University of Waterloo

(1)

NETWORK MODELLING AND SIMULATION

By
A. Bowler
M.I. Irland
S.N. Kalra
and
J.W-N. Wong

Computer Communications Networks Group

checked 10/83

1

NETWORK MODELLING AND SIMULATION

By
A. Bowler
M.I. Irland
S.N. Kalra
and
J.W-N. Wong

Computer Communications Networks Group

University of Waterloo

TABLE OF CONTENTS

I OVERVIEW (by A.T. Bowler)

1. A Proposal for the Basic Structure of the Model

1.1 Preamble

The Department of Communications requires a simulator to assist it in judging tariff applications from the carriers. As such it requires a tool to model the financial status and constraints of the carriers.

The entire construction of such a model is beyond the expertise and capabilities of CCNG. However, the basic structure of such a model already exists in the NPPS model of the long haul telephone network. Furthermore, the TCTS data network is run by the same companies, and shares common facilities with the voice net. Therefore what should be done, is to construct a model to interface with the NPPS model.

The costing and revenue portions of the model would still be outside the capabilities of CCNG. Basic estimation of traffic patterns is possible. We can provide some facilities to assist the user in generating potential traffic patterns based on gravity models and reasonable traffic estimates for various applications. CCNG has the expertise to construct something corresponding to the remainder of the NPPS operating block, which simulates the actions of the carriers in managing the network.

1.2 The Operating Block

Functionally, this is a semi-automated network design tool. Its major inputs will be basic network topology, a traffic load, and a required performance specification. It will estimate facility capacities, such as trunk speeds and node storage, required to carry the projected traffic at the specified grade of service. It will estimate required switching capacity, and hopefully estimate a good allocation of trunk capacity. Since the data and voice networks are competing for the same transmission facilities, this portion of the model should interface with the circuit allocation module of the NPPS model.

The NPPS operating block functions in a manner similar to the above. The user specifies a traffic matrix, switching hierarchy, and transmission topology. There is a facility to help the user generate the traffic matrix, and the user has standard switching and transmission networks to use as a starting point for the network topologies he wishes to describe.

The rules used by the NPPS model are built in and essentially represent experience accumulated over the last century about how to run a telephone network. The routing rules are fixed. A good rule of thumb is known for the relative capacities of regular and high usage links. There is a known formula for the amount of excess capacity required for a given grade of service.

1.3 The Network Model

The situation for the data network is not as well understood. Internal routing policies have not been disclosed by the carriers, probably because they are not convinced that the best solution is known yet. Capacity versus performance functions are not known, and there are a larger number of parameters to vary. For these reasons it will be necessary to develop a model of the network that can be appropriately instrumented to measure performance. We do not propose any specific implementation here, but will make several comments about probable features of such a model.

Essentially this model should accept as inputs a traffic load, and a network configuration as well as routing rules and protocol descriptions. It would output detailed performance measures such as delay, loss and blocking, as well as queue lengths buffer usage, or any other measure of interest to the network analyst.

The model would be used to derive information corresponding to the formulae known for the voice network. This information would be expressed in a functional relationship (analytic formulae or tables), and would be used by the operating block. To derive this information will require a series of runs of the model with a wide range of input parameters.

This model could also be used to answer some questions about the immediate impact on a fixed network configuration of trunk loss, load changes such as the addition of a new application, and situations analogous to Mother's Day on the voice net.

A detailed discrete simulation of the entire network including everything from the actions of each terminal to the operation of the major trunks is clearly too large to be handled practically in one run on current computing facilities. Therefore it will have to be structured as a hierarchical set of models of various portions of the network. Analytic approximation formulae will have to be developed and used where appropriate to model the functioning of parts of the network outside the particular submodel.

Some portions of the model will only be represented by analytic formulae. This will be true in cases where very good analytic characterizations are known and detailed simulation will not add to the accuracy. It will also be true when insufficient information is available on the behaviour of the real network to do a detailed simulation. For example, it may be that too little is known about actual transactions carried to justify any more effort than using the normal Poisson arrival pattern.

Portions of the network description such as protocols and routing strategies can only be input in the form of actual program code, as parameterization of such things is difficult. Therefore it will be extremely important to maintain flexibility in the design and coding of the network model. This may prove to be rather expensive since generality usually costs something in efficiency. Furthermore, the design of a good generalized interface normally requires many iterations.

Thought must also be given to the place of the unswitched digital services of Infodat and Dataroute in this model.

CCNG has a multiprocessor simulator which could be modified to incorporate some of the DATAPAC protocols. This could be be used for basic simulations of the backbone of the DATAPAC network. Preliminary simulations aimed at getting crude functional relations for the DATAPAC network could be done after modification of the code.

2. Outline of Following Sections

Part II of this report discusses users and services as the first step in the development of methods and languages for describing a network. One of the uses of this language will be to describe the inputs to the models outlined previously. This relates to subsections A(b) and A(c) in the contract.

Part III discusses the concepts of modelling hierarchical networks, and the use of analytic techniques for performance analysis. Two hierarchies are discussed. A three level hierarchy (national, regional and local) is proposed to model a Canadian network. The other hierarchy involves increasing level of detail in the specification of network operation (buffer allocation, flow control, etc.)

Part III corresponds to subsection A(a) in the contract.

Part IV discusses the modelling by simulation of a backbone packet network. It discusses previous simulators, and describes a large number of considerations to be kept in mind in the construction of a simulator. Part IV corresponds to subsection A(d) in the contract.

Appendix A gives a summary description of the NPPS model, and briefly discusses some interfacing considerations.

Appendix B is a paper by J.Wong on analytic modelling of flow control mechanisms.

Appendix C is a summary of a paper on cable communications, included at the request of the scientific officer for this contract.

II USER DESCRIPTION (by S.N. Kalra)

1. Goal

The object of this phase of the work is to develop a user-oriented descriptive mechanism to interface the user to design tools such as analysis and simulation. Towards this end we are going to briefly look at the user and user needs and also at the network services offered. We will also introduce a functional description of a data network and finally map the user needs through the network services on to the functional description. The last two aspects will be mainly considered in the future.

2. Users

A user is a person or a machine using network services and paying for the services used. Users possess different degrees of sophistication, and have other needs that vary widely, for example, volume and nature of data, reliability and availability requirements, need for point to point or switched service, closed user groups, etc. A user may also be defined in terms of the network services needed and the frequency of use, etc. It would thus seem that one can have a table of user needs and network services, and have a different table for each class of users. Armed with this set of tables the task of the designer would be considerably simplified. Alas, such a categorization not only proves to be very difficult and time consuming, but virtually impossible to implement with resources available to us. The surveys commissioned by the Department of Communications (e.g., Data Com 76 by Price Waterhouse Associates) may be able to fill this need.

For the purposes of this study we will consider different types of users without being very specific as to the needs of a particular user. We may thus consider a single gas station or small retailer as a user. We would also be able to consider a retail organization (e.g., department store) with many branches and warehousing operations, etc. as an example of the retail trade; banks, trust companies, or insurance companies as examples of the financial institutions; and hospitals, or police as examples of services at a national scale.

Each of the examples quoted above has a different perception of its needs and hence demands on the data network. Their needs differ in the total volume, and more important, also in terms of response time, security of data, availability of service, tolerance to error, and guarantee of delivery. It is thus sometimes easier to define network services that may be needed by a user or class of users than to define a user.

Another description of users or user needs is in terms of ownership and/or distribution of intelligence. A user with many remote terminals (interactive or remote job entry types) may wish to own his distribution system and lease bulk transmission capacity from the carriers. We do not know of any reliable rules of thumb to indicate the nature of or the number of such users. Our main interest in this class of users is the characterization of their needs. Intelligence is put in the terminal for some combination of the following:

1) reduction of transmission cost,
2) increased transmission capacity,
3) increased throughput,
4) increased flexibility,
5) increased reliability,
6) increased security.

The network description, therefore, should be such that it is possible to optimize the use or minimize the cost.

3. Network Services

We have seen above that a user is difficult to characterize. Network services on the other hand, are much easier to define. In stored program controlled logic machines there are two technical limitations, i.e., hardware and software, and one economic consideration, i.e., return on investment. Availability of network services is thus going to mainly be determined by the market place. There are certain services which seem to be planned by one or both the national common carriers. The more significant of these services are the following.

By the TCTS Datapac network:

1) permanent virtual circuits,

2) switched virtual circuits,

3) interface to intelligent terminals via SNAP,

4) interface to non intelligent terminals via NIMs,

5) closed user groups.

By CN CP Infoswitch network:

6) info exchange--a digital circuit switched service,

7) info call--a service apparently designed to interface non intelligent terminals to computer, and

8) info gram--a packet switched service. The TCTS offering Datapac is now available; the CNCP Infoswitch offering is expected to be available shortly. There seems to some debate between the two networks as to the merits of each offering. Much of this debate is due to each competitor trying to put its offering in a more advantageous position in the market place. An analysis of available information shows that both Datapac and Infoswitch are designed to meet the expected data switching needs for the next decade.

Though we have looked at the specifications of the data networks published by the carriers and the needs of the simulation programs, we find the generalized user to be rather elusive. We have investigated the applications (e.g., General Switched Data Network by S. N. Kalra--Technical Report TR3G50- 1-74 Bell-Northern Research) to data networks, and we always come to the conclusion that we can find a satisfactory solution to a specific problem, but a generalized user and a generalized network with unknown design tools is not a tractable problem.

During the remainder of the project we propose to consider a few special cases.

III CONCEPTS FOR NETWORK MODELLING (by J.W-N. Wong)

1. General Concepts for Network Modelling

1.1 Computer-Communication Networks

A computer-communication network is a collection of nodes connected together by a set of communication channels. It can be partitioned into two subnetworks <KLEI 76>: a communication subnetwork which provides message service, i.e., transmits messages from source to destination, and a resource subnetwork which provides computing service to remote terminals. The terminals and computing resources communicate with each other via the communication subnetwork. Thus we have in Figure 1 a general picture of a computer-communication network.

1.2 The Canadian Environment

In the Canadian environment, we might have a national network connecting together the major regions. Each region is represented by a major city. A possible configuration of this national network is shown in Figure 2. A node in this network can be considered as a switching centre to other regions in the nation. Within each region, we might have a regional network connecting together the cities in the same region. A possible configuration of such a network in the Toronto region is shown in Figure 3.

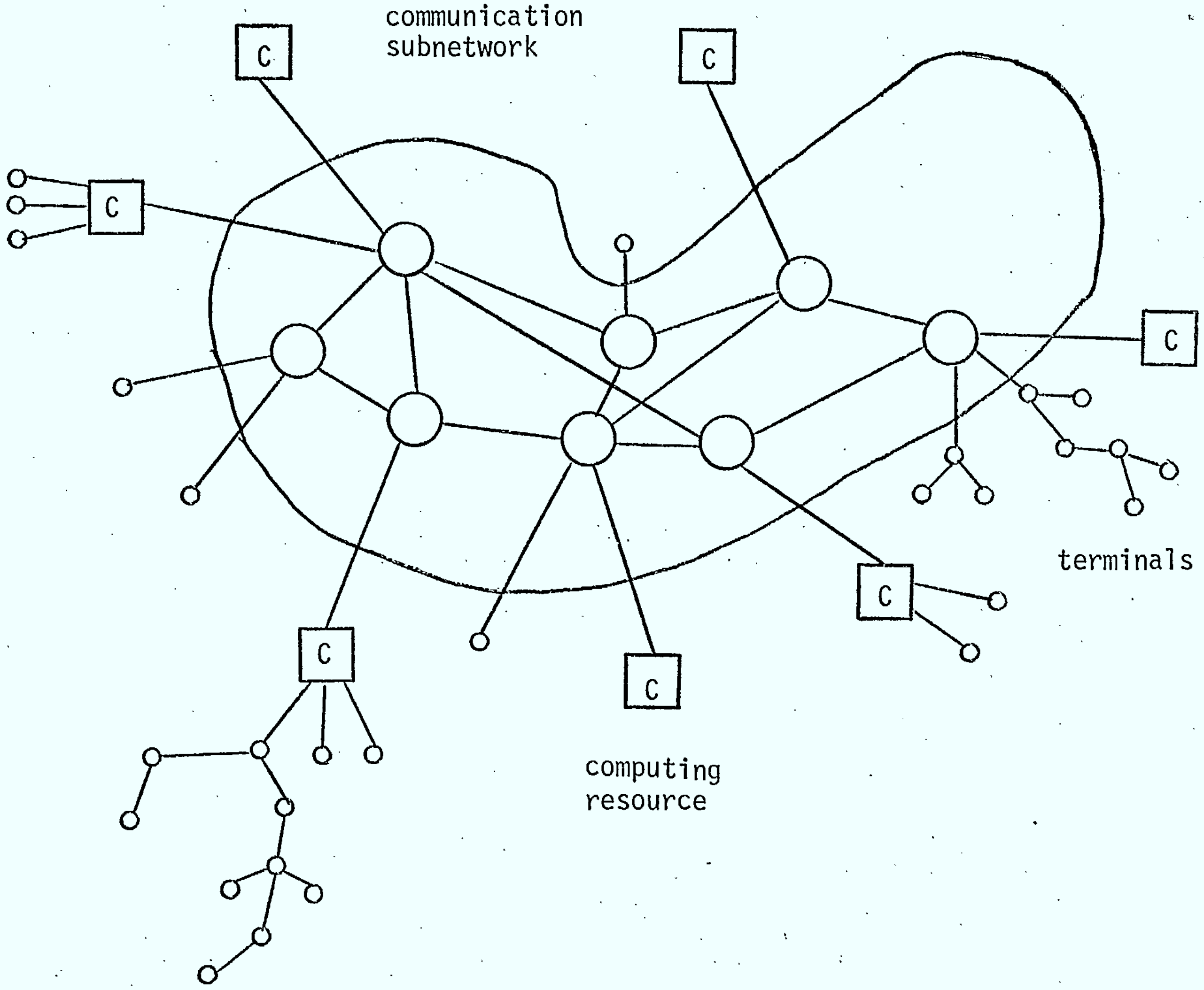


Figure 1

The Structure of a Computer-Communication Network

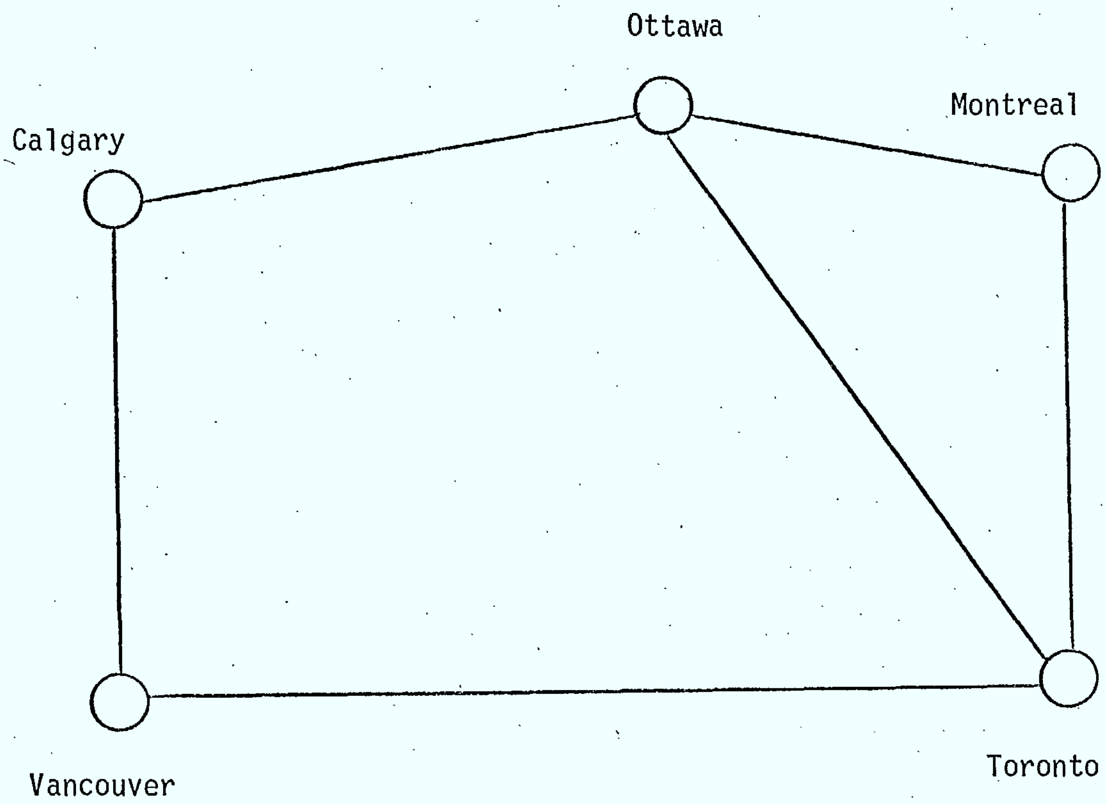


Figure 2

A Possible Configuration of the Canadian National Network

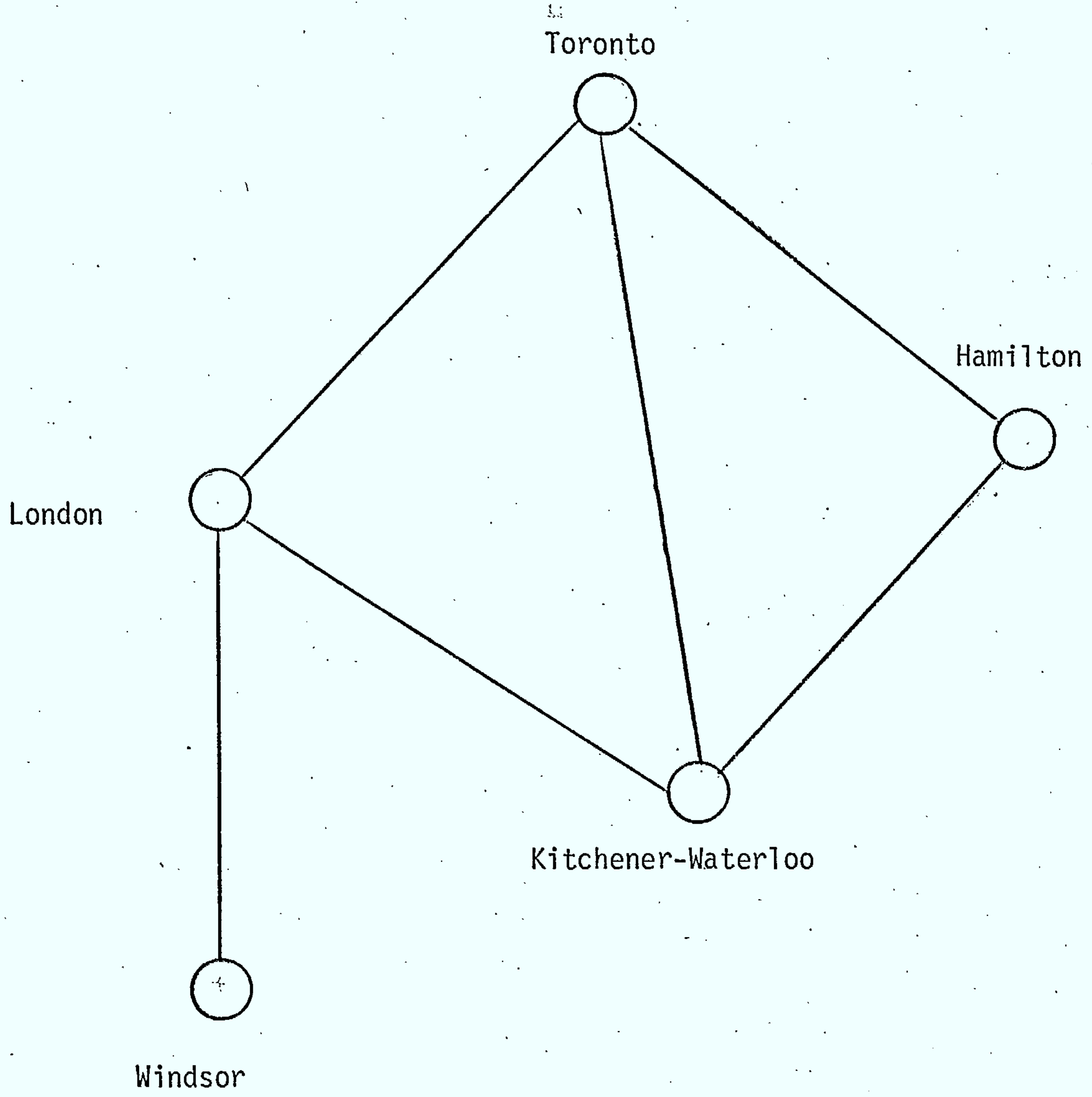


Figure 3

A Possible Configuration of the Regional Network in the Toronto Region

The regional networks have the same responsibilities as the national network in the sense that they provide message service to cities within a region. In the Canadian environment the national network and the regional networks form the communication subnetwork. We assume that each city can only belong to one regional network. This assumption implies that when a city in region A wants to communicate with another city in region B, the messages must be transmitted through the national network. Although this assumption is a little restrictive in terms of network topology, it allows us to model the communication subnetwork as a two-level hierarchy; the national network and each regional network can be analysed separately. The load on each regional network is a function of both intra-region and inter-region traffic, while the load of the national network is a function of inter-region traffic only.

Each city within a region may have its own local distribution network. This local network provides communication between terminals and computers in the same city and also entry to the regional and national networks. A general configuration of a local network is shown in Figure 4.

We have thus modelled a Canadian network as a three-level hierarchy: national network, regional networks, and local distribution networks. In Figure 5 we have displayed a hierarchical network for Canada.

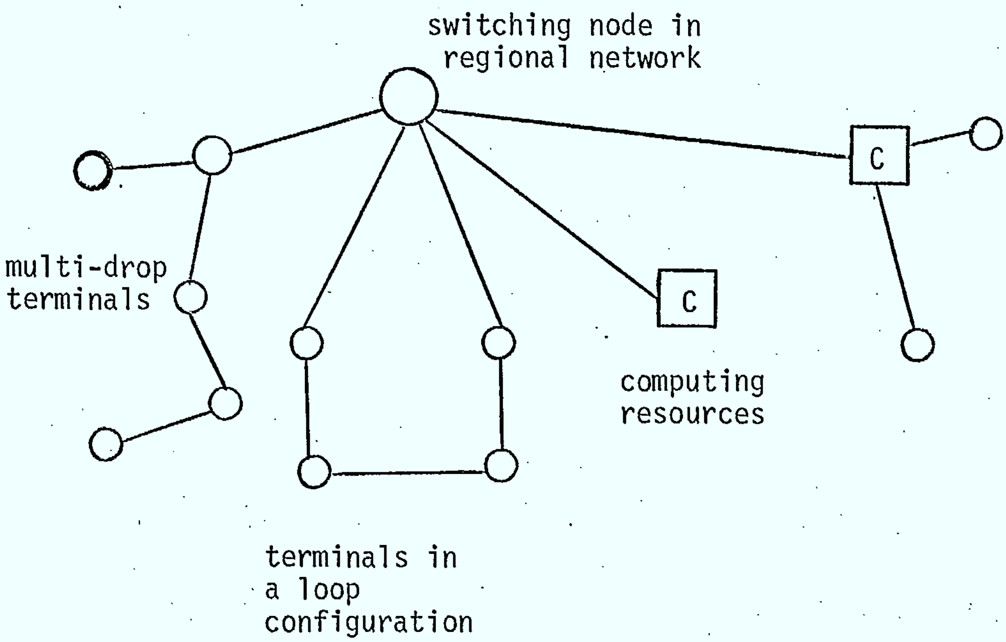


Figure 4

Local Network

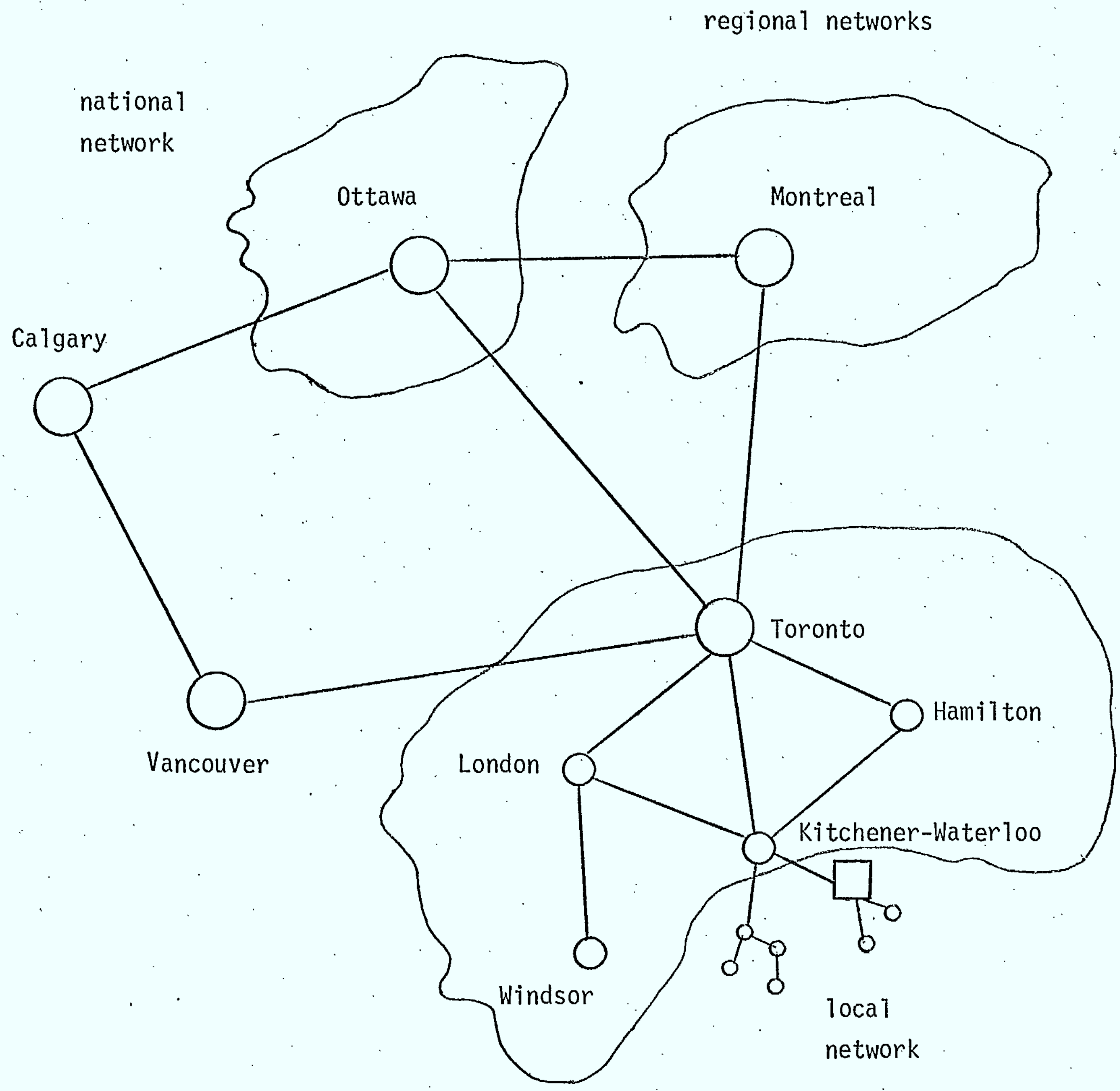


Figure 5

Hierarchical Network for Canada

2. Modelling of Communication Networks

2.1 Types of Switching Techniques

Communication networks can be divided into three types <KLEI 76>: circuit switching, message switching, and packet switching. In a circuit switching network, communication between a pair of nodes does not begin until a complete path (which may consist of a number of channels) has been established between them. This path is devoted to the two nodes until the communication between them is terminated. In message switching, messages are routed from one node to another until they reach their destination. Only one channel is used at a time for a given transmission. The messages may experience queueing delay while waiting for a free channel. Packet switching is basically the same as message switching except that messages are decomposed into smaller pieces called packets, each of which has a maximum length. This puts bounds on delay and simplifies buffer management which in turn improves switch throughput.

This report is strictly concerned with models for message-switched communication networks. Such models can be used to study the national network and each of the regional networks.

2.2 Analytic vs Simulation Models

There are two major approaches to network modelling: analysis and simulation. Simulation modelling has the advantage of being able to model networks of arbitrary complexity, but also has the disadvantage of being costly to develop and to make simulation runs. Analytic modelling based on queueing theory is attractive because it is a lot less costly to compute numerical results. The applicability of analytic modelling is often limited by the lack of solutions to complex models, but it is an extremely useful tool to gain insight into the general behaviour of computer communications networks. Moreover, any simulation model must be validated; comparison with analytic models is a powerful technique for doing so.

This part (Part III) of the report is concerned with analytic modelling. Simulation modelling will be considered in detail in Part IV.

2.3 The Basic Model

The basic model for computer networks is a queueing network model. Informally, a queueing network is a collection of servers connected together in such a way that customers (or messages) receive service from one server and move to another server according to transition probabilities. In a communication network, the servers are the switching nodes and the communication channels. The delay experienced by a message in this network is given by the sum of four components: the processing time at the switching nodes, the queueing delay for the channels, the propagation delay, and the data transfer time in the channels. The usual assumption <KLEI 64> is that the nodal processing time and the propagation delay are small in comparison to the other components and that they can be neglected in the computation of message delay.

We can thus remove the switching nodes from our model and treat each channel as a server. (Note that a full-duplex channel is considered as two channels, one in each direction.) We also assume that the channels are error free and that the queueing discipline at each channel is first-come, first-served. In our open network model, we will use M to denote the number of channels and C_i to denote the capacity of channel i.

Messages are classified according to source-destination pairs. In particular, a message is said to belong to class (s,t) if its source node is s and its destination node is t. Let R be the number of message classes. In a network with N nodes, R = N(N-1). For convenience, we assume that the message classes are numbered from 1 to R, and we use r instead of (s,t) to denote a message class.

The arrival process of class r messages from outside the network is assumed to be Poisson with mean rate $\gamma(r)$. Message lengths for all classes are assumed to have the same exponential distribution and we use $1/\mu$ to denote the mean message length. It follows from this last assumption that the mean service time at channel i is exponential with mean $1/\mu C_i$.

For the purpose of getting analytic results, Kleinrock's independence assumption <KLEI 64, KLEI 76> is required. This assumption states that each time a message enters a node, a new length is chosen from the exponential message length distribution. Kleinrock <KLEI 64> has demonstrated by simulation that this assumption gives accurate results for mean message delay.

The message routing algorithm can be fixed or random. With a fixed routing algorithm, messages belonging to a particular source-destination pair (or class) are routed through a unique path. Random routing allows alternate paths between each pair of nodes. For our discussion we will use the simple case of fixed routing. The analytic results can easily be generalized to include random routing. Finally, we assume that factors like flow control and acknowledgement of messages are not included in our model, and that there is enough buffer space in each switching node so that messages routed to a certain node are not blocked from entering that node.

We now summarize the assumptions used in our basic model:

1) M servers - each representing a channel.
2) Capacity of channel i is C_i.
3) Channels are error free.
4) Scheduling discipline at each channel is first-come, first-served.

5) Messages are classified according to source-destination pairs.

6) The external arrival processes are Poisson.

7) Message length is exponential with mean $1/\mu$.

8) Kleinrock's independence assumption is valid.

9) Fixed routing is used.

10) There is no flow control or acknowledgement of messages

11) There is buffer space in each node.

As mentioned before, this basic model can be used to study the national network and each of the regional networks described in Section 1.2. The advantage of modelling the Canadian network as a three-level hierarchy is seen here because we can treat the national network and the regional networks separately. As a result, the size of each network being analysed is much reduced. Moreover, hierarchies are better than unstructured networks (more economical, simpler routing, works well with voice network).

2.4 Input Parameters and Performance Measures

For our basic model, the input parameters are:

1) M - number of nodes

2) R - number of source-destination pairs (or classes)

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

4) $\Pi(r)$ - set of channels in the path of class r messages

5) $1/\mu$ - mean message length

6) C_i - capacity of channel i

From $\gamma(r)$, we can get: (2.1)

$$\gamma = \sum_{r=1}^{R} \gamma(r) = \text{ total external arrival rate}$$

The key performance measures are:

1) $T(r)$ - the mean delay of class r messages

2) $T = \sum_{r=1}^{R} \frac{\gamma(r)}{\gamma} T(r) =$ overall mean message delay (2.2)

3) $\gamma^*(r)$ - the throughput of class r messages

4) $\gamma^* = \sum_{r=1}^{R} \gamma^*(r) =$ total network throughput (2.3)

2.5 Exact Analysis

The basic model described in Section 2.1 belongs to the class of queueing network models studied by Baskett et al. <BASK 75>; and we can use the results of <BASK 75> in order to get the equilibrium state probabilities for our model. Let $S = (S_1, S_2, \ldots, S_M)$ be a state of the network where $S_i = (n_{i1}, n_{i2}, \ldots, n_{iR})$ and n_{ir} = number of class r messages at channel i. From <BASK 75>, the equilibrium state probabilities are given by:

$$P(S_1, S_2, \ldots, S_M) = \prod_{i=1}^{M} (1-\rho_i)\, n_i! \prod_{r=1}^{R} \frac{1}{n_{ir}!} \rho_{ir}^{\,n_{ir}} \qquad (2.4)$$

where $n_i = \sum_{r=1}^{R} n_{ir} =$ total number of messages at channel i

$$\rho_{ir} = \frac{\lambda_{ir}}{\mu C_i}, \qquad \rho_i = \sum_{r=1}^{R} \rho_{ir} \qquad (2.5)$$

and λ_{ir} = total mean arrival rate of class r messages to channel i. Since we have a fixed routing algorithm, λ_{ir} is given by:

$$\lambda_{ir} = \begin{cases} \gamma(r) & \text{if channel } i \in \Pi(r) \\ 0 & \text{otherwise} \end{cases} \qquad (2.6)$$

For the equilibrium state probabilities to exist, it is required that $\rho_i < 1$ for $i = 1,2,\ldots,M$. This is equivalent to the requirement that no channel is saturated, the condition for a stable network.

From the equilibrium state probabilities, we can get the mean message delay. Let $\bar{n}_{ir}$ be the mean number of class r messages at channel i. $\bar{n}_{ir}$ is given by <BASK 76>:

$$\bar{n}_{ir} = \sum_{n_i=0}^{\infty} \sum_{n_{i1}+\ldots+n_{iR}=n_i} n_{ir}(1-\rho_i)n_{i}! \prod_{s=1}^{R} \frac{1}{n_{is}!} \rho_{is}^{n_{is}} \quad (2.7)$$

$$= \frac{\rho_{ir}}{1-\rho_i}$$

The mean total number of class r messages in the network is therefore:

$$\bar{n}(r) = \sum_{i=1}^{M} \bar{n}_{ir} = \sum_{i=1}^{M} \frac{\rho_{ir}}{1-\rho_i} \quad (2.8)$$

Finally, using Little's result <LITT 61>, we get the following expression for the mean delay of class r messages:

$$T(r) = \bar{n}(r)/\gamma(r) \quad (2.9)$$

T(r) is then used in equation (2.2) to give:

$$T = \sum_{r=1}^{R} \frac{\gamma(r)}{\gamma} \frac{\bar{n}(r)}{\gamma(r)} = \sum_{r=1}^{R} \frac{\bar{n}(r)}{\gamma} \quad (2.10)$$

Combining equations (2.8) and (2.10), we get

$$T = \frac{1}{\gamma} \sum_{r=1}^{R} \sum_{i=1}^{M} \frac{\rho_{ir}}{1-\rho_i} = \frac{1}{\gamma} \sum_{i=1}^{M} \frac{\rho_i}{1-\rho_i} \quad (2.11)$$

Equation (2.11) is the delay expression obtained by Kleinrock <KLEI 64>, but the mean delay of messages belonging to a particular source-destination pair (equation (2.9)) cannot be obtained until the queueing network results of Baskett et al. <BASK 75> are available.

The analysis that leads to equations (2.9) and (2.11) can easily be generalized to include random routing. The only difference is the manner in which the λ_{ir}'s are determined. In particular, λ_{ir} is given by the product of $\gamma(r)$ and the fraction of class r messages that are routed through channel i.

As to the throughput of class r messages, we have

$$\gamma^{*}(r) = \gamma(r) \tag{2.12}$$

Equation (2.12) is a result of the network being stable and specifies that all messages entering the network are eventually delivered to their respective destinations.

A typical plot of T vs γ (i.e., overall mean message delay vs total input rate) is shown in Figure 6. Since $\gamma^{*} = \gamma$ for a stable network, Figure 6 is also a plot showing the delay-throughput characteristics of a communication network. It should be noted that in Figure 6 the increase in γ is due to a proportionate increase in the $\gamma(r)$'s.

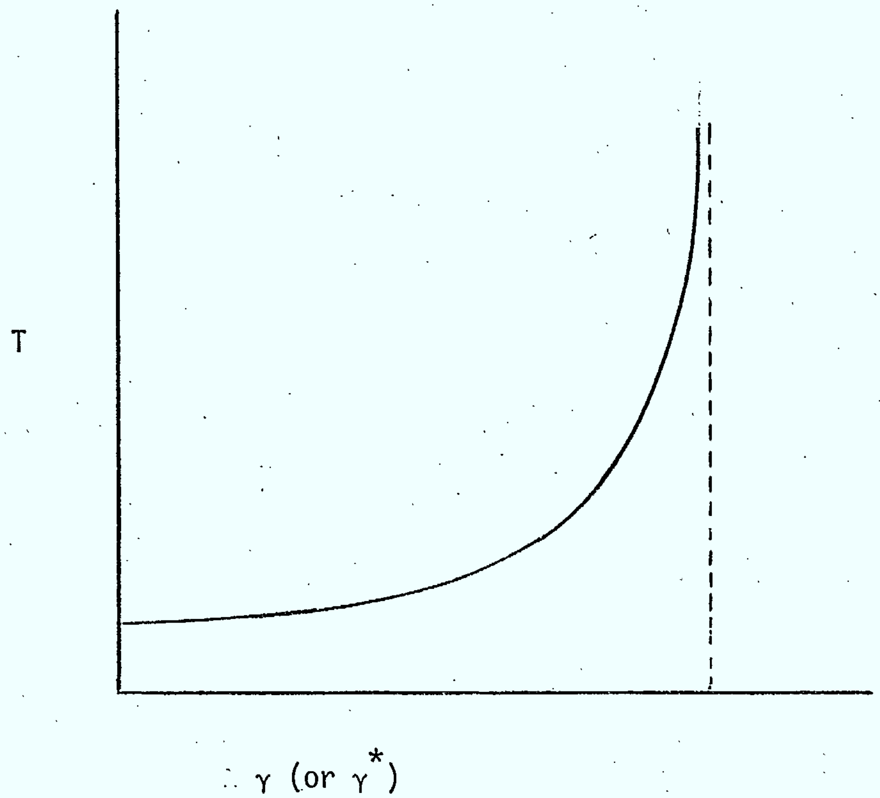


Figure 6

Delay - Throughput Characteristics
in a Communication Network

2.6 More General Models

The basic model analysed in the last section can be generalized to include more detailed features of communication networks. Some of these generalizations are discussed below:

2.6.1 Finite Buffer Space

The number of buffers in each switching node is finite and messages entering nodes are either lost or to be re-transmitted if there is no empty buffer available. The allocation of buffers to messages belonging to different source-destination pairs becomes an important factor in characterizing the network performance.

Analytic studies on network models with finite buffer space has not been plentiful. Among those available are Irland's work <IRLA 75c> on a single switch model with loss and Lam's work <LAM 76> on a network model with re-transmission.

2.6.2 Messages Belonging to Priority Classes

Messages can be classified according to their length or to their urgency. Different classes may have different arrival rates and message lengths. In a communication network, it is natural to give priority to interactive traffic over file transfer traffic. It is of interest to study the effect of the use of priority scheduling disciplines (at the channels) on network performance.

Queueing analysis of network models with priority classes is very difficult. One might have to resort to simulation techniques to obtain results for such models.

2.6.3 Flow Control

New results on the generalization of our basic model to include flow control have been derived by the author. These results are reported in CCNG Technical Report E-54, a copy of which is included in the Appendix.

2.6.4 Other Generalizations:

1) Non-Poisson arrival processes.

2) Non-exponential message length distribution.

3) Removal of Kleinrock's independence assumption.

2.7 Input Parameters and Performance Measures

As a result of the above generalizations, we have the following list of additional input parameters:

1) number of buffers in each node,

2) buffer allocation scheme,

3) arrival rate of messages belonging to each priority class,

4) mean message length of each priority class.

We also have the following performance measures:

5) buffer overflow probability,

6) mean number of buffers occupied by messages belonging to a particular source-destination pair,

7) mean delay of messages belonging to different priority classes,

8) throughput of messages belonging to different priority classes.

IV SIMULATION (by M.I. Irland)

1. Objectives and Scope

The objective of this work is to investigate issues involved in simulation of computer communications networks. These networks are used to transmit data between their subscribers, terminals and computers, in a way similar to the telephone network which serves voice communication. Designers of large networks have found that a multilevel hierarchical structure is practical, with a backbone network at the higher level and local distribution networks at the lower level. For bursty distributed data traffic a packet-switched backbone network is often used. (This network may itself exhibit hierarchical structure.) Several different techniques may be used for local distribution, such as point-to-point lines, multidrop lines (multiplexed or polled), or loops.

Similar to the design problem <NAC 76>, the simulation problem is naturally subdivided into simulation of the backbone network and of local distribution. In this report we will focus on the packet-switched backbone network.

At this stage of our work, it is not our intention to propose a specific design for a simulator, but rather to discuss the general approaches, and to anticipate problems and difficulties which may arise.

2. Why Simulate?

A packet-switched backbone network consists of communications processors (called packet switches) which are interconnected by communications links. One function of a packet switch is to interface with a local distribution network, while another is to control entry of the subscriber data into the backbone network. Within this network the subscriber data is stored and transmitted from one switch to another in a packet form (a packet is a block of data of certain fixed length, or less, plus a header).

Another function of a packet switch is to control the flow of packets within the network, according to well-defined protocols. Also, packet switches generate and propagate control information to be used for routing decisions, congestion control and accounting.

We see that a packet-switched network is a complex system whose behaviour is difficult to analyse if we were to resort only to analytical methods traditionally used for telephone networks. This difficulty arises because various network control mechanisms clearly interact with each other and may have important effects on network performance. Also, the traffic patterns are not well understood. Therefore, simulation is required as one of the design and analysis tools.

We refer to "simulation" in the narrow sense of the word here; we mean detailed simulation of the network operation where:

1) a model of the real network is implemented on a <u>digital</u> computer system;

2) actions of at least some network components are simulated step-by-step (i.e., <u>discrete</u> <u>events</u> occur);

3) some actions may follow <u>stochastic</u> rather than deterministic laws.

In particular, the NPPS system is <u>not</u> a simulation in the sense of the word defined here, in that it deals with costing and capacity assignments as well.

The inputs to the simulator will contain specifications of the network and descriptions of the subscribers' traffic in some parametric form. As outputs, we will produce a set of values which characterize the network performance for given input parameters. These may be mean values or histograms of observations which are taken during one or more simulation runs, for example, end-to-end packet delay, loss rate, and link utilization. What is measured will depend on specific applications, and may greatly affect the simulation cost.

A very important limitation of this technique is that it does not directly provide answers to the design and cost evaluation questions that are often raised. What is usually required is knowledge of various relationships between network and traffic parameters, and performance values. These can only be established by multiple simulation runs.

with different input specifications within the range of interest. For example, in order to find the relationship between link capacity and end-to-end packet delay so that a suitable capacity can be determined for a given traffic level, we will have to perform several simulation runs where link capacities are varied from run to run.

Such exhaustive studies will probably turn out to be impractical to carry out for each network under consideration. <u>It may be hoped, however, that we will be able to discover certain approximate relationships which hold for a large class of networks. These may result in convenient "rules of thumb", like those successfully used for telephone networks</u>. We suggest use of our present simulator, with modifications, to search for such relationships for the Datapac network. These relationships would be useful to subsequent activity aimed at a new comprehensive network model.

Finally, such general relationships can serve as inputs to a "simulator" of the financial aspects of computer communications, similar to the one described in the NPPS reports for the telephone system. (A concise description of the NPPS model is attached as an Appendix.) We now provide a review of the more important simulators for packet-switched networks.

3. Review of Earlier Simulators

Published information about network simulators is unfortunately very fragmentary, particularly with regard to their running times and costs. We summarize some of the more prominent simulators below.

A program designed to simulate the operation of a wide variety of communications networks was written by Kleinrock <KLEI 64> for the TX-2 (a large-scale high-speed digital computer at the Lincoln Laboratory of M.I.T.). The program could accommodate a network of up to 36 switches, and the simulation was event-driven.[1] For a 13-switch network and 10,000 simulated messages, the total running time for the program was of the order of 2 minutes. Visual displays were provided for the operator, giving current state information during the run, and displaying statistics at run termination.

A number of simulation experiments have been carried out by Fultz, using a rather detailed description of the ARPA Network and its operating procedure <KLEI 70, FRAN 72, FULT 72>. Fultz' simulation program ran on the IBM 360/91

-----

1 Simulators are normally divided into two classes. An event-driven simulator has a list of actions (events) to be simulated together with a time at which each action is to occur. This is sorted in order of time. As each action is completed the simulated clock is advanced to the time of the next event in the list. A clock-driven simulator advances its simulated clock by one time unit and checks to see if there is some action to be performed at this time. This allows more complex tests to determine if some action should be performed, but costs in wasted checks every clock tick. <GORD 69>

at UCLA, and was written in the GPSS/360 simulation language <FULT 72>; it consisted of about 2100 card images and computed a large number of statistical measures of network behaviour. The maximum simulated network size was 21 switches; most experiments were conducted on the 19-switch ARPA Network. We have no information on the simulation time or cost.

One of the best documented simulators for packet-switched networks has been developed at the National Physical Laboratory in England over a period of several years <PRIC 72, PRIC 73, PRIC 74a, PRIC 74b, PRIC 76, HEAL 73, JØLL 73>. The first NPL simulations were carried out by Healey <HEAL 73>. The main body of the simulation program was written in ALGØL, machine code being reserved for some frequently used operations; it was run on both an Elliott 4100 and an ICL KDF9 computer. The simulation was event-driven. Following Healey's work, the simulator was reprogrammed by Plessey Telecommunications Research Ltd., under contract to NPL, using the Plessey ALGØL Simulation package. It was run on an ICL 1903 computer with 32K core store and discs. With the speed improvement achieved, in one half to one hour of computer time, a system could be run for about 1 second of simulated time <JØLL 73>. Subsidiary programs were employed to analyse a disc file collected during the simulation run. The network studied had a configuration of 10 switches with links operating at 1.5 Mbits/sec (an 18-switch network was also considered).

Further simulation experiments were then carried out at NPL on isarithmic control, on a revised link and switch protocol, on the effect of link errors and link breakdown, on adaptive routing, and on a hierarchical network. (The results are available in a series of NPL reports.) To facilitate continuing work and to avoid program size limitations, the simulation package has recently been reprogrammed for transfer to a larger, general purpose computer <PRIC 76>.

At Network Analysis Corporation simulation was used as one of the network design tools <VANS 73, CHØU 74>. Simulations of a centralized computer communications system <CHØU 75> applied an interesting hybrid approach where some simulation steps were eliminated by the use of analytical formulae. This technique was effective in facilitating program development and in reducing computer running time. The hybrid and modular approach was further described by Kershenbaum <KERS 74>. The NAC simulation package was programmed for a CDC 6600 computer and required about five seconds of CPU time to simulate 1000 message transactions; the program needed about 30K words of core storage to run.

Some other projects involving simulation of computer communications networks include the following.

A GPSS program for simulation of the computer communications network ILLINET at the University of Illinois was used for evaluation of its performance under existing and proposed priority schemes <BØWD 73>. The network consisted of three nodes and used a dedicated (non-switched) communications system. Simulation results were compared with statistics gathered from the real system and close agreement was observed.

A multi-processor minicomputer Simulation Facility called SIMFAC was proposed for testing new computer communications techniques under conditions closely approximating those of the real ALOHA System at the University of Hawaii <BIND 73>. (The ALOHA system uses shared, random access to a single radio channel for transmission of messages between terminal users and a computer.) The simulator was supposed to run on three minicomputers: one simulating the channel controller-concentrator program, a second dedicated to the simulation of the communication channel characteristics, and the third to the simulation of up to 1000 user terminals. The current status of SIMFAC is not known to this author.

A simulator for a study of adaptive routing techniques was programmed on a PDP 15 by Fuchs <FUCH 74> at McGill University in Montreal. Due to limitations of main memory size, simulated switches had to be swapped into main memory, one at a time, in sequential order. This technique would be expected to result in slow (expensive) simulation. Only short simulation runs were performed.

PACKNET is a simulator of packet-switched data networks implemented on the IBM 370/155 system at the U.S. Defence Communications Engineering Center, using the SOL simulation language <ULFE 75>. It was used in studies involving transient behaviour under conditions of link degradation and outages, and involving the corrective effects of alternate routing.

A simulator for investigation of congestion and flow control in packet-switched networks was developed at the Gesellschaft fur Mathematik und Datenverarbeitung in Darmstadt, Germany <HAEN 76, RAUB 76>.

Finally, a very elegant structural approach to computer network simulation was proposed by Schneider <SCHN 75>. He recognized that a simulation model should be easily amenable to two kinds of changes: parametric changes where only certain model parameters are varied, and changes of network operating protocols; only the first is easy to perform in most simulators; the second requires extensive reprogramming. Schneider's structural approach is based on defining a number of 'protocol system modules' with clearly imposed capabilities and interfaces. The user is given skeleton modules which he is allowed to replace with his own without producing undesirable side effects, so long as he adheres to specified conventions. The entire package is written in SIMULA and is running on the CDC CYBER-74 at the University of Minnesota Computer Center. The initial runs, on a simple network of five switches with 700 messages generated in 10 seconds of simulated time, required 14 seconds of CPU time and 60K words of memory. The SIMULA program was about 1000 lines long.

4. Simulation Work at CCNG

At the University of Waterloo several projects have been directed by Dr. J. Majithia. They include simulation studies of loop networks <RAHM 72, DUBE 74, MAJI 76, YU 76>, of the routing problem in packet-switched networks <BHAR 74a, BHAR 74b, BHAR 75>, and of protocols <KING 75>. Most simulation programs were written in GPSS and executed on an IBM 360/75 computer.

Also, A. Bowler <BØWL 76> wrote a simulation program for a study of adaptive routing. About 1000 lines long, this FØRTRAN program was executed on a Honeywell 6050 computer and took about 3 minutes CPU time to simulate 30,000 packets on a 4-switch network.

Finally, the author <IRLA 73, IRLA 74a, IRLA 75a, IRLA 77> designed a distributed simulator using three minicomputers (Figure 1). Its primary advantage is cost efficiency; it is about 10 times cheaper to run than some conventional GPSS programs which use a large, time-shared computer. (A comparison of run times is shown in Table 1.[2]) To the best of our knowledge this simulator is cheaper to run than anyone else's. The simulator has been used for research and contract work which involved: determination of general network behaviour under given traffic conditions <IRLA 75b>; studies of switch-switch protocols <IRLA 74b, IRLA 76e>; simulation of a route recovery mechanism (in case

-----

2 This information was kindly provided by Mr. King <KING 74>.

network	# messages	B	C	ratio C/B	GPSS minutes per 1000 msg	multiprocessor minutes per 1000 msg
		IBM 360/75 GPSS CPU time in problem state (minutes)	multiprocessor PDP-11s total elapsed real time (minutes)			
21-node loop	500	0.91	0.92	1.0	1.8	1.8
	1000	1.63	1.78	1.1	1.6	1.8
	2000	3.16	3.36	1.1	1.6	1.7
	5000	7.91	9.03	1.1	1.6	1.8
	12000	18.23	16.97	0.9	1.5	1.4
11-node CANNET	500	0.46	0.62	1.35	0.9	1.2
	1000	0.82	1.08	1.3	0.8	1.1
	2000	1.47	2.17	1.5	0.73	1.1
	5000	3.58	4.73	1.3	0.72	0.95
	12000	9.15	11.10	1.2	0.76	0.92
19-node ARPANET	500	0.49	0.70	1.4	0.98	1.4
	1000	0.91	1.38	1.5	0.91	1.4
	2000	1.76	2.70	1.5	0.88	1.4
	5000	4.20	6.31	1.5	0.84	1.3
	12000	10.38	16.35	1.6	1.1	1.4

Table 1: Comparison of Simulation Efficiency

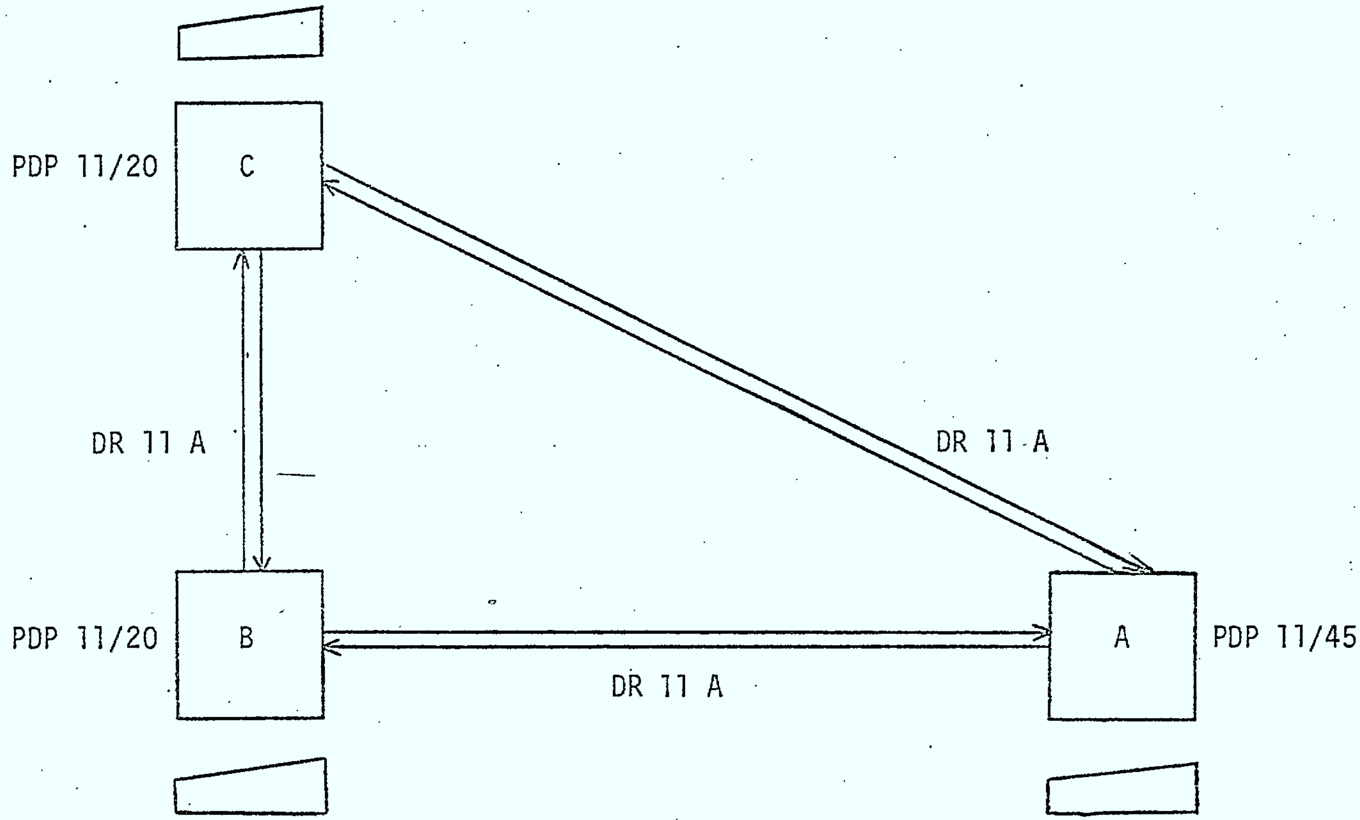


Figure 1:   Multiprocessor Simulator

of link or switch failure) <IRLA 76d>; and studies of congestion controls <IRLA 76f, IRLA 77>.

5. Validation and Confidence

Validation is the process of bringing to an acceptable level the user's confidence that any inference about a system derived from simulation is correct <SHAN 75>. We use our distributed simulator to illustrate this process which we have decomposed into the following sequence of steps:

1) validation of the simulation model,
2) validation of the simulation program,
3) validation of the simulation results.

Validating a model means establishing that it resembles its real system adequately <FISH 67>. Two aspects of the simulation model require examination. The first concerns characterization of input data, such as external arrival rate, mean packet length, and their distributions. The other aspect deals with the model structure and its operating rules.

Responsibility for appropriate specification of the input data is left with the user; he can choose the distribution functions provided by the simulator, or supply his own. The structure of each packet switch, and the operating rules, have been modelled after the CIGALE Network <POUZ 73, GRAN 75>.

Validation of the simulation program (sometimes called verification <FISH 67>) also consists of two parts -- validation of the system input and validation of structure. The system input consists of a stream of packets arriving at the network. Each packet is created using a random number generator which determines the packet length, its arrival time, and its source and destination.

In order to establish the structural correctness of the simulator (i.e., lack of programming errors) we first subjected it to manual verification <MacD 70>, through the examination of a step-by-step trace produced by the program. This approach was practical only for very small networks. We started with a single-switch network in order to examine the queueing mechanism. Then we applied the same procedure to a two-switch network to check the interaction between switches. In each case a light load was applied, followed by heavy load, in order to exercise the largest possible number of subroutines, including the drop-packet mechanism. (It should be mentioned that a simulation program tends to run in a very repetitive manner, with most subroutines executed frequently, and very few 'special cases'. This eases the problem of checking all possible control-flow paths through the program.)

Next we experimented with a five-node tandem network subjected to deterministic traffic (fixed length packets, fixed interarrival times). We could thus compare the constant transit delay for each packet against an easily calculated correct value. The numerical agreement increased our confidence in the correctness of the simulator's timing mechanism. Finally, we ran the simulator on the 19-switch ARPA Network and compared average transit delay with published simulation results <FULT 72>. The results were satisfactory (within a few percent.) The tests listed above can not guarantee the simulator's correctness; but each one gives us additional evidence that it is correct. Further evidence is provided by agreement of simulation and analytic results, discussed next.

Unfortunately, we have not been able to compare the predictions of our models with the real system because adequate measurements of real networks either have not been performed to date, or are not available. Instead, agreement between results obtained by simulation and analytic methods -- in cases where they can be applied -- may be taken as evidence that the simulator program is correct.

Since the projected simulation experiments are stochastic in nature, their results must be considered in a probabilistic rather than absolute sense. Confidence intervals provide an assessment of how representative the simulation results are <FISH 73>. A confidence interval is an interval around a sample value which contains the true value with a given probability.

Classical statistical methods are directly applicable only when independent samples can be obtained. However, a single simulation run provides a sequence of observations which is often highly autocorrelated. Multiple replication of simulation experiments being impractical because of high cost, we need a new technique for finding confidence intervals from a single run; we would accept a technique yielding even approximate results. A method proposed by Crane and Iglehart <CRAN 74> requires that the system returns to some fixed state (regeneration point) frequently, and seems to be applicable only to a system with a small number of queues. We are interested in packet-switched networks which are not such systems.

We suggest a method of batch means <GØRD 69, FISH 75>. Say we are interested in mean delay. During a simulation run, values of packet transit delay are collected to form a time series. This time series is then broken into intervals, or batches, and a mean value is derived for each batch. The batch means are checked for autocorrelation -- in our experiments we found that, if all batches were taken, a relatively high degree of autocorrelation exists. This is reasonable, bearing in mind that the state of the system at the end of one batch is the initial state for the next batch. The use of only alternate batches significantly reduces the autocorrelation. Furthermore, it was established that the distribution of alternate batch means is not skewed. Therefore, it is reasonable to assume independence of alternate batch means, and a classical statistical method can be used to determine the confidence interval for the mean delay <HØEL 62>.

There is a trade-off in the selection of a batch size. In a single simulation run of fixed length, the batch size is inversely proportional to the number of batches. Øn the one hand, we would like the batches to be long, in order to reduce the variation among the batch means, and to increase the spacing between alternate batches -- thus reducing autocorrelation. On the other hand, we want to keep the number of batches as large as possible because the width of the confidence interval is inversely proportional to the square root of this number.

We found experimentally that the choice of the batch size does not have a significant effect on the width of the confidence interval, as long as autocorrelation is small. For example, we simulated an 11-switch sample network for 120 seconds of simulated time, not counting the initial transient portion of the run which we disregarded; in total about 35,000 packets were transmitted. Using a batch size of either one or two seconds, we estimated the 90% confidence interval to lie at ±3.5% of the sample mean.

6. Expected Major Difficulties

Knowing one's enemy is half the battle. There are a number of potential problem areas in network simulation; judging from past experience we discuss some of them in the hope that early recognition will help us to avoid them.

We mention network size first. We conclude from Sections 3 and 4 that most networks simulated in the past contained less than 20 switches. It should be noted that the size of the network determines the sizes of the tables representing it for the simulator; a larger network requires a larger description. Here the important network parameters are numbers of subscribers, switches, and links; table sizes grow linearly, or quadratically, with these numbers. The memory required to store these tables quickly becomes excessive; use of auxiliary storage is usually impractical because of running time constraints. We believe that a simulator design which can accommodate a backbone network of up to twenty switches is sufficient for the proposed study. Larger networks should exhibit hierarchical structures, and may therefore admit different simulation techniques.

Run length may be measured as the the number of simulated packets or as the elapsed simulated time. The required run length can be determined by the desired width of confidence intervals, as discussed earlier. We have found that relatively long runs are needed in practice (60,000 packets has not been excessive in some cases;

250,000 may be needed in the future, depending on network size and load). This has several immediate consequences. Firstly, the simulation must be efficient so that sufficiently long runs can be carried out within practical time and cost limits. Secondly, the naming scheme should be sufficiently flexible to allow a distinct name for each packet. Thirdly, the timing mechanism should allow sufficiently large clock values.

Even a most detailed, event-by-event simulation will have to involve some conscious approximations for the sake of economy. The user should be made aware of these approximations, so he can either disregard them (if their effect is insignificant) or compensate when interpreting the simulation output. Clearly, any unintended bias of the simulator must be avoided by careful validation; the difficulties of validating a simulator were discussed in the preceding Section.

Finally, we come to the input and output considerations; that is, to the interface between the user and the simulator. It has been our experience that mistakes in input specifications are the most frequent source of simulation errors. In part this may be due to the large number of parameters which the user must define. Two kinds of mistakes occur often. First, parameter values which are inconsistent may cause the simulation to terminate abnormally; secondly, an incorrect parameter value will produce erroneous results although the simulation may run to completion. Therefore, it is important that the simulator check input data for consistency and that the format of input data be sufficiently redundant to discover most errors.

As far as the output is concerned we have to deal with two apparently inconsistent requirements. On the one hand, sufficient output must be available for a detailed analysis of the system performance (e.g., an event trace). On the other hand, it is prohibitively expensive to collect and store all intermediate data which can potentially be useful, and an excessive amount of output tends to hide significant information. In most cases only summary statistics of performance measures are required. Therefore, the user should be able to selectively gather and compact data, and display results.

We expect that the simulator users will not normally be experienced programmers; hence considerable effort must be expended on the human engineering of the user interface.

7. General Approach to Simulator Design

We assume that the simulator of packet-switched networks is intended for a wide variety of applications, some of which we cannot foresee at present. Therefore, its design will have to be flexible so that on the one hand we will be able to simulate a large class of networks with minimum effort expanded on simulator modifications and, on the other hand, we may at a later date easily add new routines to collect and process required information.

Flexibility can be achieved by imposing a modular structure, where each module is devoted to a well defined task and communicates with other modules in a well defined manner. Each modules should be easily expandible and replaceable, as the need arises, without disturbing other modules <SCHN 76>.

The tasks fall naturally into three classes:

1) initialization according to user's specifications,
2) simulation of network operation, and
3) collecting, processing and output of required information.

(Although, at first, it appears that simulation of network operation is the most difficult part of simulator design, we know from previous experience that the other two classes of tasks may pose even greater challenges.)

Modular design can be further enhanced by a _distributed_, multi-processor computing environment. We can take advantage of parallelism between some tasks, thus increasing simulation speed. We are also forced to maintain clean separation between modules because they run in different loosly-coupled processors. This results in improved programming style. Our previous experience <IRLA 75a> indicates that distributed design of a simulator, using several minicomputers and/or microcomputers, is feasible and highly cost-effective. The main disadvantage is the lack of portability, as multi-processor computing systems with the required configuration are not yet commonly available.[3] On the other hand, the distributed design will not preclude using the simulator on a single computer subject to programming language compatibility, memory and CPU requirements and probably also increased cost.

It has been the experience of other researchers that, for the sake of computational economy, certain operational details are best approximated by analytical formulae rather than simulated in detail, event-by-event. This approach is called a _hybrid_ analysis/simulation technique <CHOU 75, SAUE 76>.[4]

-----

3 The required hardware is available in the CCNG Laboratory at the University of Waterloo.

4 Unlike some some earlier simulation literature the word "hybrid" does _not_ refer to analogue/digital simulation techniques here.

8. Conclusions and Further Work

In this progress report we have discussed issues involved in simulation of packet-switched networks, without actually proposing any specific simulator design. The major problem not considered here is modelling of the local distribution networks.

We propose for further work:

1) extensive simulation of Datapac using the present CCNG simulator to discover "rules of thumb",

2) detailed design of a simulator able to replace part of the NPPS operating block, and

3) implementation of this simulator.

9. References

BASK 75 F. Baskett, K. Chandy, R. Muntz, and F. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," Journal of the ACM, Vol. 22, No. 2, April, 1975, pp. 248-260.

BHAR 74a R. Bhar and J.C. Majithia, "A Study of Routing Strategies in a Packet Switched Computer Network," University of Waterloo, Computer Communications Networks Group, Report E-20, May 1974.

BHAR 74b R. Bhar and J.C. Majithia, "Some Simulation Results for the CYCLADES Network," Reseau CYCLADES, Report MOD 501, July 1974.

BHAR 75 R. Bhar and J.C. Majithia, "Simulation Results Concerning Routing Strategies in a Proposed Canadian Network," Proc. A.I.M. International Meeting on Mini-Computers and Data-Communications, Liege, Belgium, January 1975, 217-222.

BIND 73 R. Binder and F.F. Kuo, "The Use of Multi-Processor Minicomputer for Communication System Simulation," The ALOHA System Technical Report B73-6, University of Hawaii, November 1973.

BOWD 73 E.K. Bowdon, S.A. Mamrak, and F.R. Salz, "Simulation - A Tool for Performance Evaluation in Network Computers," AFIPS Conf. Proc. National Computer Conference, Vol. 42, New York, June 1973, pp. 121-131.

BOWL 76 A. Bowler, private communication, University of Waterloo, Computer Communications Networks Group.

CHOU 74 W. Chou, "Planning and Design of Data Communications Networks," AFIPS Conf. Proc. National Computer Conference, Vol. 43, Chicago, June 1974, pp. 553-559.

CHOU 75 W. Chou, H. Frank, and R. Van Slyke, "Simulation of Centralized Computer Communications Systems," IEEE Transactions on Comunications, Vol. 23, 9 (September 1975), pp. 994-1001.

CRAN 74 M.A. Crane and D.L. Iglehart, "Simulating Stable Stochastic Systems, I: General Multiserver Queues", Journal ACM 21, 1 (January 1974), 103-113.

DAVI 72 D.W. Davies, "The Control of Congestion in Packet-Switched Networks," IEEE Transactions on Communications, Vol. 20, 3 (June 1972), 546-550.

DUBE 74 J.D. Dube, "Investigation of Message Traffic Parameters in Loop Type Data Communication Networks," M.A.S. Thesis, Department of Electrical Engineering, University of Waterloo, Waterloo, Ontario, June 1974.

FISH 73 G.S. Fishman, Concepts and Methods in Discrete Event Digital Simulation, Wiley, New York, 1973.

FISH 75 G.S. Fishman, "Batch Means in Digital Simulation", Technical Report 75-7, Operations Research and Systems Analysis, University of North Carolina at Chapel Hill, August 1975.

FRAN 72 H. Frank, R.E. Kahn, and L. Kleinrock, "Computer Communication Network Design -- Experience with Theory and Practice," AFIPS Conf. Proc. Spring Joint Computer Conference, Vol. 40, Atlantic City, May 1972, pp. 255-270.

FUCH 70 E. Fuchs and P.E. Jackson, "Estimates of Distributions of Random Variables for Certain Computer Communications Traffic Models," Communications ACM, Vol. 13, 12 (December 1970), 752-757.

FUCH 74 H. Fuchs, "Simulation Study of an Adaptive Routing Technique for Packet-Switched Communication Networks," Master of Engineering Thesis, Department of Electrical Engineering, McGill University, Montreal, March 1974.

FULT 72 G.L. Fultz, "Adaptive Routing Techniques for Message Switching Computer Communications," Doctoral Thesis, School of Engineering and Applied Science, University of California, Los Angeles, UCLA-ENG-7252 (ARPA), July 1972.

GORD 69 G. Gordon, System Simulation, Prentice-Hall, Englewood Cliffs, 1969.

GRAN 75 J.L. Grange, "CIGALE Implementation, Tools and Techniques," SHARE European Association, Dublin, September 1975; Reseau CYCLADES, Report MIT 602, September 1975.

HANL 76 J. Hanle, private communication, Gesellschaft fur Mathematik und Datenverarbeitung, Darmstadt, Germany, January 1976.

HEAL 73 R. Healey, "Computer Network Simulation Study," National Physical Laboratory Report COM 64, January 1973.

HOEL 62 P.G. Hoel, "*Introduction to Mathematical Statistics*," Third Edition, Wiley, New York, 1962.

IRLA 73 M. Irland, "Communication Networks and their Storage Requirements," University of Waterloo, Computer Communications Networks Group, Report E-9, April 1973.

IRLA 74a M. Irland, E.G. Manning, C. Britney, S. Flores, I. Mijares, and R. Payne, "Computer Networks Simulation System," University of Waterloo, Computer Communications Networks Group, Report E-25, May 1974.

IRLA 74b M. Irland, "Simulation Study of the BNR Packet-Switched Network, Final Report," unpublished report, November 1974.

IRLA 75a M. Irland and E.G. Manning, "Multiprocessor Simulation Using Minicomputers of Packet-Switched Data Networks," *Proc. A.I.M. International Meeting on Mini-Computers and Data-Communications*, Liege, Belgium, January 1975, pp. 178-184; also University of Waterloo, Computer Communications Networks Group, Report E-26, September 1974.

IRLA 75b M. Irland, "Simulation of CIGALE 1974," *Proc. ACM-IEEE Fourth Data Communications Symposium*, Quebec City, October 1975, pp. 5-13 to 5-19; also extended version "Simulation of CIGALE, Report on Assumptions and Results," University of Waterloo, Computer Communications Networks Group, Report E-32, January 1975.

IRLA 75c M. Irland, "Queueing Analysis of a Buffer Allocation Scheme for a Packet Switch," *Proc. IEEE National Telecommunications Conference*, New Orleans, December, 1975, pp. 24-8 to 23.

IRLA 76d M. Irland and J.M. Macdonald, "Simulation of Route Propagation Mechanism in CIGALE," University of Waterloo, Computer Communications Networks Group, Report E-49, May 1976.

IRLA 76e M. Irland and N.B. Cohen, "Simulation of Switch-Switch Protocol (MV8) in CIGALE," University of Waterloo, Computer Communications Networks Group, Report E-53, August 1976.

IRLA 76f M. Irland and J.L. Grange, "Notes on Simulation of CIGALE 1976: Congestion Control," University of Waterloo, Computer Communications Networks Group, Report I-27, August 1976.

IRLA 77 M. Irland, "Analysis and Simulation of Congestion in Packet-Switched Networks," Doctoral Thesis, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, January 1977.

JOLL 73 J.H. Jolly and R.A. Adams, "Simulation Study of a Data Communication Network, Part 3: Original System Programme Documentation," Report 97/73/04/TR, Plessey Telecommunications Research Ltd., January 1973.

KAHN 72 R. Kahn, and W. Crowther, "Flow Control in a Resource Sharing Computer Network," <u>IEEE Transactions on Communications</u>, June, 1972, pp. 539-546.

KERS 74 A. Kershenbaum, "Tools for Planning and Designing Data Communications Networks," <u>AFIPS Conf. Proc. National Computer Conference</u>, Vol. 43, Chicago, June 1974, pp. 583-591.

KING 74 J.T. King, private communication, University of Waterloo, Computer Communications Networks Group.

KING 75 J.T. King, "A Study of Design and Evaluation of Protocols in Packet Switching Networks," M.A.S. Thesis, Department of Electrical Engineering, University of Waterloo, Waterloo, Ontario, 1975.

KLEI 64 L. Kleinrock, <u>Communication Nets: Stochastic Message Flow and Delay</u>, McGraw-Hill, New York, 1964; reprinted by Dover, New York, 1972.

KLEI 70 L. Kleinrock, "Analytic and Simulation Methods in Computer Network Design," <u>AFIPS Conf. Proc. Spring Joint Computer Conference</u>, Vol. 36, Atlantic City, May 1970, pp. 569-579.

KLEI 76 L. Kleinrock, <u>Queueing Systems - Volume 2: Computer Applications</u>, Wiley-Interscience, New York, 1976.

LAM 76 S. Lam, "Store and Forward Buffer Requirements in a Packet Switching Network," <u>IEEE Transactions on Communications</u>, April, 1976, pp. 394-403.

LITT 61 J. Little, "A Proof of the Queueing Formula L = W," <u>Operations Research</u>, Volume 9, 1961, pp. 383-387.

MacD 70 M.H. MacDougall, "Computer System Simulation: An Introduction," <u>Computing Surveys</u> Vol 2, 3 (September 1970), 191-209.

MAJI 76 J.C. Majithia and J.D. Dube, "Simulation Results for a Loop Network with a Hybrid Message Handling Protocol," Proc. NBS-IEEE Symposium on Computer Networks: Trends and Applications, Gaithersburg, Maryland, November 1976.

NAC 76 "Local, Regional and Large Scale Integrated Networks," Sixth Semiannual Technical Report, Vol. 3, Network Analysis Corporation, Glen Cove, New York, February 1976.

PENN 75 M. Pennotti, and M. Schwartz, "Congestion Control in Store and Forward Tandem Link," IEEE Transactions on Communication, December, 1975, pp. 1434-1443.

POUZ 73 L. Pouzin, "Presentation and Major Design Aspects of the CYCLADES Computer Network," Proc. IEEE-ACM Third Data Communications Symposium, St. Petersburg, Florida, November 1973, 80-87.

PRIC 72 W.L. Price, "Survey of NPL Simulation Studies of Data Networks, 1968-72," National Physical Laboratory Report COM 60, November 1972.

PRIC 73 W.L. Price, "Simulation of Packet-Switched Data Networks Controlled on Isarithmic Principles," Proc. IEEE-ACM Third Data Communications Symposium, St. Petersburg, Florida, November 1973, 44-49.

PRIC 74a W.L. Price, "Design of Data Communication Networks Using Simulation Techniques," Computer Aided Design, Vol. 6, 3 (July 1974), 171-175.

PRIC 74b W.L. Price, "Simulation Studies of an Isarithmically Controled Store and Forward Data Communications Network," Proc. IFIP Congress, Stockholm, August 1974, pp. 151-154.

PRIC 76 W.L. Price, "Simulation of Data Networks at the National Physical Laboratory," Journees de Travail, Modelisation et Simulation de Reseaux d'Ordinateurs, IRIA, February 1976, pp. 171-197.

RAHM 72 M. Rahman, "Analysis and Simulation of a Full Duplex Loop Data Communication Network," M.A.S. Thesis, Department of Electrical Engineering, University of Waterloo, Waterloo, Ontario, December 1972.

RAUB 76 E. Raubold and J. Hanle, "A Method of Deadlock-free Resource Allocation and Flow Control in Packet Networks," Proc. Third International Conference on Computer Communication, Toronto, August 1976, pp.

483-487.

RUDI 76 H. Rudin, "Flow-Control: Session Chairman's Remarks," International Conference on Computer Communications, Toronto, Canada, August 3-6, 1976, pp. 463-466.

SAUE 76 C.H. Sauer, L.S. Woo, and W. Chang, "Hybrid Analysis/Simulation: Distributed Networks," IBM Research Report RC 6341, IBM T.J. Watson Research Center, Yorktown Heights, New York, December 1976.

SCHN 76 G.M. Schneider, "A Modular Approach to Computer Network Simulation," Computer Networks, Vol. 1, 2 (September 1976), 95-98.

ULFE 75 H.E. Ulfers, "PACKNET - A Packet Switched Data Network Simulator," Proc. IEEE International Conference on Communications, Vol. I, San Francisco, June 1965, pp. 6-22 to 6-26.

VANS 73 R. Van Slyke, W. Chou, and H. Frank, "Avoiding Simulation in Simulating Computer Communication Networks," AFIPS Conf. Proc. National Computer Conference, Vol. 42, New York, June 1973, pp. 165-169.

WONG 76 J. Wong, and M. Unsoy, "Analysis of Flow Control in Computer Networks," CCNG External Report E-54, November, 1976.

YU 76 L.W. Yu, "Design and Analysis of an Adaptive Loop-Type Computer Communication Network," M.A.S. Thesis, Department of Electrical Engineering, University of Waterloo, Waterloo, Ontario, March 76.

APPENDIX A (by A.T. Bowler)

1. Introduction

The National Policy and Planning Simulation Model (NPPS)[1] is a collection of computer programs, data files, and associated documentation. It was prepared by the Laboratoire d'Econometrie de l'Universite Laval and Sorès Inc. Montreal. The programs run on a Univac 1108 operated by the Computer Science Canada Company. The programs are mostly written in Fortran although a few are written in APL.

It should be stressed that although the model is described as a simulation model this must not be interpreted in the narrow sense of discrete simulation defined Section IV.2 of this report. Rather, the model simulates the financial affairs of the companies involved in long haul telephone transmission.

-----

1 The NPPS project was originally called the Inter-Regional accounting project. This name is perhaps more indicative of the true nature of the project.

2. Overview

The model is logically divided into four major functional blocks. Most of the programs fit clearly into one of the blocks. However, the distinction for data files is not as clear since a single file may be input to more than one block, or may form part of the interface between two blocks.

The first block is the operational block. Its programs are concerned with the physical operation of the telephone network. It has algorithms for estimating traffic, dimensioning switching facilities, and assigning logical links to physical transmission facilities. Again this is more than a simulation in the narrow sense of Section IV.2 as it involves network design decisions.

The programs in the costing block estimate the costs incurred by the carrier companies in offering their services. The programs estimate the day to day operating costs of the network, its maintenance, land taxes, etc., as well as the more flexible items such as equipment depreciation. The costing block also assigns costs to the traffic stream, although this is clearly a somewhat subjective matter.

The sharing block calculates the presettlement revenues, which are the revenues collected by the carriers from their customers. This money is then divided among the participating carriers according to some user specified mixture of settlement schemes. All the settlement plans distribute some portion of the money according to the costs incurred, as calculated by the costing block. The remainder may be distributed in some other fashion.

The accounting block is the final block of the model. Its inputs are the current financial status of the carriers and the income for the current year as calculated by the sharing and costing blocks, as well as certain 'policy' decisions of the user. Its output is the new financial status of the carriers. The relationships amongst the various financial variables are modelled by an underdetermined[2] set of equations. The policy decisions consist either of fixing certain of these values, (eg. dividend payments), or of specifying a global objective function to be optimized over the set of possible financial options.

-----

2 An underdetermined set of equations is one with more variables than equations. Such a system has many solutions.

3. Operating Block

This is the largest block in the model, and has undergone substantial alteration with each phase of the NPPS project.

The first step in the operating block is the estimation of the traffic matrix. The user must supply a typical traffic profile for a day dividing the day into 4 periods. These periods correspond to peak usage, normal hours, evening, and late night. An adjustment is made to compensate for time zone effects and weekends, and a set of traffic matrices is generated based on a gravity model.[3] Some known traffic data can also be incorporated.

The NPPS model views the telephone network as two networks.

1) The switching network is a hierarchical network with switching centres at each node. The logical links connecting the nodes are divided into regular and high usage links. The regular links run between a node and its subordinates in the hierarchy. The high usage links run between switching centres with heavy traffic flows between them.

2) The transmission network is a non-hierarchical network. The links are the physical transmission facilities (microwave, cable, etc). The facilities of the transmission network are used to implement the logical links of the switching network. A logical link of capacity C in the switching network may be implemented by two distinct paths in the transmission network of capacities $A + B \geq C$.

-----

3 A gravity model is one for which the traffic volume between two centres increases with the population of the centres and decreases with the distance between them.

The next set of programs computes the traffic flow through the switching network. The user may of course reconfigure the topology of the switching network. However, he may also request that the switching network be redimensioned before the usage of each link is calculated. This assignment of capacities is done according to the 'Economic CCS' rule, a rule of thumb that corresponds roughly to the practice employed in the industry. There is also a program to estimate the size of the switching equipment needed at each node, to be used by the costing block. Also, for the sharing block, there is a calculation of the peak usage.

The NPPS model does not calculate such performance measures as blocking and loss probabilities, or waiting times. Instead it uses known formulae for these measures to estimate capacities required for acceptable performance when dimensioning the network. It is the digital analogue to this set of rules, that we propose to search for using the network model outlined in Part I.

The next major portion of the operating block assigns circuits of the switching network to actual transmission links. This is done by solving a large linear programming problem. The user can choose one of three basic objective functions to minimize.

1) He can minimize total circuit miles used. This is equivalent to maximizing excess capacity.

2) He can minimize a function of average cost per circuit used.

3) He can minimize a function of marginal cost per circuit used.

The two cost objective functions depend on information from the costing block. Actually, the program that does this circuit allocation, CIRRES, straddles the boundary between the operating and costing blocks; information from CIRRES is used to calculate the per circuit costs. There is a facility for adding reliability constraints to the problem. The unswitched services of private line and television circuits are also included in the requirements at this time.

The CIRRES program is rather expensive to run, so some effort has been spent on finding a good starting point, and the user is given the option of choosing the first feasible solution found rather than the optimal one. To help him with this choice the program tells him the upper bound on the optimum objective value.[4]

The last program takes the circuit assignments and the traffic at peak usage, and computes for each traffic stream, a list of facilities used. This is written out for later use by the sharing block.

-----

4 The upper bound is the usual LP upper bound calculated from duality theory.

4. Costing Block

This block is concerned with estimating the assets of and costs incurred by the individual carriers. It assigns these costs to individual facilities so that the sharing block can allocate revenues, and so that the operating block has a basis on which to assign circuits.

Most of the program code that is specifically for this block is in one program. The program first estimates the value of the current assets on a per element basis. Switches are costed according to the size estimated by the operating block. For transmission links the number of microwave towers[5] is estimated, based on the distance between the endpoints. A capital cost is assigned to each. This cost includes a mark up for remote installations and for extra equipment that may be necessary such as multiplexing or branching equipment. The cost also depends of the capacity of the link.

The program then estimates aging and inflationary effects, and calculates depreciation type costs. The user is given some choice in the depreciation algorithm used. The program also considers taxes and interest on capital as well as deferred tax effects. It then estimates operating costs, maintenance, operation, land taxes, etc. These various items are then totalled by class and carrier, and printed for the user who will then have them available for use with the accounting block.

-----

5 There is an assumption that all links in the transmission network are microwave links.

The effective capacity of each link in the transmission network is estimated and the cost divided among the circuits on that link. This information is stored for use by the circuit allocation routine in the operating block.[6]

Two additional files are written out for use by the sharing block. These contain the ownership, capacity and value of the switching nodes and transmission links.

There are costing abilities in the model other than those specifically implemented in the programs described above. A user wishing to decide the cost of some service would run portions of the operating block twice: once with all the traffic, and once with a demand matrix that does not include the traffic associated with the service in question. He would then examine the estimated operating costs. The difference in the operating cost for the two runs would give the cost of this service. Other costing information is produced in the form of dual variables from the circuit allocation program.

-----

6 Actually, there is a program in this link that reformats this information before it is input to CIRRES in the operating block.

5. Sharing Block

This block computes the actual income earned by each of the carriers in long haul transmission.

There are two programs in this block. The first calculates the collected (presettlement) revenues. Its inputs are the private line data, and the switched call traffic matrices. It calculates the collected revenues in a straight forward manner by applying the tariff rules. The information may be printed for the user and/or written to a file.

The next program divides the collected revenues amongst the carriers. Its inputs are:

1) the files produced by the operating block which give the list of transmission links and switching nodes used by each traffic stream,

2) the file produced by the costing block which gives the ownership and cost of the transmission facilities, and

3) the files produced by the above program giving the revenue collected for each traffic stream.

The revenues collected from each traffic stream are then divided among the participating carriers according to the settlement scheme and the contribution of each of the carriers. The figures are then totalled by company and printed.

6. Accounting Block

This is the last block in the system. It is perhaps the most important from the regulatory point of view. Its purpose is to predict the financial status of the carrier companies.

The basic problem with modelling the accounting of the carriers appears to be the amount of flexibility available to these companies in their financial arrangements. About the only items that are not at the carrier's discretion are the amount of money collected in revenues, and a certain set of expenses that must be paid. Between these imposed limits the companies have a wide range of options open to them. They control the amount of money paid to shareholders, how long they delay paying some expenses, whether they handle certain expenses by borrowing or paying out of available funds,[7] etc.

The limitations on the actions of the companies are modelled by a set on nonlinear equations. Basically, these are conservation rules governing the flow of money between different accounts. Flexibility occurs because this is a consistent underdetermined system.

There are two approaches available to the user of the system. The first is the simultaneous equation approach. The user fixes the values of a subset of the variables so that he obtains a determined set of equations, which the

-----

7 Remember that interest paid on loans is a deductible expense when computing taxes.

system then solves. In order to simplify the task of solving the resulting set of equations the system restricts the alternatives so that the resulting set of equations is linear. The variables the user assigns values to are called BEADs,[8] and the calculated values non-BEADs.

The second approach has been named the goal programming approach. The user specifies some linear objective function to be optimized, and a linear programming problem is solved. Although, it is not explicitly stated in the NPPS reports, the user must specify some more information than just the objective function so that the set of constraints will be linear. This approach has the advantage that certain non-negativity restrictions may be explicitly stated and preserved in the solution; this is not necessarily possible with the simultaneous equation approach.

The implementation of both systems allows the user to repeat the procedure several times using the results of the previous run as a starting point. Thus the user may forecast a possible course of events for several years in the future. The user is not required to type all the BEAD variables every time as there is a facility to use a standard set available on a file.

-----

8 BEAD is an acronym for Basic Economic Accounting Data.

There is some discrepancy between the programs for the accounting block discussed in the reports, and the program described in the user guide. The reports discuss a series of APL programs of increasing complexity and give several output samples of the simultaneous equation approach. There are discussions of the goal programming approach, and mention that it had been converted from Fortran to APL. However, the user guide discusses only one program in the accounting block, which appears to implement in Fortran only the simultaneous equation approach. We did not find any detailed information about the goal programming approach, either how to use it or exactly what equations are used. As a result we are unsure about the exact functioning of the accounting block, and the reader should bear this in mind when considering the statements made above.

The accounting block is not tightly linked with the rest of the NPPS system. Data transfer is entirely manual, with the user having to type results from the costing and sharing blocks as values for certain variables in the accounting block.

7. Simulation

The NPPS model uses the term simulation quite freely with connotations different from that given previously in this report. The authors of the NPPS model distinguish four broad categories for the term. We quote[9] :

1) Validating the logic of the model and obtaining some measures of the various algorithms and interfacing devices the model contains.

2) Testing the quality of semi-realistic data by the model contained.

3) Sensitivity testing with real and semi-realistic data for benchmarks and calibration.

4) Policy simulations proper.

It would appear that the authors of the NPPS study view simulation as meaning the running of a model with some input data, and examining the output for purposes of debugging, calibrating, or actually predicting the performance of the system being modelled.

The reports on the NPPS system contain a large amount of information about results obtained in various studies done on portions of the network during the writing of the NPPS model.

The user of the system has a large number of different alternatives he may try out on the model. However, while these facilities may be of great use, it is unlikely that the parameters which he may vary as inputs to the system are related in a simple and direct manner to the question he

-----

9 See section 4.1.1 of the final report of the second phase.

wishes to ask. He must pose his question and then consider the effects of different answers in terms of measurable quantities that the model accepts as inputs. When he has finished running the model, he must decide what the outputs mean in terms of his original question. For example, if he is considering a different tariff structure he may first have to predict the daily traffic profile and pattern.

In the following we list by block, the parameters the user may alter to run his simulations.

In the operating block the user can try different traffic profiles and patterns, and see if he will require more or less equipment. He may try a different switching hierarchy, to see if it is a better or a worse one. He may try a different topology for the transmission network, or a different criteria for circuit assignment. However, there is nothing that will automatically produce an optimal switching hierarchy and transmission network. This is realistic since the carriers do not build a whole new network every time the traffic pattern changes. The fact that the model attempts an optimal circuit assignment is also realistic since reassigning capacity within the existing network is relatively easily done and it is to be expected that the carriers reassign circuits often enough so that the pattern used is nearly optimum.

In the costing block the user may change the prices of various pieces of equipment, tax and interest rates, as well as depreciation algorithms.

In the sharing block the user may change the tariff rates and settlement schemes. Note that changing the tariff rate does not affect the traffic pattern. The model itself assumes that the traffic pattern is insensitive to tariffs. Any elasticity must be supplied by the user in terms of modified traffic patterns.

In the accounting block the user has available the various decision variables. He also may attempt to predict financial status for several periods into the future.

8. Applicability

The key question is: how much of the NPPS model can be applied to the data transmission network? In out opinion only a small portion can be used directly; a somewhat larger portion could be adapted.

The accounting block is quite general and probably could be interfaced with little difficulty. It is possible that nothing more needs to be done than to change the inputs to reflect the increased expenditures and revenues.

The needs of the data network can be included when the circuit allocation routines are run. Constraints would have to be added to model some digital only links, such as the DUV microwave channels. However, this would affect the functioning of the sharing block. This should be handled by adding another data base and traffic class, much the same as is currently done for private lines. Certainly the revenue calculations must be done differently.

Costing of the switching gear is obviously quite different. The costing of the transmission links may or may not be different. Extra subroutines or programs could be added to the costing block to make it handle the data network, provided that reasonable estimates of life expectancy, hardware costs, etc. could be obtained. The model would have to be updated to do costing for digital cable systems, and DUV microwave channels.

APPENDIX B

ISSN 0384-5702

ANALYSIS OF FLOW CONTROL* IN COMPUTER NETWORKS

by

J. W. Wong and M. Unsoy

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

CCNG Report E-54
November, 1976

*This work was supported in part by the National Research Council and the Department of Communications of Canada.

ABSTRACT

A two-level flow control scheme for message-switched computer networks is considered. Using a queueing network model, analytic expressions are derived for performance measures such as throughput and mean message delay. Results based on an example network show that this two-level control scheme is capable of preventing the network performance from deteriorating when the load between a group of source-destination pairs is increased. This scheme can also be used to give preferential treatment to a particular group of source-destination pairs in terms of a higher throughput.

1. INTRODUCTION

Flow control, as defined by Rudin <1>, is a collection of algorithms which are used in a network to prevent a single user or a single user group from hoarding the resources of the network to the detriment of others. There are two basic types of flow control techniques <2>. The first type does not discriminate messages on the basis of source or destination and places a limit on the total number of messages in the network. An example of this type is the 'isarithmic' flow control technique originally suggested by Davis <3>. This technique has been implemented by Price in his computer network simulation model <4,5>. The second type of flow control places separate limits on the number of messages belonging to each source-destination pair. An example of this type is the end-to-end control scheme implemented in the Advanced Research Projects Agency (ARPA) Network <6>. In this network, a user in a host computer cannot send his next message until the RFNM (Request For Next Message) of his last message is received.

In this paper, an open queueing network model of the type studied by Baskett, et. al. <7> is used to analyse the performance characteristics of a two-level flow control scheme in a message-switched, store and forward computer network. Informally, this scheme can be described as follows. At the first level, a limit is placed on the total number of messages in the network. At the second level, disjoint groups of source-destination pairs are defined and separate limits are placed on the number of messages belonging to each group. This two-level scheme is rather general, and it includes the two basic types of flow control techniques mentioned above as special cases. More importantly, it allows us to give preferential treatment to some source-destination pairs (by adjusting the limits) while maintaining an acceptable level of message flow in the network.

Pennotti and Schwartz <2> have analysed the end-to-end flow control scheme for a store and forward tandem link in a computer network. Their work was later extended by Chatterjee, et. al. <8> to include random routing. The model analysed in this paper is different from those found in <2,8> in the sense that it is a total network model and all source-destination pairs are taken into consideration.

In section 2, our network model with a two-level flow control scheme is described. Analytic expressions for the equilibrium state probabilities and performance measures such as throughput and mean message delay are derived in section 3. Finally, section 4 is devoted to numerical examples and discussion of results.

2. MODEL DESCRIPTION

We first assume, as in <2,8,9>, that the delay experienced by a message in a store and forward network is approximated by the queueing time and data transmission time in the channels. The processing time at the switching nodes and the propagation delays are assumed to be negligible. Let M be the total number of channels, and C_i be the capacity of channel i, $i = 1,2,\ldots,M$. In our open queueing network model, each of the M channels is represented by an independent server. We assume that all channels are error-free, and the queueing discipline at each channel is first-come, first-served.

Messages are classified according to source-destination pairs. In particular, a message is said to belong to class (s,t) if its source node is s and its destination node is t. Let R be the total number of message classes. In a network with N nodes, $R = N(N-1)$. For convenience, we assume that message classes are numbered from 1 to R, and we use r instead of (s,t) to denote a message class. The arrival process of class r messages from outside the network is assumed to be Poisson with mean rate $\gamma(r)$. Message lengths for all classes are assumed to have the same exponential distribution, and we use $1/\mu$ to denote the mean message length. It follows from this last assumption that the service time of all messages at channel i is exponential with mean $1/\mu C_i$. For our analysis, Kleinrock's independence assumption <9> is required. This assumption states that each time a message enters a node, a new length is chosen from the exponential message length distribution. Kleinrock <9> has shown by simulation that this assumption gives accurate results for mean message delay.

We assume that the message classes (i.e. source-destination pairs) are divided into D disjoint groups and a message is said to belong to group u (denoted by G_u) if its class number is in group u. For a state S of our network model, let $|S|_u$ be the number of group u messages and $|S|$ be the total number of messages (from all groups) in the network. Our two-level flow control can be defined by the following limits:

First level: $|S| \leq L$

Second level: $|S|_u \leq L_u$ for $u = 1,2,\ldots,D$

With these limits, a group u message arriving from outside the network is not allowed to enter the network if $|S| = L$ or $|S|_u = L_u$. We assume that such arrivals are turned away and will not return (i.e. lost).

For convenience, we assume that a fixed routing algorithm is used. Our analysis can easily be generalized to handle random routing. Finally, we assume that there is enough buffer space in each node so that messages routed to a certain node are not blocked from entering that node.

3. ANALYSIS OF TWO-LEVEL FLOW CONTROL

We first consider a state of our network model given by $S = (S_1,S_2,\ldots,S_M)$ where $S_i = (n_{i1},n_{i2},\ldots,n_{iR})$ and n_{ir} is the number of class r messages at channel i. A feasible state is characterized by $\sum_{i=1}^{M} \sum_{r=1}^{R} n_{ir} \leq L$ and $\sum_{i=1}^{M} \sum_{r \in G_u} n_{ir} \leq L_u$. We also require that $n_{ir} = 0$ if class r messages are not routed through channel i. To get the equilibrium state probabilities, we follow the solution technique used by Baskett, et. al. <7> and find that:

$$P(S_1,S_2,\ldots,S_M) = K \prod_{i=1}^{M} n_i! \prod_{r=1}^{R} \frac{1}{n_{ir}!}\left[\frac{\lambda_{ir}}{\mu C_i}\right]^{n_{ir}} \qquad (1)$$

where $n_i = \sum_{r=1}^{R} n_{ir}$ is the total number of messages at channel i and λ_{ir} is the total mean arrival rate of class r messages to channel i conditioned on no customer being lost. Since we have assumed a fixed routing algorithm, λ_{ir} is given by:

$$\lambda_{ir} = \begin{cases} \gamma(r) & \text{if class r messages are routed through channel i} \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

K is the normalization constant obtained by summing all state probabilities and equating the sum to unity.

A complete derivation of the equilibrium state probabilities in equation (1) is given in Appendix A.

We next define a less detailed state description given by $S = (y_1, y_2, \ldots, y_M)$ where $y_i = (m_{i1}, m_{i2}, \ldots, m_{iD})$ and m_{iu} is the number of group u messages at channel i. A feasible state is now characterized by $\sum_{i=1}^{M} \sum_{u=1}^{D} m_{iu} \leq L$, $\sum_{i=1}^{M} m_{iu} \leq L_u$, and $m_{iu} = 0$ if group u messages are not routed through channel i. Using equation (1), we can write:

$$P(y_1,y_2,\ldots,y_M) = \sum_{\sum_{r\in G_u} n_{ir} = m_{iu}} K \prod_{i=1}^{M} n_i! \prod_{r=1}^{R} \frac{1}{n_{ir}!}\left[\frac{\lambda_{ir}}{\mu C_i}\right]^{n_{ir}}$$

or

$$P(y_1,y_2,\ldots,y_M) =$$

$$K \prod_{i=1}^{M} n_i! \left[\prod_{u=1}^{D} \frac{1}{m_{iu}!} \sum_{\sum_{r\in G_u} n_{ir} = m_{iu}} m_{iu}! \prod_{r\in G_u} \frac{1}{n_{ir}!} \left[\frac{\lambda_{ir}}{\mu C_i}\right]^{n_{ir}} \right] \tag{3}$$

Let

$$\xi_{iu} = \sum_{r\in G_u} \lambda_{ir}$$

Equation (3) is reduced to:

$$P(y_1,y_2,\ldots,y_M) = K \prod_{i=1}^{M} n_i! \prod_{u=1}^{D} \frac{1}{m_{iu}!} \left[\frac{\xi_{iu}}{\mu C_i}\right]^{m_{iu}} \tag{4}$$

n_i is now given by $n_i = \sum_{u=1}^{D} m_{iu}$

and the normalization constant K can be computed from:

$$K = \left[\sum_{\substack{\text{all feasible}\\ \text{states}}} \prod_{i=1}^{M} n_i! \prod_{u=1}^{D} \frac{1}{m_{iu}!} \left[\frac{\xi_{iu}}{\mu C_i}\right]^{m_{iu}} \right]^{-1} \tag{5}$$

We now derive the expressions for performance measures such as throughput and mean message delay. Let $P(m_1,m_2,\ldots,m_D)$ be the equilibrium probability that the total number of group u messages in the network is m_u, $u = 1,2,\ldots,D$.

$$P(m_1,m_2,\ldots,m_D) = \sum_{m_{1u}+\ldots+m_{Mu} = m_u,\ \forall u} P(y_1,y_2,\ldots,y_M) \tag{6}$$

The mean number of group u messages in the network is given by:

$$\overline{m}_u = \sum_{\substack{m_v \le L_v, \forall v \\ \& \; m_1 + \ldots + m_D \le L}} m_u \, P(m_1, m_2, \ldots, m_D) \tag{7}$$

The external arrival rate of group u messages is:

$$\gamma_u = \sum_{r \in G_u} \gamma(r) \tag{8}$$

and the throughput of group u messages is given by:

$$\gamma_u^* = \gamma_u \sum_{\substack{m_u < L_u \\ \& \; m_1 + \ldots + m_D < L}} P(m_1, \ldots, m_D) \tag{9}$$

Finally, we use Little's result <11> and get the following expression for the mean delay of group u messages:

$$\overline{T}_u = \overline{m}_u \,/\, \gamma_u^* \tag{10}$$

We see from equations (4) to (10) that before we can compute numerical values for the performance measures, we must first compute the normalization constant K. An efficient algorithm to compute K and $P(m_1, m_2, \ldots, m_D)$ is given in Appendix B.

4. NUMERICAL EXAMPLE AND DISCUSSION OF RESULTS

Consider the example network shown in Figure 1. This network has 5 nodes and 10 channels. There are two message groups. Group 1 contains the source-destination pairs among nodes 1,2, and 3, i.e., $G_1 = \{(1,2),(1,3),(2,1),(2,3),(3,1),(3,2)\}$. Group 2 includes all source-destination pairs not in group 1, i.e., $G_2 = \{(s,t) \mid (s,t) \notin G_1\}$.

The external arrival rate of messages belonging to each source-destination pair is given by the traffic matrix below.

$$
\begin{array}{cc}
 & \text{Destination} \\
\text{Source} &
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & 2\alpha & 2.5\alpha & 0.5\beta & 0.5\beta \\
2 & 2\alpha & 0 & 2\alpha & 0.5\beta & \beta \\
3 & 2.5\alpha & 2\alpha & 0 & \beta & 0.5\beta \\
4 & 0.5\beta & 0.5\beta & \beta & 0 & \beta \\
5 & 0.5\beta & \beta & 0.5\beta & \beta & 0
\end{array}
\end{array}
$$

These rates are expressed in terms of α and β which can vary to reflect the load on the network generated by each message group. The case $\alpha = \beta = 2.0$ corresponds to the average load. All channels are assumed to have the same capacity and the mean message length is chosen such that the mean service time at each channel (i.e., $1/\mu C_i$) has a value of 0.1.

The routing algorithm is based on the shortest path. In our example network, there is a unique shortest path between each pair of nodes.

For convenience, we use the vector (L, L_1, L_2) to denote our two-level control for the case of two message groups. The following four schemes are considered:

A: (15,15,15) -- first level control only

B: (15,12,6) -- two-level control with $L_1/L_2 = 2.0$

C: (15,9,9) -- two-level control with $L_1/L_2 = 1.0$

D: (15,6,12) -- two-level control with $L_1/L_2 = 0.5$

We first fix α at 2.0 (average group 1 load) and study the effect of an increase in β (group 2 load) on network performance. The results are shown in Figures 2 to 5. We observe from Figure 2 that with first level control only, an increase in β causes a considerable degradation to the throughput of group 1. If a two-level control scheme (B, C, or D) is used instead, the amount of degradation is much smaller. The level of group 1 throughput is also less sensitive to variations in β. Our two-level control scheme is therefore capable of preventing the network performance to one group from deteriorating when the load generated by another group is increased. This capability does not exist in the scheme with first level control only.

From Figure 3, we observe that when β is large, the scheme with first level control only gives the highest throughput to group 2. This is not surprising because the load generated by group 2 is increased and there is no second level control. Due to the large degradation in group 1 throughput, the total throughput with first level control alone is reduced significantly. This behaviour is illustrated in Figure 4. With a two-level control scheme, the total throughput is much improved. This total throughput is also less sensitive to variations in β.

The plots in Figures 2 and 3 also show the effect of the ratio L_1/L_2 on the throughput of each group. In particular, the level of group 1 (or 2) throughput is increased when a larger (or smaller) ratio is used for L_1/L_2. As to the mean message delay, we observe from Figure 5 that there is no significant degradation when β is large, and a control scheme which gives a higher throughput to a particular group also results in a longer mean message delay for the same group.

The results for the case that β is fixed at 2.0 (average group 2 load) and α (group 1 load) is allowed to vary are shown in Figures 6 to 9. The performance characteristics of the four control schemes are analogous to those shown in Figures 2 to 5. The only difference is that the total throughput under first level control only does not degrade until α is larger than 6.0.

We next study the case of a simultaneous increase in α and β (with $\alpha = \beta$). This corresponds to a balanced increase in the network load by both groups. The results are shown in Figures 10 to 13. Unlike the previous cases of α or β increasing only, the four control schemes have similar performance characteristics. The two-level scheme with $(L, L_1, L_2) = (15, 12, 6)$ gives the highest total throughput.

Our discussion so far has been limited to a fixed value of $L = 15$. To investigate the effect of L on network performance, we allow the value of L in control schemes B, C, and D to vary from 12 to 18. The results for the case of $\alpha = \beta = 2.0$ (average load for both groups) are shown in Figures 14 to 16. We observe that increasing L tends to increase the throughput of each group, but also introduces longer mean message delays.

The results of this section indicate that our two-level control scheme, with properly selected L, L_1, and L_2, can be used to give preferential treatment to a particular message group in terms of a higher throughput. These limits can also be selected such that the mean message delay of each group is maintained at an acceptable level.

5. CONCLUSION

We have used a queueing network model to analyse the performance characteristics of a two-level flow control scheme. Numerical results have shown that this scheme is capable of preventing the network performance from deteriorating when the load generated by one message group is increased. This scheme can also be used to give preferential treatment to a particular message group in terms of a higher throughput.

6. BIBLIOGRAPHY

1. Rudin, H. "Flow-Control: Session Chairman's Remarks," International Conference on Computer Communications, Toronto, Canada, Aug, 3-6, 1976, pp. 463-466.

2. Pennotti, M. and M. Schwartz. "Congestion Control in Store and Forward Tandem Links," IEEE Trans. on Comm., Vol. COM-23, No. 12, Dec. 1975, pp. 1434-1443.

3. Davies, D. "The Control of Congestion in Packet Switching Networks," IEEE Trans. on Comm., Vol. COM-20, No. 3, June 1972, pp. 546-550.

4. Price, W. "Simulation of a Packet-Switched Data Network Operating under Isarithmic Control with Revised Link and Node Protocol," NPL Report COM 71, September 1973.

5. Price, W. "A study of Bifurcated Routing in a Data Network and the Effect of Isarithmic Flow Control in this Context," NPL Report COM 72, March 1974.

6. Kahn, R. and W. Crowther. "Flow Control in a Resource Sharing Computer Network," IEEE Trans. on Comm., Vol. COM-20, No. 3, June 1972, pp. 539-546.

7. Baskett, F., Chandy, K., Muntz, R. and F. Palacios. "Open, Closed and Mixed Networks of Queues with Different Classes of Customers," Journal of ACM, Vol. 22, No. 2, April 1975, pp. 248-260.

8. Chatterjee, A., Georganas, N. and P. Verma. "Analysis of a Packet Switched Network with End-to-End Congestion Control and Random Routing," International Conference on Computer

Communications, Toronto, Canada, Aug. 3-6, 1976, pp. 488-494.

9. Kleinrock, L. "Communication Nets -- Stochastic Message Flow and Delays," McGraw-Hill, New York, 1964.

10. Reiser, M. and H. Kobayashi. "On the Convolution Algorithm for Separable Queueing Networks," Proc. International Symposium on Computer Performance Modeling, Measurement and Evaluation, Harvard University, March 1976, pp. 109-117.

11. Little, J. "A Proof of the Queueing Formula $L = \lambda W$," Operations Research, Vol. 9, 1961, pp. 383-387.

APPENDIX A

DERIVATION OF EQUILIBRIUM STATE PROBABILITIES

We first define γ_{ir} to be the mean external arrival rate of class r messages to channel i. Since we have assumed a fixed routing algorithm,

$$\gamma_{ir} = \begin{cases} \gamma(r) & \text{if class r messages are first routed through channel i} \\ 0 & \text{otherwise} \end{cases}$$

We also define

$$\phi_{ji}(r) = \begin{cases} 1 & \text{if class r messages are routed from channel j to channel i} \\ 0 & \text{otherwise} \end{cases}$$

and

$$\phi_{i}(r) = \begin{cases} 1 & \text{if class r messages are last routed through channel i} \\ 0 & \text{otherwise} \end{cases}$$

With the above definitions, we can verify that λ_{ir}, as defined by equation <2>, satisfies the following equation:

$$\lambda_{ir} = \gamma_{ir} + \sum_{j=1}^{M} \lambda_{jr} \phi_{ji}(r) \qquad i = 1, 2, \ldots, M \qquad \text{(A.1)}$$

Following the development in <7> , we first consider the state description given by $S = (Z_1, Z_2, \ldots, Z_M)$ where $Z_i = (x_{i1}, x_{i2}, \ldots, x_{in_i})$, n_i is the total number of messages at channel i, and x_{ij} is the class number of the message who is j-th in FCFS order at channel i. For those states S such that $|S| < L$ and $|S|_u < L_u$, $u = 1,2,\ldots,D$, the equilibrium state probabilities P(S) satisfy the following flow balance equation:

$$P(Z_1, Z_2, \ldots, Z_M) \left[\sum_{i=1}^{M} \sum_{r=1}^{R} \gamma_{ir} + \sum_{i=1}^{M} \delta(n_i)\, \mu C_i \right]$$

$$= \sum_{i=1}^{M} \delta(n_i)\; P(Z_1, Z_2, \ldots, Z_i^-, \ldots, Z_M)\, \gamma_{ix_{in_i}}$$

$$+ \sum_{i=1}^{M} \sum_{j=1}^{M} \delta(n_i)\, P(Z_1, \ldots, Z_i^-, \ldots, Z_j(+x_{in_i}), \ldots, Z_M)\, \mu C_j \phi_{ji}(x_{in_i})$$

$$+ \sum_{i=1}^{M} \sum_{r=1}^{R} P(Z_1, Z_2, \ldots, Z_i(+r), \ldots, Z_M)\, \mu C_i \phi_i(r) \qquad \text{(A.2)}$$

where

$$Z_i^- = (x_{i1}, x_{i2}, \ldots, x_{i,n_i-1})$$

$$Z_i(+r) = (r, x_{i1}, x_{i2}, \ldots, x_{in_i})$$

and

$$\delta(n_i) = \begin{cases} 1 & n_i > 0 \\ 0 & n_i = 0 \end{cases}$$

Equation (A.2) also applies to states with $|S| = L$ or $|S|_u = L_u$ provided that the following modifications are made:

(a) If $|S| = L$, the first sum on the LHS and the last sum on the RHS are both zero.

(b) For $u = 1,2,\ldots,D$, if $|S|_u = L_u$, γ_{ir} on the LHS and $P(Z_1, Z_2, \ldots, Z_i(+r), \ldots, Z_M)$ on the RHS are both zero for each $r \in G_u$.

The solution to the equilibrium state probabilities is given by:

$$P(Z_1, Z_2, \ldots, Z_M) = K \prod_{i=1}^{M} \prod_{j=1}^{n_i} (\lambda_{ix_{ij}} / \mu C_i) \tag{A.3}$$

where K is the normalization constant. This solution can be verified by checking that the flow balance equation (A.2) is satisfied. If we substitute (A.3) into (A.2) and divide both sides by $P(Z_1, Z_2, \ldots, Z_M)$, we obtain:

$$\begin{aligned}
&\sum_{i=1}^{M} \sum_{r=1}^{R} \gamma_{ir} + \sum_{i=1}^{M} \delta(n_i)\, \mu C_i \\
&= \sum_{i=1}^{M} \delta(n_i)\, (\mu C_i / \lambda_{ix_{in_i}})\ \gamma_{ix_{in_i}} \\
&\quad + \sum_{i=1}^{M} \sum_{j=1}^{R} \delta(n_i)\, (\mu C_i / \lambda_{ix_{in_i}})\, \lambda_{jx_{in_i}}\, \phi_{ji}(x_{in_i}) \\
&\quad + \sum_{i=1}^{M} \sum_{r=1}^{R} \lambda_{ir} \phi_i(r)
\end{aligned} \tag{A.4}$$

When there is no loss of customers, the total external arrival rate must equal the total departure rate. We thus have:

$$\sum_{i=1}^{M} \sum_{r=1}^{R} \gamma_{ir} = \sum_{i=1}^{M} \sum_{r=1}^{R} \lambda_{ir} \phi(r)$$

After cancelling the first sum on the LHS with the last sum on the RHS, it is easy to see that (A.4) is reduced to the relationship in equation (A.1).

We observe from (A.3) that the equilibrium state probabilities are not affected by the FCFS order of messages at each channel. We thus define, as in <7>, a less detailed state description given by $S = (S_1, S_2, \ldots, S_M)$ where $S_i = (n_{i1}, n_{i2}, \ldots, n_{iR})$ and n_{ir} is the number of class r messages at channel i. Using the results of Baskett, et. al. <7>, $P(S_1, S_2, \ldots, S_M)$ is given by:

$$P(S_1, S_2, \ldots, S_M) = K \prod_{i=1}^{M} n_i ! \prod_{r=1}^{R} \frac{1}{n_{ir}!} \left[\frac{\lambda_{ir}}{\mu C_i} \right]^{n_{ir}}$$

APPENDIX B

EFFICIENT COMPUTATION OF K AND $P(m_1, m_2, \ldots, m_D)$

The normalization constant K, as given by equation (5), can be computed from:

$$K^{-1} = \sum_{\substack{\text{all feasible}\\ \text{states}}} F_1(y_1)\, F_2(y_2) \ldots F_M(y_M)$$

where

$$F_i(y_i) = n_i! \prod_{u=1}^{D} \frac{1}{m_{iu}!} \left[\frac{\xi_{iu}}{\mu C_i} \right]^{m_{iu}}$$

and a feasible state is characterized by $\sum_{i=1}^{M} \sum_{u=1}^{D} m_{iu} \le L$ and $\sum_{i=1}^{M} m_{iu} \le L_u$ for $u = 1,2,\ldots,D$.

The efficient algorithm to compute K is based on the work of Reiser and Kobayashi <10>. Let $y_i^* = (m_{i1}^*, m_{i2}^*, \ldots, m_{iD}^*)$ where $m_{iu}^* = \sum_{j=1}^{i} m_{ju}$ is the total number of group u messages in channels $1,2,\ldots,i-1$ and i. The set of feasible y_i^*'s is characterized by $\sum_{j=1}^{i} \sum_{u=1}^{D} m_{ju} \le L$ and $\sum_{j=1}^{i} m_{ju} \le L_u$ for $u = 1,2,\ldots,D$. We also define:

$$F_i^*(y_i^*) = \sum_{y_1 + \ldots + y_i = y_i^*} F_1(y_1)\, F_2(y_2) \ldots F_i(y_i)$$

It is easy to see that $F_1^*(y_1^*) = F_1(y_1^*)$, and for $i > 1$, $F_i^*(y_i^*)$ can be computed from the following recursive formula <10>:

$$F_i^*(y_i^*) = F_{i-1}^*(y_i^*) + \sum_{u=1}^{D} F_i^*(y_i^* - \ell_u)(\xi_{iu}/\mu C_i)$$

where

$$\ell_u = (0, \ldots, 0, \overset{\overset{\text{u-th}}{\downarrow}}{1}, 0, \ldots, 0)$$

K can then be computed from:

$$K^{-1} = \sum_{\text{all feasible } y_M^*} F_M^*(y_M^*)$$

The amount of computation is $O(MDL_1L_2\ldots L_D)$, and the storage requirement is $O(L_1L_2\ldots L_D)$.

To compute $P(m_1,m_2,\ldots,m_D)$, the equilibrium probability that the total number of group u messages in the network is m_u, $u = 1,2,\ldots,D$, we see from equation (7) that $P(m_1,m_2,\ldots,m_M)$ can be rewritten as:

$$P(m_1, m_2, \ldots, m_D) = K \; F_M^*(y_M^*) \Big|_{y_M^* = (m_1, m_2, \ldots, m_D)}$$

Both $F_M^*(y_M^*)$ and K can be computed efficiently.

Fig. 1. Example Network

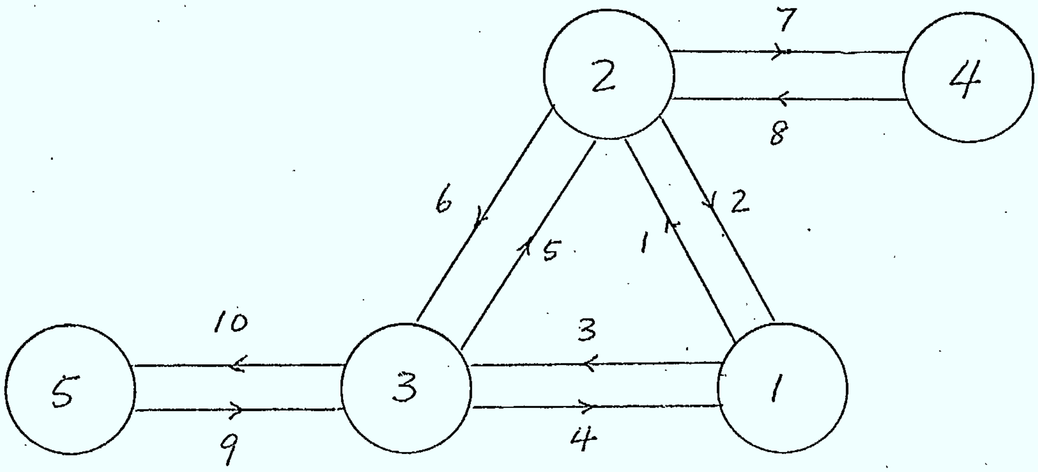


Fig. 2. Throughput of group 1 vs. β

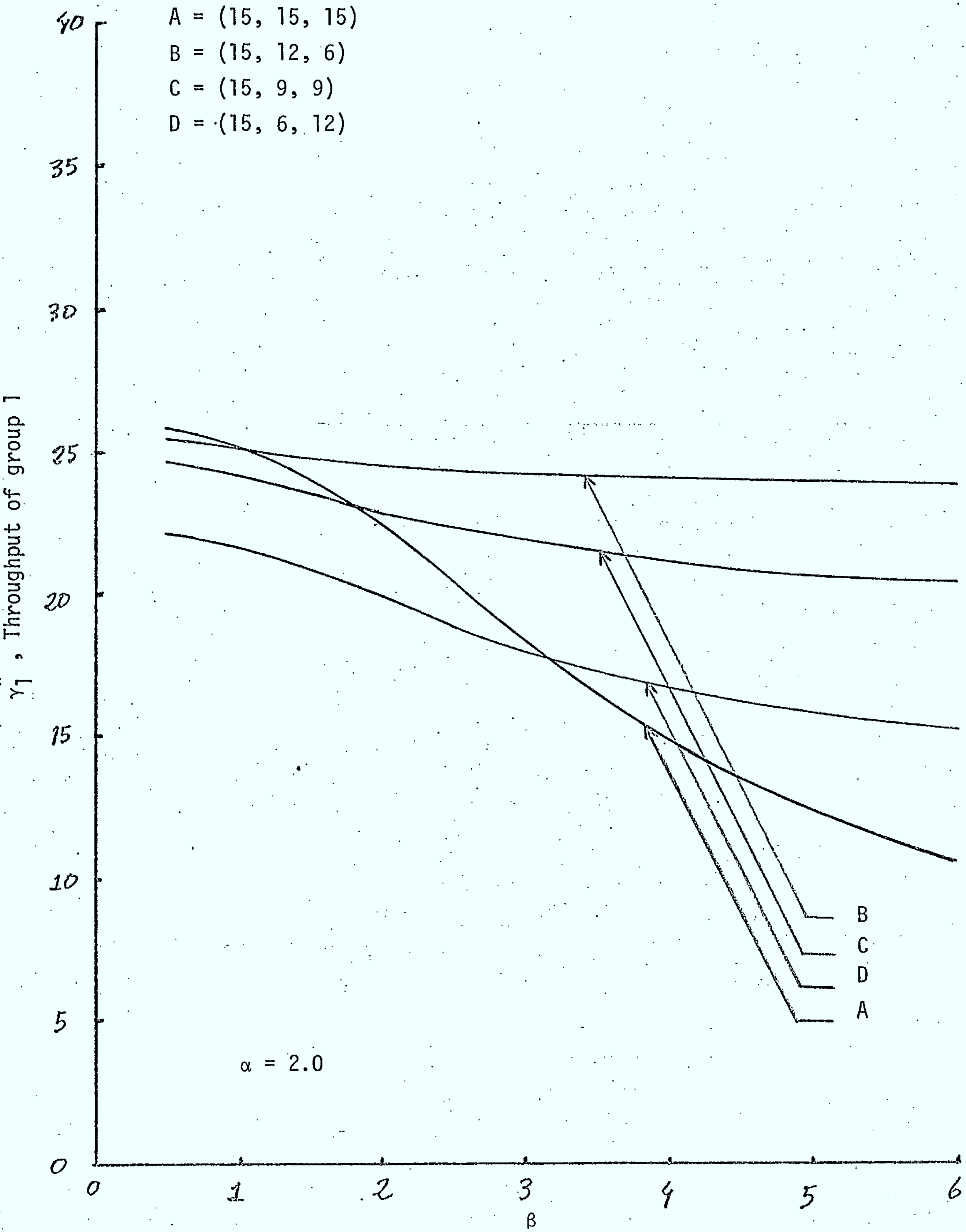


Fig. 3. Throughput of group 2 vs. β

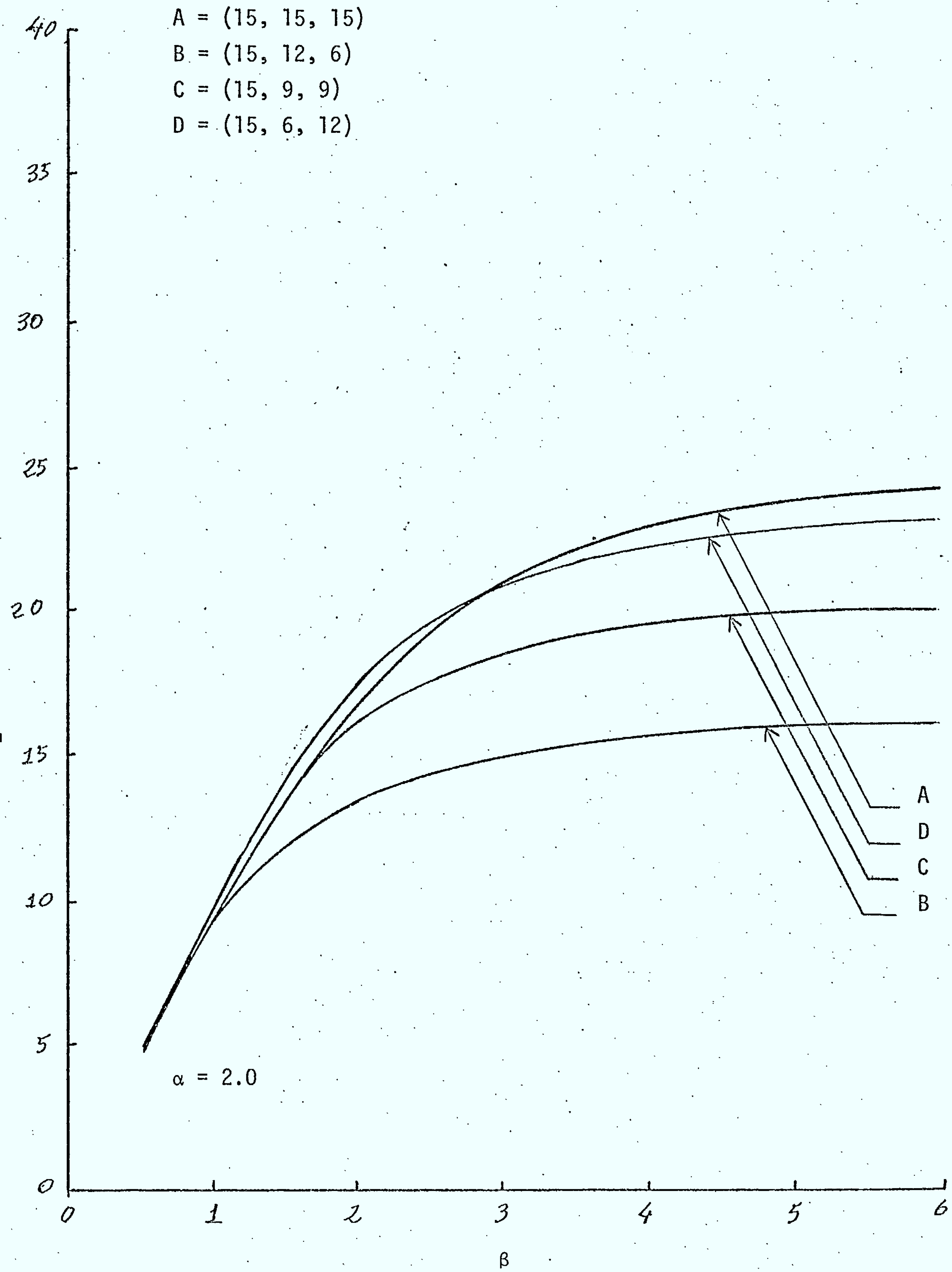


Fig. 4. Total throughput vs. β

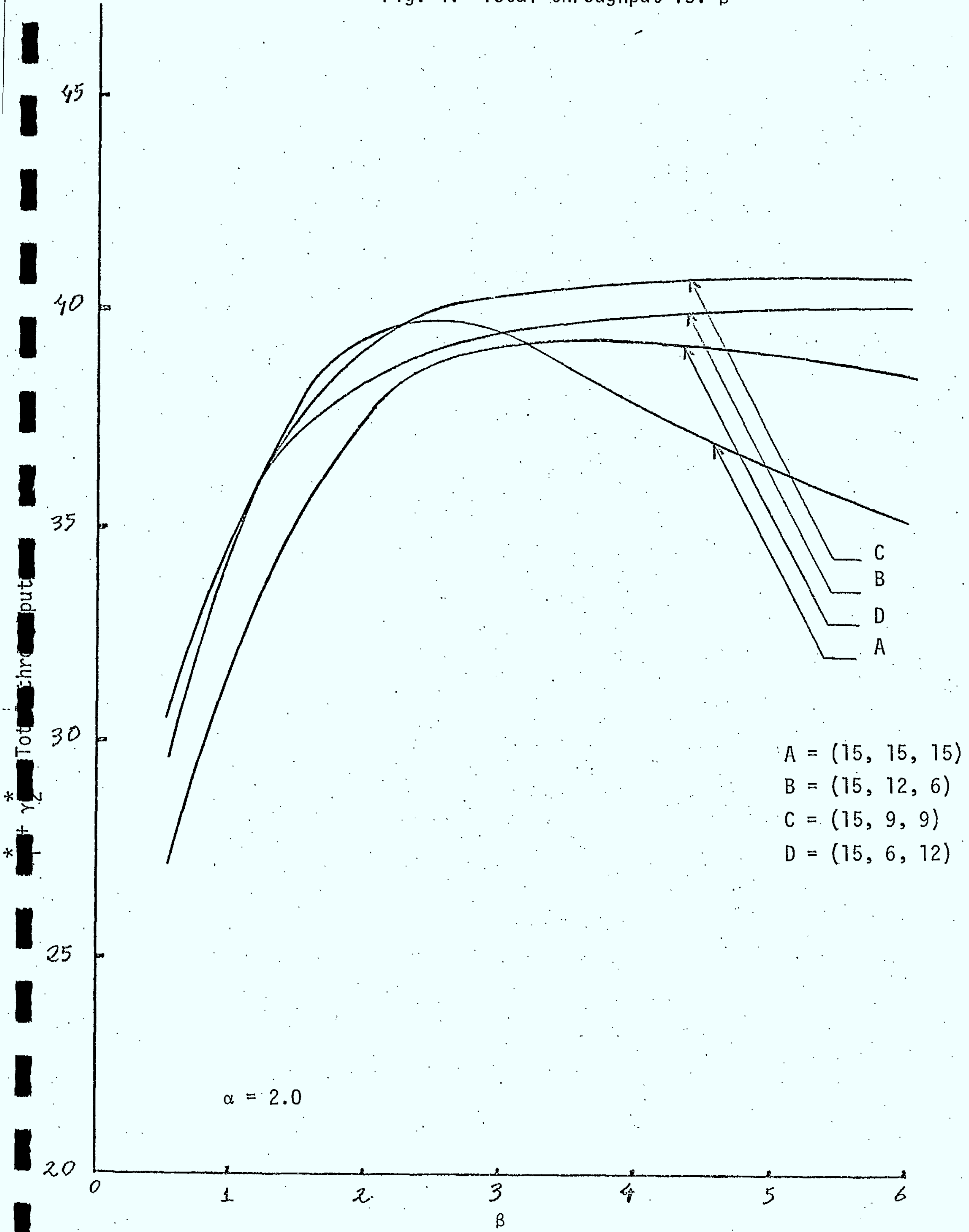


Fig. 5. Mean message delays vs. β

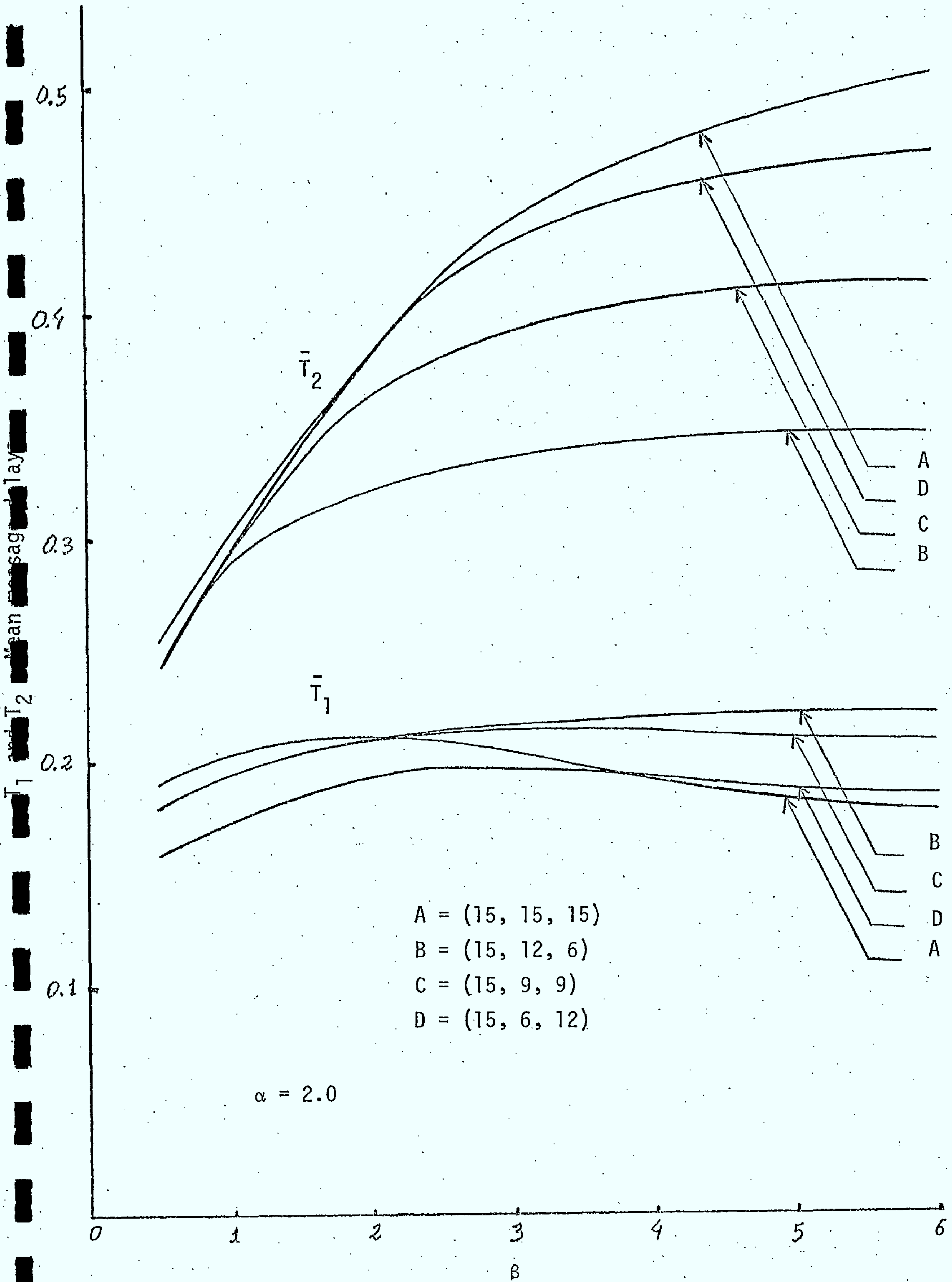


Fig. 6. Throughput of group 1 vs. α

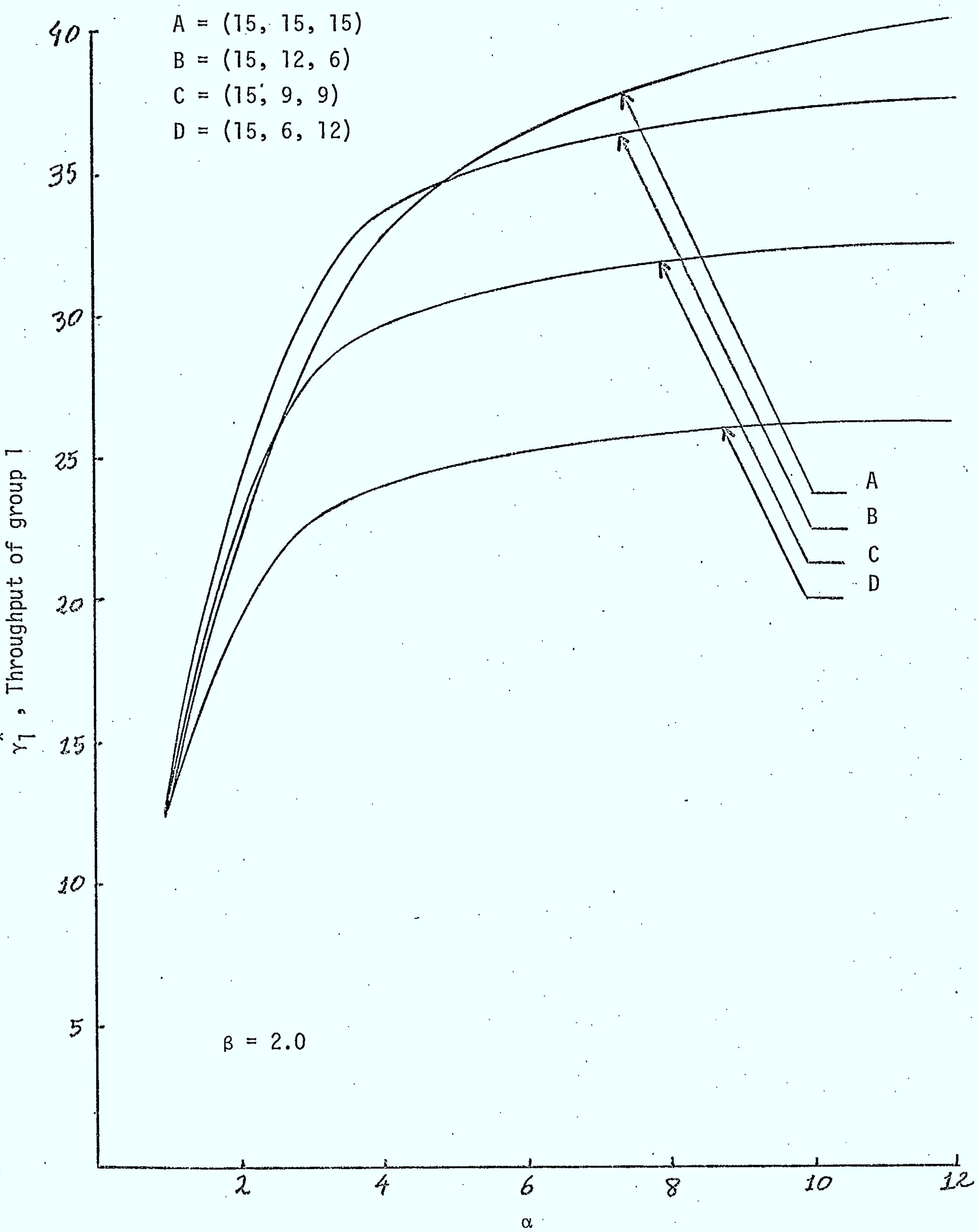


Fig. 7. Throughput of group 2 vs. α

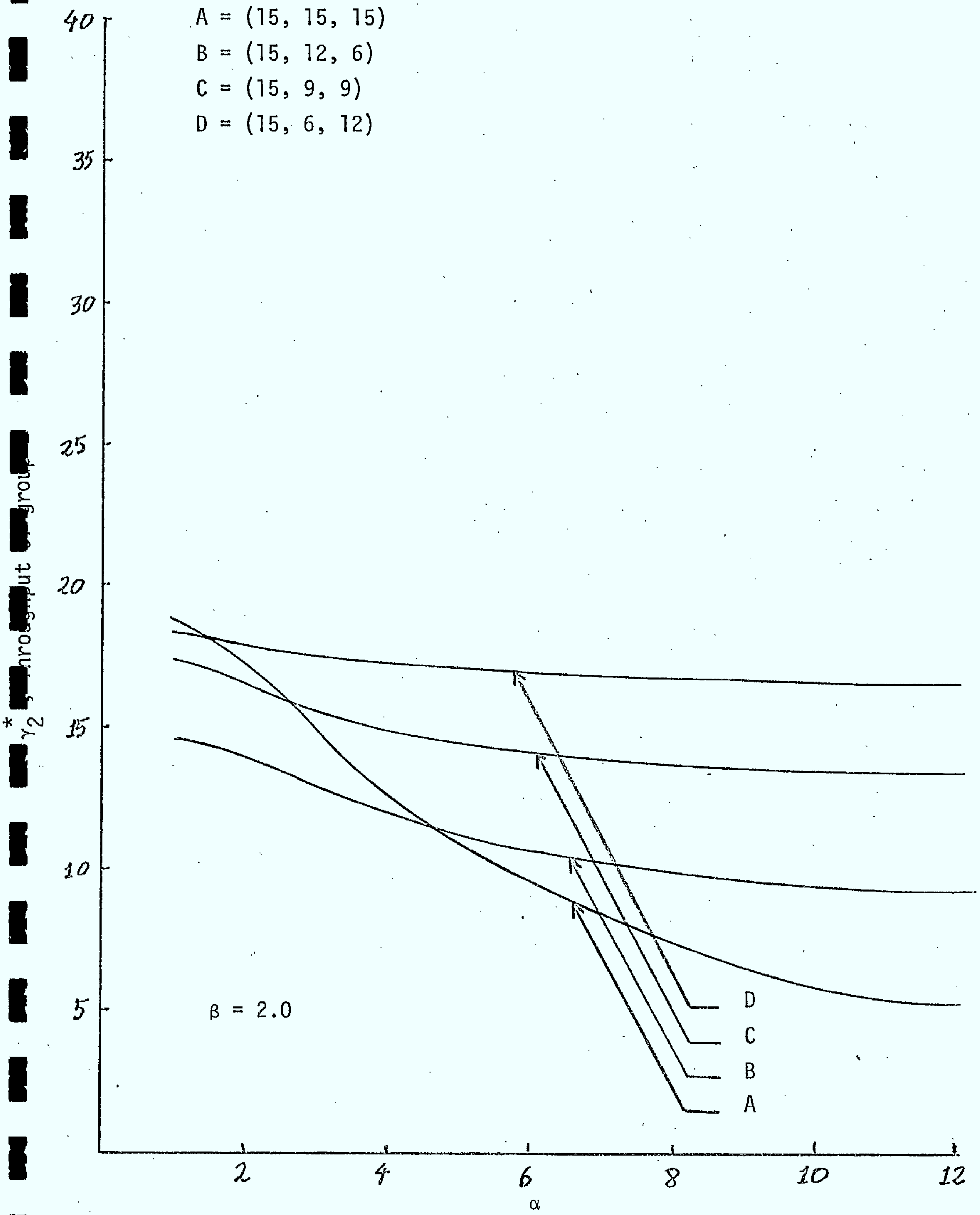


Fig. 8. Total throughput vs. α

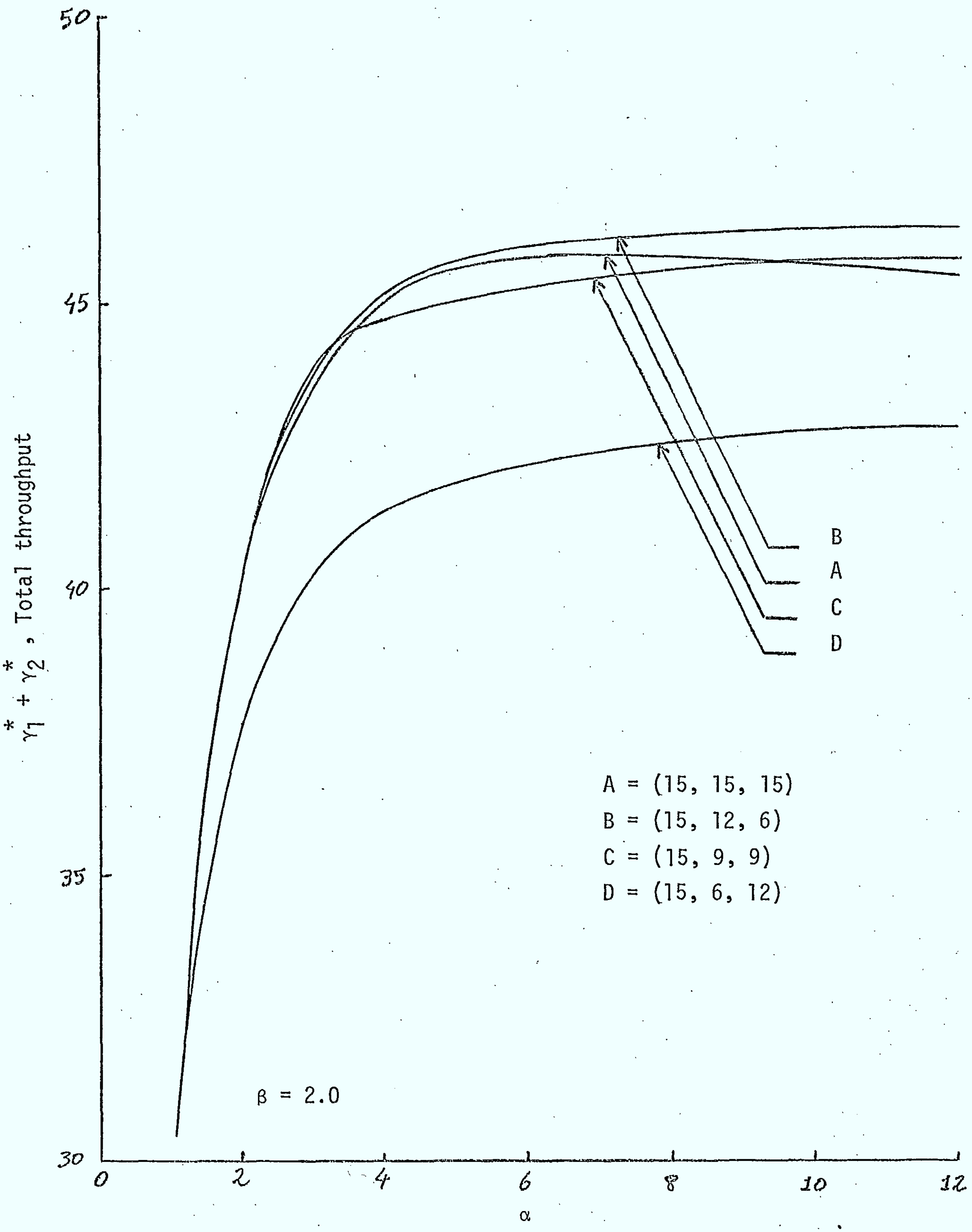


Fig. 9. Mean message delays vs. α

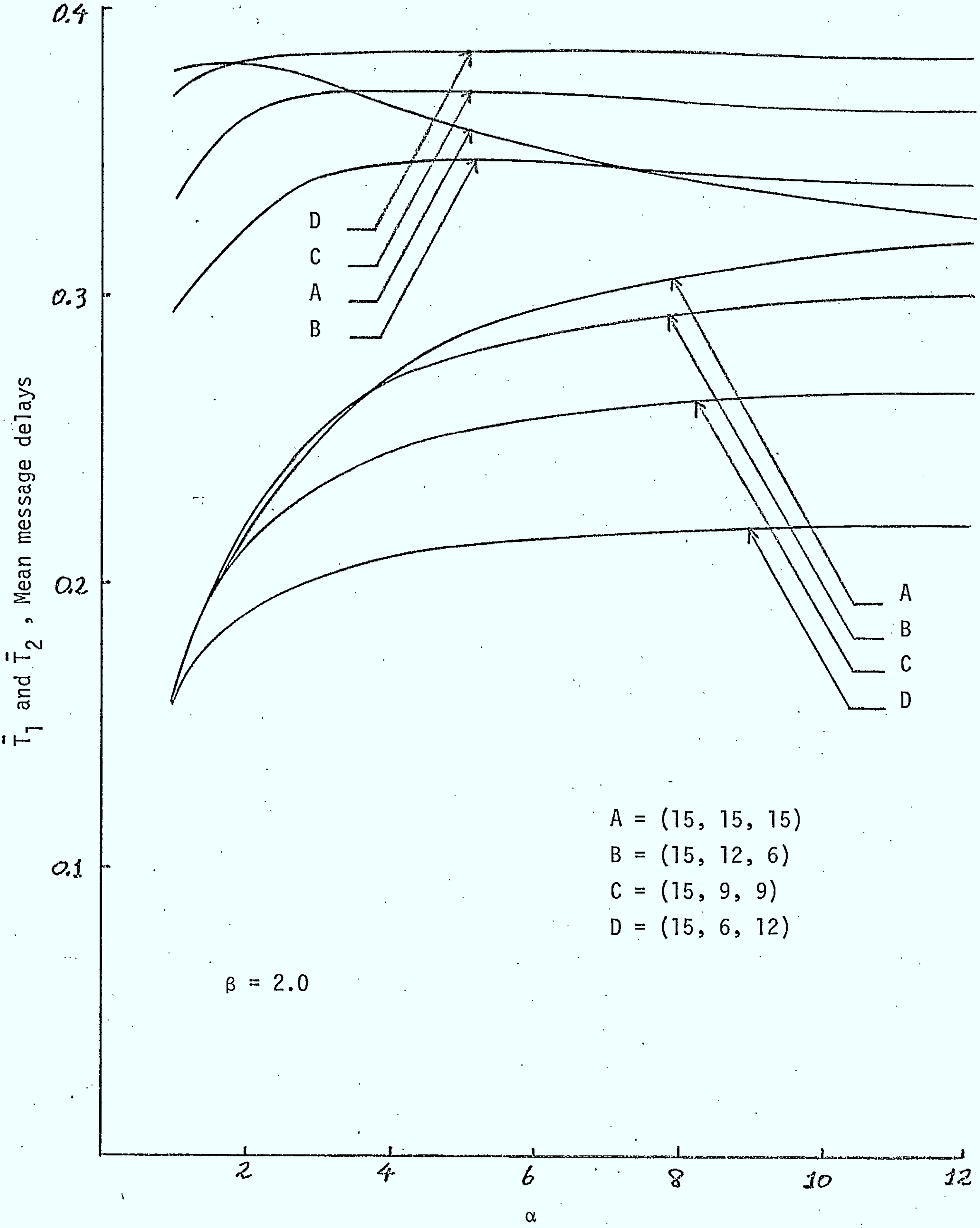


Fig. 10. Throughput of group 1 vs. α, β

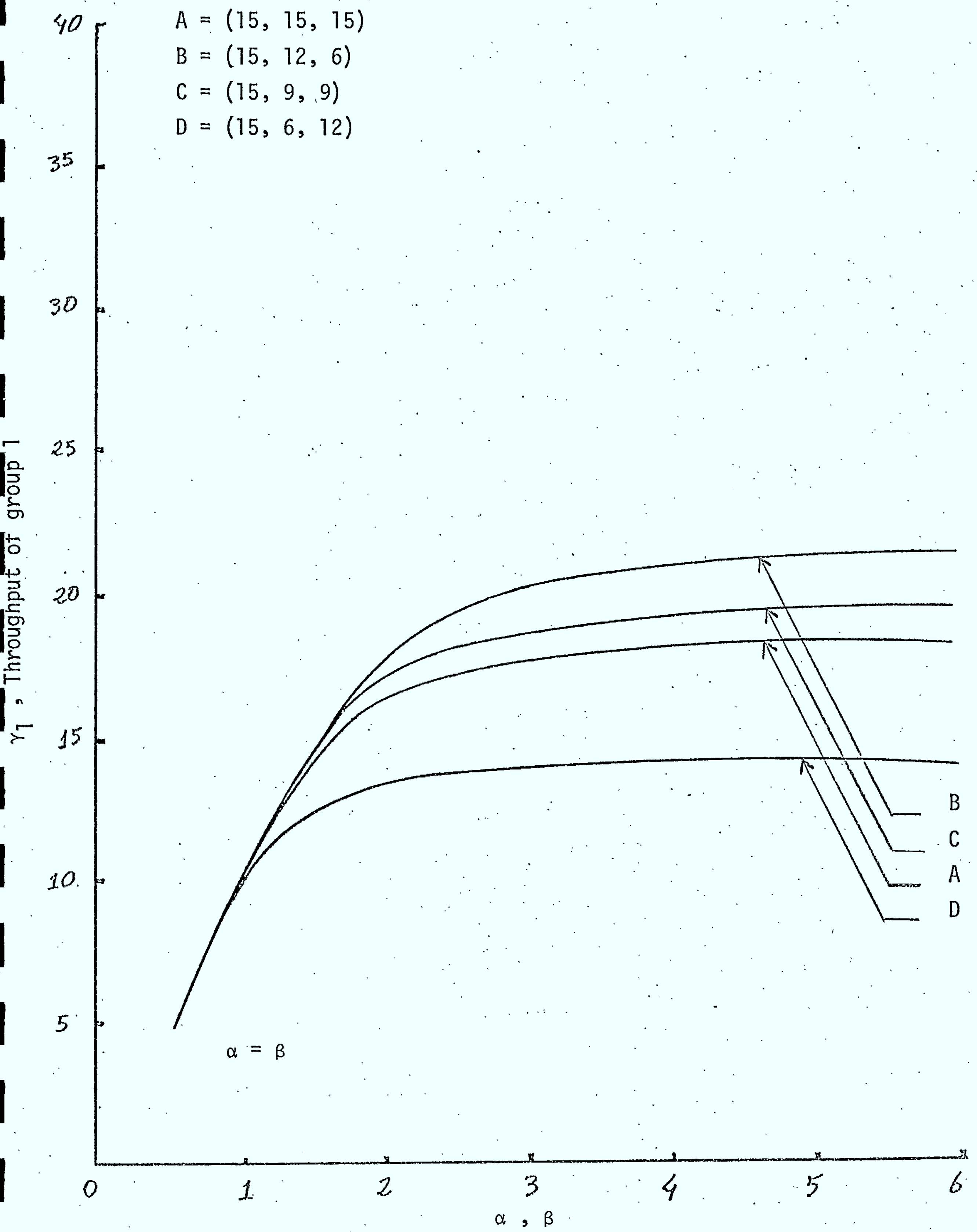


Fig. 11. Throughput of group 2 vs. α, β

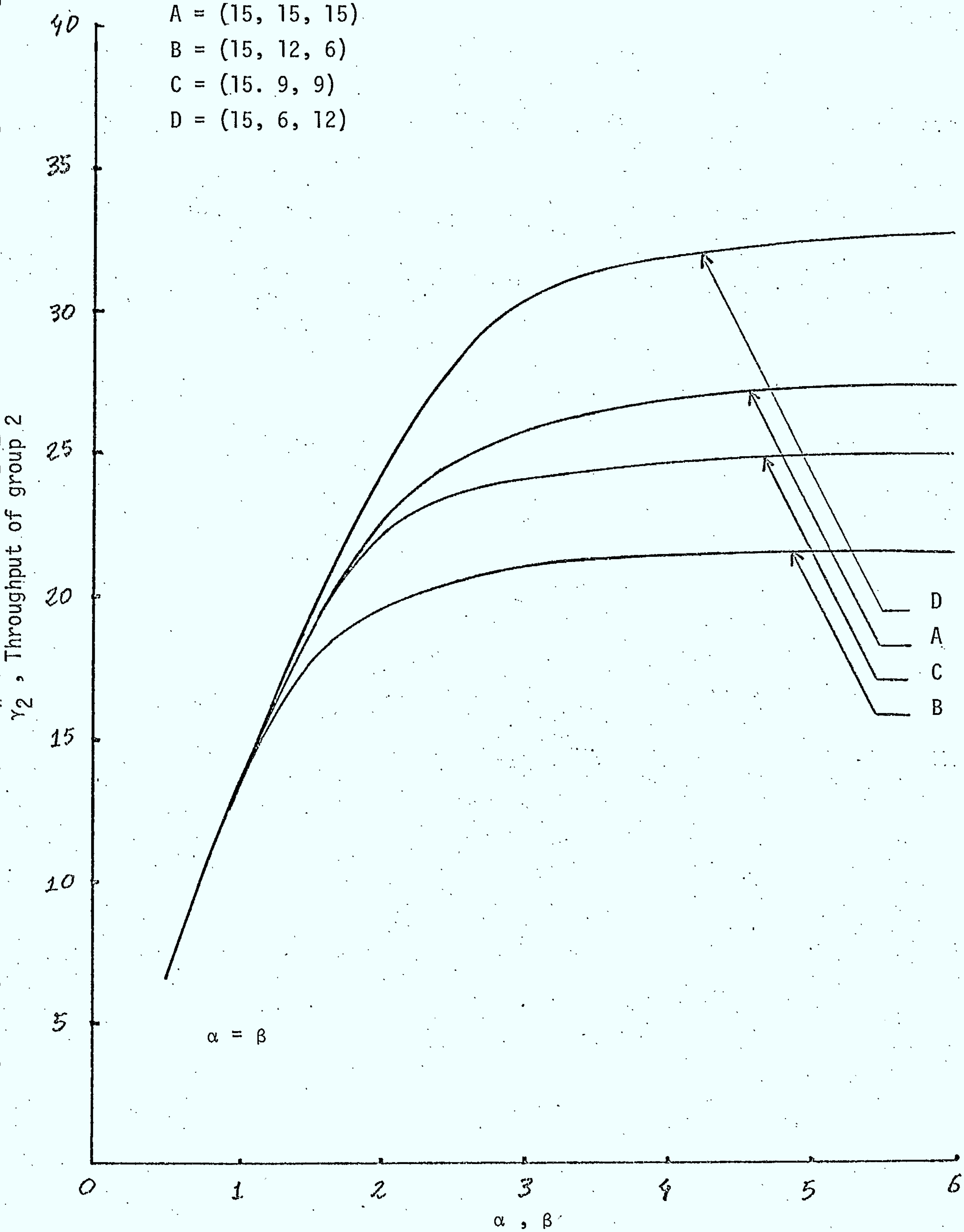


Fig. 12. Total throughput vs. α, β

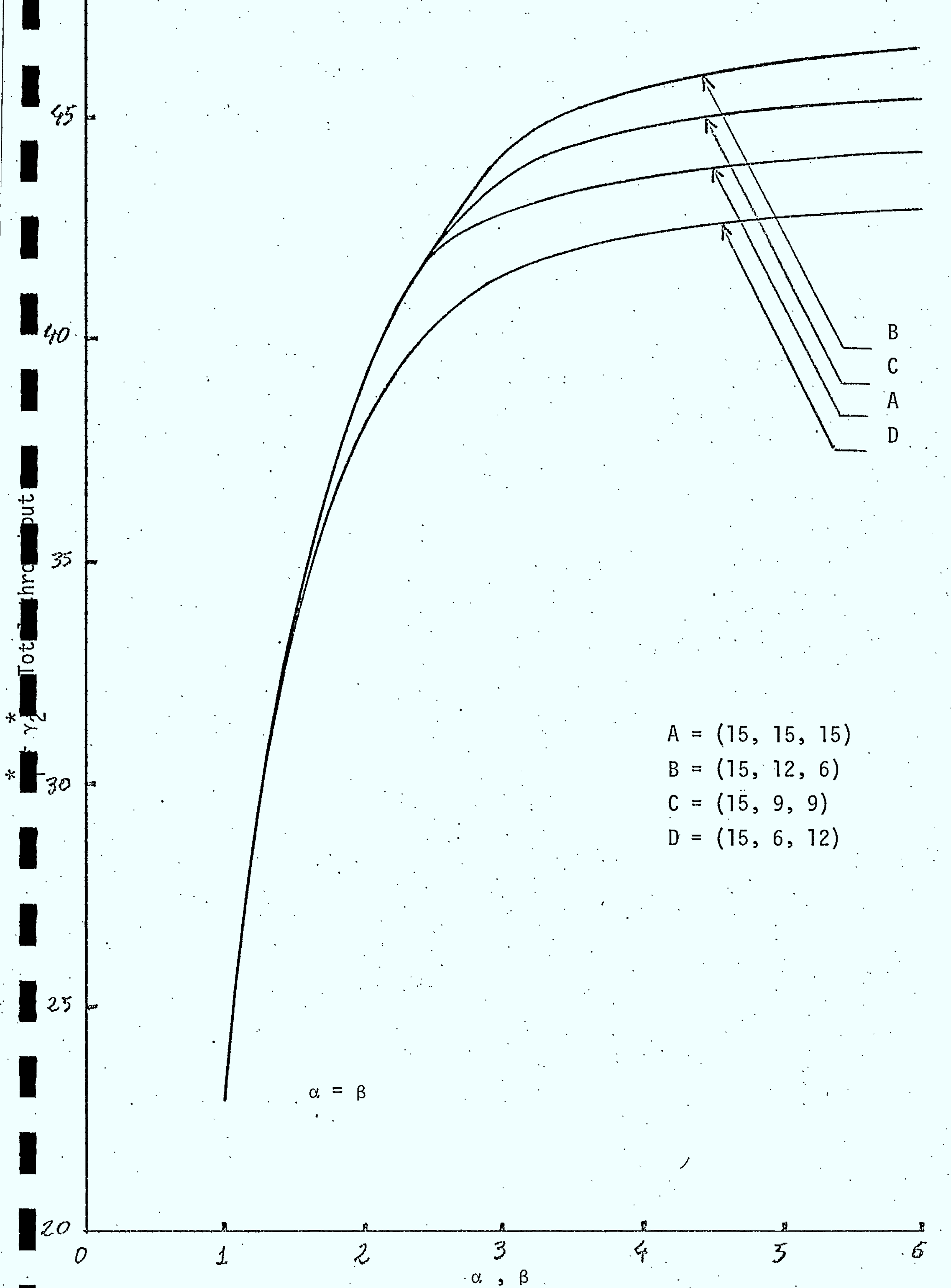


Fig. 13. Mean message delays vs. α, β

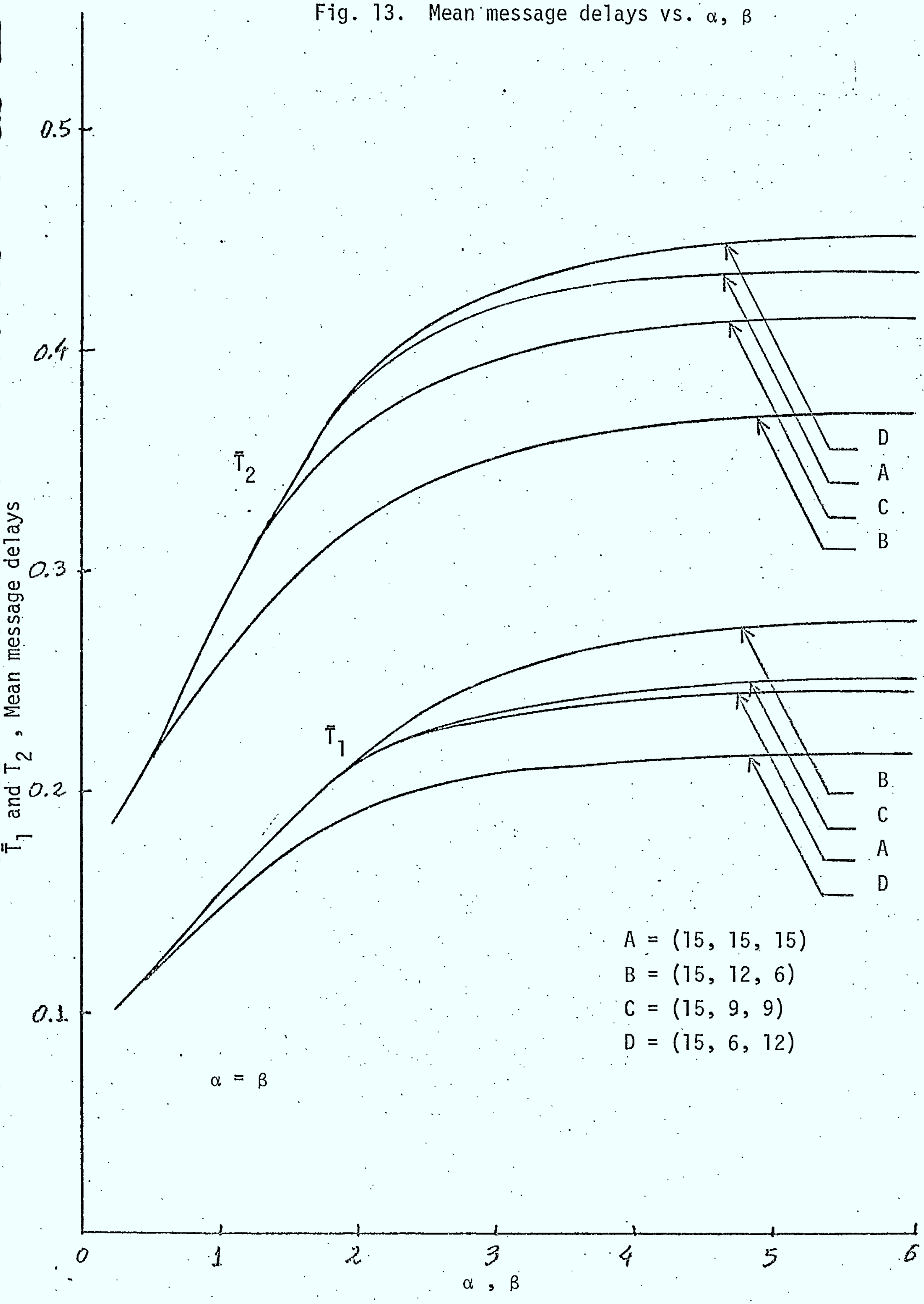


Fig. 14. Throughput of group 1 and group 2 vs. L

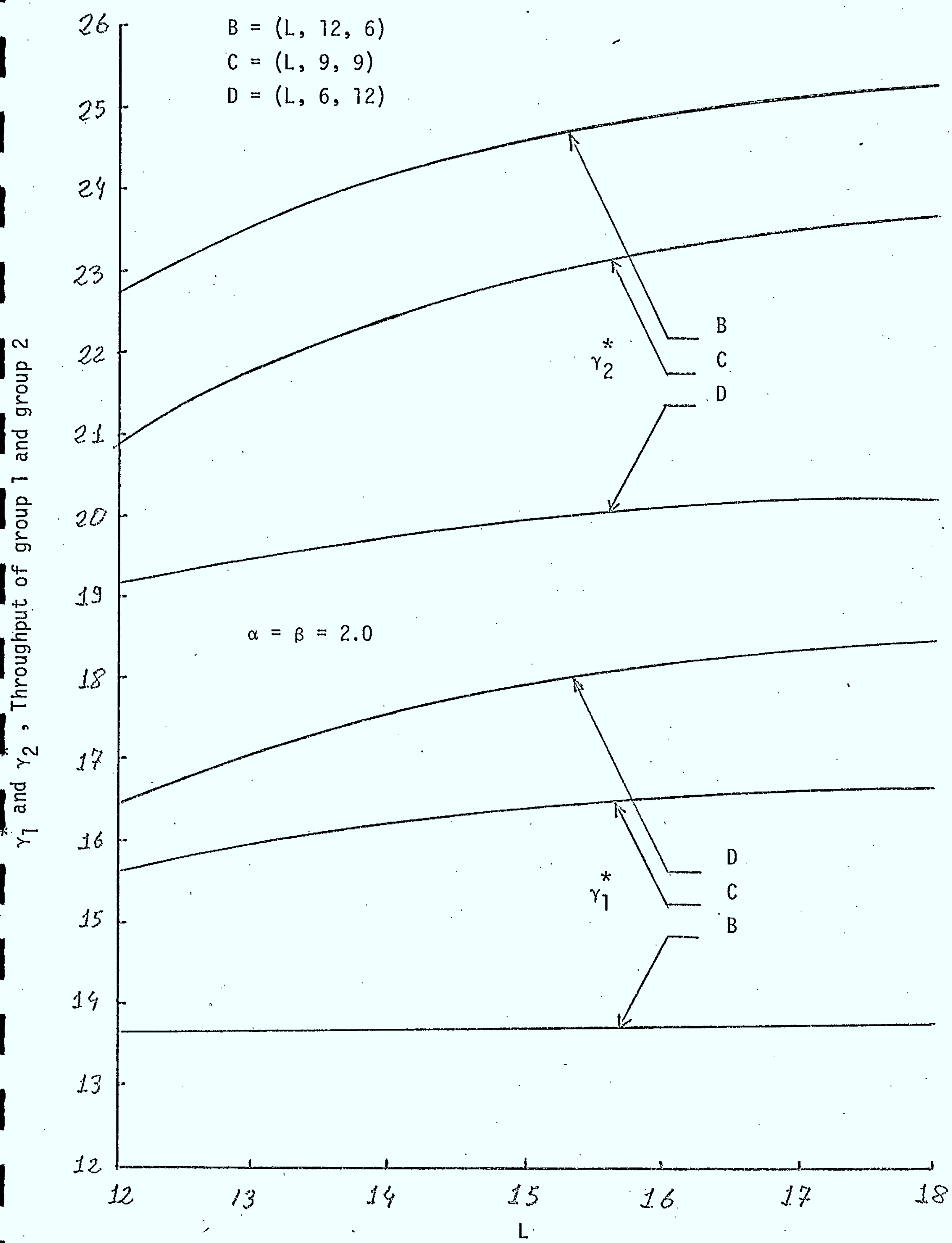


Fig. 15. Total throughput vs. L

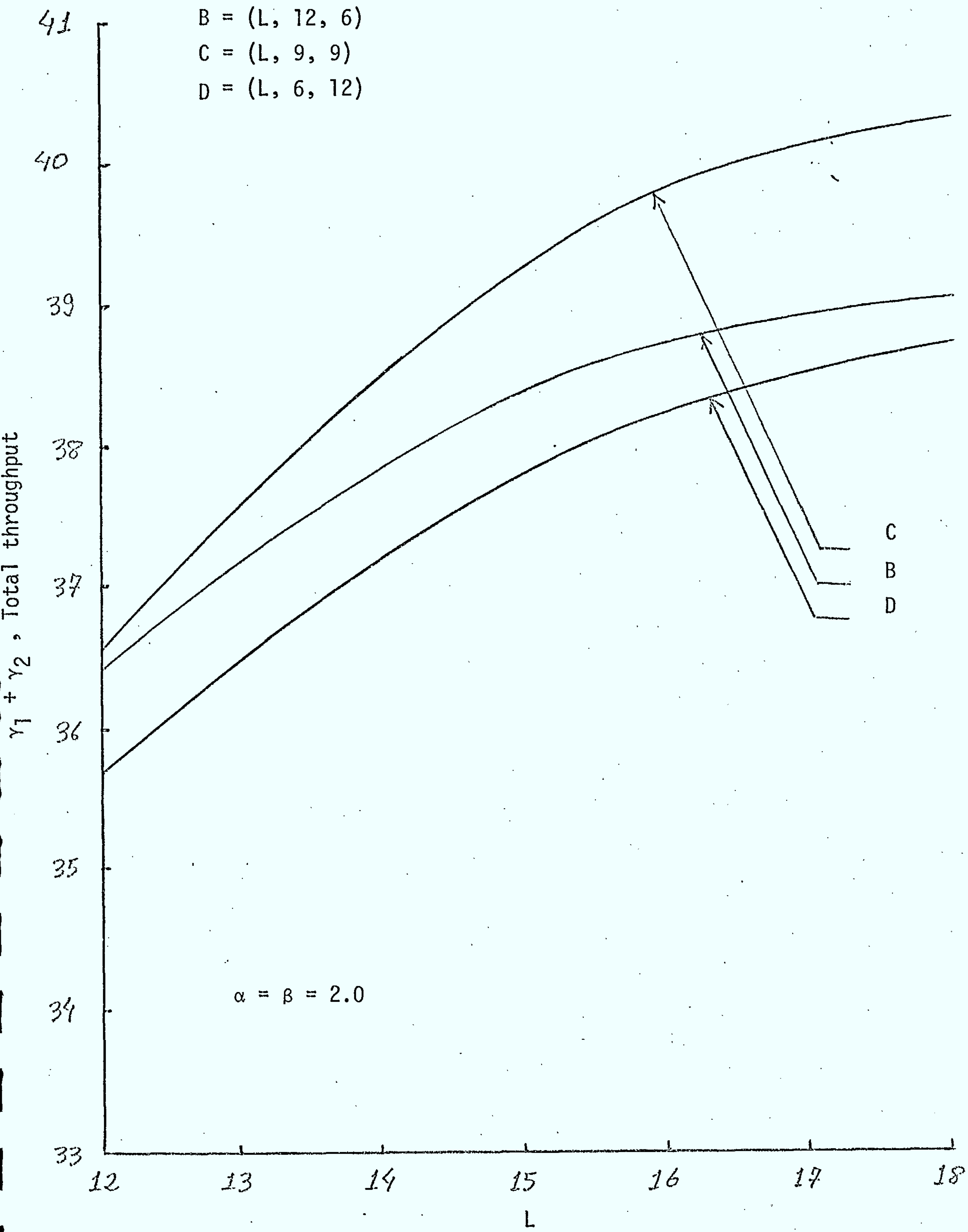
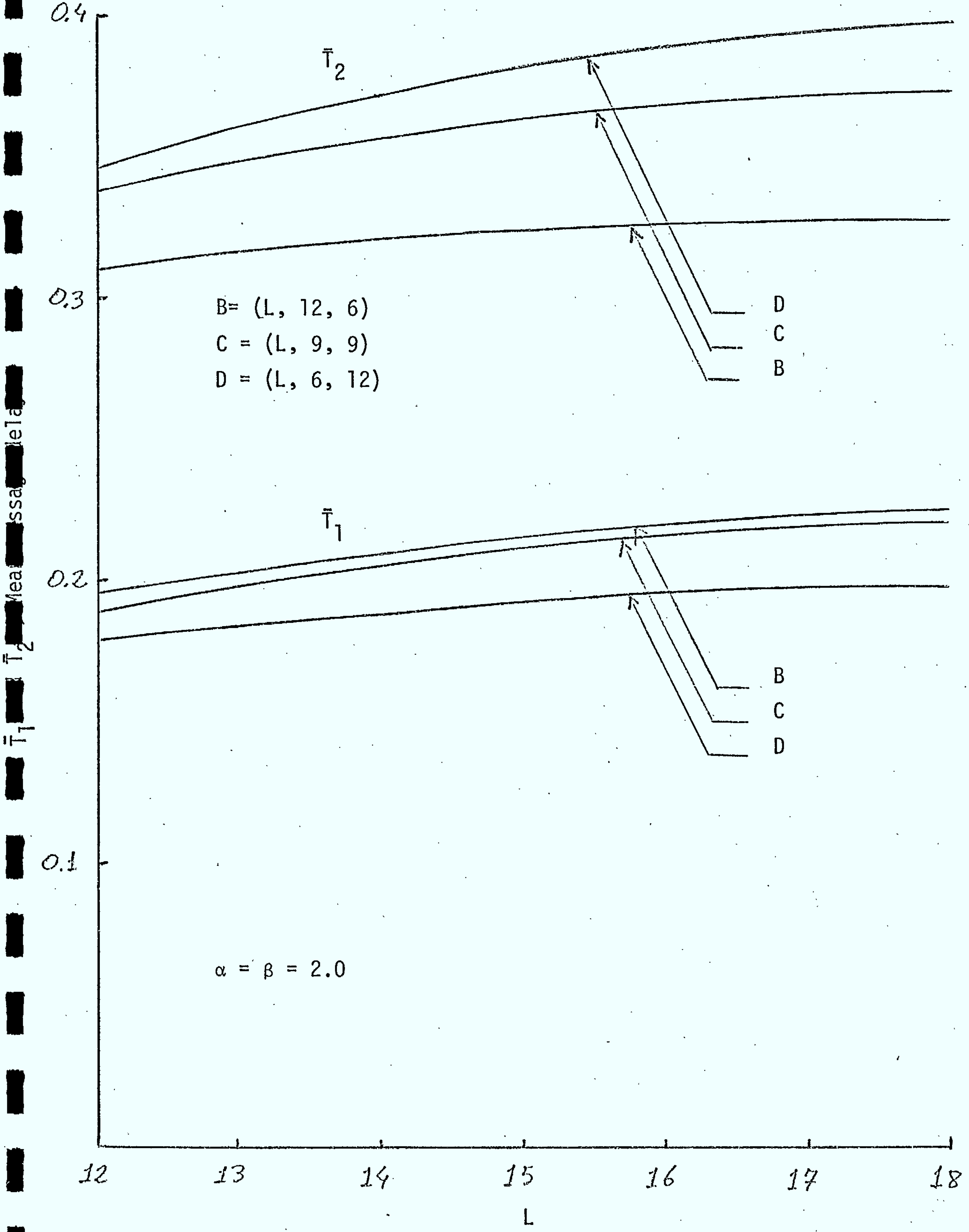


Fig. 16. Mean message delays vs. L



APPENDIX C (by A.T. Bowler)

1. Cable Communications Policy Issues: An Overview

by Kan Chen (A Summary)

IEEE Transactions on Systems, Man, and Cybernetics,

Volume 6 (11), 1976

This paper gives a general discussion of cable potentials for local loop type applications.

The first section of the discusses a set of possibilities from a "wired city" scenario. These are listed here with a short description and some of the more obvious policy questions raised.

Application: A Community Information System
This would provide, on demand, reports on items of local interest. This should be more economical than trying to distribute everything to everyone via radio, TV or newspaper.
Questions: Who will pay? How is accuracy ensured?

Application: A Consumer Information System
This would function in a manner like the above but would have price and availability data, as well as Consumer Reports type information.
Questions: Same as above but the answers are probably different.

Application: Teleshopping and Electronic Funds Transfer
The author predicts that teleshopping is a fair distance in the future, but EFT is just around the corner.
Questions: Will equipment costs, and the cost of reorganizing the delivery system force all but the very largest firms out of competition?

Application: Tele-education
The author believes that this will be used mainly (not exclusively) for non-instructive purposes that faculty currently do "reluctantly and poorly". He believes the problems will be mainly how to maintain high quality software. Such software has a high initial cost and can become obsolete quickly.

Questions: There are some questions about job security for teachers, and copyright problems.

Application: Telemedicine
In this a paramedic would use remote telemetry and television equipment to allow a doctor to examine a patient. This could be useful in remote areas, or for consultation with a specialist.
Questions: How are roles divided between nurse, doctor and paramedic? How is legal responsibility divided? Is this cost effective in comparison with mobile clinics, or patient pickup services?

Application: Social Experimentation
The author seems to be referring to trial runs of services such as the previous five.
Questions: Who pays for experiments? How much is experiment, and how much is demonstration? How much should you rely on surveys and expert opinion?

In the next section of the paper the author discusses values wanted from the cable system. He mentions equity, privacy, efficiency and diversity, and then proceeds to discuss the last two.

The author seems to feel that the level of efficiency is a function of the level of regulation and the locale. Therefore the level of regulation must vary from city to city. (One has visions of a growing bureaucracy to regulate regulatory commissions).

The author then discusses the word "diversity" and its interpretation. The U.S. FCC has interpreted it as implying separate ownership. Most studies interpret it as program classification, sports, news, drama etc. He then effectively proposes a study into the nature of diversity and policy options that may be used to achieve it.

The last portion of the paper then goes on to discuss the possibility of using fiber optics. He predicts the possibility of a fiber optic network replacing both the current cable and telephone nets. Finally he states that much study is needed to determine the feasibility, practicality, need and implications of such a system.

2. Comments

This summary was done at the explicit request of the scientific officer for this contract.

Many of the potential services discussed seem to have a major drawback in the system security area. They involve placing sophisticated terminal equipment in a vulnerable, hard to service location (the subscriber's home). There is a high potential for a malfunctioning piece of equipment to impact a sizable portion of the system.

NETWORK MODELLING AND SIMULATION