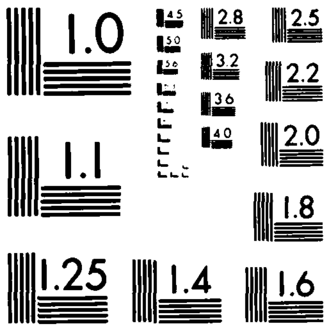
AD-A131 914 RESEARCH IN NETWORK MANAGEMENT TECHNIQUES FOR TACTICAL DATA COMMUNICATION NETWORKS(U) POLYTECHNIC INST OF NEW YORK BROOKLYN R R BOORSTYN ET AL. 01 MAR 81 1/1

UNCLASSIFIED DAAK80-80-K-0579 F/G 17/2.1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Polytechnic Institute of New York

(1)

AD A131914

SEMIANNUAL TECHNICAL REPORT

SEPTEMBER 1, 1980 TO FEBRUARY 28, 1981

"RESEARCH IN NETWORK MANAGEMENT TECHNIQUES FOR TACTICAL DATA COMMUNICATION NETWORKS"

FUNDED BY U.S. ARMY CORADCOM

CONTRACT NO. DAAK 80-80-K-0579

PRINCIPAL INVESTIGATORS:

Robert R. Boorstyn

Aaron Kershenbaum

Polytechnic Institute of New York

333 Jay Street

Brooklyn, N.Y. 11201

83 08 22 072

SEMIANNUAL TECHNICAL REPORT

SEPTEMBER 1, 1980 TO FEBRUARY 28, 1981

"RESEARCH IN NETWORK MANAGEMENT TECHNIQUES FOR TACTICAL DATA COMMUNICATION NETWORKS"

FUNDED BY U.S. ARMY CORADCOM

CONTRACT NO. DAAK 80-80-K-0579

PRINCIPAL INVESTIGATORS:

Robert R. Boorstyn

Aaron Kershenbaum

Polytechnic Institute of New York

333 Jay Street

Brooklyn, N.Y. 11201

TABLE OF CONTENTS

I. INTRODUCTION

In this semi-annual report we describe progress in each of the four major areas of our research. These areas are dynamic routing, multihop packet radio networks, new switching techniques in networks, and algorithms.

We have already reported on our proposed dynamic routing technique. In section IIA we describe the results of simulations that have confirmed predictions about performance and deepened our understanding of the problem.

Our previous work on multihop packet radio has been extended to include the effects when passive acknowledgements are not heard. We are able to extend our analytic model to include this phenomenon and evaluate the performance of the network for a variety of protocols. A new protocol is proposed that essentially guarantees that passive acknowledgements will be heard. Details are given in section IIB.

The third area of our research is to explore new switching techniques in networks. There are many applications for which packet switching is inefficient. Fortunately modern networks have the resources - processing power, memory, distribution of intelligence - to utilize much more sophisticated communication techniques. We have been exploring this subject by investigating various techniques and applications and by finding bounds on the best that can be achieved. In section IIC we describe our results for a particular application. Packet switching is inefficient, but is still used for lack of an alternative, when a large number of users transmit short messages with a tight time delay requirement. Here we find an optimum packeting strategy and compare it with others. In later work we will propose alternative switching schemes for this application.

In section IID we describe a generalization to our approach to algorithms. We have already reported on many algorithms used to design networks. Here we explore the asymptotic and probabilistic behavior of algorithms.

Research in each of these four areas is continuing.

II. RESEARCH SUMMARIES

IIA. A Simulation Study of A Dynamic Routing Scheme

1. Introduction

A dynamic routing scheme has been previously proposed. Instead of assigning fixed paths or randomly selecting paths, permissible paths (e.g., minimum hop paths) are pre-assigned to each node pair. A packet arriving at a node, and destined for another node, is assigned to one of two types of queues. A dedicated queue is a queue whose packets are served by one and only one dedicated server. A shared queue is a queue whose packets may be served by two or more specified servers. A server is equivalent to an output link. The purpose of pre-selecting paths is to avoid long paths and resultant increased network traffic that would occur in a random routing scheme. The purpose of the shared queue is to let a portion of the nodal traffic experience the delay of a multiple server queue and therefore reduce the average packet delay at a node. The objective of the model is therefore to reduce both nodal as well as network average packet delay.

We have conducted a series of simulation studies to evaluate the nodal as well as the network delay under the proposed dynamic routing scheme. Section 2 describes the nodal and network models as well as the different service disciplines used. Section 3 gives the results of nodal and network delay performance. A summary is given in Section 4.

2. The Model

The Nodal Model

The traffic from each node pair is preassigned to one or more pre-selected paths (and therefore server(s) at a node). Two types of queue can be constructed at a node. A dedicated queue can only be served by one dedicated server. There are at most n such queues at a node with n links. A shared queue can be served by any of several preassigned servers. There could be as many as 2^n-n-1 shared queues at a node. A two server, three queue nodal model is shown in Figure IIA.1.

In Figure IIA.1, traffic arrives at the different queues with arrival rates λ_i ($i = 1,2,3$) packets/sec. Queue 1 and queue 3 are competing with server 1 for service while queue 2 and queue 3 are competing for service by server 2. The service rate is μ packets/sec. If one of the competing queues (e.g., queue 1 or queue 3) is empty (e.g., queue 1), and the server for these queues is idle, a packet in the non-empty queue is served. If both queues are not empty when the server has just completed service, a pre-defined service discipline will determine which queue to serve next. When both competing queues are empty (e.g., queue 1 and queue 3) and the server for these queues is idle, the server will stay idle until there is a new arrival to one of these queues. The idle server will not serve the queue which is not pre-assigned to it for service. This is the major difference between our nodal model and a multiple server queueing system. The operation of a node with more communications links is basically the same as that of a two link node. Shared queues will be served by a subset of the servers.

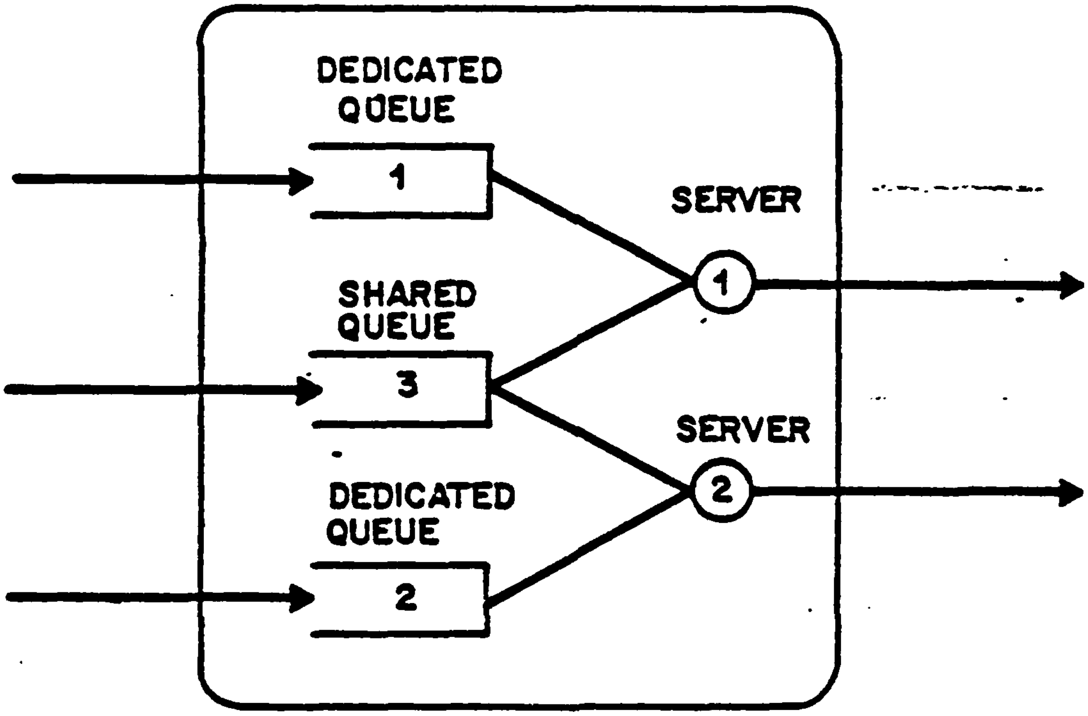


FIGURE IIA.1. A NODAL QUEUEING MODEL WITH TWO SERVERS

Service Disciplines

The nodal model described in the previous section allows different types of queues to compete for service. A service discipline must therefore be specified. We will describe four different service disciplines for the nodal model of Figure IIA.1.

a. Priority Service Discipline (PSD)

A server always searches the dedicated queue for service prior to its service of the shared queue. For example, server 1 in Figure IIA.1, after completing a service, will always look to queue 1 for more service. Packets in queue 3 will be served by server 1 only if queue 1 is empty when server 1 has just completed a service. Therefore, server 1 is busy as long as there are packets to be served in queue 1 or queue 3. Conversely, server 1 is idle and stays idle when both queue 1 and queue 3 are empty. The instantaneous queue length of queue 2 does not effect any of the decision rules at server 1 described above. A packet entering the shared queue when both servers are idle is served randomly by either server 1 or server 2.

b. Alternate Service Discipline (ASD)

A server at the node in Figure IIA.1, whenever possible, alternatively serves n packets from the dedicated queue and m consecutive ones from the shared queue. The server is forced to serve more than n (or m) packets from the dedicated (or the shared) queue if the other queue happens to be empty. Under this situation, alternation of service will begin whenever there is a new arrival to the empty queue. The process is started over again at this point.

c. Serve The Longer Queue Discipline (SLQD)

The queue with the longer queue length among the two competing queues (the shared and the dedicated queues) gets service first. A server, upon a service completion, compares the queue lengths of the two competing queues, and serves a packet from the longer queue. The SLQD tries to equalize all the queue lengths at a node.

d. Random Service Discipline (RSD)

A server (e.g., server 1) upon completing a service, serves packets from one of two competing queues according to the following algorithm: it serves queue 1 if queue 3 is empty and vise versa; it randomly selects a queue if both queue 1 and queue 3 are non-empty. This is a totally "uncontrolled" service discipline.

The Network Model

In order to describe the dynamics of a packet switching network employing the proposed dynamic routing scheme, one must (1) construct nodal queueing models similar to the one in Figure IIA.1 for every node in the network, (2) pre-select a set of "permissible" paths for each source-destination node pair, and (3) assign the data flow according to the preselected paths between adjacent nodes. In order to simplify description of the network model, we have chosen a four node network to illustrate the dynamics of our network model.

A four node network is shown in Figure IIA.2. Every node is connected to two links, i.e., is of degree two. Therefore, the nodal queueing model at each node is exactly the same as that of the two server, three queue model described previously. The permissible paths between sources and destinations are selected based on a shortest path (minimum hop) algorithm. Adjacent nodes will only use the one hop path between them. Non-adjacent nodes will use either one of the two two-hop paths. Therefore, traffic between adjacent nodes is pre-assigned to the dedicated queues. Traffic between non-adjacent nodes, on the other hand, is pre-assigned to the shared queue at the source node and the appropriate dedicated queue at the intermediate node.

Each server at the node serves the two competing queues (the dedicated queue and the shared queue) according to a pre-specified service discipline. We have simulated the network for the four service disciplines discussed previously. A service discipline where the condition of the next node is "known" to all servers at the current node and used to determine routing is also studied.

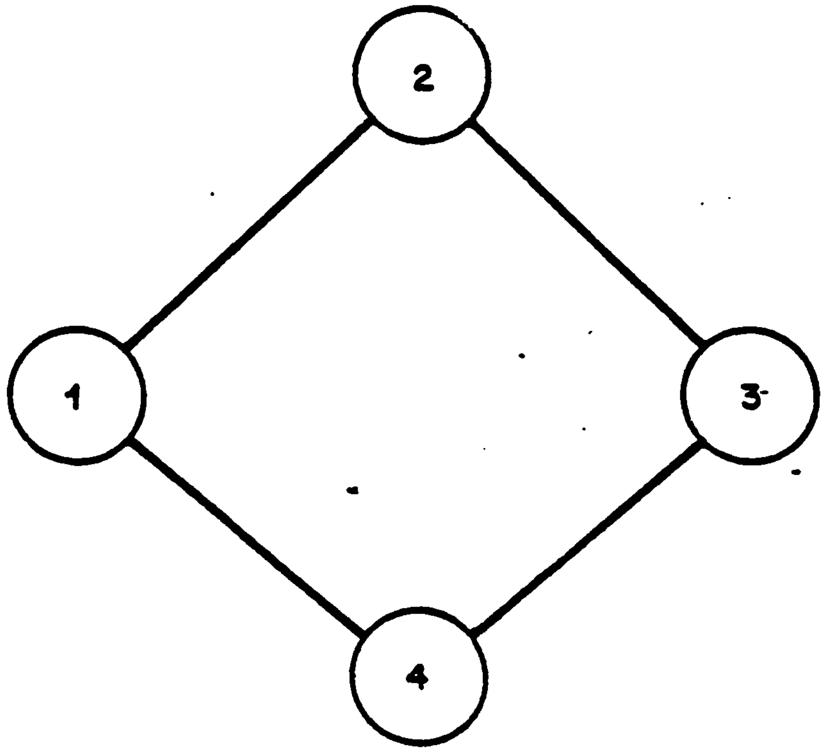


FIGURE IIA.2 A FOUR NODE NETWORK

3. Results

Simulation programs in PL/1 have been written to evaluate the delay of the nodal model of Figure IIA.1 and the network model of Figure IIA.2 operating under the different service disciplines. All simulation results are based on statistics collected over a sufficient time interval so as to represent the steady state. The statistics collected for the PSD scheme are based on at least one million (sometimes four million) simulation events -including both arrivals and departures. In order to conserve computing time, the statistics collected for other service disciplines are based on at least two hundred thousand simulation events. In our simulation program, all arrival processes to source nodes are assumed to be Poisson and all service distributions are assumed to be exponential and independent of each other. No assumption is made about arrivals processes to intermediate nodes in the network model.

Nodal Delay Performance

The percentage of the traffic that is shared at a node is an important parameter affecting the nodal delay, We have studied nodal delay based on three sets of nodal traffic mixes. They are low (approximately 10% to 15%), moderate (25%), and high (50%) percentages for the shared traffic at a node respectively. Fairly extensive simulation runs were conducted throughout the whole range of nodal utilization (0 to 1) for the PSD and a moderate percentage of shared traffic. We assumed that the input processes to the node were Poisson.

TABLE IIA.1 Nodal Delay Of The PSD With Low, Moderate, And High Percentages Of Traffic Being Shared ($\rho=\frac{\lambda_1+\lambda_2+\lambda_3}{2\mu}$, μ = 4 packets/sec)

Shared	Arrival Rate to Queue (packets/sec.)			Utilization	Nodal Delay (Seconds)		
Traffic	λ_1	λ_1	λ_3	ρ	PSD	M/M/2	M/M/1
Low	2.8	2.8	1	0.825	1.01	0.78	1.43
	3.0	3.0	1	0.875	1.33	1.07	2.00
	3.2	3.2	1	0.925	2.08	1.73	3.33
Moderate	2.475	2.475	1.65	0.825	0.91	0.78	1.43
	2.625	2.625	1.75	0.875	1.18	1.07	2.00
	2.775	2.775	1.85	0.925	1.87	1.73	3.33
High	1.65	1.65	3.3	0.825	0.80	0.78	1.43
	1.75	1.75	3.5	0.875	1.09	1.07	2.00
	1.85	1.85	3.7	0.925	1.78	1.73	3.33

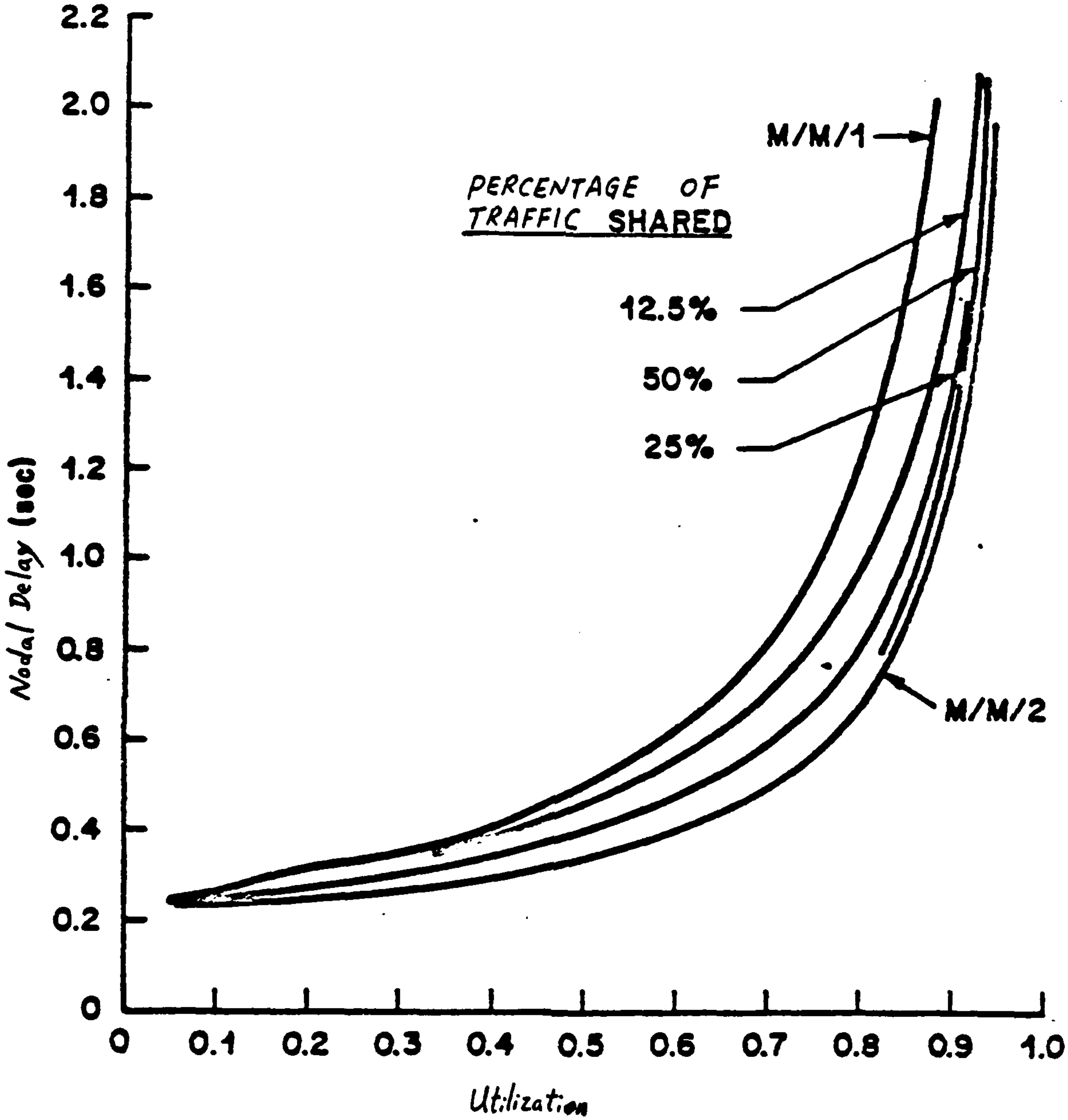


FIGURE IIA.3. DELAY-THROUPUT RELATIONSHIP OF NODAL MODEL

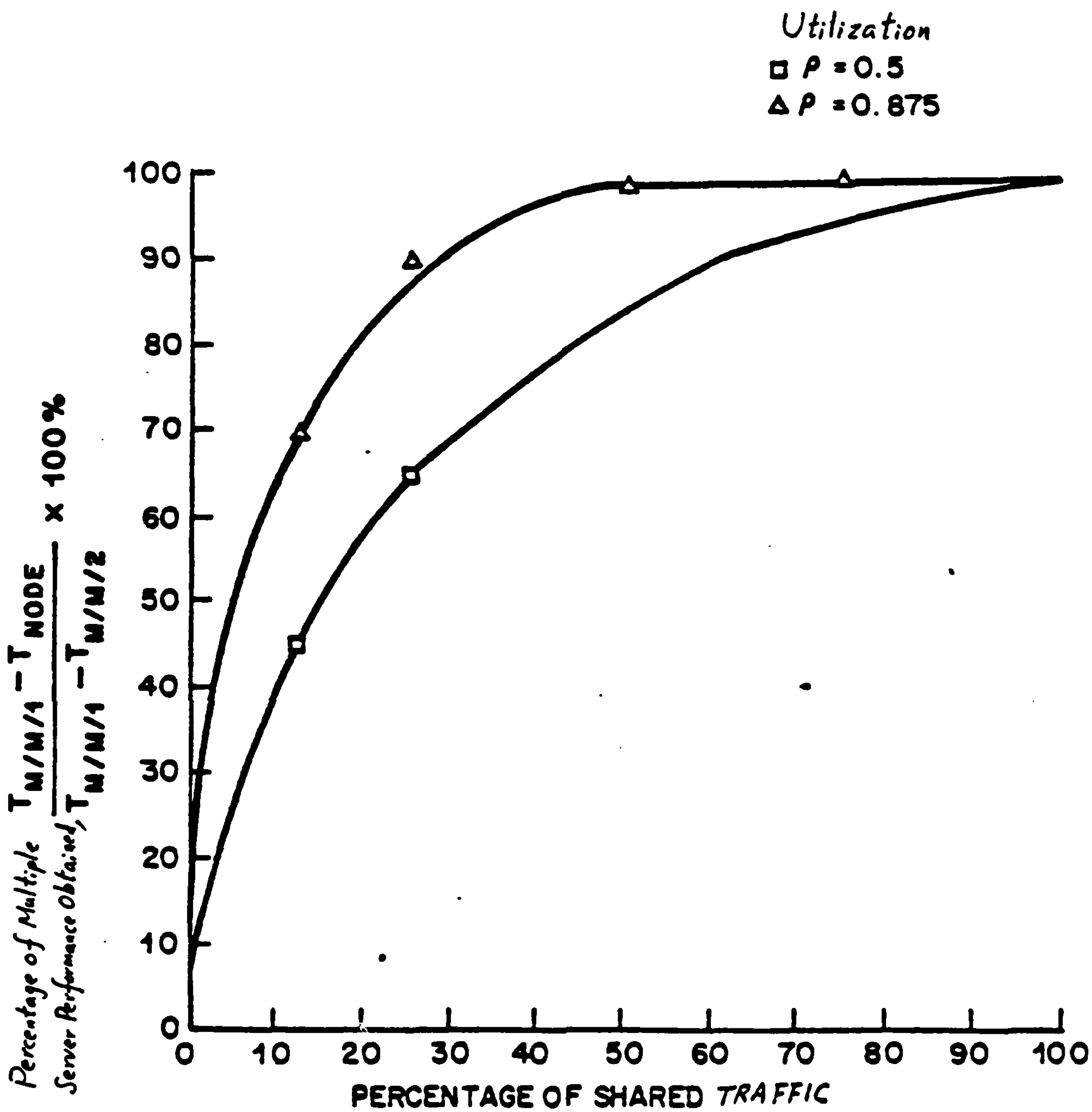


FIGURE IIA.4. COMPARISON OF PSD AND MULTIPLE SERVER QUEUES

Simulation Results Of The Priority Service Discipline

The effects of the percentage of the traffic that is shared on nodal delay are tabulated in Table IIA.1. Notice that the higher the percentage, the closer the nodal delay is to that of an M/M/2 queueing system. Figure IIA.3 plots the delay-throughput relationship of the PSD for different percentages of shared traffic. The delay for the M/M/2 and the M/M/1 queueing systems are also plotted for reference.

In Figure IIA.4, a measure of the "closeness" of the PSD to M/M/2 queueing system performance, is plotted against the percentage of shared traffic for a node operated under the PSD scheme. Two utilizations are studied. The ordinate is the percentage of multiple server performance that is obtained by the PSD. At high traffic utilization (ρ=0.875), only 25% of the nodal traffic need to be shared to achieve 90%. At higher nodal utilization, the same effect is reached with less shared traffic. At moderate nodal traffic utilization (ρ=0.5), th PSD achieves 65% of the multiple server performance with only 25% of the traffic being shared.

In Figure IIA.5, we have redrawn the plot of Figure IIA.3 for the average nodal delay in the range from 0 to 0.5 seconds for the case of 25% shared traffic. For a fixed delay of 0.5 second, the utilization of the PSD scheme is about 25% greater than that of the M/M/1 model (a commonly used fixed routing nodal nodel). Thus for the same throughput a significant decrease in required channal capacity is obtained.

Results For Other Service Disciplines

In this section, we discuss nodal delay for other service disciplines. For these disciplines, we only studied the nodal performance for ρ=0.875 and 50% of the traffic being shared.

Table IIA.2 tabulates the nodal delay from our nodal simulation for the ASD, the SLQD, and the RSD. We can make the following observations from Table IIA.2:

a. A general characteristic of the results in Table IIA.2 is that: at high utilizatons (ρ=0.875 in this case) with a high percentage of the traffic shared (50%), the nodal delay is closer to that of the M/M/2 than the M/M/1 queueing system regardless of the service disciplines.

b. The PSD has the best nodal delay among all service disciplines.

c. The nodal delay of the SLQD, with the modification of not serving (whenever possible) two packets in a row from the shared queue, is the closest to that of the PSD.

d. The ASD with parameter n = 1 and the threshold at the shared queue set to zero, has almost the worst nodal delay performance.

e. The RSD, being that it is an "uncontrolled" service discipline, does perform the worst among all service disciplines. However, the RSD performs better than the M/M/1 queuing system by almost 40%.

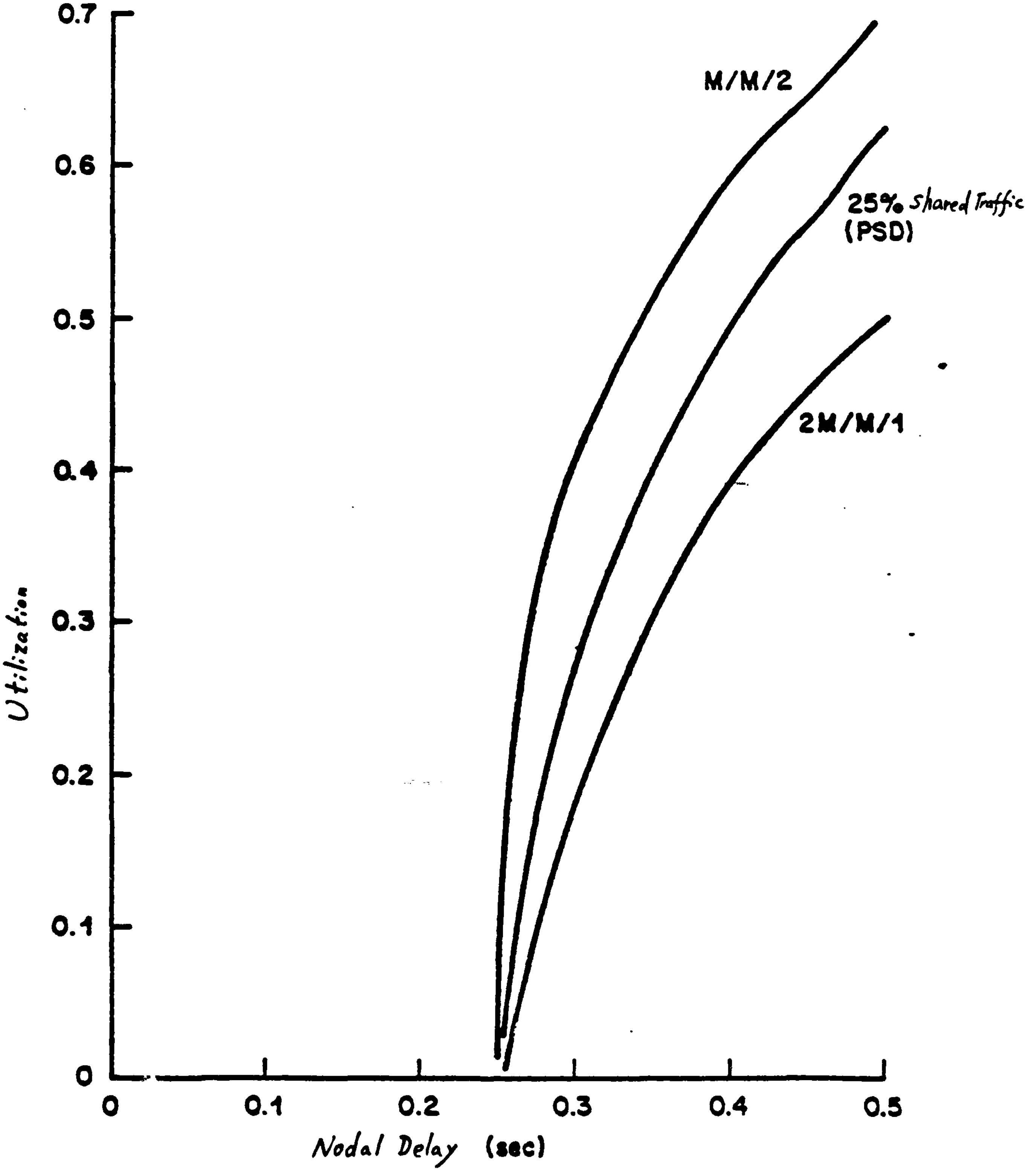


FIGURE IIA.5 DELAY-THROUGHPUT RELATIONSHIP FOR NODAL MODEL

TABLE IIA.2 Nodal Simulation Results

($\lambda_1=\lambda_2=1.75$ packets/sec, $\lambda_3=3.5$ packets/sec, $\mu=4$ packets/sec, and $\rho=0.875$)

Service Discipline			Nodal Delay (Seconds)
PSD			1.03
SLQD			
Serve the Longer	$y = 1$		1.04
Queue but not y	$y = 2$		1.09
consecutive "shared	$y = 3$		1.08
queue" packets	$y = \infty$		1.13
		$n = 1$	1.11
ASD	TH = 0	$n = 2$	1.12
Alternate Service:		$n = 3$	1.05
one from shared		$n = 1$	1.09
queue and n	TH = 1	$n = 2$	1.05
from dedicated		$n = 3$	1.07
queue but do not		$n = 1$	1.08
serve shared	TH = 2	$n = 2$	1.05
queue when its		$n = 3$	1.04
queue length $\leq$ TH		$n = 1$	1.06
	TH = 3	$n = 2$	1.07
		$n = 3$	1.05
RSD Random			1.13
M/M/2			1.01
M/M/1			2.00

Network Delay Performance

Analysis

In this section, we study the average network delay performance. In our routing strategy, instead of treating each communications link separately, we focus our interest on the whole node, including all its communications links. As a result, our focus is on nodal delay as opposed to link delay, which is usually studied.

Let us assume that we have an expression for nodal delay at node j, which is denoted by T_j. The average number of packets N_j at node j can therefore be written as $N_j = \lambda_j T_j$, where λ_j is the total traffic rate (from both external and internal traffic sources) at node j. The average number of packets in the network (N) is therefore the sum of the average number of packets at each node. That is, $N = \sum_{\text{all nodes}} N_j$. The average network delay (T) is therefore

$$T = \frac{1}{\gamma} \sum_{\text{all nodes}} \lambda_j T_j \qquad \text{(IIIA-1)}$$

where $\gamma = \sum_{j,k} \gamma_{jk}$ is the total external network traffic demand and γ_{jk} is the offered rate of traffic between pairs (j,k). Equation (IIA.1) is therefore a general expression for the average network delay.

To analyze the network delay we make the following assumptions. We assume the node models used previously. The traffic is found by applying the routing implied by the preassigned paths. We can then calculate the rate in packets per second of arrivals to the shared and dedicated queues. In our network analysis we assume these arrivals to be a Poisson process and compute the nodal delay for each node using the techniques described above. Equation (IIA.1) gives us the network delay. Below we compare this technique with the results of two simulations. The first dispenses with the Poisson assumption and accurately models arrivals to intermediate nodes in paths. To make the simulation simpler, packet lengths are independently reassigned at each node in a path. The second simulation dispenses with this approximation and retains the initial packet length for a packet throughout its path.

Simulation

In this section, we will focus our discussion on two PL/1 simulations which were written to evaluate the network delay performance. The interarrival time of external arrivals to network nodes is assumed to be exponentially distributed. The packet length is assumed to be exponentially distributed. All servers are assumed to have the same service capacity (for convenience chosen as unity). Therefore, the average service rate of each server is the same. In most of our simulation studies, we have used $\rho=0.875$, 50% shared traffic at the nodes, and a symmetrical network. Other traffic utilizations, nodal traffic mixes, and asymmetric network traffic demands are also studied but not as extensively as the above.

The first simulation allows the packet lengths to be reassigned at the next node. It does not make any assumptions about the output process from a server. A server serves its competing queues according to a pre-specified service discipline. After the server finishes service of a packet, the packet either exits from the network or joins the next queue for more service. Once it arrives at the next queue (in the next node), a new packet length from the same exponential distribution is assigned to it. It then is put on the appropriate queue and waits for service. We compare the results of our network simulation with our analysis based upon nodal simulation results. In the latter we assume all inputs to the node to be Poisson.

The second simulator simulates the true service situation in an operating packet network. That is, a particular packet when served by the next server will require the same amount of service time. When the results obtained from the second simulation are compared with our analysis by using our nodal simulation results, we can evaluate the effect of both the Poisson output stream and the packet length reassignment assumptions.

The First Network Simulation

Local Service Disciplines

All four service disciplines are studied. The simulation results in Table IIA.3 and in previous work reveal the following:

a. The average network delay from the network simulation is significantly colser to that of the M/M/2 nodal model than to the M/M/1 nodal model.

b. The difference between the analytical results using the nodal model and those obtained from the network simulation, for the PSD, increases as the percentage of shared traffic increases at a node.

c. With the same traffic mix at a node, the difference between the network delay found analytically and those obtained from the network simulation for the PSD increases as the traffic load increases.

d. The network delay obtained from the network simulation for the Alternate Service Discipline (ASD) and the Serve the Longer Queue Discipline (SLQD) are lower than that of the PSD scheme. The difference in these delay are at most 11%. The network delay of the Random Service Discipline (RSD) scheme is approximately the same as that of the PSD scheme.

e. The difference between the results obtained analytically and from our network simulation for other service disciplines is substantially large (12 to 26 percent difference).

f. The parameter n (the number of packets to be served from the dedicated queue before the server alternates its service) in the ASD has an effect on network delay.

g. In the ASD, the threshold parameter set at the shared queue to avoid the shared queue becoming empty before the dedicated queues, also has an effect on network delay.

h. The network delay of the RSD - an "uncontrolled" and "unpredictable" service discipline - is about the same as that of the PSD and the "worst cases" of the ASD and the SLQD.

Although we are operating at high utilization, we are getting most, but not all, of the expected improvement. Many different nodal strategies have only a modest effect on resolving this gap. We find that it was easy to generate strategies that produced multiple server behavior. But some of them have a profound effect on the output process from the server, and thus affect the Poisson assumption for inputs to the next node. The PSD is notorious in this regard. Usually any traffic one hop away from its destination will be placed in a dedicated queue. Thus, only continuing traffic will be found in nondedicated queues. These have lower priority and will be transmitted only when the dedicated queue is empty. But then a burst of several packets is likely to be sent. A bursty arrival process results in much poorer queue performance than a Poisson process. This contributes significantly to the gap in delay. The design of a nodal strategy must consider not only multiserver performance at a node,

TABLE IIA.3 Network Simulation Results
($\lambda_{i1}=\lambda_{i2}=0$ packet/sec, $\lambda_{i3}=3.5$ packets/sec,
$\mu_{i1}=\mu_{i2}=4.0$ packets/sec. $\rho=0.875$)

Service Discipline			Network Delay (seconds) Nodal Model	Simulation	% Difference
PSD			2.06	2.75	33%
ASD Alternate Service: one from shared queue and n from dedicated queue but do not serve shared queue when its queue length $\leq$ TH	TH = 0	n = 1	2.22	2.65	20%
	TH = 0	n = 2	2.24	2.51	12%
	TH = 0	n = 3	2.10	2.58	23%
	TH = 1	n = 1	2.18	2.50	15%
	TH = 1	n = 2	2.10	2.50	19%
	TH = 1	n = 3	2.14	2.53	18%
	TH = 2	n = 1	2.16	2.59	20%
	TH = 2	n = 2	2.10	2.52	20%
	TH = 2	n = 3	2.08	2.62	26%
	TH = 3	n = 1	2.12	2.51	18%
	TH = 3	n = 2	2.14	2.49	17%
	TH = 3	n = 3	2.10	2.55	21%
SLQD Serve the Longer Queue but not y consecutive shared queue packets	y = 1		2.08	2.45	18%
	y = 2		2.18	2.58	19%
	y = 3		2.16	2.67	24%
	y = ∞		2.26	2.67	19%
RSD			2.26	2.71	20%
M/M/2			2.02	-	-
M/M/1			4.00	-	-

$$* \quad \% \text{ Difference} = \frac{T_{\text{NETWORK}}\text{(Simulation)} - T_{\text{NETWORK}}\text{(Nodal Model)}}{T_{\text{NETWORK}}\text{(Nodal Model)}}$$

but the resultant output process as well. The usual way such a burst could occur is when the packets are short. But these should have made the next node's job easier. Unfortunately, the second aspect of the independence assumption now comes into play - packet lengths are reassigned at each node. This accentuates the effect of burstiness. When the simulation is altered to take into account dependence between nodes, hence dispensing with the independence assumption, the difficulty disappeared.

<u>A Look Ahead Service Scheme</u>

A common situaton of all the "local" service disciplines (The PSD, the ASD, the SLQD, and the RSD) is that a packet may be waiting to be served in the shared queue at one node while there is an idle server at the next node. Conversely, a packet may wait in the dedicated queue for service when there is no need to serve a shared queue packet because the next server has plenty of packets to serve at the next node. Both phenomena cause degradation in network performance. Both are due to the fact that each server, upon completing a service, does not know what the next server wants. Consequently, the server may serve a "wrong" packet from time to time. Therefore, it is not the selection of a queue but the timing of this selection process which influences the service decision. In all of the "local" service disciplines which we have discussed so far, the timing factor was not considered. We have designed a look ahead service discipline and have examined its performance. The basic algorithm is shown in Figure IIA.6.

Table IIA.4 shows our results. We make the following observations:

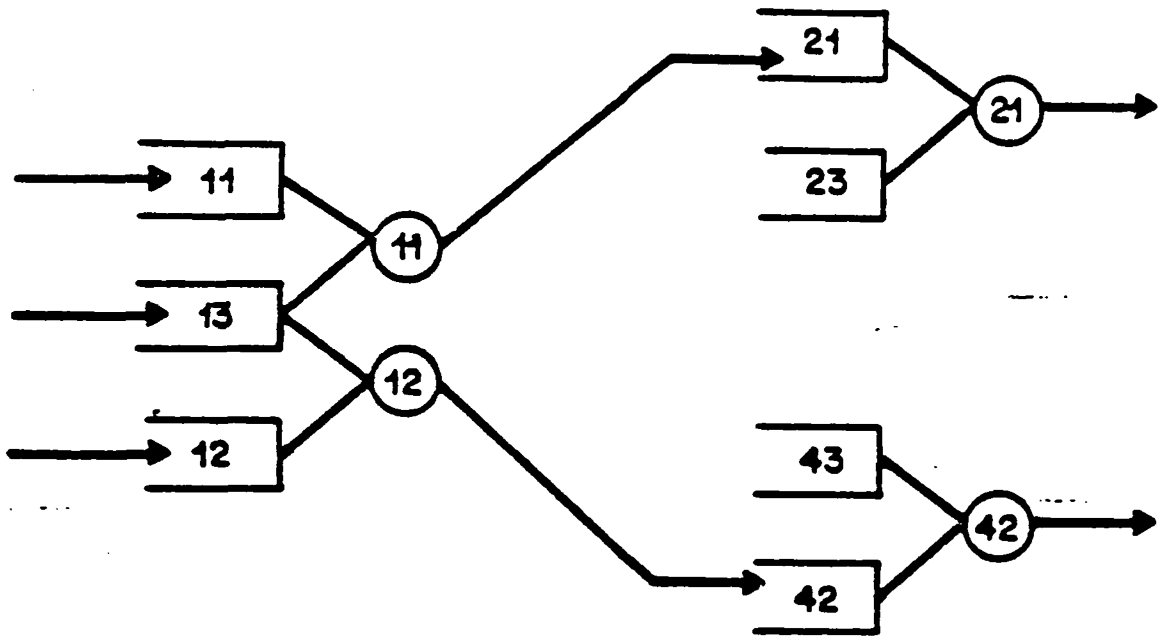
a. The network delay of the best look ahead scheme has an improvement of about 11% over that of the best local scheme.

b. The look ahead service scheme performs consistently better than the local service disciplines.

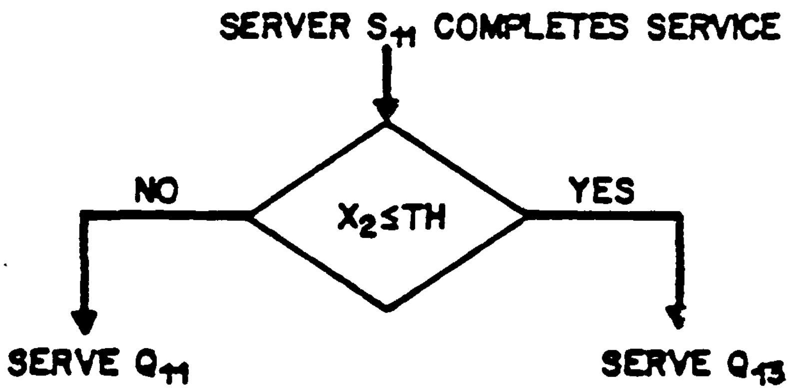
c. When both servers are empty, it is better to serve an arrival to the shared queue on the basis of downstream information, than to do it randomly.

The Second Network Simulation - No Packet Length Reassignment

The analytically obtained network and our first simulation differ not only in the PSD but also in the ASD and the SLQD where the output stream should be somewhat smoother. Furthermore, the packet length reassignment assumption does not work well in a small network like our four node network with very few traffic streams flowing at each node. This motivated us to build the second network simulator whereby the packet length reassignment assumption is not made.



(a) A PART OF THE FOUR NODE NETWORK



$X_2 = Q_{21}$ OR $Q_{21} + 1/2\, Q_{23}$

(b) THE LOOK AHEAD ALGORITHM

FIGURE IIA.6 A LOOK AHEAD SERVICE SCHEME

TABLE IIA.4 Network Simulation Results Of A Look Ahead Routing Strategy ($\lambda_{i1}=\lambda_{i2}=0$ packet/sec, $\lambda_{i3}=3.5$ packets/sec, $\mu_{i1}=\mu_{i2}=4$ packets/sec, $\rho=0.875$)

Threshold	Network Delay (seconds)		
	$\chi_2 = Q_{21}$*	$\chi_2 = Q_{21}$**	$\chi_2 = Q_{21} + \frac{1}{2}Q_{23}$**
0 (PSD)	2.75	2.75	--
1	2.33	2.26	2.27
2	2.36	2.29	2.19
3	2.37	2.32	2.26
4	2.35	2.42	2.32
5	--	--	2.35
6	--	--	2.41

* When both S_{11} and S_{12} are not busy and there is an arrival to Q_{13}, this packet will be served by either S_{11} or S_{12}.

** Under the same situation as above, χ_2 will be compared with χ_4 (the same parameter at node 4).

The packet in Q_{13} will be served by S_{11} (or S_{12}) if $\chi_2 < \chi_4$ (or $\chi_2 > \chi_4$)

Dynamic Routing Scheme With Local Node Service Disciplines

We have used our second simulation to evaluate the network delay performance under the various service disciplines (PSD, ASD, SLQD, and RSD) discussed previously. Each node, operates independently, serves its queues according to the pre-specified service discipline. Traffic information at each node is not exchanged between adjacent nodes. Therefore, only "local" node service disciplines are discussed in this section. We have also examined the average network delay under two fixed routing strategies.

Table IIA.5 tabulates our results. We make the following observations:

a. Removing the packet length reassignment assumption greatly improves the accuracy of our analytic nodal method.

b. The packet length reassignment assumption has more effect on the network delay performance for the PSD than any other service disciplines.

c. The PSD is the best "local" service disciplines among all other local service disciplines.

d. The network delay of our dynamic routing scheme under any one of the service disciplines is significantly better than either of the benchmark fixed routing strategies.

Dynamic Routing With Look Ahead Service Scheme

The "clock driven" network simulation keeps track of how much service time a packet requires from a server. This enables a server to know how much work (in terms of service time) the next server has at any given instant of time. A server can make an accurate decision to keep the network from losing its service capacity. A server, upon completing a service, compares the total amount of work

TABLE IIA.5 Network Simulation Results For Different Service Disciplines - Local Decision Rules
($\lambda_{i1}=\lambda_{i2}=0$ packet/second, $\lambda_{i3}=3.5$ packets/second, $\mu_{i1}=\mu_{i2}=4$ packets/second, $\rho=0,875$)

Service Discipline	Network Delay (seconds) First Simulation	Second Simulation	Analytic
PSD	2.75	2.16	2.06
ASD	2.65	2.38	2.22
SLQD	2.45	2.32	2.08
RSD	2.73	2.56	2.26
Fixed Routing #1	--	3.84	--
Fixed Routing #2	--	3.26	--

at the next server (the remaining service time for the packet still in service at the next server, the total service time in the dedicated queue at the next node, the total service time in the shared queue at the next node, and a threshold parameter which incorporates the fact that the shared queue packets are served by both servers at the next node and there are new arrivals to the next node during the service time) to the sum of the service times of the first packet in each of its competing (shared and dedicated) queues. The server serves the shared queue packet if the total amount of work at the next server is less than the sum of the service times of the first packet in each of its competing queue. Should the server not serve the shared queue packet immediately, the next server could be idle for a period of time equal to the difference between these two quantities. The server would otherwise serve the dedicated queue packet. This look ahead service scheme would be ideal in our dynamic routing if the threshold parameter could be correctly set. The network delay with both exponentially distributed and constant packet lengths are evaluated. The following observations can be made (see Table IIA.6):

a. The PSD is equivalent to setting the threshold value to negative infinity.

b. The network delay is insensitive to the threshold values near its "optimal" setting.

c. The difference between the network delay of the "best" local service discipline (the PSD) and the look ahead service scheme with the "best" choice of the threshold value is about 10%.

Assymetric Traffic Demands And Service Rates

We have illustrated significant improvements in network delay using our adaptive techniques for the four node network. However, only symmetric traffic loads have been studied so far. In order to demonstrate the robustness of our routing strategy, a certain degree of asymmetry was introduced into the network. We have evaluated the network delay of the four node symmetric network with asymmetric traffic and with different link capacities.

We make the following observations from the results of Tables IIA.7 and IIA.8:

The RSD performs the worst among all service disciplines; the PSD performs the best among all routing strategies based on local traffic information; the "best" look ahead scheme performs

TABLE IIA.6 Network Simulation Results For A Look Ahead Routing Strategy ($\lambda_{i1}=\lambda_{i2}=0$ packet/second, $\lambda_{i3}=3.5$ packets/second, $\mu_{i1}=\mu_{i2}=4$ packets/second, $\rho=0/875$)

Routing Strategy			Network Delay (seconds)	
			Exponential Packet Lengths	Constant Packet Lengths
Local		PSD	2.16	1.23
		RSD	2.56	1.31
Look Ahead	TH	-0.25	1.96	1.17
		0.00	1.94	1.14
		0.05	1.96	--
		0.10	1.94	--
		0.15	1.95	--
		0.25	1.96	1.20
		0.50	2.01	1.27
		0.75	2.07	1.33
		1.00	2.13	1.38

TABLE IIA.7 Network Delay With Asymmetric Traffic Demands
(λ_{i1}=2 packets/sec, λ_{i2}=1 packet/sec, λ_{i3}=2 packets/sec,
μ_{ij}=4 packets/sec, ρ=0/875)

Routing Strategy			Network Delay (seconds)
Local	PSD		1.91
Local	SLQD		1.96
Local	RSD		2.67
Look Ahead	TH	-0.25	1.81
Look Ahead	TH	-0.125	1.81
Look Ahead	TH	0.05	1.82
Look Ahead	TH	0.10	1.81
Look Ahead	TH	0.15	1.83
Look Ahead	TH	0.25	1.84
Look Ahead	TH	0.50	1.87
Look Ahead	TH	0.75	1.91
Look Ahead	TH	1.00	1.95

TABLE IIA.8 Network Delay With Asymmetric Service Rates
($\lambda_{i1}=\lambda_{i2}=0$ packets/sec, $\lambda_{i3}=3.5$ packets/sec,
$\mu_{11}=\mu_{22}=\mu_{31}=\mu_{42}=5$ packets/sec,
$\mu_{12}=\mu_{21}=\mu_{32}=\mu_{41}=3$ packets/sec, $\rho=0.875$)

Routing Strategy			Network Delay (seconds)
Local		PSD	2.23
		-0.25	2.07
Look		0	2.03
	TH	0.25	2.04
Ahead		0.75	2.15
		1.00	2.22

better than the PSD in terms of average network delay; and the threshold value set in the look ahead scheme is insensitive to the network delay peformance over a wide range.

4. Conclusion

We have studied a dynamic routing scheme for packet switching networks. Its network delay performance is superior to that of fixed routing. We have also developed and evaluated an analytic technique for evaluating adaptive routing performance. Extensive simulations confirm these results. A locally adaptive scheme performs almost as well as one that has more information.

IIB. The Effect of Imperfect Acknowledgments on the Throughput of Packet Radio Networks.

1. Introduction

In this section we extend our previous work on throughput in multihop packet radio networks to include the effects of passive acknowledgments not being heard. We also evaluate the performance of a new protocol which greatly increases the probability of hearing these acknowledgments.

All terminals (nodes) operate under a CSMA protocol. Node i will transmit a scheduled packet to j, a neighbor of i, if i and all i's neighbors are idle. Denote the set of all neighbors of i, including i itself, as N_i. The transmission will be heard by j, assuming perfect capture, if all nodes in N_j are also idle. Let P(A) be the probability that all nodes in the set A are idle. Let $G_{i,j}$ be the rate of scheduled traffic from i to j. Let $S_{i,j}$ be the rate of successfully transmitted packets from i to j. Then since the probability of transmission is $P(N_i)$, and the probability of successful reception is $P(N_j|N_i)$, we have

$$S_{i,j} = G_{i,j}P(N_j|N_i)P(N_i) = G_{i,j}P(N_i,N_j) \qquad \text{(IIB-1)}$$

When j transmits the packet to the next node in the route, i can hear the transmission, and thus receive a passive acknowledgment, if all of its neighbors (except j) are idle. If it does not hear the transmission, then after a suitable time-out period it retransmits the packet. Previously we assumed all passive acknowledgments were heard. Here we see that a passive acknowledgment is heard with probability $P(N_i|N_j)$. We assume that the time-out period is selected so that all nodes have the opportunity of hearing exactly one transmission. If they fail to hear it, the packet is retransmitted. This is repeated until an acknowledgment is heard. The assumption that exactly one transmission may be heard in a time-out period is equivalent to assuming that the same time out period is used for all nodes and that it is long enough to insure one transmission.

The rate $S_{i,j}$ in equation (IIB-1) includes repeated packets, because of failure to hear acknowledgments. Let $S^*_{i,j}$ be the rate of packets for which passive acknowledgments are heard. Then

$$S^*_{i,j} = q_{i,j} S_{i,j} \tag{IIB-2}$$

where $$q_{i,j} = P(N_i | N_j) = \frac{P(N_i, N_j)}{P(N_j)} \quad . \tag{IIB-3}$$

We have already shown how to solve sets of equations like (IIB-1) for the $G_{i,j}$ as a function of the $S_{i,j}$, and found the maximum possible $S_{i,j}$ which we defined as the throughput. For example, in a chain, all $S_{i,j} = S$ for $j = i$ and $i - 1$. Here we see how to modify this procedure to account for passive acknowledgments. The $S^*_{i,j}$ are determined from the offered traffic and the routing.

We identify two situations. Packets are retransmitted if they are unsuccessfully transmitted or if the passive acknowledgment is not heard. In the latter case, a duplicate packet is sent. If the receiving node recognizes the duplicate packet and deletes it, then $S^*_{i,j}$ is the desired rate of traffic between i and j. If a simpler node does not have this capability, then duplicate packets are retransmitted, and after several hops many duplicates exist. In this latter case, $S^*_{i,j}$ includes these duplicate packets and must be related to the desired rate of traffic. We will first consider a chain.

2. A chain network

Consider the chain shown in figure IIB-1 consisting of n nodes. Assume that duplicate packets are not detected.

```
0————————0————————0— . . . . . —0————————0
1        2        3             n-1      n
```

Figure IIB-1. A chain network

We assume that node 1 and n wish to send S packets per second to each other. All other nodes are just repeaters. When node 2 relays a packet to node 3, node 1 must be idle, and all passive acknowledgments are always heard. Thus $q_{1,2} = 1$ and,

$$S^*_{1,2} = S_{1,2} = S \tag{IIB-4}$$

At the other end, node n does not repeat the packets, so passive acknowledgments are not possible. However every transmission must be successful since there are no interfering nodes. Thus

$$S^*_{n-1,n} = S_{n-1,n} \tag{IIB-5}$$

Node i attempts to transmit all $S_{i-1,i}$ successful packets to i+1. Thus

$$S^*_{i,i+1} = S_{i-1,i} \quad , \quad i = 2, \ldots, n-1 \tag{IIB-6}$$

Similarly, in the reverse direction, we have

$$S^*_{n,n-1} = S_{n,n-1} = S$$

$$S^*_{2,1} = S_{2,1} \tag{IIB-7}$$

$$\text{and} \quad S^*_{i+1,i} = S_{i,i-1} \quad , \quad i = 2, \ldots, n-1$$

Equations (IIB-4 to 7) relate $S_{i,j}$ and $S_{i,j}^*$ to the desired traffic, S. Note, by symmetry, we obtain relations like equations (IIB-7) by replacing i and j by n-i+1 and n-j+1, respectively.

As an example we evaluate the throughput of a four node chain. First we assume perfect reception of acknowledgments and use only equation (IIB-1) with all $S_{i,j}=S$. Then

$$S = G_{12}P(1,2,3) = G_{23}P(1,2,3,4) = G_{21}P(1,2,3) \qquad \text{(IIB-8)}$$

We let $G_{12} = G_1$, $G_{21} + G_{23} = G_2$, and note by symmetry that $G_3 = G_2$ $G_4 = G_1$, and $P(1,2,3) = P(2,3,4)$. From our previous work we have

$$P(1,2,3) = \frac{1+G_4}{D} = \frac{1+G_1}{D}$$

$$P(1,2,3,4) = \frac{1}{D} \qquad \text{(IIB-9)}$$

$$D = 1+2G_1+2G_2+2G_1G_2+G_1^2 = (1+G_1)^2+2G_2(1+G_1)$$

Thus we get

$$G_2 = G_1 + G_1(1+G_1) = G_1(2+G_1)$$

$$S = \frac{G_1}{1+G_1+2G_2} = \frac{G_1}{(1+G_1)+2G_1(2+G_1)} \qquad \text{(IIB-10)}$$

We can solve equation (IIB-10) for the maximum value of S, the throughput. We obtain $S_{max} = .128$ when $G_1 = .71$.

To include the effects of passive acknowledgments we use equations (IIB-2 to 6) to obtain for arbitrary length chains,

$$S_{12} = S$$

$$S_{i,i+1} = \frac{S}{q_{2,3}q_{3,4}\cdots q_{i,i+1}}, \quad i=2,\ldots,n-2 \qquad \text{(IIB-11)}$$

$$S_{n-1,n} = S_{n-2,n-1}$$

Also,

$$S_{12}^* = S_{23}^* = S$$

$$S_{i,i+1}^* = \frac{S}{q_{2,3}\cdots q_{i-1,i}} \qquad \text{(IIB-12)}$$

Returning to the example for n=4 we use equations (IIB-11) and (IIB-3) in (IIB-1) to obtain

$$S = G_{12}P1,2,3) = q_{23}G_{23}P(1,2,3,4) = q_{23}G_{21}P(1,2,3)$$

$$= G_{12}P(1,2,3) = G_{23}\frac{[P(1,2,3,4)]^2}{P(1,2,3)} = G_{21}P1,2,3,4) \qquad \text{(IIB-13)}$$

or $\quad G_2 = G_1(1+G_1)+G_1(1+G_1)^2 = G_1(1+G_1)(2+G_1)$

Thus

$$S = \frac{G_1}{1+G_1+2G_2} = \frac{G_1}{(1+G_1)+2G_1(1+G_1)(2+G_1)} \qquad \text{(IIB-14)}$$

The throughput here is S_{max} = .098 when G_1 = .37, a reduction of 23%.

In Table IIB-1 we compare the effects of passive acknowledgments for various length chains. We see that as the chain becomes long the throughput vanishes. Also shown in that table is the

Table IIB-1. Effect of Passive Acknowledgments

Length of Chain	Throughput: Perfect Acknowledgments	Throughput: Duplicates Detected	Throughput: Duplicates Not Detected
4	.128	.106	.098
5	.111	.083	.069
6	.102	.072	.053
7	.097	.066	.044
8	.094	.063	.038
9	.092	.061	.034
10	.091	.060	.031
∞	.086	.057	0

throughput for chains when duplicate packets are detected and not transmitted to the next node. We analyze this case now.

If duplicate packets are detected and not transmitted further, then $S_{i,j}{}^*$ is just the desired traffic to be sent from i to j. As before we find $S_{i,j}{}^*$ from the offered traffic and the routing. For example, in a chain $S^*_{i,i+1} = S^*_{i+1,i} = S$ for $i = 1,\ldots,n-1$. Thus equations (IIB-1 to 3) combine to give

$$S_{i,j}{}^* = G_{i,j} \frac{[P(N_i, N_j]^2}{P(N_j)} \tag{IIB-16}$$

except at terminal nodes where no further transmission is necessary and $q_{ij} = 1$. This can be solved as before to yield the $G_{i,j}$ as a function of the $S_{i,j}{}^*$ and then the maximum $S_{i,j}{}^*$ or throughput can be found. Details of the procedure have already been reported.

We present the analysis for the four node chain analyzed above. Here $S_{i,i+1}^* = S_{i+1,i}^* = S$ for $i = 1,2,3$. Again we use symmetry to obtain

$$S = G_1 \frac{(1+G_1)}{D} = G_{23} \frac{1}{(1+G_1)D} = G_{21} \frac{1+G_1}{D}$$
$$D = (1+G_1)^2 + 2G_2(1+G_1) \tag{IIB-17}$$

as before.

But $G_2 = G_1 + G_1(1+G_1)^2$, so

$$S = \frac{G_1}{(1+G_1)+2G_1[1+(1+G_1)^2]} \tag{IIB-18}$$

Solving we get $S_{max} = .106$ when $G_1 = .42$

The throughput for various length chains are given in Table IIB-1. We see here that there is a 34% reduction in throughput for long chains when the effects of passive acknowledgments are included and duplicate packets are detected and not transmitted further.

We can obtain an asymptotic solution for the infinitely long chain. First assume perfect acknowledgments. We have already discussed this in a previous report, but present a different analysis here. We assume $S_{i,i+1} = S_{i+1,i} = S$ and $G_{i,i+1} = G_{i+1,i} = \frac{1}{2} G$. Then

$$S = \frac{1}{2} G \; P[i,i+1,i+2,i+3]$$

$$= \frac{1}{2} G \; \frac{SP[1,\ldots,i-1,i+4,\ldots,n]}{SP[1,\ldots,n]} \qquad \text{(IIB-19)}$$

Here SP[ ] is the sum of products notation previously introduced. We have

$$SP[1,\ldots,i-1,i+4,\ldots,n] = SP[1,\ldots,i-1]SP[i+4,\ldots,n]$$

and

$$SP[1,\ldots,n]=GSP[1,\ldots,i-1]SP[i+3,\ldots,n]+SP[1,\ldots,i]SP[i+2,\ldots,n]$$

Thus $S = G/2X$ where

$$X = G\,\frac{SP[i+3,\ldots,n]}{SP[i+4,\ldots,n]} + \frac{SP[1,\ldots,i]}{[1,\ldots,i-1]} \; \frac{SP[i+2,\ldots,n]}{SP[i+4,\ldots,n]} \qquad \text{(IIB-20)}$$

Furthermore,

$$SP[i+2,\ldots,n] = GSP[i+4,\ldots,n] + SP[i+3,\ldots,n].$$

Therefore if we let

$$Q = \frac{SP[1,\ldots,i]}{SP[1,\ldots,i-1]} = \frac{SP[i,\ldots,n]}{SP[i+1,\ldots,n]} \; . \qquad \text{(IIB-21)}$$

then asymptotically for large i and n, we obtain

$$X = GQ + Q(G+Q) = 2GQ + Q^2$$

or

$$S = \frac{G}{2(Q^2+2GQ)} \tag{IIB-22}$$

We can solve for Q by noting that

$$SP[1,\ldots,i] = GSP[1,\ldots,i-2] + SP[1,\ldots,i-1].$$

Thus

$$Q = G\frac{1}{Q} + 1$$

or

$$G = Q(Q-1) \tag{IIB-23}$$

Note that as G goes from 0 to ∞, Q goes from 1 to ∞. We finally obtain

$$S = \frac{Q-1}{2Q(2Q-1)} \tag{IIB-24}$$

This has a maximum of $S_{max} = .086$ at $Q = 1+1/\sqrt{2}$ or $G = (1+\sqrt{2})/2$.

If the effect of passive acknowledgments is included in the preceding analysis we get

$$S = \frac{1}{2}G \quad \frac{P[i,i+1,i+2,i+3]^2}{P[i+1,i+2,i+3]} \tag{IIB-25}$$

But

$$\frac{P[i,i+1,i+2,i+3]}{P[i+1,i+2,i+3]} = \frac{SP[1,\ldots,i-1]}{SP[1,\ldots,i]} = \frac{1}{Q} \quad . \tag{IIB-26}$$

Thus

$$S = \frac{Q-1}{2Q^2(2Q-1)} \tag{IIB-27}$$

This has a maximum $S_{max} = .057$ when $Q = 1.4$ and $G = .56$.

3. Star Networks

We next analyze the effect of passive acknowledgments on throughput in star networks. We will consider two topologies - one where all legs are unconnected from each other, the second where the nodes closest to the center are fully connected. We have studied these networks in our previous work and assumed there that all passive acknowledgments were heard. We assume further here that duplicate packets are detected and not retransmitted. See Figure IIB-2 for the configuration of the star netwrrk with zero connectivity, L legs, and K nodes in each leg. For full connectivity the nodes 0,1,K+1,2K+1,...,(L-1)K+1 are fully connected.

The equations used in the previous section are still valid here. For j, a neighbor of i,

$$S_{i,j} = G_{i,j}\, P(N_i,N_j) \tag{IIB-1}$$

$$S_{i,j}^* = q_{i,j}\, S_{i,j} \tag{IIB-2}$$

$$q_{i,j} = P(N_i,N_j)/P(N_j) \tag{IIB-3}$$

When duplicate packets are detected, all $S_{i,j}^* = S$, the throughput in each leg. The end conditions are

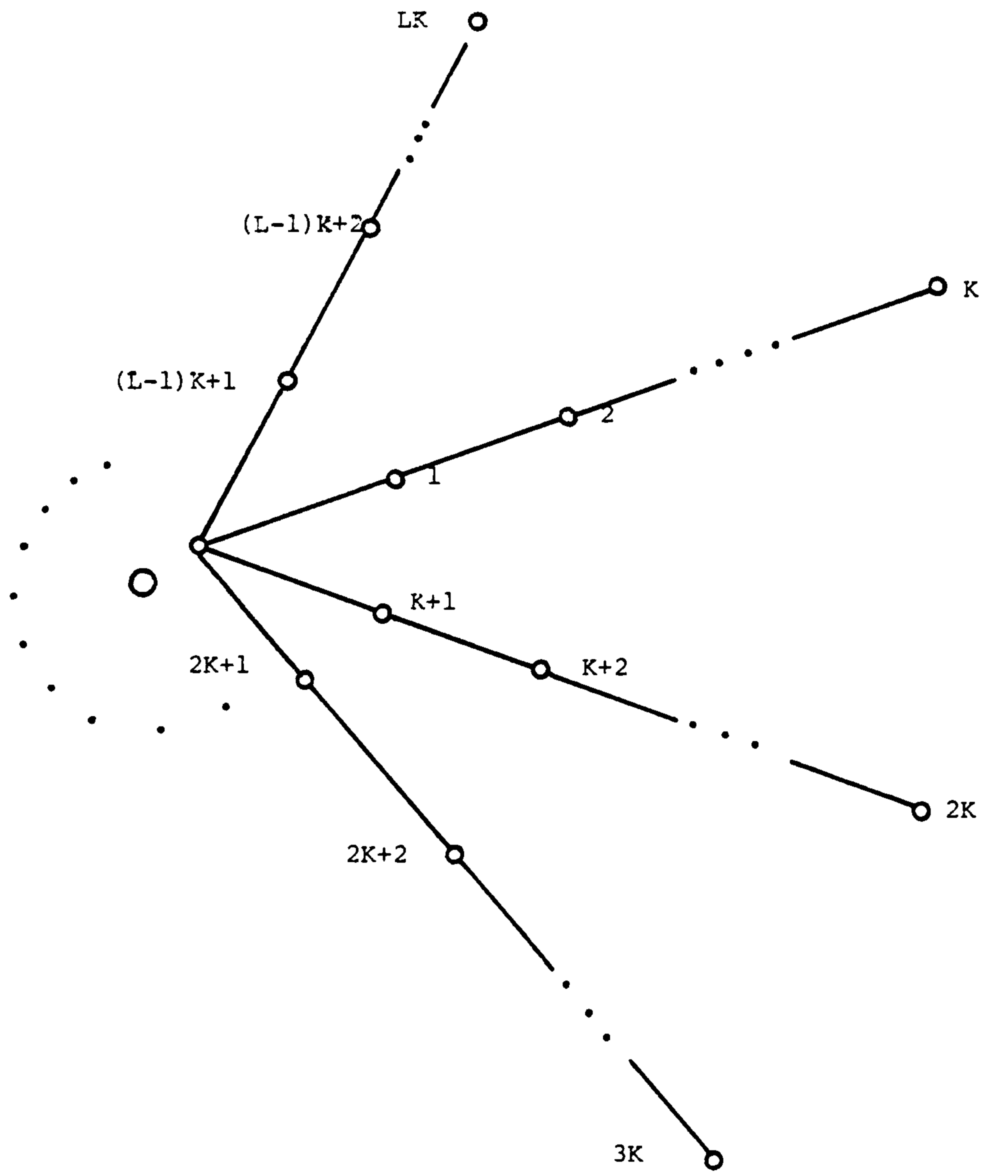
$$q_{\ell K+L-1,(\ell+1)K} = 1 \quad , \quad \ell = 0,\ldots,L-1 \tag{IIB-28}$$

since transmissions to the end nodes are always heard and passive acknowledgments are not sent. This can be found from equation (IIB-3). We further assume that

$$q_{\ell K+1,0} = 1 \quad , \quad \ell = 0,\ldots,L-1 \tag{IIB-29}$$

Transmissions to the center node are not always heard. Retransmissions due to collisions are included in our analysis. However since

Figure IIB-2: A star network with L legs and zero connectivity.



the center node does not retransmit, passive acknowledgments are not available. We assume a perfect end-to-end acknowledgment is operating.

Equations (IIB-1 to 3) have been solved with perfect passive acknowledgments and with imperfect passive acknowledgments and duplicate packet detection. Two topologies have been studied - zero connectivity and full connectivity around the center. The resultant maximum throughputs per leg are shown in Table IIB-2 and 3. The maximum throughput from the center node is given in Table IIB-4. We see from these results that there is again a 34% reduction in throughput for long legs and zero connectivity. For full connectivity we find a 43 to 46% reduction in throughput due to the failure to hear passive acknowledgments.

From our previous work we found that except for short legs, the fully connected topology is better when passive acknowledgments are not considered. This is because inward transmissions from neighbors of the center are not collided with. With zero connectivity passive acknowledgments for transmissions from the first ring to the second are interfered with only by the center node. With full connectivity all other first ring nodes add to this interference. Thus the first ring will have to retransmit more. In Table IIB-4 we compare the maximum throughputs from the center node for full and zero connectivity and see that indeed zero connectivity is better.

Table IIB-2: Effects of Passive Acknowledgements on Star Networks. Zero Connectivity (Number of Legs, L=5) - Duplicates Detected

Number of Hops per Leg, K	Maximum throughput per Leg, S_{max} Perfect Acknowledgments	Imperfect Acknowledgments
1	.058	.058
2	.045	.033
3	.044	.0290
4	.044	.0286
5	.044	.0286
6	.044	.0286

Table IIB-3: Effects of Passive Acknowledgements on Star Networks. Full Connectivity - Duplicates Detected

Number of Legs, L	Number of Hops per leg, K	Maximum throughput per Leg, S_{max} Perfect Acknowledgments	Imperfect Acknowledgments
6	3	.04	.0233
	4	.04	.0230
	≥ 5	.04	.0230
9	3	.0286	.0155
	4	.0280	.0154
	≥ 5	.0280	.0154

Table IIB-4: Effect of Passive Acknowledgments on Star Networks. Maximum Throughput from the Center.

Number of Legs, L	Number of Hops per leg, K	Connectivity	Maximum throughput per Leg, S_{max} Perfect Acknowledgments	Imperfect Acknowledgments
5	2	Zero	.225	.165
	≥ 3		.220	.145
6	3	Full	.240	.140
	≥ 4		.240	.138
9	3	Full	.257	.140
	≥ 4		.252	.139

4. A new protocol for acknowledgments

In previous sections we studied the effects of passive acknowledgments on the maximum obtainable throughput for various topologies. We found that the throughput decreased by 34 to 46%. Disadvantages of the protocol studied are:

1) each node has only one chance to hear acknowledgments from its neighbors,
2) failure to hear passive acknowledgments causes increased channel traffic since packets must be repeated,
3) the next node must recognize duplicate packets or performance will be drastically worse, and
4) a node must wait a time-out period before retransmitting, thus incurring further delays.

We propose the following protocol. A node in general has many packets to transmit, in addition to repeating a packet just received. These packets are intended for all its neighbors. The old protocol requires one to wait until that particular packet is repeated. Here we enlarge the header of every packet so that it includes acknowledgment information for all recently received packets. Thus a "passive" acknowledgment can be received on the next packet transmitted. The acknowledgment for a particular packet can be included in the next several packets, so there are several attempts to hear the acknowledgment. At the expense of increased header and more logic in the nodes, the effect of passive acknowledgments can be made negligible.

We describe the protocol more carefully. A node i sends a packet to one of its neighbors, node j. This packet, if heard, is to be relayed to a neighbor of j, say k. Node j has other neighbors. The next m transmissions, after receipt of the packet in question, from node j to any of its neighbors includes in the header a specific acknowledgment for the packet. Node i is listening and has m chances to hear the acknowledgment. Node i sets a time-out period, which is the expected time for node j to transmit m times. Note that the acknowledgment is in general received (or the time out period reached) before the packet is actually relayed. A preliminary analysis of this protocol indicates that for m = 3 or 4 the effect of passive acknowledgments can be reduced drastically for many topologies. Full details will be presented in the next report.

IIC. Switching For a Large Number of Users With Short Packets and a Tight Time Delay Constraint.

1. Introduction

One of the main areas of our research is to investigate new communications techniques within networks. The motivation for this it two-fold. First, to take advantage of the vast increases in processing power and storage now available in networks. Second, packet switching has been found to perform well for certain classes of problems but not for all. Other techniques are better suitable for file transfers, facsimile, integrated data and voice, very short packets, telemetered data, etc. For very long messages some form of circuit switching is preferable. For non-bursty traffic some form of dedicated assignment is preferable. Finally, for very short messages the overhead in packet switching becomes burdensome.

We concentrate here on the latter problem. In this report we will analyze the performance of packet switching techniques and develop optimum packetizing strategies for this problem. In subsequent reports we will develop and analyze more efficient communication schemes.

The problem we consider is that of a large number of users communicating from remote terminals to a host computer. They input characters, one at a time, in a bursty fashion - short bursts of characters with varying times between characters, followed by long pauses between bursts. The echoing of characters is performed by the host computer. The round trip delay from the time a character is typed until it is received must be short or further input is impeded.

One solution (a common one) is to form a packet for each character. Each packet must contain a header which includes synchronization information, addresses of the large number of users, error checking, etc. Thus the overhead is excessive. Longer packets reduce the overhead but at the expense of packetization delay. We will develop and analyze optimum packet switching strategies as a prelude to investigating other switching techniques.

In our model we consider all users connected to a concentrator which is then connected to the host computer. Except for the echoing we ignore the return messages from the host to the users. The total delay experienced includes packetizing delay, queuing delay in the concentrator, and transmission time over the link to the host. Other delays, such as host turnaround time, are considered either negligible or invariant to the schemes we compare.

Our goal is to find the maximum number of users that can communicate with an allowable maximum delay. To that end we analyze and compare various packetization strategies below. We also develop a packetization strategy which is optimum.

2. Model Descriptions

First consider one user. It generates characters in a bursty fashion with long idle periods. We consider time slotted in Δ second periods. (Δ is the minimum time between characters, say 1/4 second.) When the user is generating characters we assume it produces a character in a slot with probability p. With probability 1-p there is no character in that slot. Successive slots are independent. Thus the rate of generating characters, when busy, is p/Δ characters per second. (For $p = 1/4$ and $\Delta = 1/4$ this becomes one character per second.) The average time between characters is Δ/p seconds.

The characters are collected in a buffer until a packet is formed. Then a header is added to the packet, which is sent over an error-free and synchronous channel with capacity C characters per slot. The packet is received by the host, which then echoes the characters, i.e., sends them back to the user. They then appear on the user's terminal as confirmation of proper receipt.

We ignore the return message from host to user in this analysis, and consider only the inbound packets and echoed characters. We consider only the delays in packetization and queuing (at the buffer) and in transmission over the line to the host. Other delays, such as host turnaround time, are either negligible or invariant to the schemes we compare. We also ignore the effect of users switching between active and idle states.

Our model is shown in Figure IIC-1.

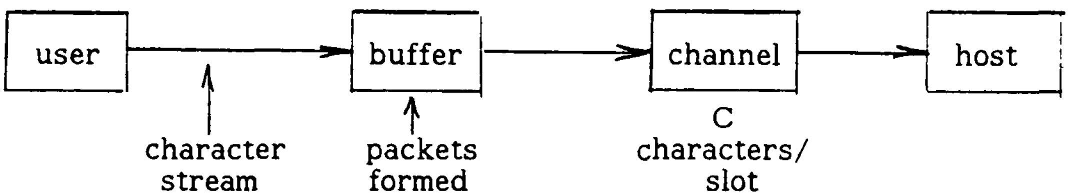


Figure IIC-1. A single user model.

We are interested in finding the number of users, such as the one described above, that can be handled by the system with tolerable delay. Assume that an any time (or for a short period of many slots) M of a larger population of identical users are actively generating characters. We want to find the maximum tolerable value for M. The size of the larger population influences only the size of that part of the header needed for addressing. This is not sensitive and will not be considered further here.

We model each of the users the same way we modeled the single user above. The M users are generating characters independently of each other. The characters are collected in the buffer and formed into separate packets for each user. They are then queued and transmitted.

This model is shown in Figure IIC-2.

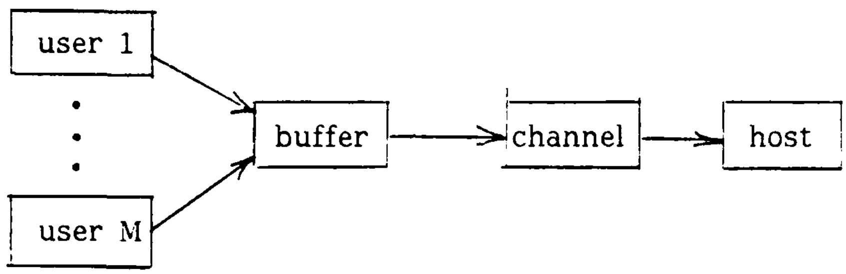


Figure IIC-2. A Multi-User Model.

3. Analysis of Fixed Length Packet Switching

We first analyze a fixed length packet switching scheme for a single user. Here the buffer waits until y characters are received and then forms a packet by added h characters as header. Thus the packet length is $x = y+h$ characters. The delay we are concerned with is given by

$$\bar{D} = \overline{PD} + \bar{W} + \bar{S} \tag{IIC-1}$$

where $\bar{D}$ is the total average delay per character,

$\overline{PD}$ is the average packetization delay per character,

$\bar{W}$ is the average waiting time for the packet, and

$\bar{S}$ is the service time or transmission time for the packet.

All times are in slots.

For fixed length packets of length x, $\bar{S} = x/C = (y+h)/C$. To find the average packetization delay note that the average time between generation of characters is $1/p$ slots. The packet begins to be formed upon generation of the first character and is considered formed when the y^{th} character is in the buffer. Thus the first character must wait on the average $(y-1)/p$ slots, the last character does not wait at all, and the k^{th} character waits $(y-k)/p$ slots. The average packetization delay per character is

$$\overline{PD} = \frac{1}{y} \sum_{k=1}^{y} \frac{y-k}{p}$$

or

$$\overline{PD} = \frac{y-1}{2p} \qquad \text{(IIC-2)}$$

Other parameters of interest are

$\bar{a}$ = average time between packets = y/p

λ = packet arrival rate = $1/\bar{a} = p/y$

ρ = line utilization = $\dfrac{\lambda(y+h)}{C} = \dfrac{p(y+h)}{yC}$.

To find the average waiting time we must first find the distribution of the time between packets. The time between characters is geometric and has moment generating function $pz/(1-qz)$, where $q=1-p$. The y successive times between packets are independent, so

A(z) = moment generating function of time between packets

$$= \left(\frac{pz}{1-qz}\right)^y \ . \qquad \text{(IIC-3)}$$

The distribution of the time between packets is given by

$$P[a=\ell] = \binom{\ell-1}{y-1} q^{\ell-y} p^{y} \ , \ \ell \geq y. \qquad \text{(IIC-4)}$$

This can be found by using $A(z) = \sum_{\ell=y}^{\infty} P[a=\ell]z^{\ell}$ and expanding equation (IIC-3) or by noting that the ℓ slots after a packet is formed must be filled by y-1 characters in the first ℓ-1 slots. The last (ℓ^{th}) slot must contain the last character.

Consider the problem of packets arriving to a queue with inter-arrival times given by equations (IIC-3) or (IIC-4). The service length is a constant given by $\bar{S} = (y+h)/C$ slots. This is a G/D/1 queue. The generating function for the waiting time distribution can be found as

$$W(z) = \frac{N(z)}{[(z-q)^{y}-p^{y}z^{x}]} \qquad \text{(IIC-5)}$$

where $$N(z) = q^{x} \sum_{i=0}^{y-1} \frac{W^{(i)}(q)}{i!} \sum_{j=0}^{y-1-i} \binom{x}{j} \frac{1}{q^{j}} R_j(z)$$

$$R_j(z) = p^{i+j}(z-q)^{y} - p^{y}(z-q)^{i+j}$$

$$x = y+h \quad \text{and} \quad q = 1-p \quad .$$

The average waiting time is given by

$$\bar{W} = W^{(1)}(1) = \frac{N''(1)-p^{y-2}y(y-1)+p^{y}x(x-1)}{2p^{y-1}(y-px)} \qquad \text{(IIC-6)}$$

where $$N''(1) = \left.\frac{d^2N(z)}{dz^2}\right|_{z=1} = q^{x} \sum_{i=0}^{y-1} \frac{W^{(i)}(q)}{i!} \left\{ \sum_{j=1}^{y-1-i} \binom{x}{j} \frac{1}{q^{j}} p^{y+i+j-2} \right.$$

$$\left. \cdot \ [y(y-1) - (i+j)(i+j-1)] \right\}$$

We can solve for $\bar{W}$ by the following method. First find the (y-1) roots of $(z-q)^{y}-p^{y}z^{x} = 0$ (from equation IIC-5) which are inside the unit circle, i.e., $|z| < 1$. For these values of z, the numerator $N(z)$ must vanish. Thus $N(z) = 0$ for these y-1 values of z. This gives y-1 linear simultaneous equations for the y values of $W^{(i)}(q)$. The last equation is obtained by noting that $z = 1$ is also a root, $R_j(1) = 0$, so $N(z) = 0$ also. But $W(1) = 1$. L'Hospital's rule can be used to find this last relation.

The average (inbound) character delay can now be found from equation (IIC-1).

4. Variable Length Packet Switching Techniques

In this section we present three different variable length packet switching techniques. In all cases we use the same user model as presented above. In the next sections we will analyze the performance of these techniques and compare them with fixed length packet switching.

Variable Length Packet Switching Technique I (VLPS I).

In this technique a packet is generated every r slots. However if any r slot interval contains no characters, then the next r slot interval starts when the next character is produced.

Variable Length Packet Switching Technique II (VLSP II).

Here a packet is generated every r slot. If any r slot interval is empty then it is skipped and the next r slot interval is considered.

The two schemes differ in the following manner. In VLSP II time is blocked into r slot units. A packet is sent at the end of the block if at least one character has been produced. In VLSP I, when a block has no characters, the next block is not formed until a new character is produced. A third scheme, in which all blocks start when the first character is produced, is not considered here.

Modified Fixed Length Packet Switching (MFLPS)

A packet is produced upon the arrival of either n or n+1 characters. The choice is made randomly. Thus, let 1-Q be the probability that the packet contains n characters. With probability Q, the packet will be formed when the next ($n+1^{st}$) character arrives. The packet, of either length, is created immediately upon the arrival of the last character. We have proven that this scheme is optimum in the sense that for a fixed average packetization delay it yields the maximum average packet length. We will describe this further in the next report. We will describe in the next section a procedure for finding the optimum values of n and Q.

The fixed length packet switching technique (FLPS) was described and analyzed in the previous section.

5. Analysis of Variable Length Packet Switching Techniques

Variable Length Packet Switching Technique I

The following are the results of our analysis. Most of the parameters have already been defined in section 3.

$\bar{y}$ = average packet length = $(1+q^r)rp + q^{r+1}$

$\bar{a}$ = average time between packets = $\bar{y}/p$.

λ = arrival rate of packets = $1/\bar{a}$.

$q = 1-p$

$\overline{y^2}$ = second moment of packet length = $\bar{a}p(pr+q) + q^{r+1}p(r-1)$

$x = y + h$

$\bar{x} = \bar{y} + h$

$\overline{x^2} = \overline{y^2} + 2h\bar{y} + h^2$

$\overline{PD}$ = average packetization delay

$$= \lambda \left\{ \frac{(r-1)(r-2)(1+q^r)}{2} + (r-1)\left(1 + \frac{q^r}{p}\right) \right\} \qquad \text{(IIC-7)}$$

$\bar{S}$ is the average service time per character. Although each character in a packet must wait for the entire packet to be received we consider that service time to be apportioned over all characters in the packet. A packet with y characters has a service time of $(y+h)/c$. Thus

$$\bar{S} = \frac{\overline{y(y+h)/c}}{\bar{y}} = \frac{1}{c}\,\frac{\overline{y^2}}{\bar{y}} + \frac{h}{c} \quad . \qquad \text{(IIC-7a)}$$

We consider M identical users. We approximate their waiting time at the buffer by assuming it acts as an M/G/1 queue. Thus the total average delay per character is given by

$$\bar{D} = \overline{PD} + \bar{S} + \frac{M\lambda\bar{x}^2}{2C^2(1-p)} \qquad \text{(IIC-8)}$$

where ρ = utilization = $\frac{M\lambda(\bar{y}+h)}{C} < 1$.

Equation (IIC-8) can be written so as to give M as a function of $\bar{D}$. We will find the value of the parameter r to maximize M for a fixed $\bar{D}$.

Variable Length Packet Switching Technique II.

Using the same definitions as above, the results of the analysis of this scheme are given by:

$$\bar{a} = \frac{r}{1-q^r} = 1/\lambda$$

$$\bar{y} = \bar{a}p = \frac{rp}{1-q^r}$$

$$\overline{y^2} = \frac{rp}{1-q^r}(rp+q) \qquad \text{(IIC-9)}$$

$$\overline{PD} = \frac{r-1}{2}$$

We can again use Equation IIC-8 to find $\bar{D}$ or to find the maximum M for a fixed value of $\bar{D}$.

Modified Fixed Length Packet Switching

Recall that the packet length is either n, with probability 1-Q, or n+1 with probability Q. The packets are created upon receipt of the last character. The values of n and Q are to be determined. (We have already mentioned the optimality of this technique.) The results of the analysis are:

$$\begin{aligned}
\bar{a} &= \frac{n+Q}{p} = 1/\lambda \\
\bar{y} &= n+Q \\
\overline{y^2} &= (n+Q)^2 + Q(1-Q) \\
\rho &= \frac{Mp(n+Q+h)}{C(n+Q)}
\end{aligned} \qquad \text{(IIC.10)}$$

From these results we can again get an expression for $\bar{D}$, and for M. We can then find the values of the parameters y,r,n and Q in the various schemes to maximize M for a given value of $\bar{D}$. This is done in the next section.

6. Comparison of Different Techniques

Each of the four schemes described above have been analyzed. For each scheme we found the optimum value of their parameters (y,r,n,Q respectively) so as to maximize M for a given value of $\bar{D}$. Thus we seek the maximum number of users that can be handled while providing an allowable total character delay.

We have considered an example where p = 1/4, C = 300 characters per slot, and a slot is 1/4 second long. Thus the line speed is 1200 characters per second. One source produces on the average one character per second with a peak of 4 per second. Without overhead and ignoring delay at most 1200 users can be handled. However both overhead and delay constraints reduce this. We have considered header sizes of 1 or 2 characters per packet and a maximum delay of D = 4 slots or 1 second. The results are shown in Table IIC-1.

Table IIC-1 Comparison of Different Packet Switching Techniques

Technique	Optimum Value of Design Parameter	$\bar{a}$	$\bar{y}$	$\overline{PD}$	h	M	M*
Fixed Length PS	y=2	8	2	2	1	797	800
					2	597	600
Variable Length PSI	r=8	9.1	2.27	3.61	1	821	833
					2	626	638
Variable Length PSII	r=8	8.89	2.22	3.5	1	817	827
					2	622	631
Modified Fixed Length PS	n=2,Q=0.77	11.08	2.77	3.67	1	864	881
	n=2,Q=0.79	11.16	2.79	3.70	2	679	698

We can see from the results that the modified fixed length packets switching scheme performs best. Indeed we have proven that this scheme is optimum when considering only packetization delay and service time. The variable length schemes perform about the same and better than the fixed length scheme. The value M* in the tables is that for which $\rho = 1$ and serves as an upper bound on M. As M approaches M* the waiting time dominates and becomes large.

7. Conclusions

We have analyzed four different packet switching techniques in an environment where packet switching is an inefficient technique. We considered many users generating individual characters and requiring a fast response. Small packets waste overhead, while large packets consume the response time. The optimization is very tight. We have found a modified fixed length packet switching technique which has some optimal properties and performs the best of those schemes studied.

Subsequent work will further explore the optimality of the modified fixed length packet switching technique, and explore the performance of packet switching techniques over a wide variety of situations and design parameters -- source models, time delay constraints, header sizes, etc.

We intend to focus attention on other, more sophisticated, techniques to transmit the characters from many users. Thus we seek alternatives to packet switching in these environments.

IID. Probabalistic Analysis of Algorithm Performance

The problem of finding an optimal set of repeater locations covering a given set of terminal sites, or set of potential repeater locations and a covering matrix specifying which terminal sites can be covered by each repeater location, is an important problem in the design of multihop packet radio networks. The problem in an instance of the classic set covering problem which has many other applications as well. The set covering problem is known to be NP-complete and as such it is unlikely that an algorithm will be found which can guarantee an optimal solution and also guarantee a reasonable running time, that is a runtime which grows polynomially in the number of terminals and repeaters.

The theory of nondeterministic polynomial completeness (NP-completeness) basically states that there is a large class of problems, which includes almost all optimization problems of interest in the area of network design, which have the property that a solution to one could be used to obtain a solution to all the others in a reasonable amount of time. It has been shown that one problem, the Satisfiability Problem, has the property that all problems in this class can be transformed into an instance of it. Thus, an algorithm which could obtain an optimal solution to the Satisfiability Problem, could be used to solve all the others. Techniques have been developed for proving that many other problems are NP-complete. Proofs have been given for the NP-completeness of over 200 other well-known problems.

In our previous work we reported on results obtained using a new set covering algorithm. The algorithm obtained optimal solutions for moderate sized problems with up to 300 node networks and coverage related to distance. That is, if the average matrix reflected the fact that a repeater was far more likely to cover a nearby terminal than one farther away, then the algorithm worked very well, converging to an optimal solution very quickly.

If on the other hand, the coverage matrix was random, the algorithm ran much more slowly and only problems with up to 50 terminals could be treated efficiently. In examining why the algorithm had difficulty treating problems with random data, we observed that the difficulty stemmed from the large number of optimal and near optimal solutions available. The algorithm had no difficulty finding an optimal solution but, rather, had difficulty in verifying the solution was optimal in a reasonable amount of time due to the presence of a large number of alternate solutions of comparable quality. This led us to conjecture that it might be possible to develop algorithms which had a reasonable running time and which could find optimal time or near-optimal solutions with high probability. It is known that asymptotically, as the size of the problem becomes infinite, there exists algorithms which give optimal results almost always. These results are an outgrowth of the same symmetry which makes it difficult to verify the optimality of the solutions produced by our set covering algorithm.

These results are encouraging but of no direct use in solving network design problems since real problems are of finite size and nothing was said about how fast the algorithms converge probablistically to an optimum; i.e., what the probability is of their finding the optimum, or a solution within some bound of the optimum. We thus sought to investigate what could be said about the probabalistic performance of known heuristics for problems of moderate size. We chose the set covering problem mentioned above as the first problem to investigate. The probabalistic model of this problem follows.

We are given a set $T=\{t_1, t_2, \ldots t_N\}$ of N terminals, a set $R=\{r_1, r_2, \ldots r_M\}$ of M potential repeater sites, and an N x M covering matrix, C, where c_{ij} is 1 if t_i can be covered by r_j. We seek a subset of R containing as few repeaters as possible and covering all terminals. The elements of C are chosen randomly and independently. Specifically, we assume $Pr\{c_{ij}=1\}=p$ and $Pr\{C_{ij}=0\}=1-p=q$.

We define R_j, the set of terminals covered by r_j, by:

$$R_j = \{t_i | c_{ij} = 1\}$$

Similarly, we define T_i, the set of repeaters covering t_i, by:

$$T_i = \{r_j | c_{ij} = 1\}$$

A cover of the terminals is then defined as a subset $S \subseteq R$ satisfying:

$$\bigcup_{r_j \varepsilon S} R_j = T$$

A minimum cover, S^*, is then a cover containing as few elements as possible, i.e.:

$$\bigcup_{r_j \varepsilon S^*} R_j = T$$

and $|S^*| \leq |S|$ for all S such that $\bigcup_{r_j \varepsilon S} R_j = T$.

Note that the minimal cover is not in general unique. Indeed our approach rests on the existence of a large number of minimal and near minimal covers.

In order to analyze the probabalistic behavior of a heuristic we must do two things. First, for a given set of values of N, M, and p, we must find the probability that $|S^*| = K$, that is, the probability that there exists a minimal covering containing exactly K repeaters. Let

$$P(N,M,K) = \Pr\{|S^*| = K\} \text{ for given } N,M, \text{ and } p.$$

Next we must find $P_{ALG}(N,M,K)$, defined as the probability that the algorithm will find a solution with exactly K repeaters. A figure of merit for evaluating the probabalistic behavior of the algorithm is then

$$F_{ALG} = \frac{\sum_{K=0}^{M} KP_{ALG}(N,M,K) - \sum_{K=0}^{M} KP(N,M,K)}{\sum_{K=0}^{M} KP(N,M,K)} \qquad \text{(IID.1)}$$

This is simply the relative error made by the algorithm, i.e., the difference between the avarage number of repeaters in a covering found by the algorithm and the minimum number required, normalized by the minimum number required. We believe that $\lim_{N\to\infty} F_{ALG} \to 0$ for any M, and p, and any reasonable algorithm where a reasonable algorithm is defined as one which stops when it has a cover. In particular, we believe $F_{ALG}\to 0$ for an algorithm which picks repeaters randomly until a covering is obtained.

We begin by evaluating Q(N,M,K), the probability that a randomly chosen set of K repeaters covers all N terminals. Notice that this is not the same as the probability that an algorithm picking repeaters at random will stop after K repeaters since in some cases fewer than K will also suffice and the algorithm will find them. Also, Q(N,M,K) is a cumulative distribution, that is it relates to the probability that K or fewer repeaters are required for a covering rather than exactly K. We do have, however, that

$$Q(N,M,K) \leq \sum_{L=0}^{K} P(N,M,K) \qquad \text{(IID.2)}$$

that is, the probability of a randomly chosen set of K repeaters covering all N terminals is less than or equal to the probability of the existence of a covering containing K or fewer terminals. Thus we have,

$$P(N,M,K) \geq Q(N,M,K) - Q(N,M,K-1), \; K>1,$$

that is, the difference between two successive Q's is a lower bound on the corresponding P.

Q(N,M,K) is easy to evaluate because repeaters picked randomly are picked independently of one another and our random model assumed their original characteristics are independent. The probability of a single terminal not being covered by a single repeater is $q(=1-p)$. The probability of a single terminal not being covered by any of the K repeaters is q^K. The probability of at least one repeater of K covering a given terminal is then $1-q^K$. Finally, the probability that all N terminals are covered by the K repeaters is $(1-q^K)^N$. We thus have that

$$Q(N,M,K) = (1 - q^K)^N \qquad \text{(IID.3)}$$

Note that this is not a function of M except in the trivial sense that M must be no smaller than K. Figure IID.1 shows values of Q(N,M,K) for p=.5. Figure IID.2 shows values of Q(N,M,K) - Q(N,M,K-1) for the same examples. Both show a tendency for solutions to cluster about a narrow range of K. This is encouraging in that it implies that probabalistically, simple algorithms should do well.

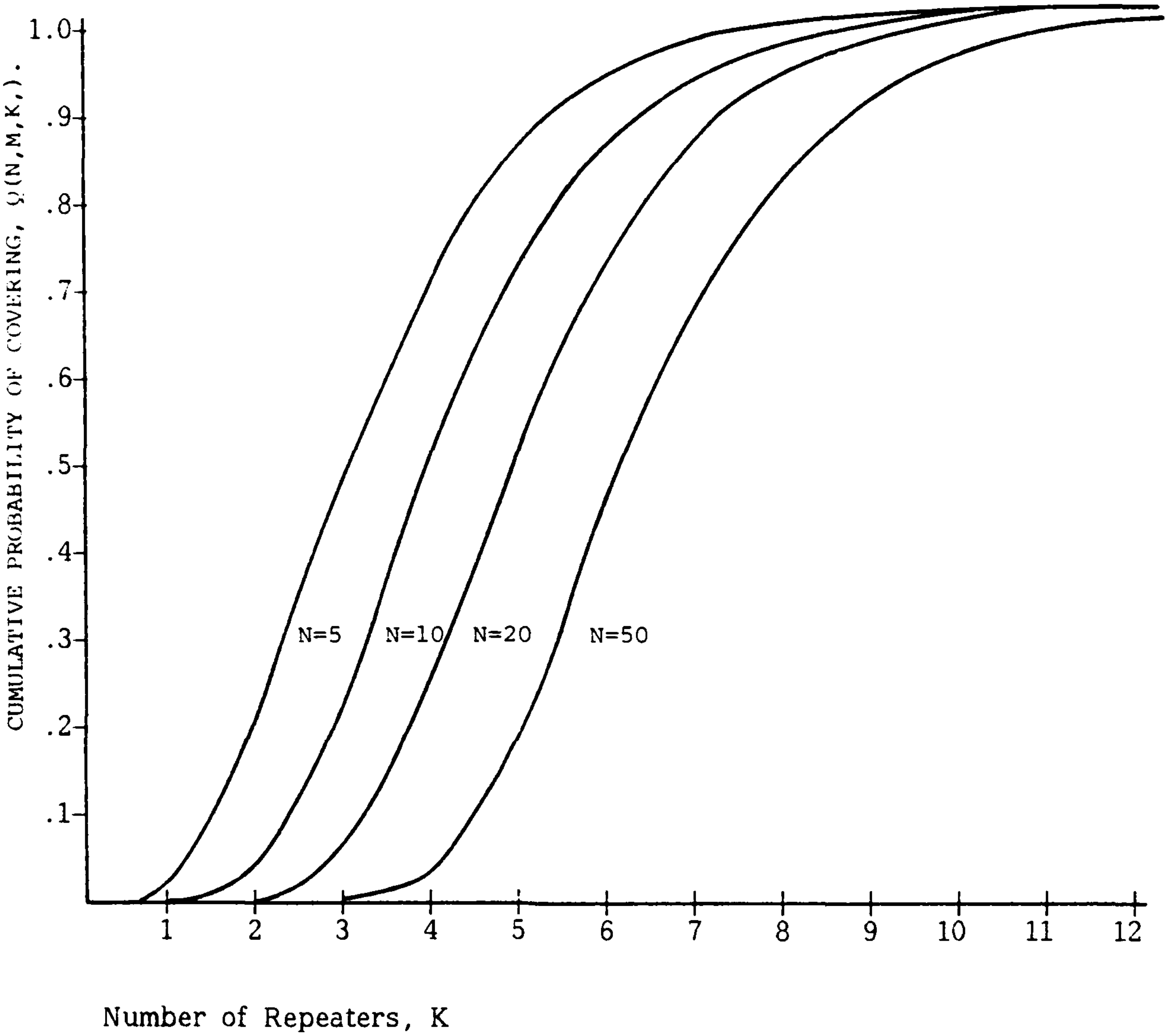


FIGURE IID.1 CUMULATIVE PROBABILITY OF A COVERING FOR RANDOM ALGORITHM (p=0.5).

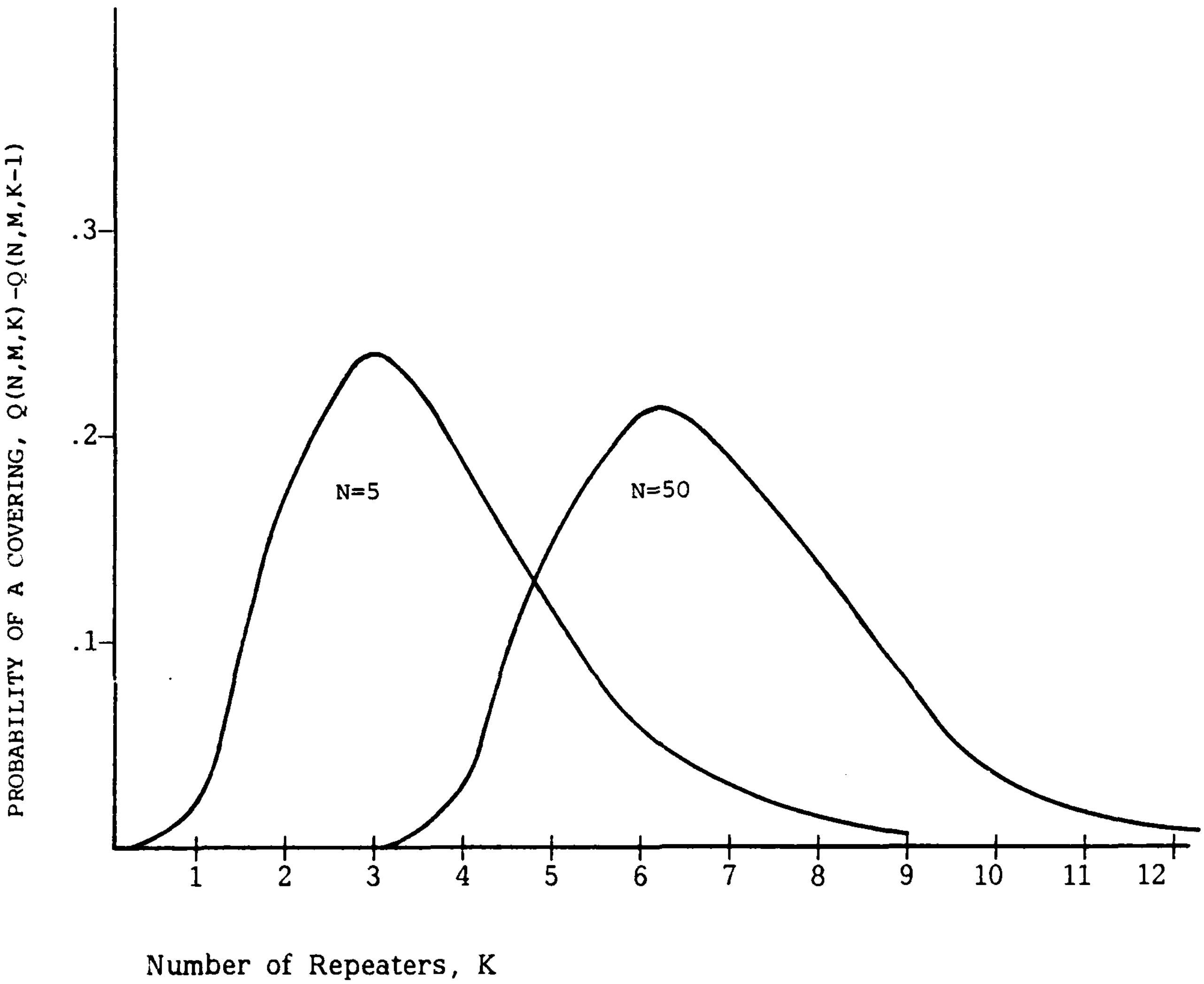


FIGURE IID.2 PROBABILITY OF A COVERING FOR RANDOM ALGORITHM (p=0.5).

As yet, we do not have an exact analytic expression for P(N,M,K). We can get a tighter lower bound on it by evaluating the distribution of solutions from increasingly sophisticated algorithms. A simple minded algorithm would be to pick repeaters randomly until one obtained a covering. The probability of obtaining a solution with K or fewer repeaters using this algorithm is $P_1(N,M,K)$ given by:

$$P_1(N,M,K) = \sum_{i=0}^{N} \binom{N}{i} p^i q^{N-i} P_1(N-i,\ M-1,K-1) \qquad \text{(IID.4)}$$

since the $K^{\underline{th}}$ repeater will cover i terminals with probability $\binom{N}{i}p^i q^{N-i}$ and the remaining K-1 of M-1 repeater must cover the remaining N-i terminals. Note that this is, again, not a function of M except that M must be no smaller than K. The probability of obtaining a covering with exactly K repeaters is given by $P_1(N,M,K)-P_1(N,M,K-1)$.

A somewhat more sophisticated algorithm is to consider repeaters in a random order and to select a repeater only if it improves the chances of obtaining a covering more by picking it than by not picking it. Letting $P_2(N,M,K)$ be the probability of obtaining a covering containing K or fewer repeaters using this algorithm, we have

$$P_2(N,M,K) = \sum_{i=0}^{N} \binom{N}{i} p^i q^{N-i} \{\max\ (P_2(N-i,\ M-1,K-1),\ P_2(N,M-1,K))\} \qquad \text{(IID.5)}$$

$P_2(N,M,K)$ is plotted for N=5 and N=10 in Figure IID.3. Along with it, the probability of obtaining a covering using K randomly chosen repeaters is plotted. Notice that this algorithm's performance is a substantial improvement over randomly picking repeaters. Notice also, that $P_2(N,M,K)$ is a function of M. This algorithm is very simple and has a reasonable running time even for very large N and

M. Our next objective is to compare it with an upper bound on P(N,M,K) to see if it already has converged to near optimal performance probabalistically.

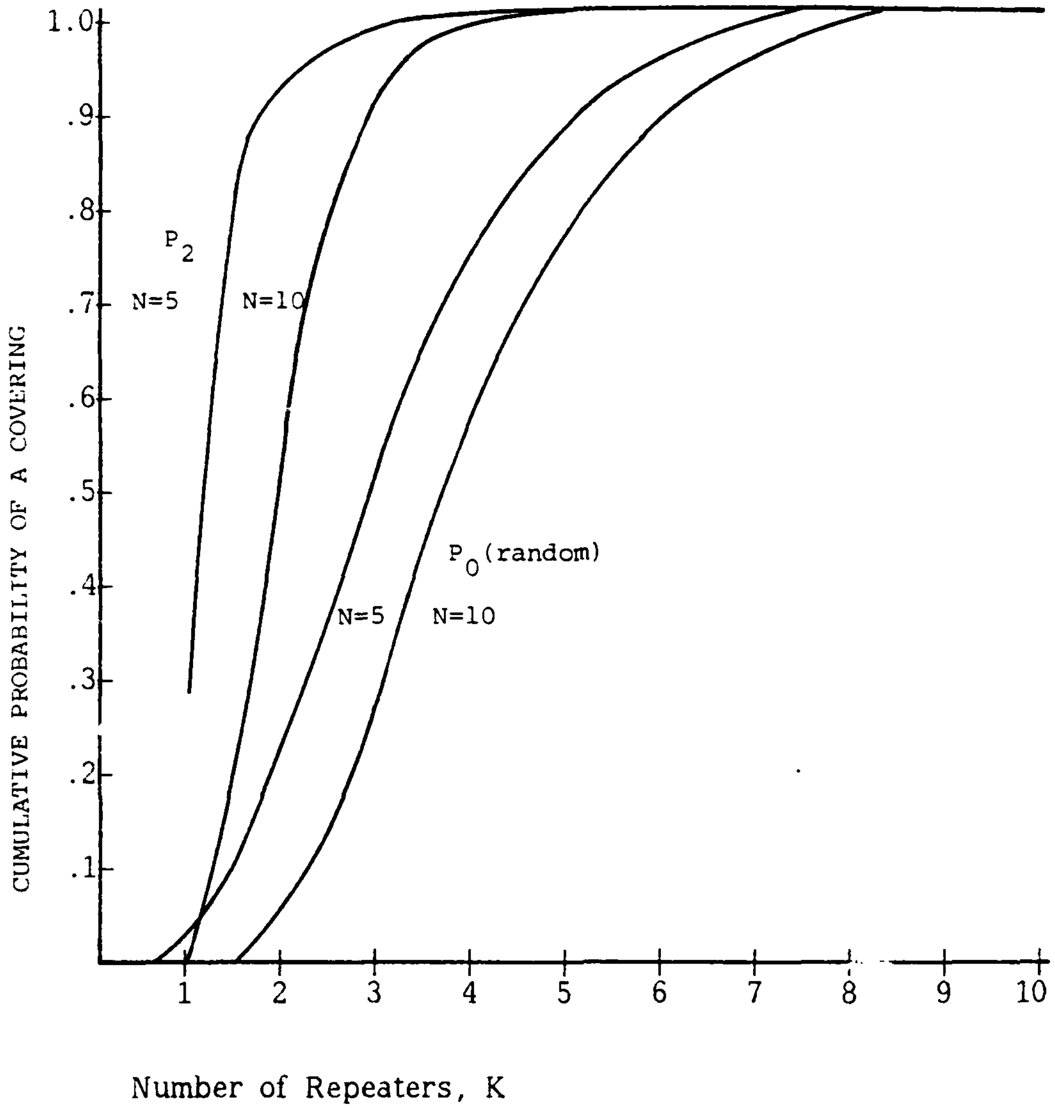


FIGURE IID.3 CUMULATIVE PROBABILITY OF A COVERING FOR MORE SOPHISTICATED ALGORITHM (p=0.5,M=10).

III. REFERENCES

Most of the previous work referred to in section II is described in our final report: October 1, 1979 to September 30, 1980, prepared for USARMY CORADCOM. Published papers that contain reference material to the discussions in section II are:

(section IIA). "A Technique for Adaptive Routing in Networks," R. Boorstyn and A. Livne, IEEE Transactions on Communications, April 1981.

(section IIB). "Throughput Analysis of Multihop Packet Radio Networks," R. Boorstyn and A. Kershenbaum, International Conference on Communications, IEEE, Seattle, June 1980.

(section IIC). "Delay and Overhead in the Encoding of Bursty Sources," R. Boorstyn and J. Hayes, International Conference on Communications, IEEE, Seattle, June 1980.

(section IID). "A Network Shortest Path Approach to the Knapsack Problem, International Conference on Circuits and Computers, IEEE, New York, October 1980.

IV. PERSONNEL

R. Boorstyn, Professor

A. Kershenbaum, Associate Professor

Students

Paul Chu (Adaptive Routine)

Veli Sahin (Multihop Packet Radio)

David Tsao (New Switching Techniques)

William Chuang (Algorithms)

Other Students

William Chen

H.K. Chao

Rachel Mendelsohn

V. ACTIVITIES

The paper "Second Order Greedy Algorithms for Centralized Teleprocessing Networks", by Profs. Boorstyn and Kershenbaum was published in the IEEE Transactions on Communications, October 1980.

Two papers by Prof. Kershenbaum, "An Algorithm for Designing Circuit Switched Networks," and "A Network Shortest Path Approach to the Knapsack Algorithm," were presented at the ICCC Conference in Rye, New York, September 1980, and published in the Proceedings.

A paper on "Analysis of Multihop Packet Radio Networks" was presented by Prof. Boorstyn at the URSI Conference, Boulder, Colorado, January 1981.

Professor Boorstyn presented two talks on "Analysis of Multihop Packet Radio Networks" at Stevens Institute and the University of Waterloo. Prof. Kershenbaum presented a talk on "Combinational Optimization" at the University of Rochester.

Paul Chu defended his Ph.D. thesis on a "Simulation Study of Dynamic Routing".

END

FILMED

9-83

DTIC