, pp.2-29. In this section, a more extensive survey on the application of queueing models to computer communication networks is given.

This survey is strictly dealing with the modelling, analysis, and design of communication networks because the performance analysis of user-resource networks has received little attention. Structurally, a communication network is a number of switching nodes connected together by a set of communication channels. The basic technique to deliver messages in this network is message-switching [1]. It involves the routing of messages from one node to another in a store-and-forward manner until they reach their destination. In the ARPA, CYCLADES, and DATAPAC networks, a message is further divided into packets. Each packet has a maximum size and each is independently routed through the network. This is known as packet-switching [1]. It has the advantage that packets of a message can be forwarded to their next node before the whole message is received, thus speeding up the delivery of messages from source to destination. Since packets can be treated as small messages, the same type of models have been used to analyse both message-switched and packet-switched networks [1,4].

1.1 THE BASIC MODEL

The first model for message-switched (or packet-switched) networks is developed by Kleinrock [1,4]. It is an open queueing network model of the type analysed by Baskett, et.al. [5]. Each server in this model is used to represent a communication channel. The processing time at the switching nodes and the propagation delay are assumed to be zero. The buffer space at each node is infinite, and the queueing discipline at each channel is first-come, first-served.

Messages are classified according to their source-destination node pair. In particular, messages originating at node s and destined for node d are called s-d messages. The arrival process of s-d messages from outside the network is assumed to be Poisson with mean rate γ_{sd}. All messages are assumed to have the same exponential message length distribution. Let $1/\mu$ be the mean message length and C_i the capacity of channel i. The service time (or data transfer time) at channel i is then exponential with mean $1/\mu C_i$. The key assumption in Kleinrock's model is the independence assumption [1,4] which states that each time a message enters a switching node, a new length is chosen from the exponential message length distribution. This assumption is required for the mathematical analysis to be tractable. Kleinrock [4] has shown by simulation that it gives accurate results to mean end-to-end (or source-to-destination) delay.

The routing algorithm is restricted to fixed or random. In fixed routing, a unique path (or ordered set of channels) is defined for messages belonging to each s-d pair. Random routing allows the possibility of alternate paths, and the routing algorithm selects one of these paths according to a probability distribution.

1.2 END-TO-END DELAY

Using the basic model, Kleinrock [1,4] derived an expression for the mean end-to-end delay. This expression is given by:

$$\overline{T} = \frac{1}{\gamma} \sum_{i=1}^{M} \frac{\lambda_i}{\mu_i - \lambda_i} \tag{1}$$

where $\gamma = \sum_{s,d} \gamma_{s,d}$ and λ_i is the mean rate of messages routed through channel i. λ_i is determined from:

$$\lambda_i = \sum_{s,d} \gamma_{s,d} * \text{fraction of s-d messages routed through channel i}$$

Eq.(1) has been used extensively in optimization problems for network design. A detail account of these problems is given in [1,7]. Of particular interest is the topological design problem where the location of the switching nodes and the message arrival rates are given. The problem is to select (a) the location and capacity of the channels and (b) the fraction of messages routed through each channel such that a constraint on the mean end-to-end delay is not violated and the cost of the network is minimized. Due to the combinatorial nature of network topology, the amount of computation required to solve this problem increases exponentially with the number of nodes. For relatively large networks, the optimal solution is computationally infeasible. Techniques to get sub-optimal solutions are discussed in [1,7]. These techniques have been used to provide guidelines for the design of the ARPA network [7,15].

A more general class of design problems has been considered by Chandy, et.al. [8]. They allow constraints on more detailed network performance measures. Examples of such measures are mean end-to-end delay by message class, variance and all moment of queue lengths, and the probability of exceeding a specified queue length.

In a model validation study, Kleinrock [1,6] extended his basic model to include nodal processing time, propagation delay, and other features pertinent to the ARPA network; and derived an approximate expression for the mean round-trip time. In the ARPA network, a RFNM (Request For Next Message) [1,18] is returned to the source node when a message is successfully delivered at its destination. The round-trip time is then the total delay experienced by a message and its RFNM. It is found that the mean round-trip time calculated from the model is reasonably close to that derived from the measurement data [1,6]. This study indicates that queueing network models are useful for performance prediction of existing networks.

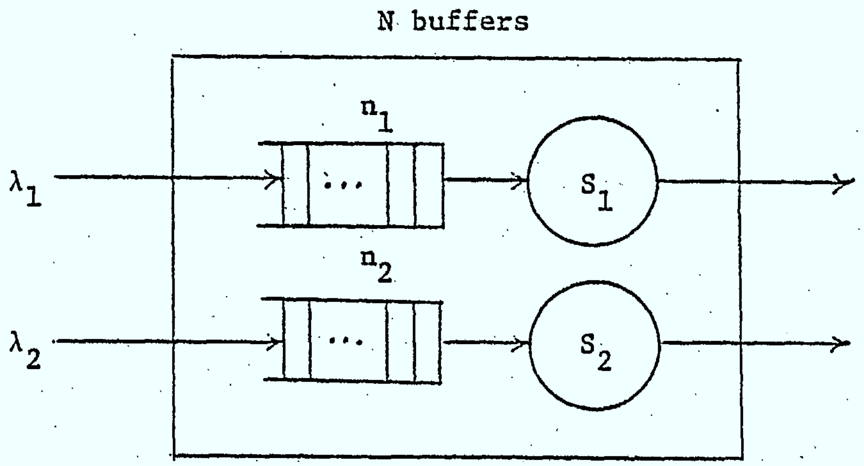
As mentioned in the last section, the independence assumption is needed for the mathematical analysis to be tractable. The special case of deterministic message length, however, does not require this assumption because the message length is always the same. Network models with deterministic service times and a first-come, first-served discipline at each node do not yield to exact analysis except for the case of a tandem queue [10]. Labetoulle and Pujolle [11] have considered the application of diffusion approximation to these models. Via an example network, they show that the approximate analysis compares favorably with results of simulation.

1.3 BUFFER MANAGEMENT

As a message is routed through a message-switched network, it occupies buffer space in the intermediate nodes. If the buffers are not properly managed, an increase in demand from one message class would reduce the availability of buffers to the others. This may lead to degradations in network performance.

The basic model described in section 1.1 assumes that the buffer space at each node is infinite. It is not adequate for performance analysis of buffer management schemes. Models with finite buffers are therefore needed. Irland [12] and Kamoun [13] have analysed a finite buffer model for a single switching node. An example of this model is shown in Figure 3. Messages are classified according to which channel they are routed out of the switch. Two of the schemes analysed in [12,13] are unrestricted sharing and restricted sharing. Let N be the number of buffers. In unrestricted sharing, the N buffers are allocated to messages on a first-come, first-served basis, no discrimination is made on the basis of message class. A message which finds all buffers in use is assumed to be turned away and will never return (i.e., lost). In restricted sharing, messages routed to channel i $(i = 1,2)$ can have a maximum of N_i buffers. By selecting $N_i < N$, this scheme can prevent one message class from occupying all the buffers.

For the case of Poisson arrivals and exponential service time distributions, Irland [12] and Kamoun [13] have derived the equilibrium joint queue length distribution using the technique of independent balance [5]. This distribution is then used to determine the switch throughput. Let λ_1 be the mean arrival rate of messages to channel 1. A comparison of the two buffer management schemes as λ_1 increases is shown



Class 1 arrivals: lost if $n_1 \leq N_1$
or $n_1 + n_2 \leq N$

Class 2 arrivals: lost if $n_2 \leq N_2$
or $n_1 + n_2 \leq N$

Figure 3. Single node model with finite buffers

in Figure 4. With unrestricted sharing, the throughput is first increased, and then degraded significantly. The initial improvement is due to the increased utilization of channel 1. However, when λ_1 is large, almost all buffers are occupied by messages to channel 1. The utilization of channel 2 is much reduced. Consequently, the switched throughput is degraded significantly.

With restricted sharing, messages routed to channel 2 are guaranteed at least $N - N_1$ buffers. This would prevent a significant reduction in the utilization of channel 2. As a result, the switch throughput stays at a relatively high level when λ_1 is large. Restricted sharing is therefore a better scheme for buffer management.

The analysis of a total network model with finite buffers at each node is very difficult. Irland [12] has attempted to solve this model by an iterative approximation technique and found that the network throughput under unrestricted and restricted sharing at each node has essentially the same properties as those shown in Figure 4.

Lam [16] has also used an iterative approximation technique to solve a network model with finite buffers at each node. His model includes a positive acknowedgement scheme [14] where a message is kept in a buffer until an acknowledgement is received from the node to which this message is routed. If the acknowledgement is not received within a time-out period, the message is re-transmitted.

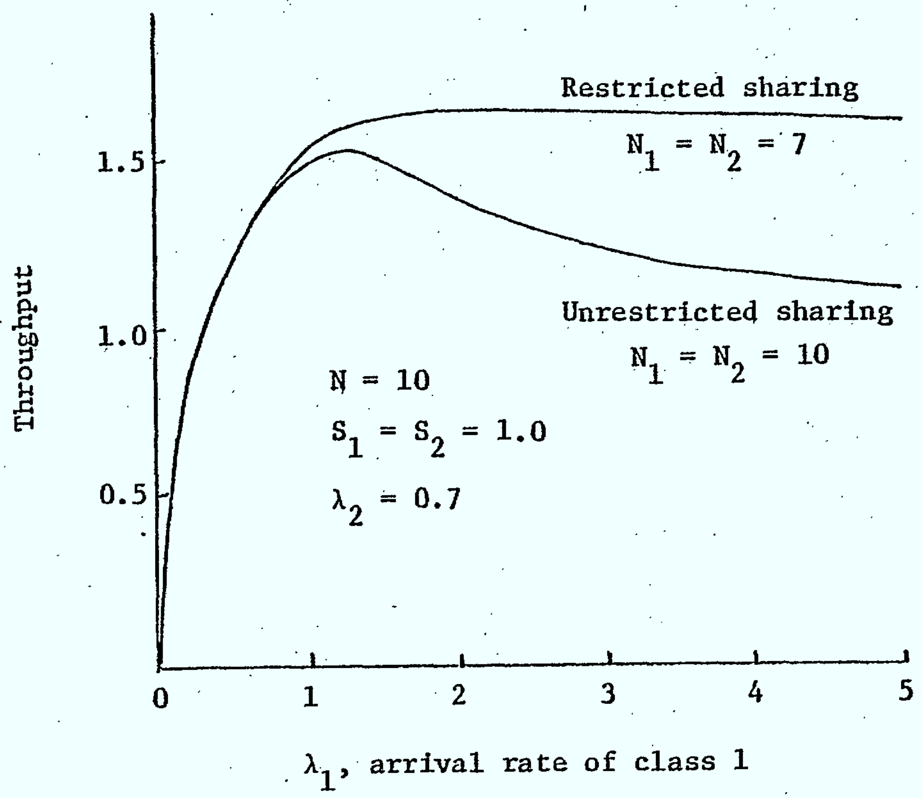


Figure 4. Single node model: throughput vs. λ_1

In a separate study, Lam and Reiser [17] have analysed a buffer management scheme based on input buffer limits. This scheme distinguishes messages according to whether they are input messages or transit messages, and a limit is placed on the fraction of buffers that the input messages can occupy. The advantage of this scheme is to devote a fraction of the buffers to messages already in transit so that network throughput can be improved.

1.4 FLOW CONTROL

In a message-switched network, flow control is a mechanism to prevent congestion by regulating the entry of messages to the network. A common flow control technique is the end-to-end control where a limit is placed on the number of messages in each virtual circuit. In a computer communication network, a virtual circuit [3] is a logical channel connecting together two users in the network. Examples of end-to-end control are the RFNM feature in the ARPA network [1,18], and the window technique in DATAPAC [3].

Pennotti and Schwartz [22] have used a tandem queue model to analyse the end-to-end control scheme. An example of this model is shown in Figure 5. There are two classes of messages. Link messages are those routed through all channels while external messages are routed through one channel only. End-to-end control is modelled by placing a

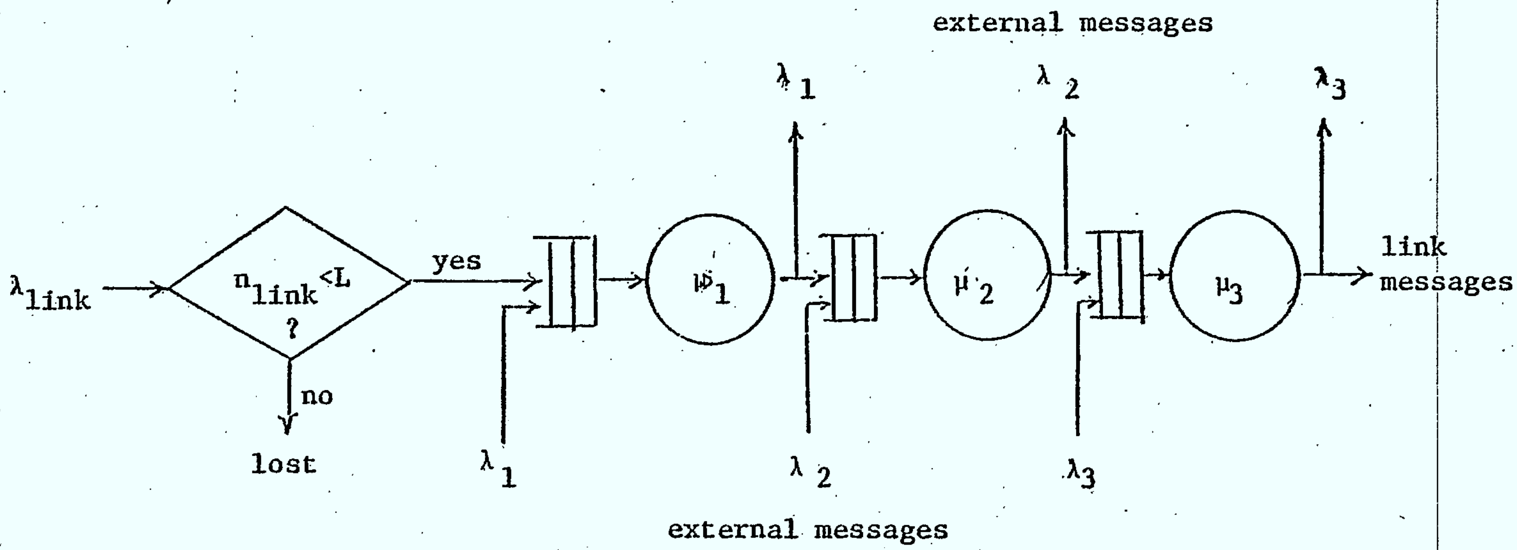


Figure 5. Tandem queue model for end-to-end control

limit L on n_{link}, the number of link messages in the network. A link arrival which finds n_{link} = L is assumed to be lost. With Poisson arrivals and exponential message length, Pennotti's model is a special case of the general queueing network model analysed by Baskett, et.al. [5]. Using Baskett's result, Pennotti [22] derived an expression for a link-loading factor which is the fractional increase in the mean delay of external messages due to the presence of link messages. Let λ_{link} be the mean arrival rate of link messages. A plot of the link-loading factor against λ_{link} is shown in Figure 6. With end-to-end control, the link-loading remains finite and is a slowly increasing function of λ_{link}. The corresponding curve for the case of no control is also shown. It shows that the link-loading becomes unbounded as λ_{link} increases.

Pennotti's work was later extended by Chatterjee, et.al. [23] to include random routing.

A second technique for admission control is the isarithmic technique originally suggested by Davies [19]. This technique places a limit on the total number of messages in the network, no discrimination is made on the basis of source or destination. This is done by circulating a fixed number of permits in the network, and requiring a message to secure a permit before it can be admitted to the network. An isarithmic control scheme has been implemented by Price [20] in his network simulation model.

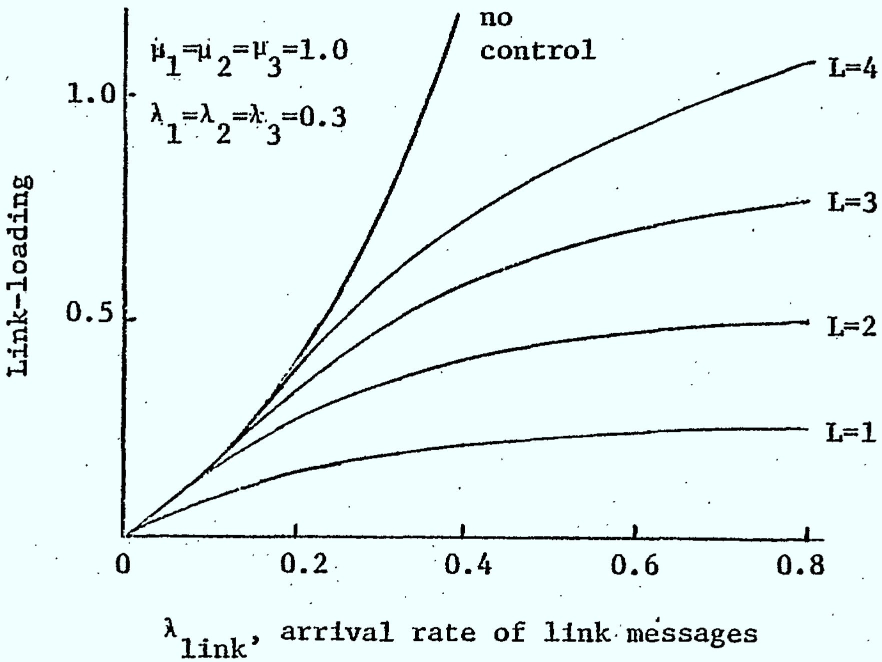


Figure 6. Tandem queue model: link-loading vs. λ_{link}

Davies [19] remarked that isarithmic control alone may not be effective, and suggested that it should be used as a supplement to other flow control techniques. This suggestion has motivated the analysis of a two-level control by Wong and Unsoy [21]. At the first level, a limit is placed on the total number of messages in the network. At the second level, disjoint groups of virtual circuits are formed and separate limits are placed on the number of messages belonging to each group.

Wong's model [21] is essentially a generalization of the basic mode in section 1.1 to include flow control. Message classes are assigned to user groups. Let n_u be the number of group u messages and n be total number of messages in the network. The two-level control is modelled by requiring that $n \leq L$ and $n_u \leq L_u$ for all u. A group u arrival is assumed to be lost if $n = L$ or $n_u = L_u$. By properly defining the groups and their limits, this model can be used to study Davies' isarithmic and the end-to-end control schemes.

Similar to the model analysed by Irland [12] and Kamoun [13], the network model with two-level control is solved by the technique of independent balance [21]. Results from numerical examples have indicated that Davies' isarithmic control is not capable of preventing throughput degradation when the demand from one user group is increased. Throughput degradation can be avoided if a two-level control is used. This observation is consistent with Davies' suggestion [19] that isarithmic control should be used as a supplement to other flow control schemes.

As a final note, Lam [24] has analysed a general queueing network model with population size constraints. The switch models for buffer management [12,13,16,17] and the network models for flow control [21,22,23] are special cases of Lam's model.

PART 2 -- COMMUNICATION NETWORKS

2.1 DISTRIBUTION OF END-TO-END DELAY

In a message-switched communication network, the end-to-end delay is the elapsed time from the arrival of a message at its source node to the successful delivery of this message at its destination. In this section, we establish a theorem regarding the distribution of end-to-end delay of messages belonging to a particular s-d pair. This is a more detailed characterization of network delay than Kleinrock's classical formula in Eq.(1). It allows us to determine statistics such as the mean, variance, and percentiles of end-to-end delay.

This result is also important for the analysis or simulation of user-resource networks. The message-switched communication network can be treated as a "black-box" with an appropriate delay distribution. This would reduce (a) the complexity of analysis and (b) the cost of simulation.

2.1.1. MODEL DESCRIPTION

Our derivation is based on Kleinrock's model described in section 1.1. The model assumptions are summarized below:

(a) M servers, each representing a communication channel.

(b) Capacity of channel i, i = 1,2,...,M, is C_i.

(c) Queueing discipline at each channel is FCFS.

(d) Nodal processing time and propagation delay are negligible.

(e) Unlimited buffer space at each node.

(f) Messages are classified according to s-d pairs.

(g) Arrival process of messages belonging to each class is Poisson.

(h) Message length distribution (same for all classes) is exponential with mean $1/\mu$.

(i) Kleinrock's independence assumption [1,4] is used.

(j) Fixed or random routing.

For convenience in derivation, we also use the following notation:

(a) r -- message class (i.e., s-d pair)

(b) R -- total number of classes; in a network with N nodes, R can be as large as N(N-1).

(c) $\gamma(r)$ -- mean arrival rate of class r messages

(d) Fixed routing:

a(r) -- the path (or ordered set of channels) for class r

(e) Random routing:

k_r -- number of alternate paths for class r

$a_j(r)$ -- set of channels in j-th path, $j = 1,2,\ldots,k_r$

$q_j(r)$ -- probability that j-th path is selected

The routing algorithms defined above are path-oriented algorithms. When a message enters the network at its source node, a path is immediately selected, and this message will be routed through the selected path. There is no feedback of messages to a previously visited node.

2.1.2 QUEUEING ANALYSIS

We first consider the case of fixed routing. Let λ_{ir} $(i = 1,2,\ldots,M;\ r = 1,2,\ldots,R)$ be the mean arrival rate of class r messages to channel i. With fixed routing, λ_{ir} is given by:

$$\lambda_{ir} = \begin{cases} \gamma(r) & \text{if channel } i \in a(r) \\ \emptyset & \text{otherwise} \end{cases} \tag{2}$$

Let ρ_{ir} be the utilization of channel i by class r messages,

$$\rho_{ir} = \lambda_{ir} / (\mu C_i) \tag{3}$$

The total utilization of channel i (denoted by ρ_i) can then be written as:

$$\rho_i = \sum_{r=1}^{R} \rho_{ir} \tag{4}$$

We require that $\rho_i < 1$ for $i = 1,2,\ldots,M$. This is equivalent to the requirement that no channel is saturated, the condition for existence of a stochastic equilibrium.

Let $t_r(x)$ be the probability density function (pdf) of the end-to-end delay of class r messages, and $T_r^*(s)$ be its Laplace Transform, i.e.,

$$T_r^*(s) = \int_0^\infty e^{-sx}\, t_r(x)\, dx \tag{5}$$

The main result can be stated as follows:

<u>Theorem</u>: For our model of a message-switched network with fixed routing,

$$T_r^*(s) = \prod_{i \varepsilon a(r)} \frac{\mu C_i(1 - \rho_i)}{S + \mu C_i(1 - \rho_i)} \tag{6}$$

Proof:

We first prove two lemmas.

Lemma 1:

Let $N_r(z)$ be the generating function of the total number of class r messages in the network at equilibrium,

$$N_r(z) = \prod_{i \varepsilon a(r)} \frac{1 - \rho_i}{1 - \rho_i + \rho_{ir}(1 - z)} \tag{7}$$

Proof:

Let $(S_1, S_2, \ldots, S_M)$ be the state of our network model where $S_i = (n_{i1}, n_{i2}, \ldots, n_{iR})$ is the state of channel i and n_{ir} is the number of class r messages (in queue or in transmission) at channel i. Since our model is a special case of the general queueing network model analysed by Baskett, et.al. [5], we apply Baskett's result and get the following expression for the equilibrium state probabilities:

$$P(S_1, S_2, \ldots, S_M) = P_1(S_1)\, P_2(S_2) \ldots P_M(S_M) \tag{8}$$

where

$$P_i(S_i) = (1 - \rho_i) \left[\sum_{r=1}^{R} n_{ir} \right] ! \prod_{r=1}^{R} \frac{1}{n_{ir}!} \rho_{ir}^{n_{ir}} \tag{9}$$

and ρ_{ir} and ρ_i are defined in Eqs.(3) and (4) respectively.

$P_i(S_i)$ is also the marginal probability that channel i is in state S_i. Let $N_{ir}(z)$ be the generating function of the number of class r messages at channel i. $N_{ir}(z)$ can be written as:

$$N_{ir}(z) = \sum_{k=0}^{\infty} z^k \sum_{\substack{\text{all states} \\ S_i \text{ s.t. } n_{ir}=k}} P_i(S_i) \tag{10}$$

Using Eq.(9) in Eq.(10), and after simplification, we get:

$$N_{ir}(z) = \frac{1 - \rho_i}{1 - \rho_i + \rho_{ir}(1 - z)} \tag{11}$$

From Eqs.(8) and (11), it is easy to see that $N_r(z)$ has the product form in Eq.(7) because the equilibrium state probabilities are the same as if the state variable of the M channels are mutually independent.

Lemma 2:

Let $p_n(r)$ be the equilibrium probability that the number of class r messages in the network is n, and

$d_n(r)$ the probability that a class r departure left behind n class r messages.

Then

$$p_n(r) = d_n(r) \qquad n = 0,1,2,\ldots \tag{12}$$

Proof:

Let $\alpha_n(r)$ be the probability that a class r arrival sees n class r messages in the network. Since the number of class r messages in the network changes by unit step values only, it follows from [26] that:

$$\alpha_n(r) = d_n(r) \qquad n = 0,1,2,\ldots \tag{13}$$

Since the arrival process of class r messages is Poisson, we also have [27]:

$$\alpha_n(r) = p_n(r) \qquad n = 0,1,2,\ldots \tag{14}$$

Combining Eqs.(13) and (14), we get Eq.(12).

We now prove our theorem. Let $D_r(z)$ be the generating function of the number of class r messages left behind by a class r departure. Eq.(12) implies that:

$$N_r(z) = D_r(z) \tag{15}$$

Since we have assumed fixed routing and a FCFS discipline at each channel, the number of class r messages left behind by a class r departure must equal to the number of class r arrivals during the stay of the departing message in the network. Since we have also assumed a Poisson arrival process, $D_r(z)$ is given by [27]:

$$D_r(z) = T_r^*\,(\gamma(r) - \gamma(r)z) \tag{16}$$

Substituting s for $\gamma(r)-\gamma(r)z$, Eq.(16) is reduced to:

$$T_r^*(s) = D_r(1 - s / \gamma(r)) \tag{17}$$

Finally, using Eqs.(7) and (15) in Eq.(17), we get:

$$T_r^*(s) = \prod_{i \varepsilon a(r)} \frac{\mu C_i(1 - \rho_i)}{s + \mu C_i(1 - \rho_i)}$$

QED

Let $|a(r)|$ be the number of channels in a(r). Our theorem indicates that the end-to-end delay of class r messages is given by the sum of $|a(r)|$ independent random variables. The i-th variable in this sum is exponential with mean $(\mu C_i(1-\rho_i))^{-1}$; it can be interpreted as the delay (queueing time + data transfer time) at the i-th channel of a(r). It is of interest to note that the mean of this i-th random variable is a function of ρ_i and not ρ_{ir}, implying that all messages routed through a particular channel have the same delay distribution at this channel.

Reich [28,29] has considered the special case of a tandem queue with one class of customers. He proved that the output process of each server is Poisson and the delays at the individual servers are mutually independent. The product form for the Laplace Transform of end-to-end delay is therefore obvious. We have not attempted to prove similar properties for our network model although our theorem indicates that the distribution of end-to-end delay is the same as if the delays at the individual channels are mutually independent.

$T_r^*(s)$ can easily be inverted, by using partial fractions [27], to give $t_r(x)$. The mean $\overline{T}_r$ and variance σ_r^2 of class r delay can also be obtained from $T_r^*(s)$. They are given by:

$$\overline{T}_r = \sum_{i\varepsilon a(r)} \frac{1}{\mu C_i(1-\rho_i)} \qquad (18)$$

and

$$\sigma_r^2 = \sum_{i\varepsilon a(r)} \frac{1}{[\mu C_i(1-\rho_i)]^2} \qquad (19)$$

2.1.3. GENERALIZATION TO RANDOM ROUTING

With random routing, a class r message can be routed through one of k_r alternate paths, and the j-th path is selected with probability $q_j(r)$. Our analysis in the last section is applicable if we treat each alternate path as a separate message class. We thus replace class r by k_r artificial classes. Let these classes be $r_1, r_2, \ldots, r_{k_r}$, then

$$\gamma(r_j) = \gamma(r)\, q_1(r) \qquad (20)$$

$$\gamma_{ir_j} = \begin{cases} \gamma(r_j) & \text{if channel } i \;\varepsilon\; a_j(r) \\ \emptyset & \text{otherwise} \end{cases} \qquad (21)$$

and

$$\rho_i = \sum_{r=1}^{R} \sum_{j=1}^{k_r} \lambda_{ir_j} / (\mu C_i) \tag{22}$$

Applying our theorem in the last section, we get:

$$T^*_{r_j}(s) = \prod_{i\varepsilon a_j(r)} \frac{\mu C_i(1 - \rho_i)}{s + \mu C_i(1 - \rho_i)} \tag{23}$$

$T^*_r(s)$ can then be obtained by removing the artificial class from our model, i.e.,

$$T^*_r(s) = \sum_{j=1}^{k_r} q_j(r)\, T^*_{r_j}(s) \tag{24}$$

Similar to the case of fixed routing, this Laplace Transform can also be inverted to give $t_r(x)$. As to the mean and variance of class r delay, we have:

$$\overline{T}_r = \sum_{j=1}^{k_r} q_j(r) \sum_{i\varepsilon a_j(r)} \frac{1}{\mu C_i(1 - \rho_i)} \tag{25}$$

and

$$\sigma_r^2 = \sum_{j=1}^{k_r} q_j(r) \left[\sum_{i\varepsilon a_j(r)} \frac{1}{[\mu C_i(1 - \rho_i)]^2} - [\sum_{i\varepsilon a_j(r)} \frac{1}{\mu C_i(1 - \rho_i)}]^2 \right] - \overline{T}_r \tag{26}$$

2.1.4. GENERALIZATION TO MESSAGE GROUPS

It is often useful to consider the end-to-end delay of messages belonging to a group of source-destination pairs. For example, we can study the delay characteristics of (a) messages sent among a subset of the nodes, (b) messages sent from a particular source node, or (c) messages sent to a particular destination. We thus define a group G to contain a number of message classes, and a message is said to belong to group G if its class membership is in G. It is easy to see that our result for random routing is directly applicable to message groups. We thus have the following result for $T_G^*(s)$, the Laplace Transform of the pdf of group G delay:

$$T_G^*(s) = \sum_{r \in G} \frac{\gamma(r)}{\gamma_G} T_r^*(s) \qquad (27)$$

where $T_r^*(s)$ is given by Eq.(6) or (24), and $\gamma_G = \sum_{r \in G} \gamma(r)$.

In the special case that all classes of messages belong to a single group, we have the Laplace Transform of the pdf of the overall end-to-end delay:

$$T^*(s) = \sum_{r=1}^{R} \frac{\gamma(r)}{\gamma} T_r^*(s) \qquad (28)$$

where $\gamma = \sum_{r=1}^{R} \gamma(r)$

2.1.5. NUMERICAL EXAMPLES

Our numerical examples are based on the hypothetical network shown in Figure 7. This network has 5 nodes and 10 channels. The external arrival rate of messages belonging to each source-destination pair is given by the traffic matrix in Figure 8. All channels are assumed to have the same capacity, and the mean message length is chosen such that the mean data transfer time at each channel (i.e., $1/\mu C_i$) has a value of 0.1.

We first consider the case of fixed routing and assume that the routing algorithm is based on the shortest path. In our example network, there is a unique shortest path between each pair of nodes. Suppose we are interested in the end-to-end delay from node 1 to node 2. Denoting this source-destination pair by class 1, we apply Eq.(6) and get:

$$T_1^*(s) = [3 / (s + 3)]\,[2 / (s + 2)]\,[4 / (s + 4)] \tag{29}$$

This Laplace Transform can be inverted to give:

$$t_1(x) = -24\ell^{-3x} + 12\ell^{-2x} + 12\ell^{-4x} \tag{30}$$

A plot of $t_1(x)$ is shown in Figure 9. The mean, variance, and 90-percentile of class 1 delay are also shown.

We next consider the case of random routing and assume that 25% of class 1 messages are shifted to the path {1,3,8,9}. This implies that the remaining 75% are sent over the shortest path {1,5,9}. Applying Eq.(24), and

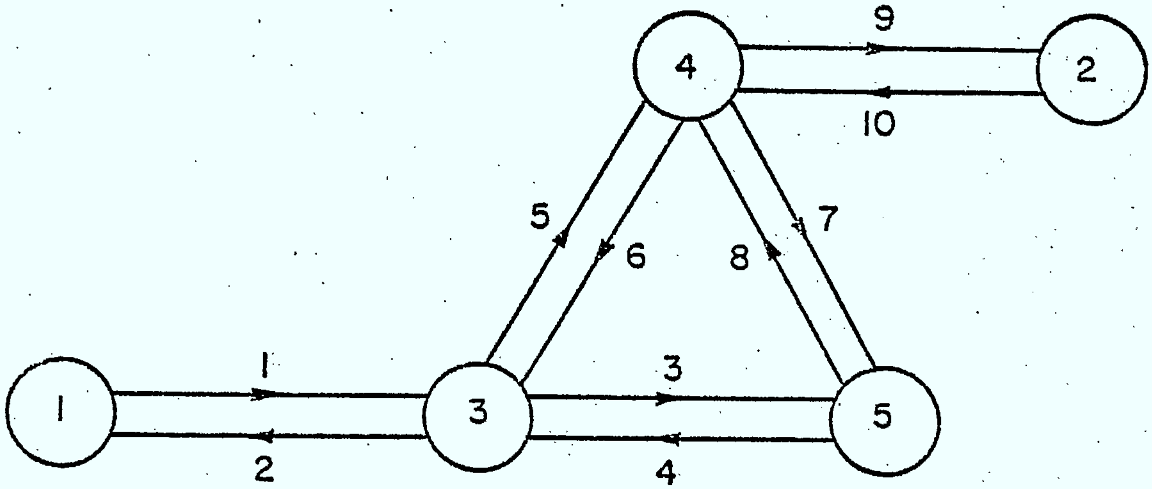


Figure 7. Hypothetical Network

DESTINATION

SOURCE	1	2	3	4	5
1	0	2	1	3	1
2	2	0	1	2	1
3	1	1	0	2	4
4	3	2	2	0	3
5	1	1	4	3	0

Figure 8. Traffic Matrix

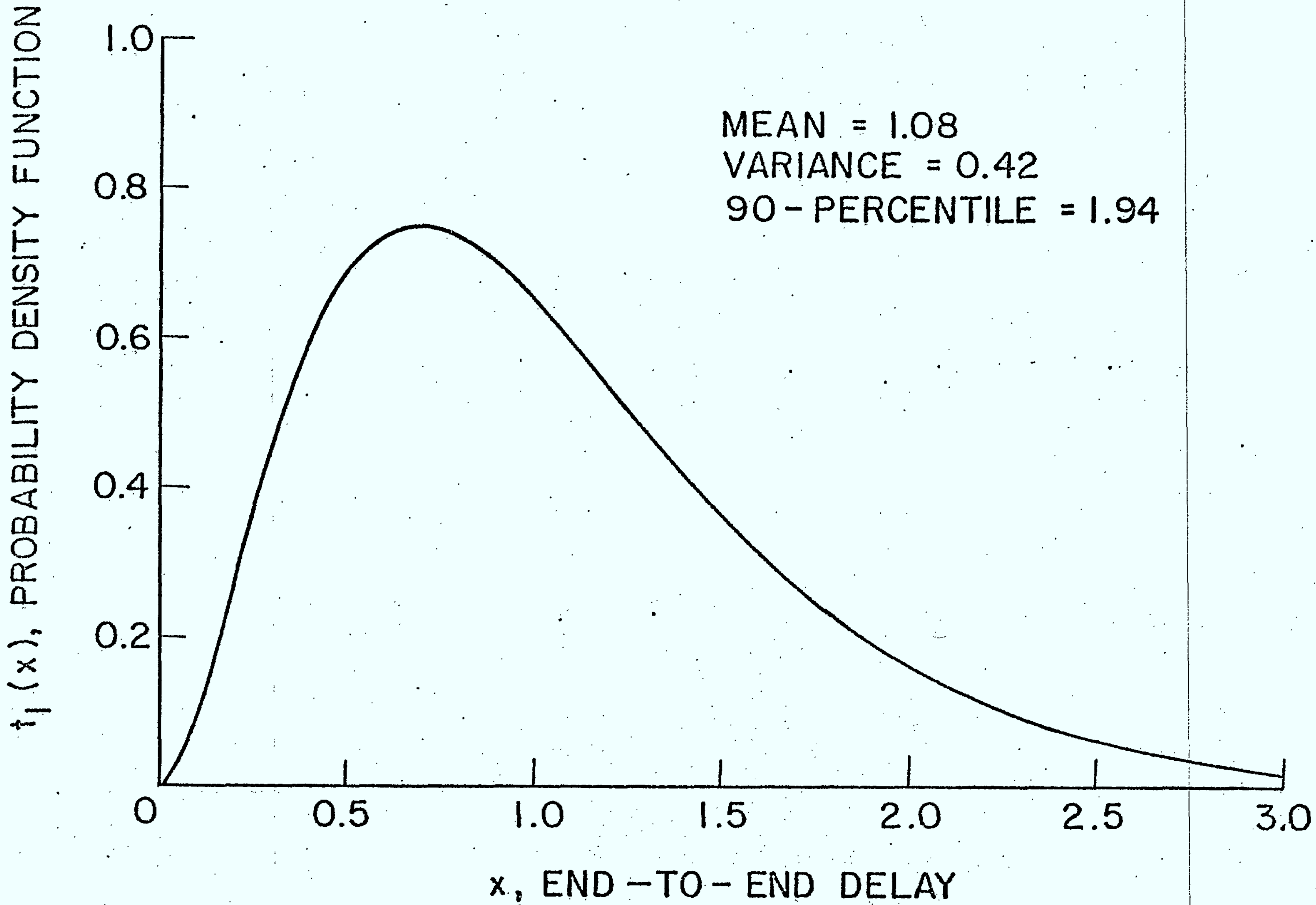


Figure 9. End-to-End Delay from Node 1 to Node 2 (Fixed Routing)

inverting the resulting $T_1^*(s)$, we get:

$$t_1(x) = -25.2\ell^{-3x} + 30\ell^{-2.5x} + 99\ell^{-4.5x} - 19.8\ell^{-5.5x} - 84\ell^{-4x} \quad (31)$$

A plot of this pdf, together with its mean, variance, and 90-percentile, are shown in Figure 10. A comparison between Figures 9 and 10 indicates that the mean class 1 delay under random routing is smaller. This is due to the fact that a fraction of traffic has been directed from a more heavily utilized channel (channel 5) to a couple of less heavily utilized channels (channels 3 and 8).

As a third example, we consider the end-to-end delay of all messages originated from node 1 under fixed, shortest-path routing. Applying our results for message groups (Eq.(27)), we get the plot shown in Figure 11.

Finally, in Figure 12, we show the pdf of the end-to-end delay over all messages under fixed, shortest-path routing.

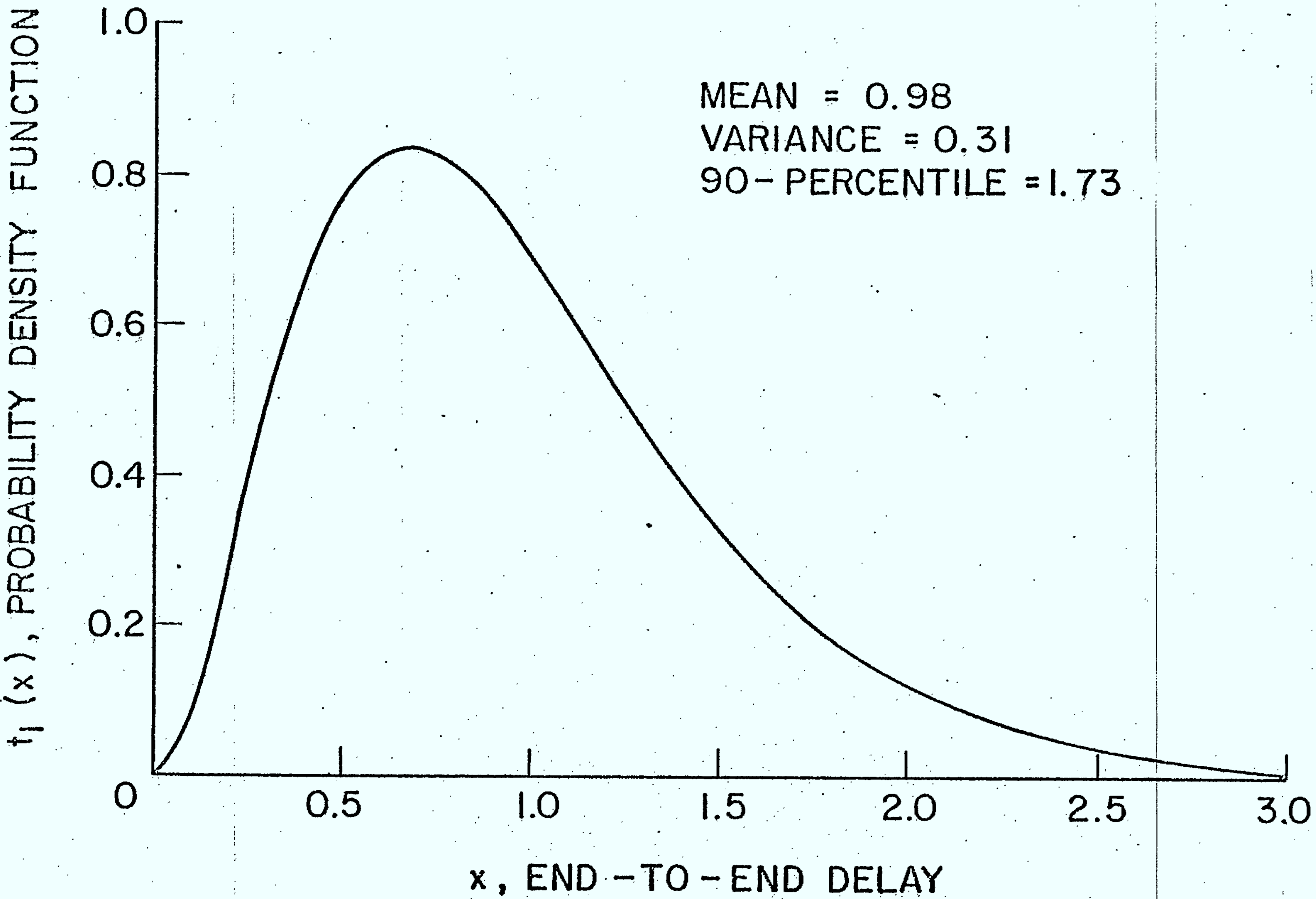


Figure 10. End-to-End Delay from Node 1 to Node 2 (Random Routing)

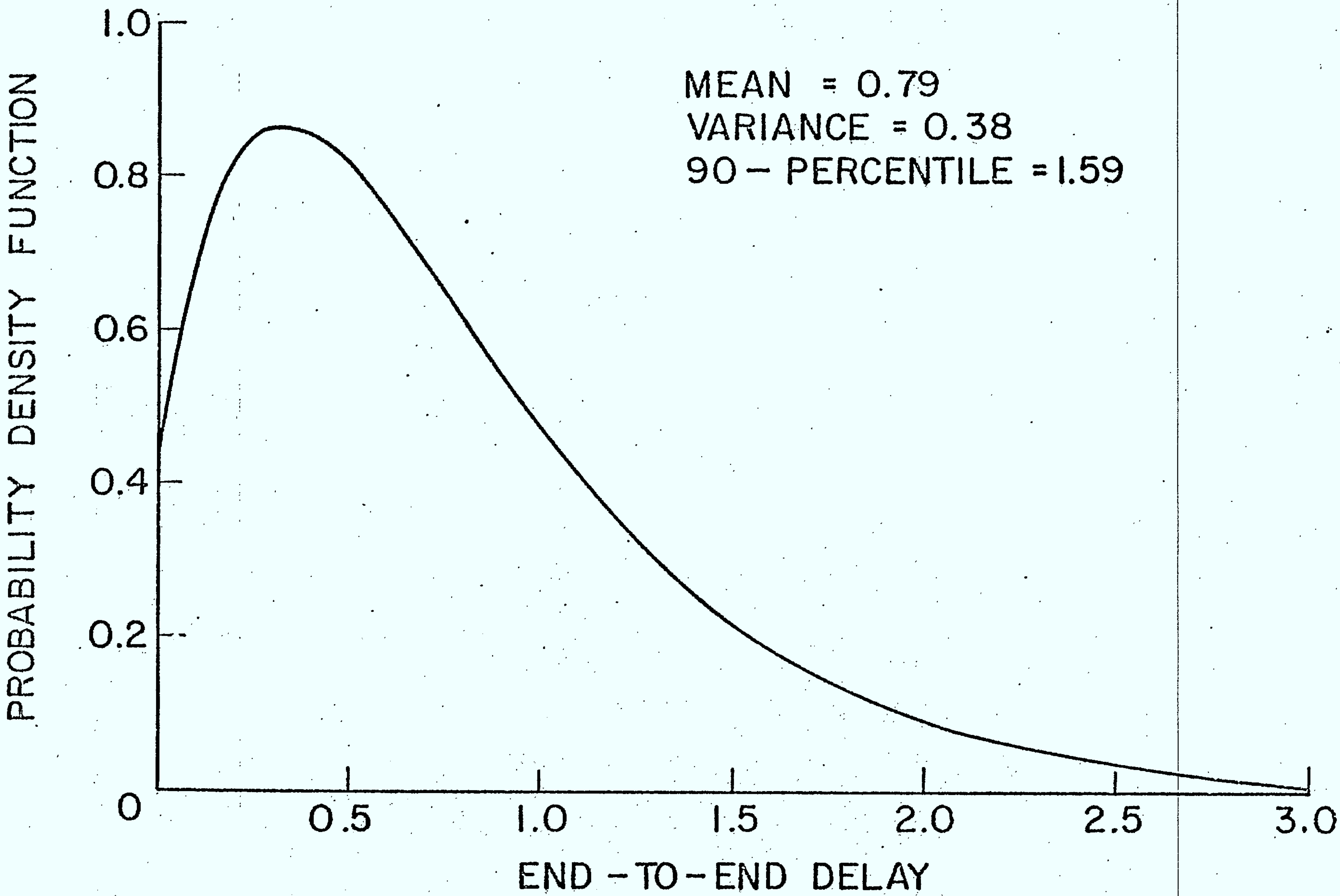


Figure 11. End-to-End Delay of Messages Originated from Node 1 (Fixed Routing)

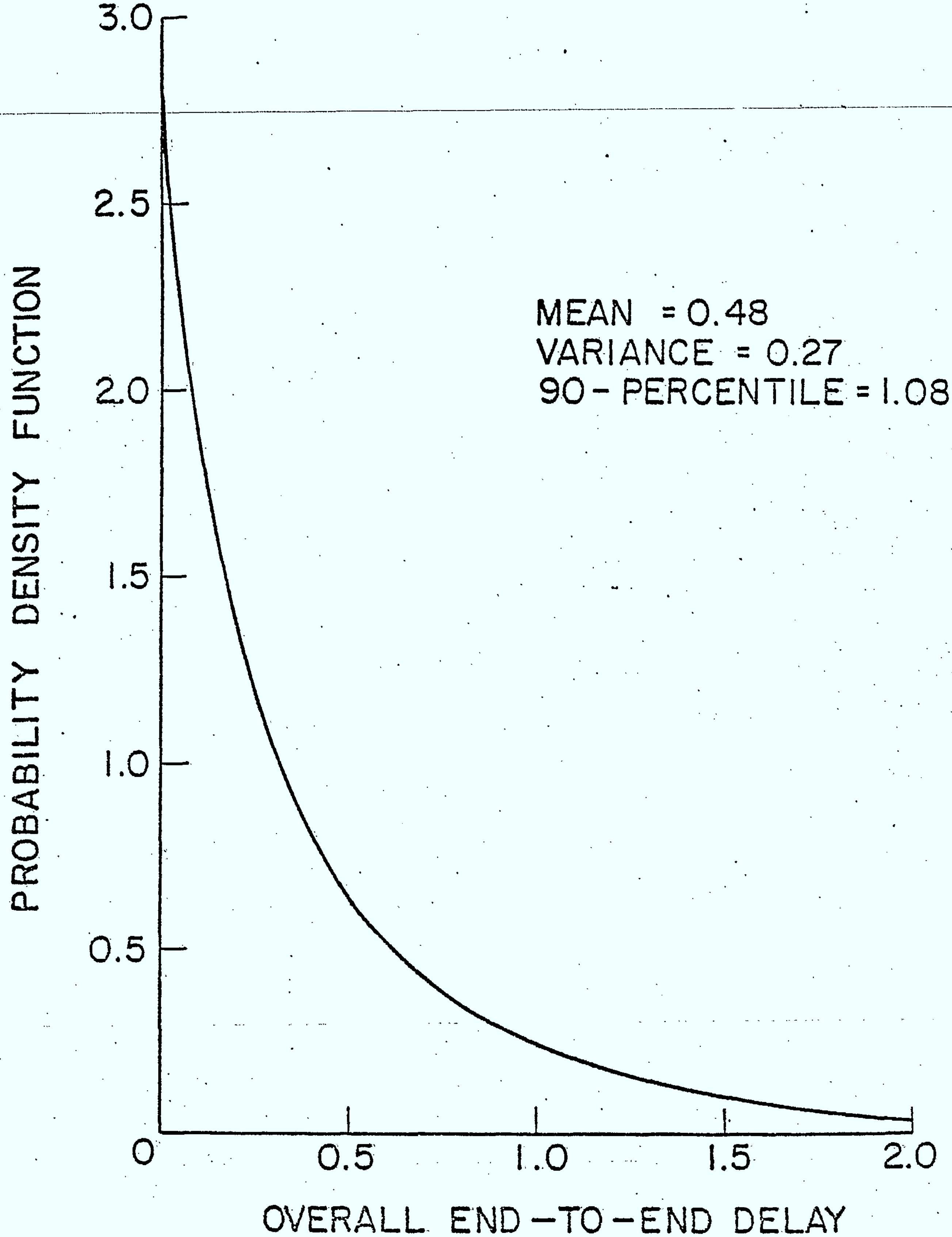


Figure 12. Overall End-to-End Delay (Fixed Routing)

2.2 HIERARCHICAL NETWORKS

At the present time, communication networks for computer communication are relatively small in size, e.g., less than 100 nodes. In the future, these networks are expected to grow, and the number of nodes will be on the order of hundreds, or even thousands. One technique to organize large networks is to use a multi-level hierarchy. An example of a two-level hierarchy has been shown in Figure 2. In this organization, nodes are divided into clusters, and clusters are connected together by "exchange" nodes. This section is a discussion of some preliminary result on the problems of routing and flow control in hierarichical networks.

2.2.1 ROUTING

In a hierarchical network, messages are distinguished according to whether they are intra-cluster messages or inter-cluster messages. With respect to routing algorithms, the approach of area routing [13] is most appropriate. This approach has the advantage of using a much smaller table for routing information, an extremely desirable feature when the network is large.

Due to the hierarchical structure, intra-cluster messages are seldom routed to an intermediate node in another cluster. Inter-cluster messages, on the other hand, must be routed through the exchange nodes. The applicability of queueing theory is still restricted to the case of fixed or random area routing. The case of adaptive routing does not yield to exact analysis.

2.2.2 FLOW CONTROL

The hierarchical organization introduces new issues in congestion and flow control. Messages sent between nodes in different clusters may be routed through a relatively large number of channels. Flow control schemes such as end-to-end control may not be adaquate because congestion can build up inside a cluster. This would result in a substantial increase in network delay. Such congestion is usually caused by one or more slow or heavily utilized channel in the cluster. In what follows, we will illustrate the potential problem of end-to-end control and show how this problem can be overcome by a hierarchical control scheme.

2.2.2.1 END-TO-END CONTROL

Our example of a hierarchical network is re-drawn in Figure 13 to show the path from node s in cluster 1 to node d in cluster 2. To study end-to-end control, we use Pennotti's approach [22] and model this path by the tandem queue model shown in Figure 14. The rest of the network is represented by the external messages.

As discussed in reference [22] and section 1.4, s-d messages are called link messages. End-to-end control is applicability of queueing theory is still restricted to the case of fixed or random area routing. The case of adaptive routing does not yield to exact analysis.

2.2.2 FLOW CONTROL

The hierarchical organization introduces new issues in congestion and flow control. Messages sent between nodes in different clusters may be routed through a relatively large number of channels. Flow control schemes such as end-to-end control may not be adaquate because congestion can build up inside a cluster. This would result in a substantial increase in network delay. Such congestion is usually caused by one or more slow or heavily utilized channel in the cluster. In what follows, we will illustrate the potential problem of end-to-end control and show how this problem can be overcome by a hierarchical control scheme.

2.2.2.1 END-TO-END CONTROL

Our example of a hierarchical network is re-drawn in Figure 13 to show the path from node s in cluster 1 to node d in cluster 2. To study end-to-end control, we use Pennotti's approach [22] and model this path by the tandem queue model shown in Figure 14. The rest of the network is represented by the external messages.

As discussed in reference [22] and section 1.4, s-d messages are called link messages. End-to-end control is

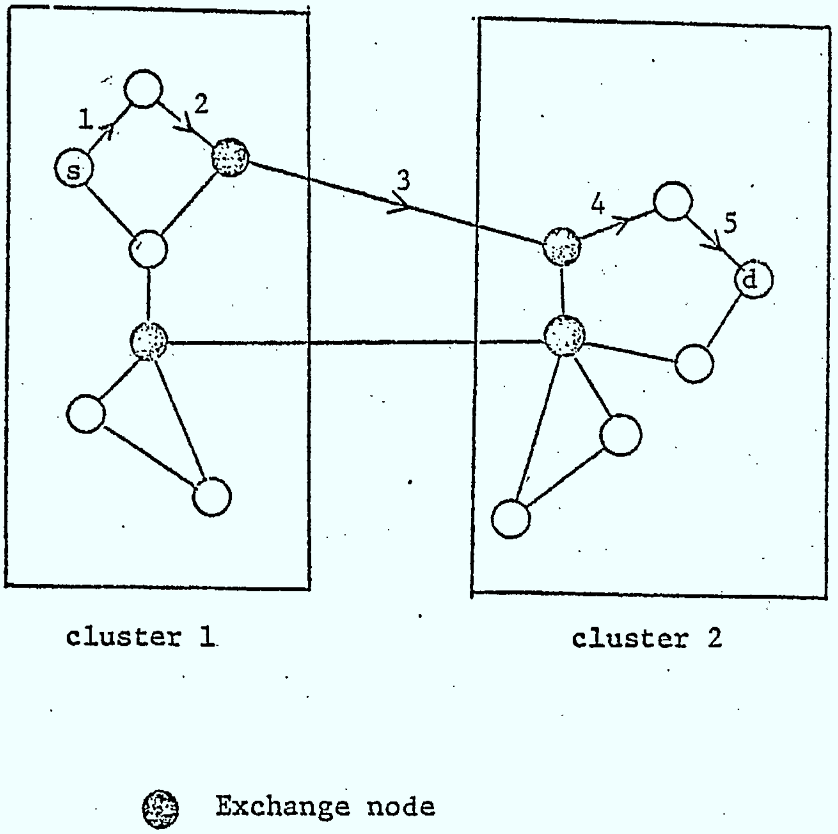


Figure 13. Hierarchical Network

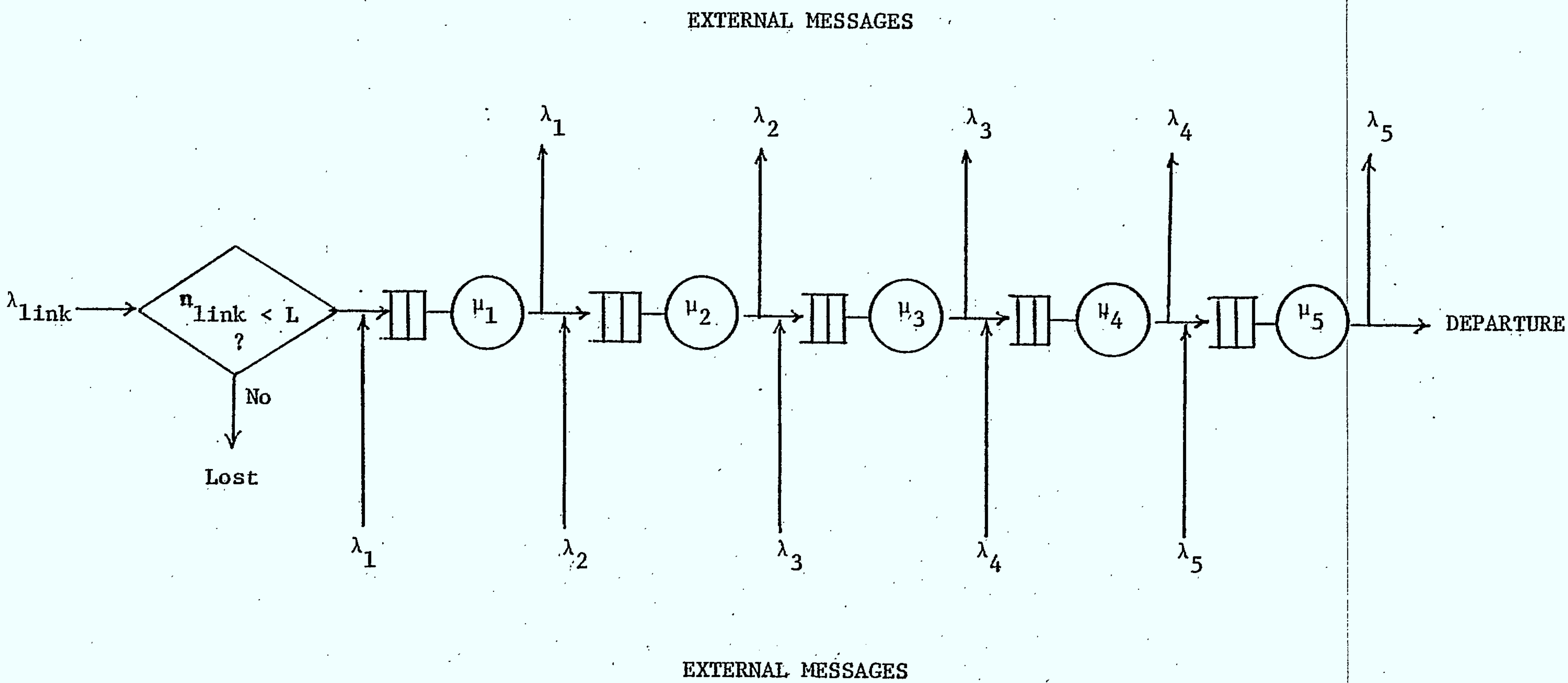


Figure 14. Tandem Queue Model for End-to-End Control

modelled by requiring that n_{link} be less than or equal to L. Let λ_{link} be the mean arrival rate of link messages. We study the effect of an increase in λ_{link} on the mean delay of external messages in cluster 2.

The queueing analysis can be found in reference [22]. It is summarized below.

Let $((n_{11},n_{12}),(n_{21},n_{22}),\ldots,(n_{51},n_{52}))$ be the state of the model where n_{i1} and n_{i2} are respectively the number of link and external messages at channel i. The equilibrium state probability is given by [22]:

$$P((n_{11}, n_{12}), (n_{21}, n_{22}), \ldots, (n_{51}, n_{52}))$$

$$= C \prod_{i=1}^{5} \frac{(n_{i1} + n_{i2})!}{n_{i1}!\, n_{i2}!} \rho_{i1}^{n_{i1}} \rho_{i2}^{n_{i2}} \qquad (32)$$

where $\rho_{i1} = \lambda_{link}/\mu_i$ and $\rho_{i2} = \lambda_i/\mu_i$.

C is the normalization constant obtained by summing all state probabilities and equating the sum to 1. Let $\hat{n}_1 = (n_{11},n_{21},\ldots,n_{51})$ and $n_1 = n_{11}+n_{21}+\ldots+n_{51}$. C is given by:

$$C = \left[\sum_{\substack{\text{all } \hat{n}_1 \\ \text{s.t. } n_1 \le L}} \sum_{n_{12}=0}^{\infty} \cdots \sum_{n_{52}=0}^{\infty} \prod_{i=1}^{5} \frac{(n_{i1} + n_{i2})!}{n_{i1}!\, n_{i2}!} \rho_{i1}^{n_{i1}} \rho_{i2}^{n_{i2}} \right]^{-1}$$

$$= \left[\sum_{\substack{\text{all } \hat{n}_1 \\ \text{s.t. } n_1 \le L}} \prod_{i=1}^{5} \rho_{i1}^{n_{i1}} \frac{1}{(1 - \rho_{i2})^{n_{i1}+1}} \right]^{-1} \qquad (33)$$

Let $Q(n_{11}, n_{21}, \ldots, n_{51})$ be the marginal probability that the number of link messages at channel i is n_{i1}, $i = 1, 2, \ldots, 5$. $Q(n_{11}, n_{21}, \ldots, n_{51})$ is given by:

$$Q(n_{11}, n_{21}, \ldots, n_{51}) = C \prod_{i=1}^{5} \rho_{i1}^{n_{i1}} \frac{1}{(1 - \rho_{i2})^{n_{i1}+1}} \tag{34}$$

It is of interest to note that this marginal probability is the same as the case of no external messages but the capacity of channel i is reduced by a fraction equal to ρ_{i2}.

From Eq.(34), we get the following expression for the mean number of link messages at channel i:

$$\bar{n}_{i1} = \sum_{k=0}^{L} k \sum_{\substack{\text{all } \hat{n}_1 \text{ s.t.} \\ n_1 \le L \ \& \ n_{i1}=k}} Q(n_{11}, n_{21}, \ldots, n_{51}) \tag{35}$$

It has been shown in [22] that $\bar{n}_{i2}$, the mean number of external messages at channel is related to $\bar{n}_{i1}$ by the following formula:

$$\bar{n}_{i2} = \frac{\rho_{i2}}{1 - \rho_{i2}} (1 + \bar{n}_{i1}) \tag{36}$$

Finally, we use Little's formula [30] and get the following expression for the mean delay of external messages at channel i:

$$\bar{T}_{i2} = \bar{n}_{i2} / \lambda_i = \frac{1/\mu_i}{1 - \rho_{i2}} (1 + \bar{n}_{i1}) \tag{37}$$

The mean delay over all external messages in cluster 2 is then given by:

$$\overline{T}_2 = \frac{\lambda_4 \overline{T}_{42} + \lambda_5 \overline{T}_{52}}{\lambda_4 + \lambda_5} \tag{38}$$

We now illustrate the performance of end-to-end control by a numerical example. The parameter values for this example are:

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 0.3$$

$$\mu_1 = \mu_2 = \mu_3 = 1.0$$

$$\mu_4 = 0.8 \qquad \mu_5 = 0.5$$

These values imply that the mean arrival rates of external messages to all channels are the same and the channels in cluster 1 are faster than those in cluster 2. Also, channel 5 is the slowest channel.

In Figure 15, we have plotted the mean delay of external messages in cluster 2 against λ_{link} for the case L = 8. We observe that as λ_{link} increases, the mean delay in cluster 2 approaches a constant value. To explain this phenomenon, we plot in Figure 16 the mean number of link messages at channels 4 and 5. These plots show that when λ_{link} is large, most of the 8 messages are in queue or in service at channel 5. The long queue in front of this channel will cause a substantial increase in the delay of external messages. We thus conclude that end-to-end control is not capable of preventing congestion from building up inside a cluster. Better flow control schemes are therefore needed.

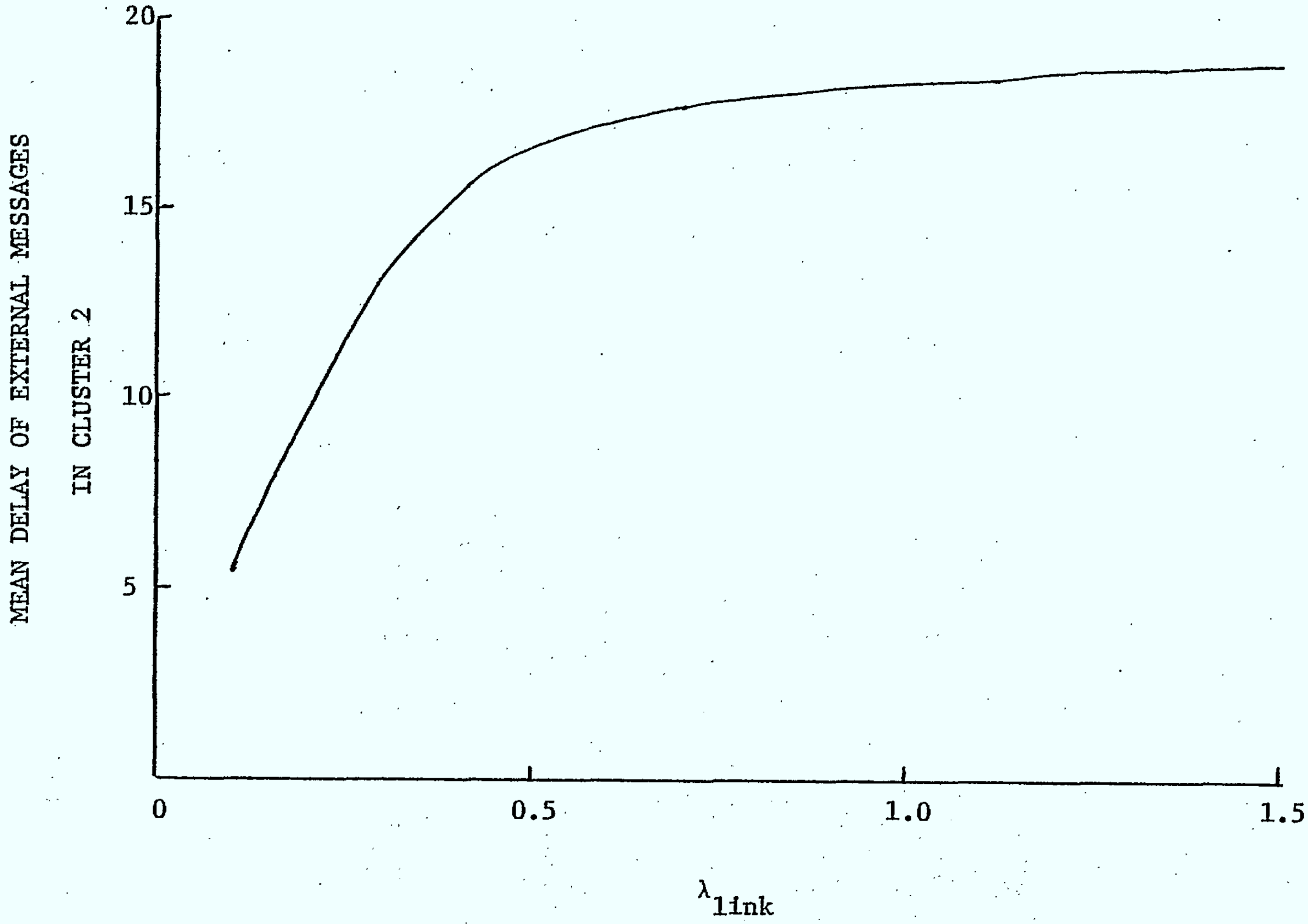


Figure 15. Mean Delay of External Messages in Cluster 2 vs λ_{link}

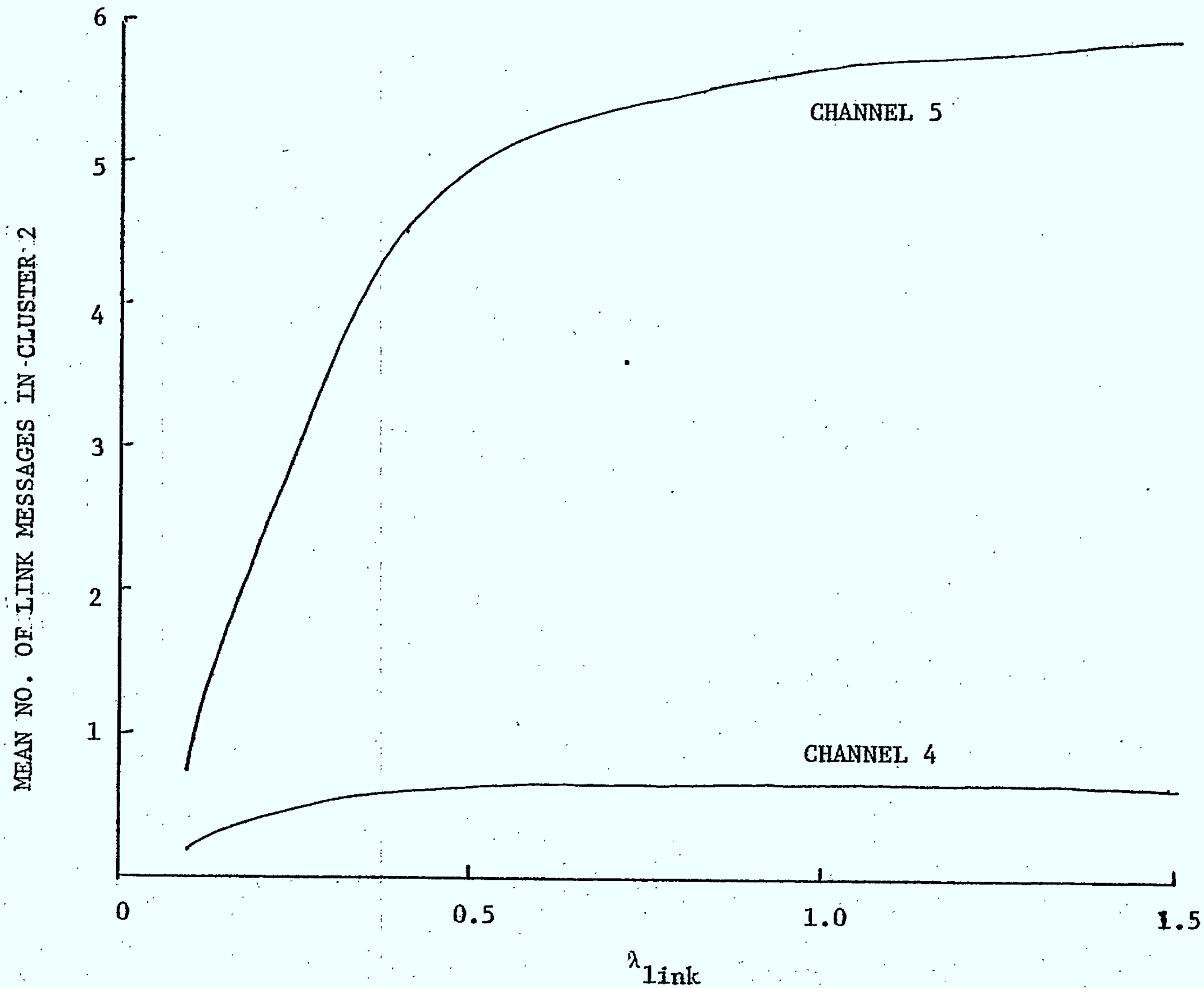


Figure 16. Mean Number of Link Messages in Cluster 2 vs λ_{link}

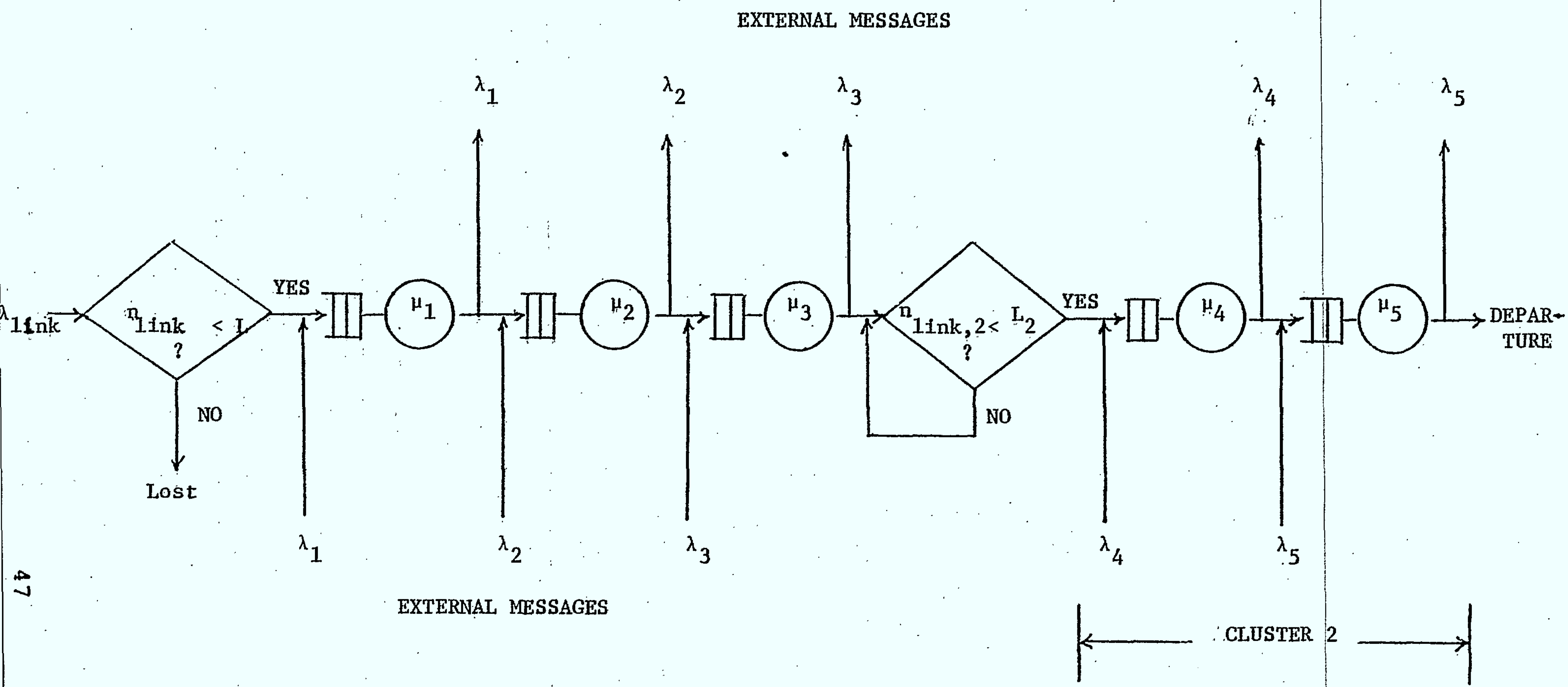
In the next section, we define a hierarchical control scheme and illustrate its advantage over end-to-end control by the same numerical example.

2.2.2.2 HIERARCHICAL CONTROL

The hierarchical control scheme under consideration can be divided into 3 levels: end-to-end control, cluster control, and inter-cluster control. This scheme allows each cluster to make its own flow control decisions. These decisions are coordinated by the inter-cluster control.

We will use the tandem queue model in Figure 17 to study the performance of a combined end-to-end and cluster control scheme. The limit L is imposed by end-to-end control. Cluster control is modelled by requiring the number of link messages in cluster i, i = 1,2, to be less than or equal to L_i.

With the addition of L_1 and L_2, the model does not yield to exact analysis. Instead of providing an approximate analysis to this model, we note that as λ_{link} increases, the number of link messages in cluster 2 will approach L_2. This is the case because channel 5 is the slowest channel. Assuming that there are always L_2 link messages in cluster 2, we can get an upper bound on the mean delay of external messages in this cluster. We thus reduce the model in Figure 17 to the cyclic queue model in Figure 18 and derive an expression for this upper bound.



$n_{link,\,2}$ = NUMBER OF LINK MESSAGES IN CLUSTER 2

Figure 17. Tandem Queue Model for Hierarchical Control

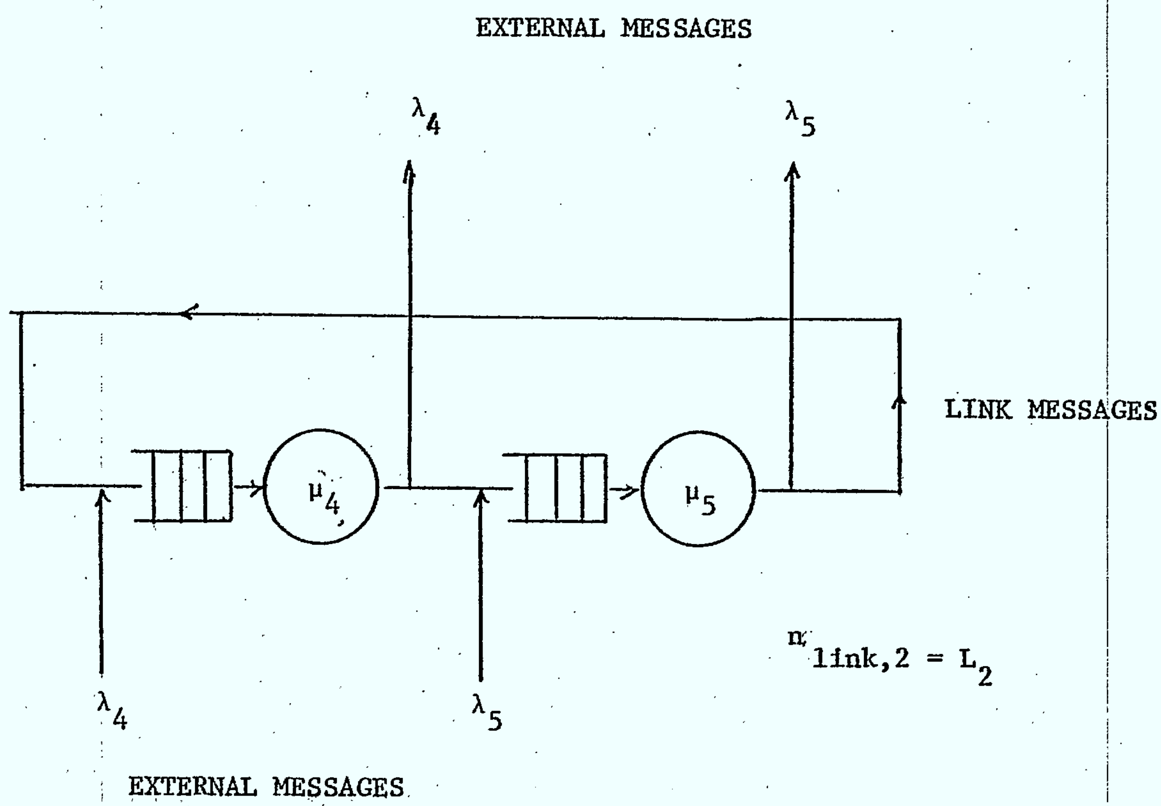


Figure 18. Cyclic Queue Model for Cluster 2 when λ_{link} is large

Let $((n_{41}, n_{42}), (n_{51}, n_{52}))$ be the state of this model. The equilibrium state probability is given by:

$$P((n_{41}, n_{42}), (n_{51}, n_{52})) = C_1 \prod_{i=4}^{5} \frac{(n_{i1} + n_{i2})\,!}{n_{i1}\,!\; n_{i2}\,!} \rho_{i1}^{\,n_{i1}} \rho_{i2}^{\,n_{i2}} \tag{39}$$

where

$$C_1 = \left[\sum_{n_{41}+n_{51}=L_2} \prod_{i=4}^{5} \rho_{i1}^{\,n_{i1}} \frac{1}{(1 - \rho_{i2})^{n_{i1}+1}} \right]^{-1} \tag{40}$$

Following the analysis in the last section, we get the following expressions for the performance measures of interest:

$$Q(n_{41}, n_{51}) = C_1 \prod_{i=4}^{5} \rho_{i1}^{\,n_{i1}} \frac{1}{(1 - \rho_{i2})^{n_{i1}+1}} \tag{41}$$

$$\bar{n}_{i1} = \sum_{k=0}^{L_2} k \sum_{\substack{n_{41}+n_{51}=L_2 \\ \& \; n_{i1}=k}} Q(n_{41}, n_{51}) \tag{42}$$

$$\bar{T}_{i2} = \frac{1 / \mu_i}{1 - \rho_{i2}} (1 + \bar{n}_{i1}) \qquad i = 4, 5 \tag{43}$$

$$T_2 = \frac{\lambda_4 \bar{T}_{42} \;\; \lambda_5 \bar{T}_{52}}{\lambda_4 + \lambda_5} \tag{44}$$

For our numerical example, the same parameter values as

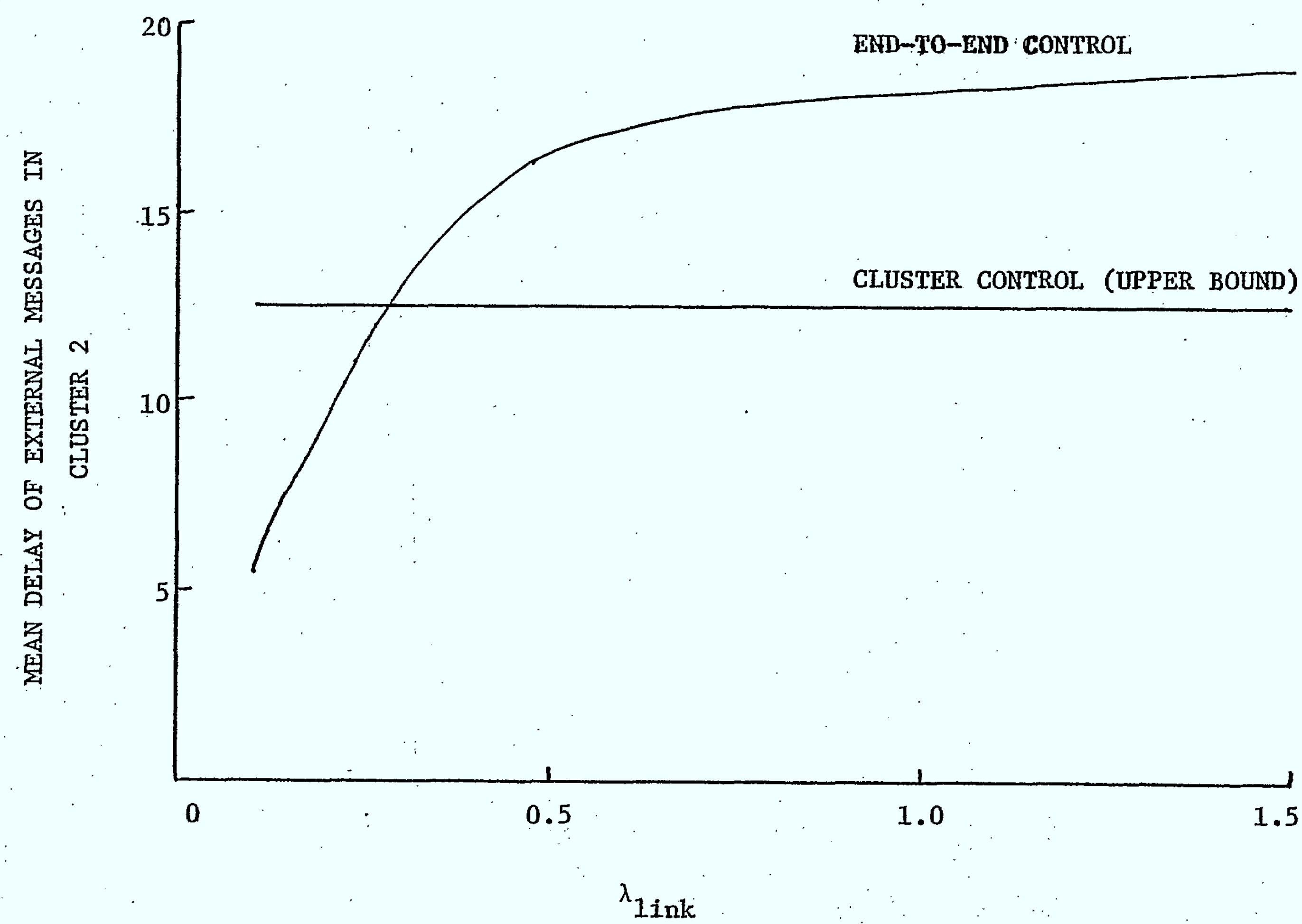


Figure 19. Mean Delay of External Messages in Cluster 2 vs λ_{link}

those for end-to-end control are used. The results for $L_1 = L_2 = 4$ are plotted in Figure 19. The corresponding result for end-to-end control (from Figure 15) is also shown. We observe that the delay in cluster 2 is much reduced. This reduction is due to the limits imposed by cluster control. We thus conclude that hierarchical control is more effective than end-to-end control alone.

We now present an approximate analysis to the model of hierarchical control shown in Figure 17. The basic technique is similar to that used by Irland [12] where a link message is assumed to be lost if $n_{link,2} = L_2$. Also, the arrival rate of link messages to cluster 2 is assumed to be Poisson. With these assumptions, the model for clusters 1 and 2 are shown in Figure 20.

Consider cluster 1 alone, the analysis for end-to-end control with limits $L - L_2$ can be used to obtain $P(n_{11}, n_{21}, n_{31})$, the equilibrium probability that the number of link messages at channel i (i = 1,2,3) is n_{i1}. From this analysis, we get the following expression for λ'_{link}, the mean arrival rate of link messages to cluster 2:

$$\lambda'_{link} = \lambda_{link} \sum_{n_{11}+n_{21}+n_{31} \leq L} P(n_{11}, n_{21}, n_{31}) \qquad (45)$$

Cluster 2 is now treated as a tandem queue model. The arrival process is Poisson with mean rate λ'_{link}. We can easily determine the mean number of link messages at each

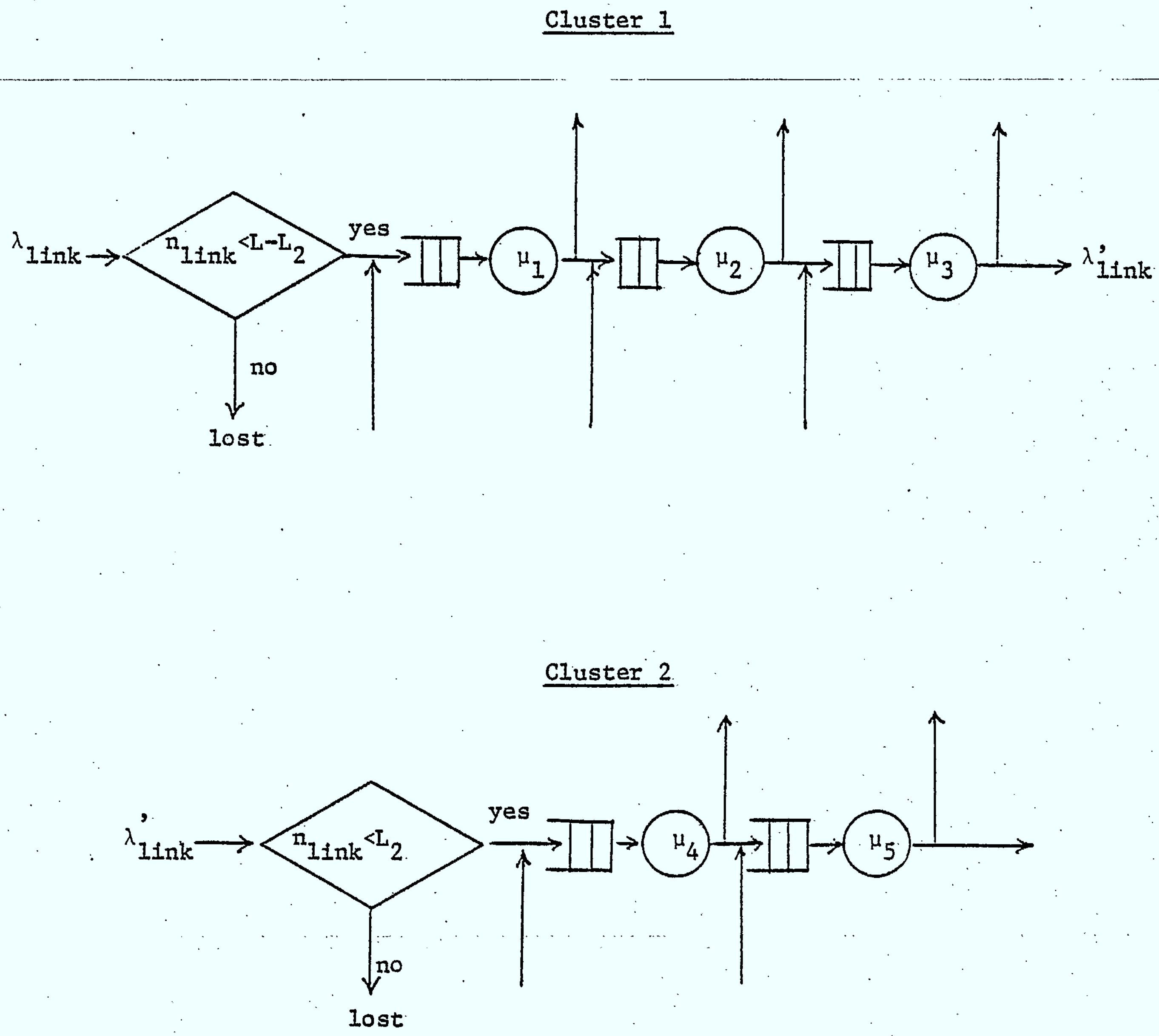


Figure 20. Approximate Models for Hierarchical Control

channel in cluster 2. Let them be $\overline{n}_{41}$ and $\overline{n}_{51}$ respectively. The mean delay in cluster 2 is then given by Eq.(44), i.e.,

$$\overline{T}_2 = \frac{\lambda_4 \overline{T}_{42} + \lambda_5 \overline{T}_{52}}{\lambda_4 + \lambda_5} \tag{46}$$

where

$$T_{i2} = \frac{1/\mu_i}{1-\rho_{i2}} (1 + \overline{n}_{i1}) \qquad i=4,5$$

A plot of the approximate expression for T_2 is shown in Figure 21.

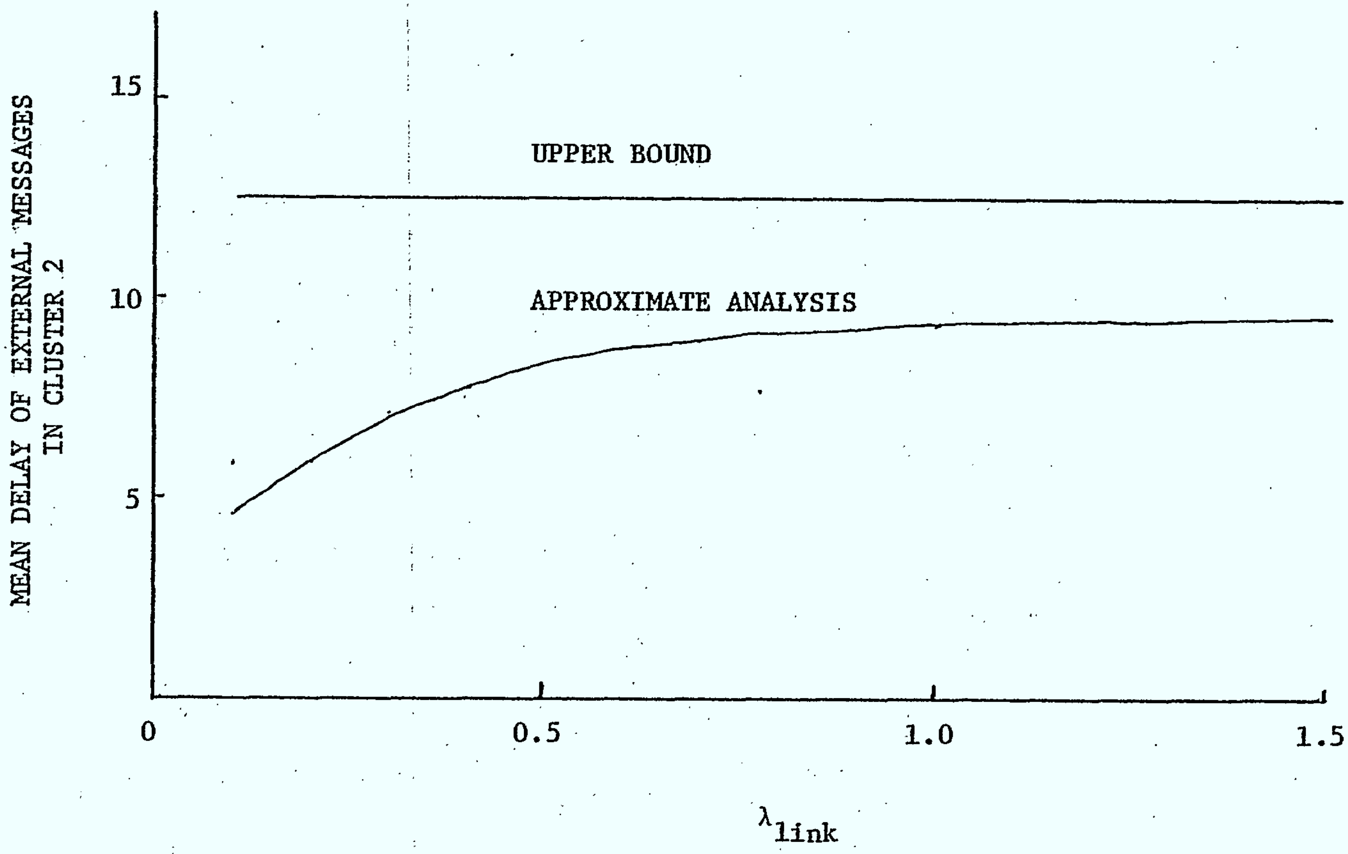


Figure 21. Mean Delay of External Messages in Cluster 2 vs. λ_{link}

PART 3 -- USER-RESOURCE NETWORKS

A user-resource network, as shown in Figure 22, is a collection of user terminals interacting with remote computers via a communication network. To synchronize the action of the user and the remote computer, a set of protocols is defined. To comply with these protocols, software modules must be executed. This introduces overhead in the computer system, and as a result, the response time to local and remote users is degraded. In this section, a queueing network model is developed to study the effect of protocol overhead on mean response time. This model is based on the ARPA network [1] where remote terminals attached to a TIP [31] are entering requests to a host computer.

3.1 PROTOCOLS IN THE ARPA NETWORK

In the ARPA network, a host-to-host protocol [32] is defined for the host computers to communicate with each other. A TIP is an interface unit with this protocol implemented. It acts as a data concentrator for local terminals to interact with remote host computers.

Messages in the ARPA network are transmited over unidirectional logical paths, known as links. A flow control mechanism is implemented in this path where the next message cannot be sent until the RFNM of the last message is received.

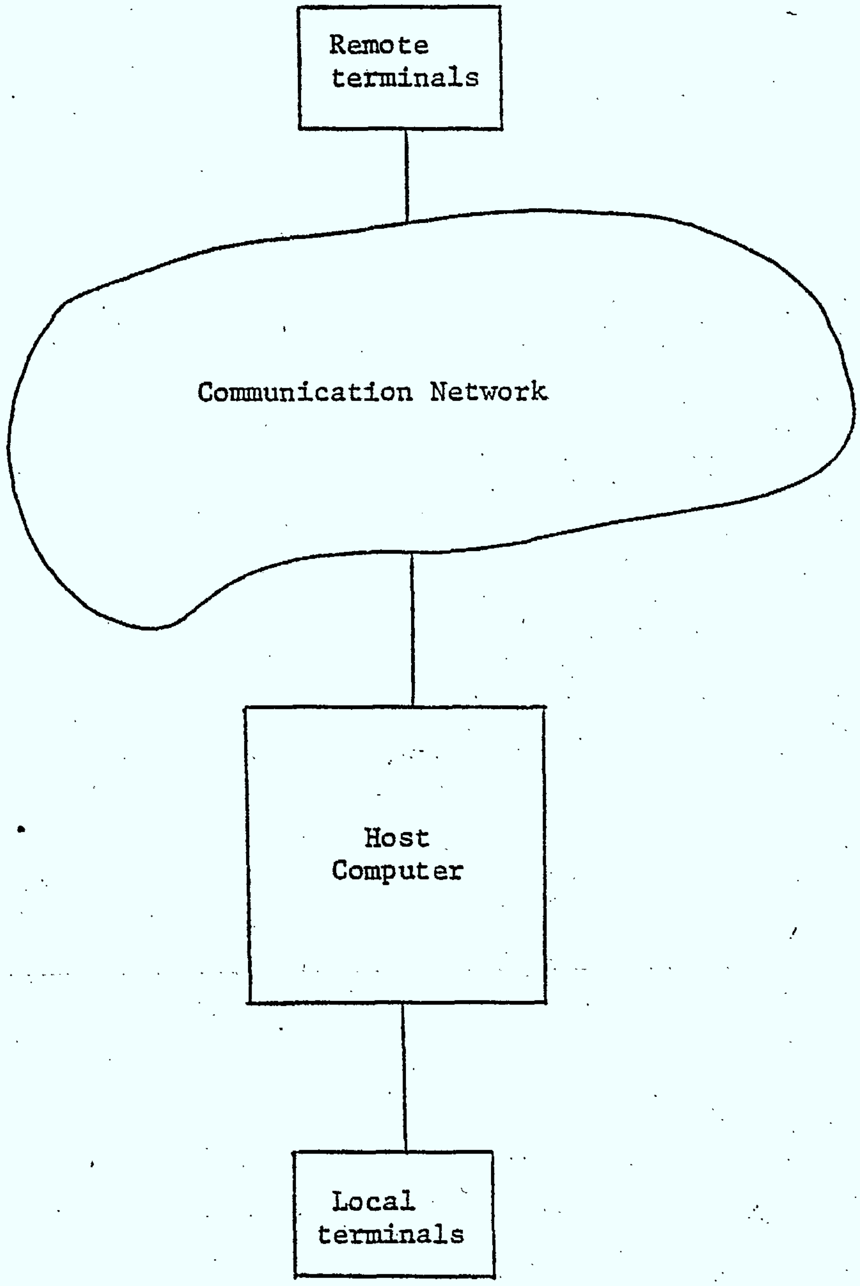


Figure 22 User-Resource Network

Due to the multiprogramming environment of most host computers in the ARPA network, it is conceivable that remote users at different nodes may want to use the same host computer. The notion of a connection is introduced to provide for process addressing within the host. (A process is a program in execution.) A connection is an extension of link to the process level. It is a simplex connection between a sender and a receiver. Two such connections (one in each direction) are required for two processes in different machines to communicate.

Connections are established, monitored, and destroyed by a Network Control Program (NCP). An excellent survey of NCP's at the different nodes in the ARPA network can be found in [33]. A control link (link number 1) is reserved for NCP communications. It is used for connection establishment, flow control, and connection termination.

To facilitate the establishment of connections between processes in different machines, an Initial Connection Protocol (ICP) [34] is defined. This protocol, together with the Telnet protocol [35] allows a terminal user to interact with a remote computer. The Telnet protocol is responsible for going through the ICP sequence, sending and receiving of data, interrupting the remote host computer, and terminating a connection. A complete description of the various types of protocols can be found in [36].

After the initial connection sequence, the flow of messages between sender and receiver is controlled by the "allocate" mechanism in the host-to-host protocol. Under this mechanism, the receiver NCP notifies the sender NCP as to the number of bytes he can send on a particular connection. This notification is done by an "allocate" message over the control link. The receiver NCP is therefore able to regulate the rate of receiving data. This often lowers the rate at which a terminal user can enter requests to a remote computer. The "allocate" message also introduces more traffic in the network, and as a result, the response time to remote users is increased. In what follows, we will develop a queueing network model to study the response time of local and remote users. A parameter to represent overhead due to the host-to-host protocol is also identified.

3.2 QUEUEING NETWORK MODEL

The basic model is shown in Figure 23. The communication network is modelled by two tandem queues, one for each direction. The rest of the communication network is once again, modelled by the external messages. Attached to these tandem queues are remote and local terminals, and a computing resource. The model assumptions and notation are given below:

(a) For the local users, the mean think time at terminal is $1/h_1$, and the mean service time at CPU is $1/\mu_{c1}$.

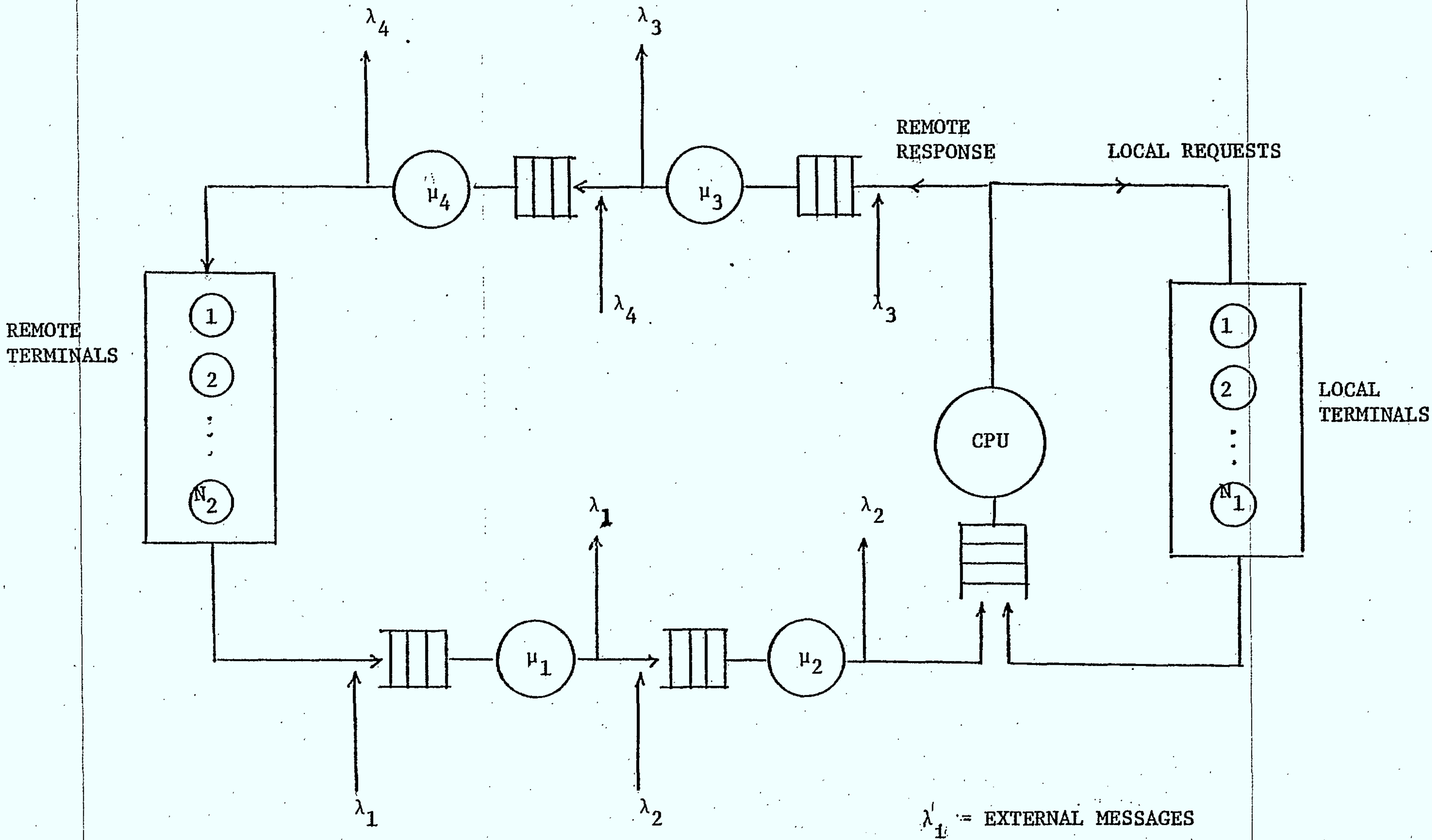


Figure 23 Queue Network Model for User-Resource Network

(b) The corresponding means for the remote users are $1/h_2$ and $1/\mu_{c2}$ respectively.

(c) $1/\mu_{c2} = (1 + \alpha)/\mu_{c1}$. α is a parameter related to protocol overhead, it is the ratio of CPU time for protocol handling to that for actual processing.

(d) The external messages are removed from the model by a proper reduction to the capacity of each channel.

(e) The data transfer time at channel i, $i = 1,2,3,4$, is exponential with mean $1/\mu_i$.

(e) The scheduling discipline at the CPU is processor sharing [5].

A state of this model is given by $((n_{t1},n_{t2}),(n_{c1},n_{c2}),(n_1,n_2,n_3,n_4))$ where

n_{t1} and n_{t2} are respectively the mean number of local and remote users thinking at terminals,

n_i, $i = 1,2,3,4$, is the number of messages to/from remote terminals at channel i, and

n_{c1} and n_{c2} are respectively the mean number of local and remote requests at the CPU.

With N_1 local and N_2 remote users, a feasible state is characterized by:

$$n_{t1} + n_{c1} = N_1$$

$$n_{t2} + n_{c2} + n_1 + n_2 + n_3 + n_4 = N_2$$

3.3 QUEUEING ANALYSIS

Our model for user-resource networks belongs to the class of network models analysed by Baskett,et.al [5]. Using Baskett's result, the equilibrium state probability is given by:

$$P((n_{t1}, n_{t2}), (n_{c1}, n_{c2}), (n_1, n_2, n_3, n_4)) = C_2 \frac{1}{n_{t1}!\, n_{t2}!} \left(\frac{1}{h_1}\right)^{n_{t1}} \left(\frac{1}{h_2}\right)^{n_{t2}}$$

$$\cdot \frac{(n_{c1} + n_{c2})!}{n_{c1}!\, n_{c2}!} \left(\frac{1}{\mu_{c1}}\right)^{n_{c1}} \left(\frac{1}{\mu_{c2}}\right)^{n_{c2}} \prod_{i=1}^{4} \left(\frac{1}{\mu_i}\right)^{n_i} \qquad (47)$$

C_2 is the normalization obtained by summing all state probabilities and equating the sum to 1.

From Eq.(47), we can use the following formulas to compute $\bar{n}_{t1}$ and $\bar{n}_{t2}$, the mean number of local and remote users thinking at terminals:

$$\bar{n}_{t1} = \sum_{k=0}^{N_1} k \sum_{\substack{\text{all feasible states} \\ \text{s.t. } n_{t1}=k}} P((n_{t1}, n_{t2}),(n_{c1}, n_{c2}),(n_1, n_2, n_3, n_4)) \qquad (48)$$

$$\bar{n}_{t2} = \sum_{k=0}^{N_2} k \sum_{\substack{\text{all feasible states} \\ \text{s.t. } n_{t2}=k}} P((n_{t1}, n_{t2}),(n_{c1}, n_{c2}),(n_1, n_2, n_3, n_4)) \qquad (49)$$

For the local users, the mean number of requests at the CPU is $N - \bar{n}_{t1}$, and the mean rate of requests entering the CPU is $\bar{n}_{t1}h_1$. Using Little's result [30], the mean response time of local users is given by:

$$R_1 = \frac{N_1 - \bar{n}_{t1}}{\bar{n}_{t1}\, h_1} \tag{50}$$

Similarly, for the remote users, the mean response time is:

$$R_2 = \frac{N_2 - \bar{n}_{t2}}{\bar{n}_{t2}\, h_2} \tag{51}$$

3.4 NUMERICAL EXAMPLES

In our numerical examples, the model parameters are as follows:

$1/h_1 = 1/h_2 = 10.0$

$1/\mu_{c1} = 1.0$

$1/\mu_1 = 1/\mu_2 = 1/\mu_3 = 1/\mu_4 = 0.25$

$\alpha = 0.5$, 1.0, and 2.0

The mean response time of local users for the case of $\alpha = 1.0$ is shown in Figure 24. We observe that as N_1 increases, this response time increases slowly at first, and then becomes a linear function of N_1. This behaviour is typical of queueing network models with a finite number of user terminals [37]. Figure 24 also shows the effect of the presence of the remote users on the response time of local users.

The plots for $\alpha = 0.5$ and 2.0 are not shown because they are almost identical to those in Figure 24. This indicates that under processor sharing, the amount of protocol

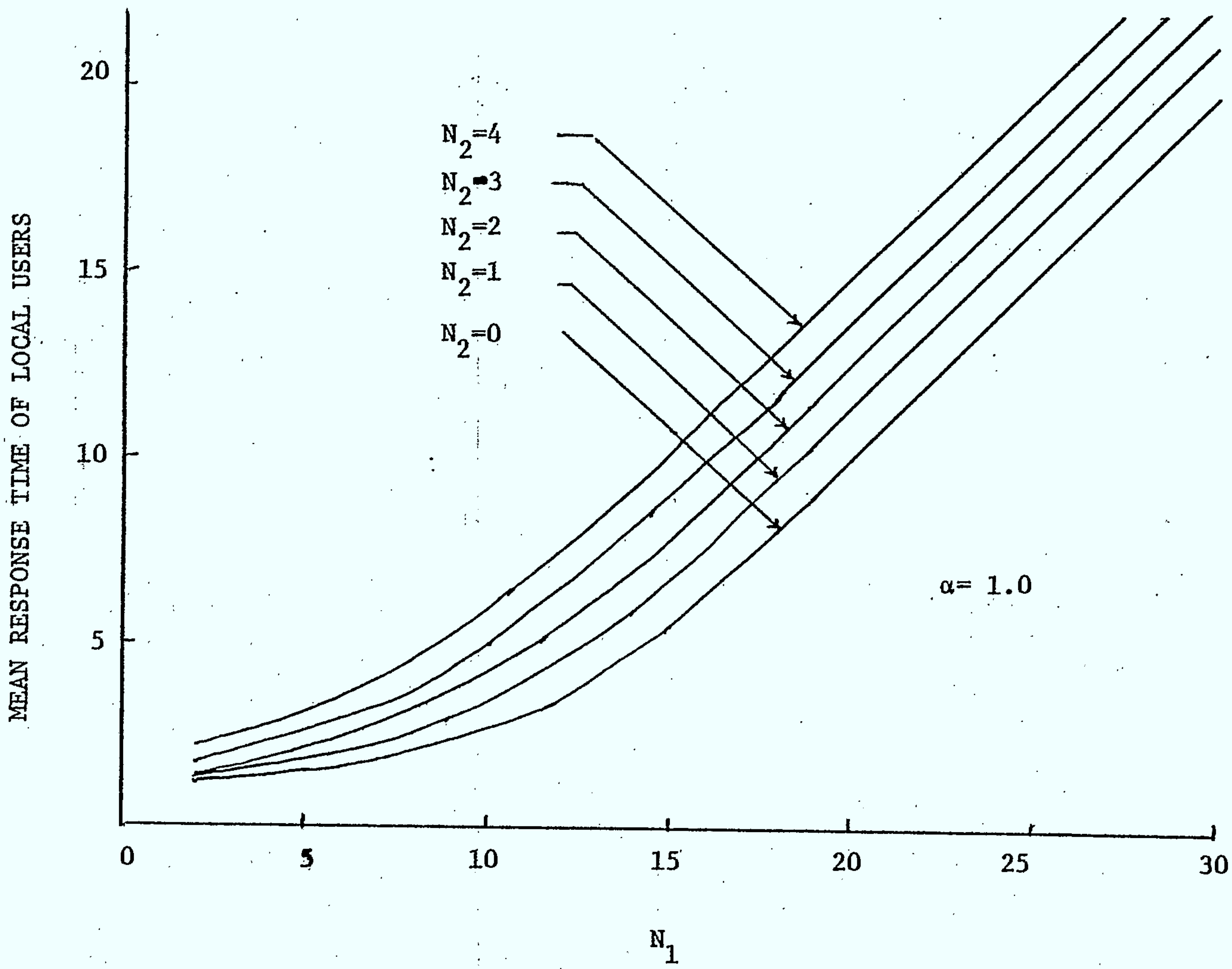


Figure 24 Mean Response Time of Local Users vs N_1

overhead has little effect on the response time of local users. Its effect on the response time of remote users, on the other hand, is more pronounced. This can be seen in Figures 25 to 27 where the remote response time is plotted against N_1 for the three values of α.

The above examples are for the processor sharing scheduling disciplines only. The results will be quite different if a different discipline is used. Also, the modelling of the computer system can be refined by the addition of secondary storage devices and main memory limitations.

3.5 APPROXIMATE ANALYSIS

We now include the "allocate" messages into our model and illustrate how the theorem on end-to-end delay in section 2.1 can be used to obtain approximate expressions for the local and remote response times. We assume that each time a data message is received and processed by the NCP (in the TIP at node 1 or the computer system at node 2), an allocate message is returned to the sending NCP to set up the transfer of the next data message.

For our approximate analysis, we let R_2 be the mean response time of remote users. The mean arrival rate of messages to the communication network is given by:

$$\lambda_2 = \frac{2N}{1/h_2 + R_2} \qquad (52)$$

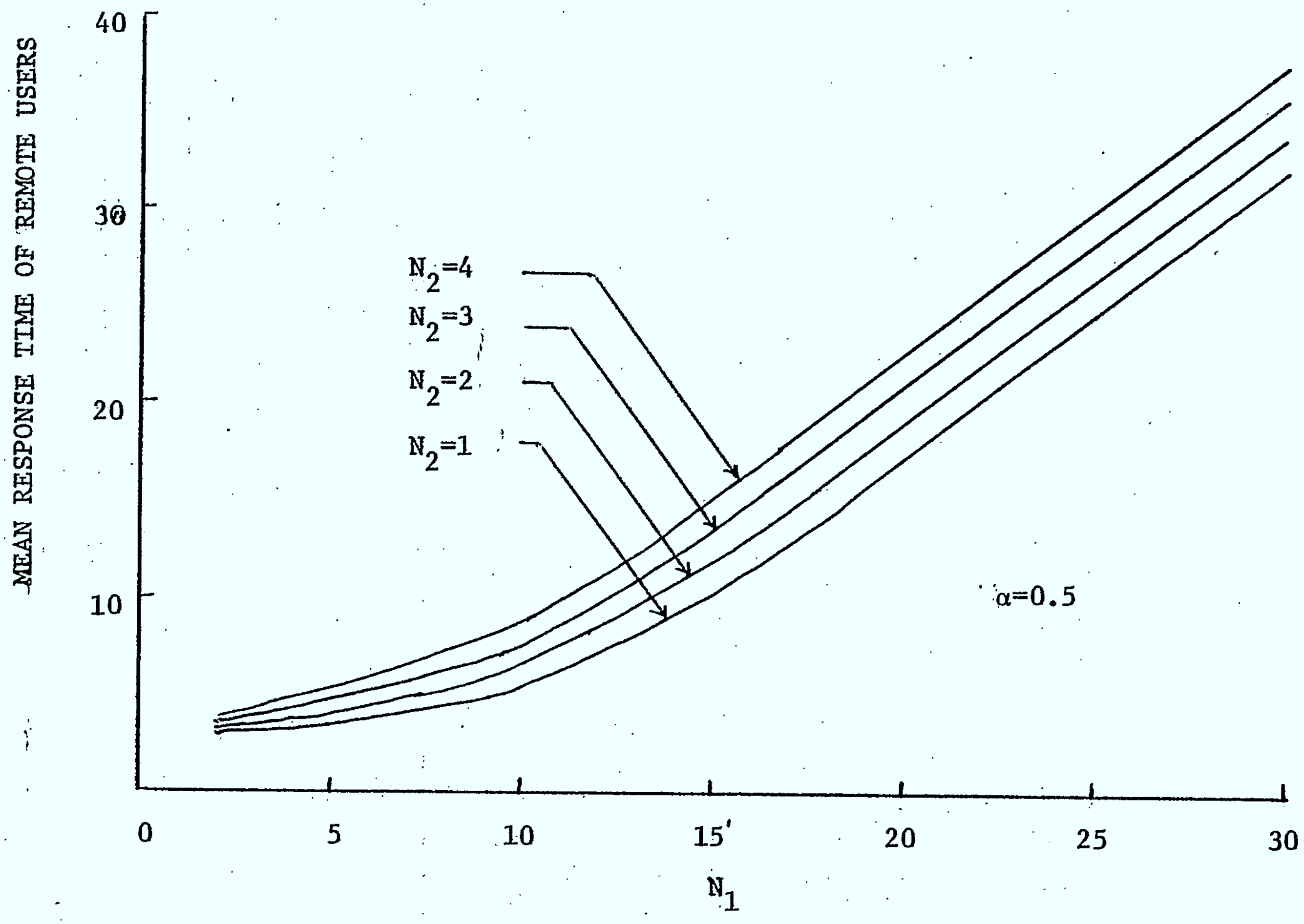


Figure 25 Mean Response Time of Remote Users vs N_1

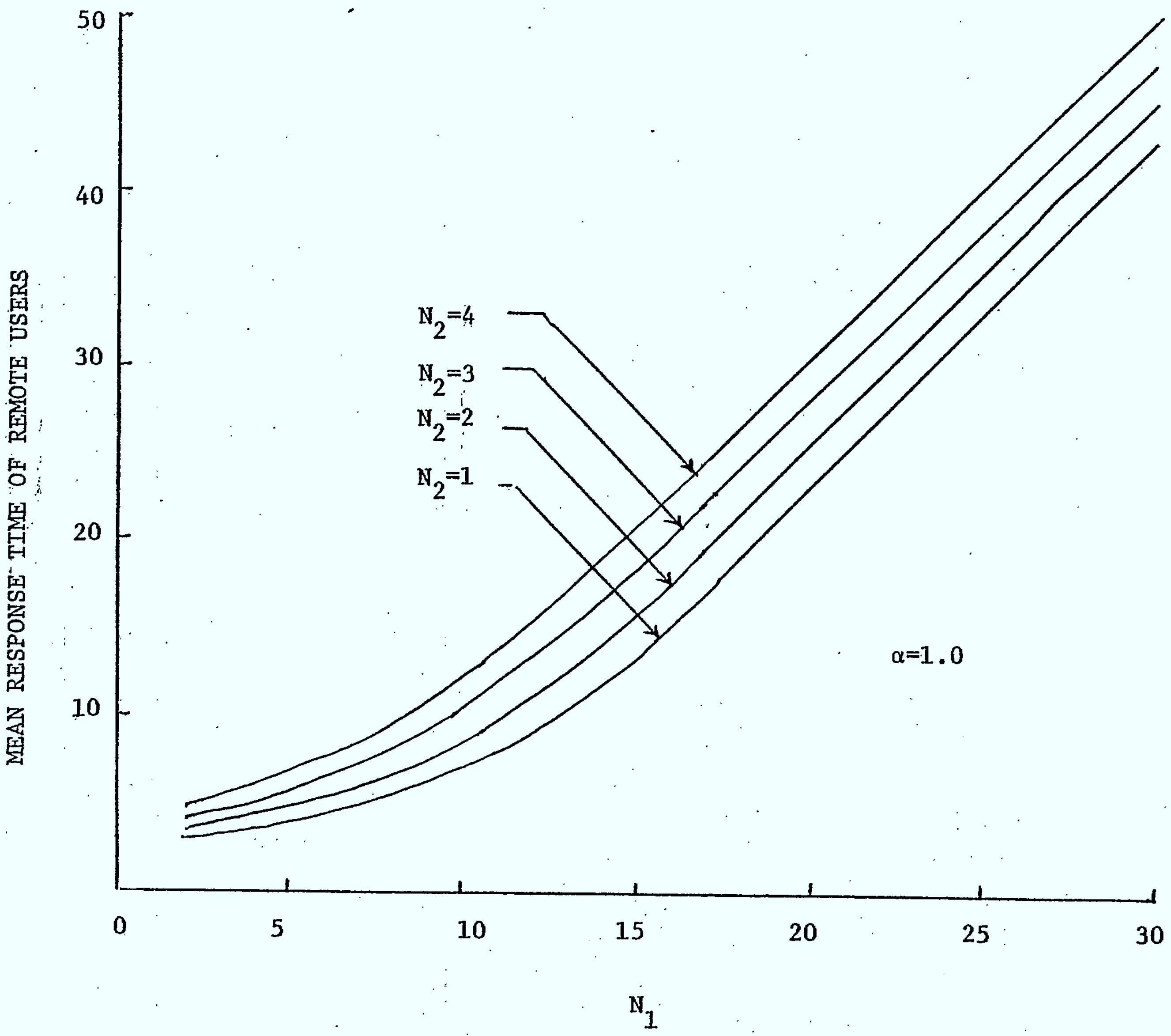


Figure 26 Mean Response Time of Remote Users vs N_1

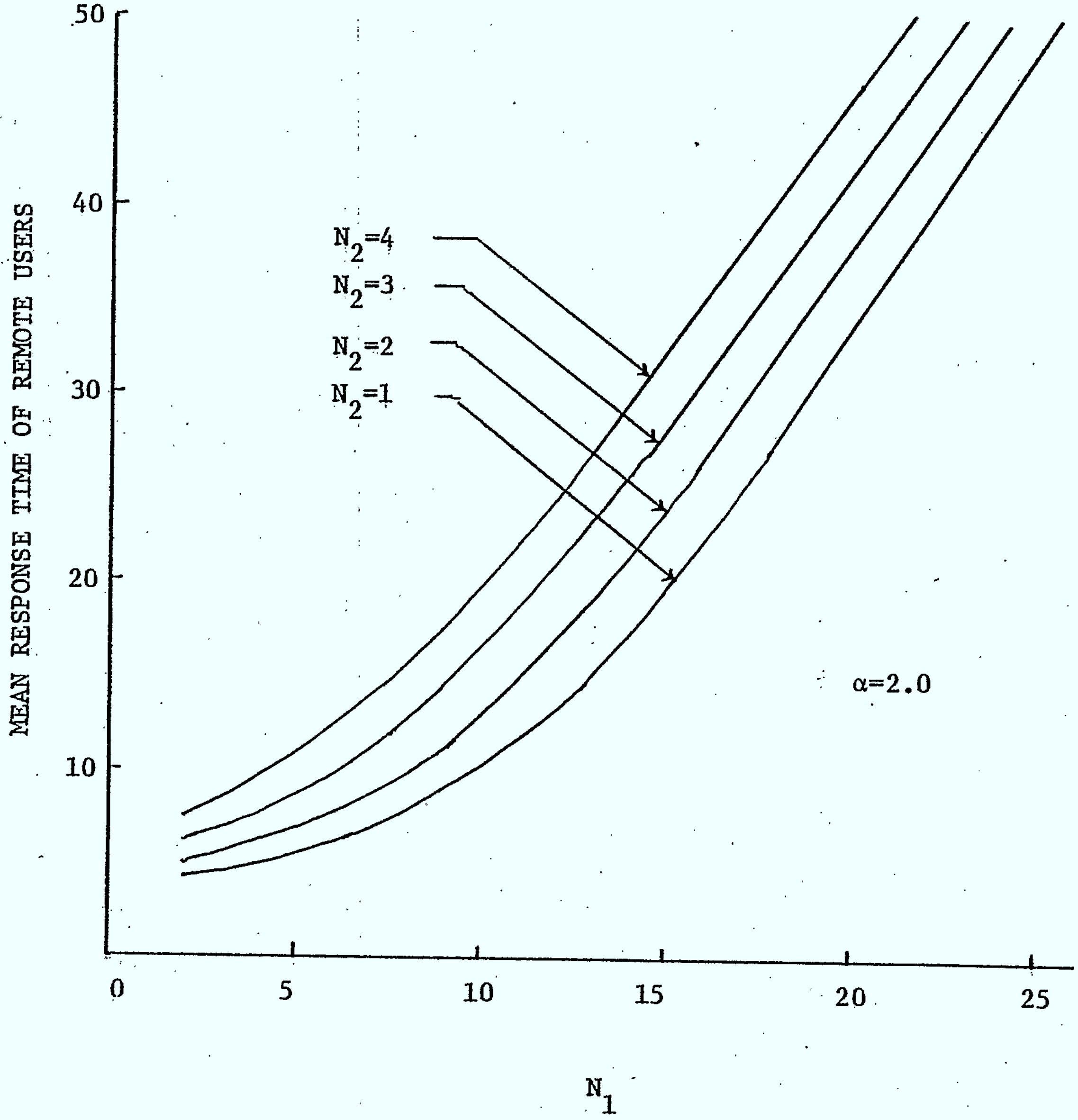


Figure 27 Mean Response Time of Remote Users vs N_1

June 1971.

35. Postel, J. Official Telnet-Logger Initial Connection Protocol, Network Information Center 7103, Stanford Research Institute, June 1971.

36. Crocker, S., Heafner, J., MetCalfe, R., and Postel, J. Function-oriented Protocols for the ARPA Computer Network, AFIP Conference Proceedings 40 (1972) 271-279.

37. Muntz, R. and Wong, J. Asymptotic Properties of Closed Queueing Network Models, Proc. 7th Princeton Conference on Information Sciences and Systems, (1974) 348-352.

38. Rybczynski, A., Wessler, B., Despres, R., and Wedlake, J. A New Communication Protocol for Accessing Data Networks: The International Packet Mode Interface. Proc. ICC (1976), 20.7-20.11.

39. Kermani, P. Switching and Flow Control Techniques in Computer Communication Networks. Ph.D. Thesis, University of California at Los Angeles, 1977.

40. Fayolle, G., Gelenbe, E., and Pujolle, G. An Analytic Evaluation of the Performance of the "Send and Wait" Protocol. IEEE Trans. on Commun. COM-26, 3 (Mar 1978), 313-319.

41. Gallager, R. Basic Limits on Protocol Information in Data Communication Networks. IEEE Trans. on Information Theory, IT-22, 4 (July 1976), 385-398.

The factor of 2 is due to the fact that each data message is followed by an allocate message. This is also the mean rate of messages returning to the remote terminals. Applying our theorem on end-to-end delay in section 2.1, the total mean delay in the communication network is given by:

$$\overline{T} = \sum_{i=1}^{4} \frac{1/\mu_i}{1 - \lambda_2/\mu_i} \qquad (53)$$

With respect to the remote users, the effective mean think time is:

$$1/h_2' = 1/h_2 + \overline{T} \qquad (54)$$

The model for the computer system at node 2 is thus reduced to a finite population model with two classes of customers and processor sharing discipline. The mean time spent by remote jobs in the system can easily be determined. Let this time be R_2'. We then have:

$$R_2 = R_2' + \overline{T} \qquad (55)$$

This equation can be solved iteratively for R_2.

In Figures 28 and 29, we have plotted the mean response time to local and remote users for the case $\alpha = 1$. We see similar behaviour as that shown in Figures 24 to 27.

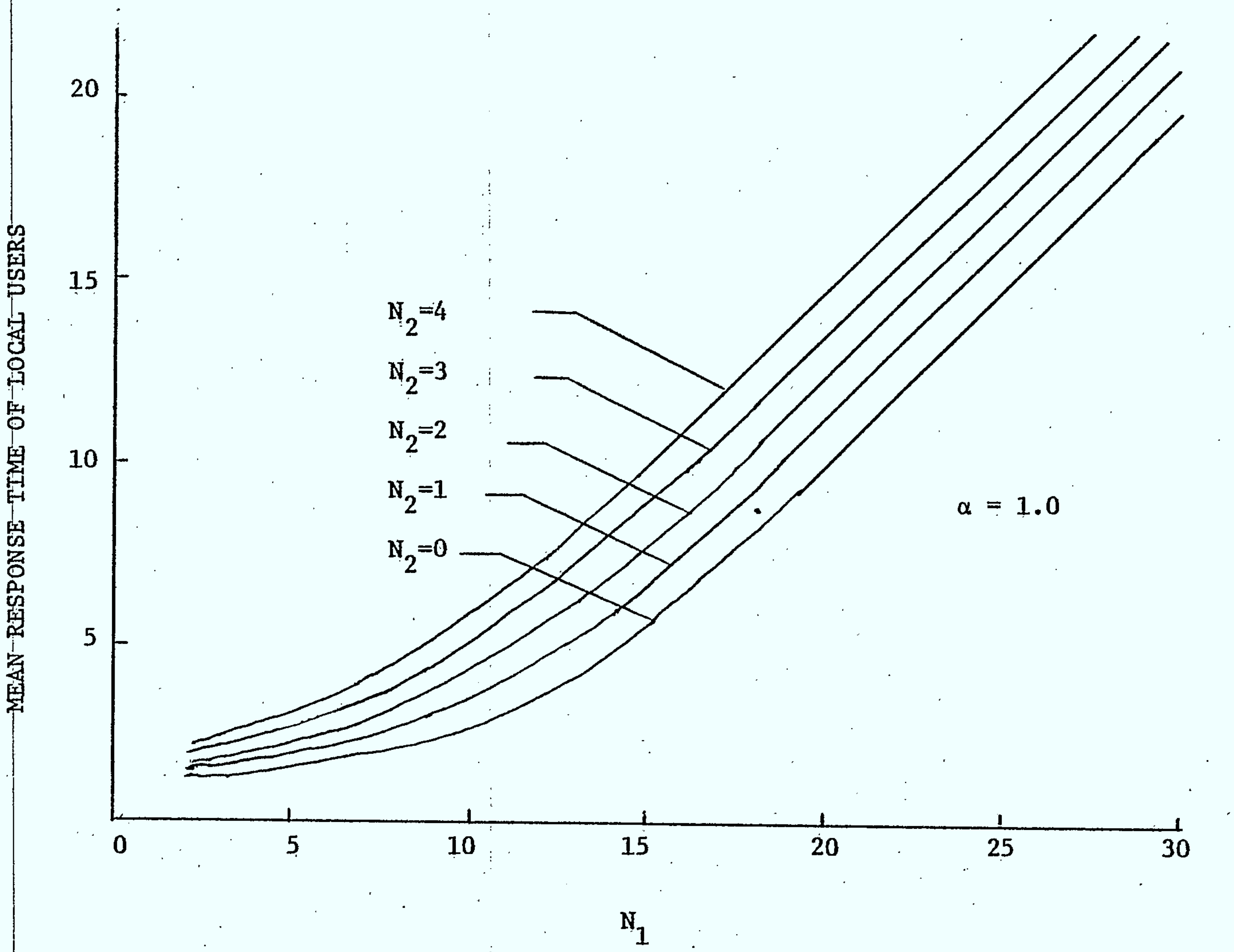


Figure 28. Mean Response Time of Local Users vs. N_1

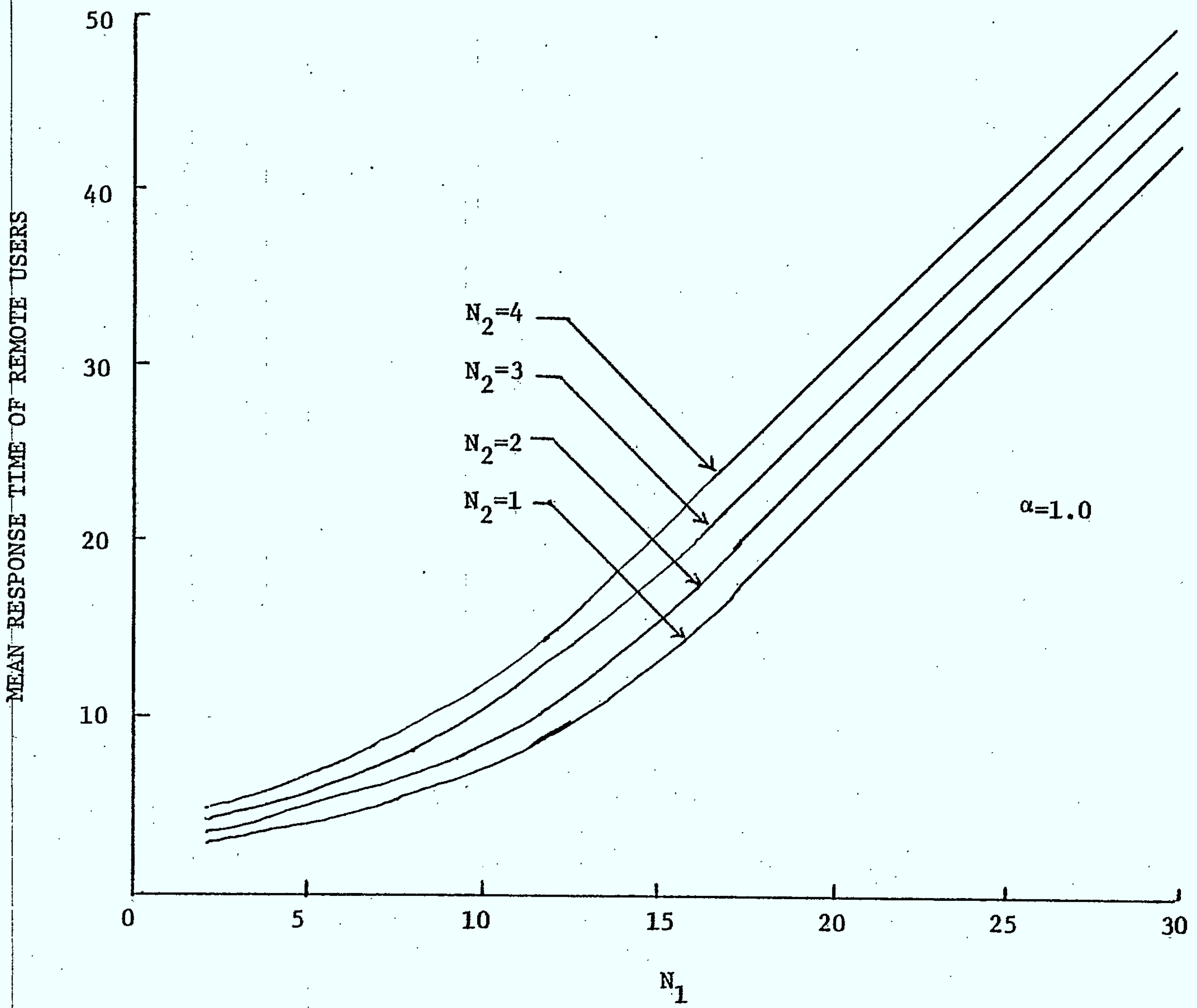


Figure 29. Mean Response Time of Remote Users vs. N_1

PART 4 -- MISCELLANEOUS ITEMS AND CONCLUDING REMARKS

4.1 COMMENTS ON THE ANALYSIS OF X.25 PROTOCOL (REQUESTED BY DR. Y.F. LUM)

The X.25 protocol is established by CCITT as an international standard for network access. A description of X.25 can be found in [38]. Of particular interest is the packet level protocol which is dealing with the establishment, maintainence, and termination of virtual circuits.

We consider the environment that a virtual circuit is already set up and data transfer is in progress. This environment is sufficient to characterize the network performance because the time spent in establishing and terminating a virtual circuit is small compared to that spent in data transfer.

A key feature in the X.25 packet level protocol is the window mechanism where the number of unacknowledged packets on a particular virtual circuit is limited by the window size. Let the window size be W. A packet is not allowed to enter the network if the window is closed, i.e., the number of unacknowledged packets is equal to W. This packet must wait for the window to open. The selection of window size is important because a small window size may result in low throughput while a large window size may result in unacceptable end-to-end delay.

The window mechanism also implements a flow control scheme because it regulates the rate at which packets (or messages) are admitted to the network. The modelling and analysis of flow control in [21,22,23] can be used to study the performance of the X.25 window mechanism <u>if</u> we assume that there is no error in data transfer and the acknowledgement is returned in zero time.

The assumption of zero acknowledgement delay is unrealistic. However, the analysis of a flow control model with end-to-end acknowledgement is very complex. At the present time, the only case which yields to exact analysis is that of a single virtual circuit established between nodes connected together by a single channel. This analysis is done by W. Yu and J. Majithia and is not available in open literature. The generalization of this analysis to more than one virtual circuit or more than one channel is not possible. A more extensive study of the window mechanism (with acknowledgement) can be found in a recent thesis by P. Kermani from UCLA [39]. He provides an approximate analysis of a tandem link model with end-to-end acknowledgement.

A feature related to acknowledgement is the time-out mechanism. In this mechanism, a packet is re-transmitted if the acknowledgement of this message is not received within a time-out period. This may due to (a) the original packet is lost, (b) the acknowledgement is lost, or (c) the acknowledgement is delayed by traffic in the network. Fayolle, Gelenbe, and Pujolle [40] have analysed a time-out mechanism for a window size of one. They called it the "send and wait" protocol. Unfortunately, their analysis is only approximate because of the way the acknowledgement packets are handled. Specifically, when a packet is re-transmitted, the acknowledgement of the original packet is not taken into consideration. This would give a pessimistic result for the mean end-to-end delay. The optimal time-out period which minimizes the mean packet delay is also derived. The generalization of Fayolle's result to a window size of greater than 1 is very difficult.

Kermani [39] has also considered the time-out mechanism and provided approximate results for network throughput and mean packet delay.

In conclusion, queueing analysis has been used successfully to study the performance of protocols such as X.25 which uses the window mechanism. The results obtained so far are either exact analysis for special cases or approximate for more general cases. Queueing analysis will continue to be the fundamental approach for performance analysis of the window mechanism in X.25.

4.2 COMMENTS ON R. GALLAGER'S WORK ON PROTOCOL (REQUESTED BY DR. M. SABLATASH)

Gallager [41] studied the amount of protocol information that must be transmitted in a data network to keep track of source and destination addresses and starting and stopping of messages. A lower bound on the required information per message is derived. This lower bound is found to depend on the distribution of message length, the message arrival rate, and the delay in transmitting the message.

Gallager's work is an information theoretic approach to the study of protocols. It is quite different from the other materials in this report which deal with the queueing theoretic approach. It gives us answers to the amount of protocol information required, but not to performance measures such as network throughput and end-to-end delay. In terms of protocols such as X.25, Gallager's study is related to the number of bits in a message for header information and not related to the effect of window size on network performance.

4.3 CONCLUDING REMARKS

Queueing network models have been used extensively in the performance analysis of message-switched (or packet-switched) communication networks. A survey of the application of these models to the analysis of end-to-end delay, buffer management, and flow control has been presented. The exact analysis of network models is restricted to the class of models analyzed by Baskett et al. [5]. Analytical results for models with more complex features, e.g., adaptive routing algorithms and finite buffer space, are therefore not presently available. These models are usually studied by approximation techniques and discrete simulation.

A theorem on the distribution of end-to-end delay in a message-switched network has been established. This theorem has been published in Computer Networks, vol. 2, no. 1 (Feb 1978) 44-49. It allows us to obtain statistics such as mean, variance, and percentiles of end-to-end delay. It also finds application in the analysis of user-resource networks.

Due to the increased interest in computer networking, it is conceivable that the size of networks will grow. The problems of routing and flow control in large, hierarchical networks have been considered. The analysis of routing algorithm is still limited to fixed or random routing. A hierarchical flow control technique has been suggested. Via a simple network example, this technique is shown to be more effective than the traditional end-to-end control in terms of a smaller network delay.

As mentioned in the introduction, the performance analysis of user-resource networks has received little attention. A queueing network model has been developed for a simple user-resource network. The mean response time of a computer system to local and remote users has been illustrated by means of a numerical example. This result, together with a state of the art survey, will appear in the September 1978 issue of the ACM Computing Surveys.

REFERENCES

1. Kleinrock, L. Queueing Systems -- Volume 2: Computer Applications. John Wiley & Sons, New York, 1976.

2. Pouzin, L. Presentation and Major Design Aspects of the CYCLADES Computer Network. Proc. Third Data Communications Symposium, St. Petersburg, Florida (Nov. 1973), 80-87.

3. Twyver, D., and Rybczynski, A. DATAPAC Subscriber Interfaces. Proc. International Conference on Computer Communications, Toronto (Aug. 1976), 143-149.

4. Kleinrock, L. Communication Nets -- Stochastic Message Flow and Delays. McGraw-Hill, New york, 1964.

5. Baskett, F., Chandy, K., Muntz, R., and Palacios, F. Open, Closed and Mixed Networks of Queues with Different Classes of Customers. Journal ACM 22, 2 (Apr. 1975), 248-260.

6. Kleinrock, L., and Naylor, W. On Measured Behavior of the ARPA Network. AFIP Conference Proceedings, 43 (1974), 767-780.

7. Gerla, M. The Design of Store-and-Forward (S/F) Networks for Computer Communications. Report UCLA-ENG-7319, University of California at Los Angeles, Jan. 1973.

8. Chandy, K., Hogarth, C., and Sauer, C. Selecting Capacities in Computer/Communication Systems. IEEE Trans. on Software Engineering 3, 4 (July 1977), 290-295.

9. Wong, J. Distribution of End-to-End Delay in Message-Switched Networks. to appear in Computer Networks.

10. Rubin, I. Path Delays in Communication Networks. Applied Mathematics and Optimization 1, 3 (1975), 193-221.

11. Labetoulle, J., and Pujolle, G. A Study of Queueing Networks with Deterministic Service and Applications to Computer Networks. Proc. International Symposium on Computer Performance Modeling, Measurement and Evaluation, Harvard University, Cambridge, Mass. (Mar. 1976), 230-240.

12. Irland, M. Analysis and Simulation of Congestion in Packet-Switched Networks, Ph.D. Thesis, University of Waterloo, Waterloo, Ontario, Canada, 1977.

13. Kamoun, F. Design Consideration for Large Computer Communication Networks. UCLA-ENG-7642, University of California at Los Angeles, Apr. 1976.

14. McQuillan, J., Crowther, W., Cosell, B., Walden, D., and Heart, F. Improvements in the Design and Peformance of the ARPA Network. AFIP Conference Proceedings 39 (Fall 1972) 741-754.

15. Network Analysis Corporation. Analysis and Optimization of Store-and-Forward Computer Networks. NAC Fourth Semiannual Technical Report, Defense Documentation Center, Dec. 1971.

16. Lam, S. Store-and-Forward Buffer Requirements in a

Packet Switching Network. IEEE Trans. on Commun. COM-24, 4 (Apr. 1976), 394-403.

17. Lam, S. and Reiser, M. Congestion Control of Store-and-Forward Networks by Input Buffer Limits. Proc. National Telecommunications Conference, Los Angeles (Dec. 1977).

18. Kahn, R. and Crowther, W. Flow Control in Resource Sharing Computer Network. IEEE Trans. on Commun. COM-20, 6 (June 1972), 539-546.

19. Davies, D. The Control of Congestion in Packet Switching Networks. IEEE Trans. on Commun. COM-20, 6 (June 1972), 546-550.

20. Price, W. Simulation of a Packet-Switched Data Network Operating under Isarithmic Control with Revised Link and Node Protocol. NPL Report COM 71, National Physical Laboratory, Sep. 1973.

21. Wong, J., and Unsoy, M. Analysis of Flow Control in Switched Data Networks. Proc. IFIP Congress 77 (Aug. 1977), 315-320.

22. Pennotti, M., and Schwartz, M. Congestion Control in Store and Forward Tandem Links. IEEE Trans. on Commun. COM-23, 12 (Dec. 1975), 1434-1443.

23. Chatterjee, A., Georganas, N., and Verma, P. Analysis of a Packet Switched Network with End-to-End Congestion Control and Random Routing. Proc. International Conference on Computer Communications, Toronto, Canada (Aug. 1976), 484-494.

24. Lam, S. Queueing Networks with Population Size Constraints. IBM Journal of Research and Development 21, 4 (July 1977), 370-378.

25. Mamrak, S. Dynamic Response Time Prediction for Computer Networks. Comm. ACM 20, 7 (July 1977), 461-468.

26. R. Cooper, Introduction to Queueing Theory (MacMillan, New York, 1972).

27. L. Kleinrock, Queueing Systems, Volume 1: Theory (Wiley-Interscience, New York, 1975).

28. E. Reich, Waiting Times when Queues are in Tandem, Ann. Math. Statistics 28 (1957) 768-773.

29. E. Reich, Notes on Queues in Tandem, Ann. Math. Statistics 34 (1963) 338-341.

30. Little, J. A Proof of the Queueing Formula $L = \lambda W$, Operations Research 9 (1961) 383-387.

31. Ornstein, S. et.al. The Terminal IMP for the ARPA Computer Network, AFIP Conference Proceedings 40 (1972) 243-254.

32. McKenzie, A. Host/Host Protocol for the ARPA Network, Network Information Center 7104, Stanford Research Institute, Jan 1972.

33. Postel, J. Survey of Network Control Programs in the ARPA Computer Network, Report MTR-6722, Mitre Corporation, Oct 1974.

34. Postel, J. Official Initial Connection Protocol, Network Information Center 7101, Stanford Research Institute,

June 1971.

35. Postel, J. Official Telnet-Logger Initial Connection Protocol, Network Information Center 7103, Stanford Research Institute, June 1971.

36. Crocker, S., Heafner, J., MetCalfe, R., and Postel, J. Function-oriented Protocols for the ARPA Computer Network, AFIP Conference Proceedings 40 (1972) 271-279.

37. Muntz, R. and Wong, J. Asymptotic Properties of Closed Queueing Network Models, Proc. 7th Princeton Conference on Information Sciences and Systems, (1974) 348-352.

38. Rybczynski, A., Wessler, B., Despres, R., and Wedlake, J. A New Communication Protocol for Accessing Data Networks: The International Packet Mode Interface. Proc. ICC (1976), 20.7-20.11.

39. Kermani, P. Switching and Flow Control Techniques in Computer Communication Networks. Ph.D. Thesis, University of California at Los Angeles, 1977.

40. Fayolle, G., Gelenbe, E., and Pujolle, G. An Analytic Evaluation of the Performance of the "Send and Wait" Protocol. IEEE Trans. on Commun. COM-26, 3 (Mar 1978), 313-319.

41. Gallager, R. Basic Limits on Protocol Information in Data Communication Networks. IEEE Trans. on Information Theory, IT-22, 4 (July 1976), 385-398.