CANADIAN THESES ON MICROFICHE

THÈSES CANADIENNES SUR MICROFICHE

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

NL 339 (r. [illegible]/09)

Canada

MODELLING AND DYNAMIC ROUTING FOR COMPUTER QUEUEING NETWORK

by

Yong Wha Lee

A thesis
presented to the University of Ottawa
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
The Department of Electrical Engineering
Faculty of Science and Engineering

OTTAWA, Ontario, 1985

The University of Ottawa requires the signatures of all persons using or photocopying this thesis. Please sigh below, and give address and date.

In memory of my father,

Doo Ryun Lee

ABSTRACT

In this thesis, a simple stochastic dynamic model for computer queueing network with finite buffer, time-varying stochastic input and channel errors has been presented. Also it is proposed a dynamic routing policy that minimizes the total size of queue in the network for each short interval of time based on the immediate past state information and the expected number of external arrivals during the interval. The procedure is repeated for each interval of time and hence it provides an adaptive strategy. Compare to static assignments, the strategy proposed in this study has the advantages in being able to deal with transients and congestion and to adapt to the changing environment. Numerical algorithm is quite simple and is applicable in practical systems. Simulation has been carried out with a simple 3 node network. The results show that the policy can provide good performance of the network.

ACKNOWLEDGEMENTS

The author wishes to express his deepest gratitude to his thesis supervisor Prof. N.U. Ahmed for his generous encouragement; continued assistance and invaluable guidance throughout this work, without which this thesis would not have been possible.

The author also wish to express his sincere thanks to the member of the system and control group, especially to Mr. S. Biswas, Mr. T. Dabbous, and Mr. S.S. Lim for their stimulating discussion, very helpful suggestions, and moral support.

Special thanks to all graduate students and staff members of Department of Electrical Engineering, University of Ottawa.

The financial assistance of the Natural Science and Engineering Research Council and the University of Ottawa during the period of this research is gratefully acknowledged.

Finally, the author wishes to highlight the very especial support he received from his beloved wife Yun Sook Lee.

CONTENTS

Chapter I

INTRODUCTION

1.1 THE NECESSITY FOR DYNAMIC ROUTING

We frequently encounter situations involving traffic problem in our daily living. The examples are automobile traffic in down town area, the passage of customers through the checkout counter of a supermarket, the transmission of messages, and so on.

These have a fundamental property in common; they represent processes involving the flow of some commodity through a channel or a network of channels which has a limit to the traffic it can carry.

One of the most extensively studied problems in transportation theory is the optimization of static network flows. Most of the studies and achievements have been concentrated on the analysis of steady-state assignment under non-congested conditions [8] [17] [19] [22] [33]. Mathematical techniques, such as graph theory, linear programming and integer programming [23], have been applied successfully to these problems, which in turn have inspired further development of these techniques.

However, many network problems are naturally characterized by time-varying and stochastic behavior. Delay and congestion have become common phenomena. One of the examples is the management of automobile traffic during rush hour for city downtown areas. The capacity of the streets may be momentarily unable to accommodate the peak demand, and as a result waiting queues of cars would form at street intersections. Another example is the data-routing problem in computer communication networks. The packet arrival rate and the requested amount of service also vary with the time. Sudden surges in demand by certain users may also cause difficulties in the queueing buffers. In all such cases, dynamic strategies are needed to direct traffic flows and to minimize the congestion. With a proper feedback mechanism, these strategies should be adaptive to the uncertain and changing environment. Clearly, classical optimization techniques assuming steady-state or deterministic models are not adequate for such problems.

1.2 REVIEW OF MODELS FOR DYNAMIC ROUTING

Recently, Gazis [14] suggested that a congested transportation system can be viewed as a store and forward network. From this point of view all delays due to congestion are handled by assuming that queue can build up inside the network when demand exceeds capacity. Thus, traffic is assumed to move between nodes of the network at constant velocity and is stored in queues at the nodes until it can proceed onward toward its destination. In this way the time it takes to pass from node to node is spilt into a constant delay portion and a portion associated with delay of queue. Taking this into consideration, D'Ans and Gazis [15] formulated a dynamic optimization problem for minimizing the aggregate delay in a network with no-turn intersections. They showed that an open loop solution can be obtained when the problem is discretized in time. In [11], Chu and Gazis considered the problem of optimum dynamic route allocation for a congested network consisting of parallel channels and obtained a feedback allocation strategy. Also Chu [10] has developed a decentralized control algorithm for large networks. Sarachick [29] [30] [31] found a local dynamic strategy for a congested node with two alternative path to multidestinations of a traffic.

In 1976, an analytical model for problems of dynamic routing in data-communication has been introduced by Segall [37]. He considered a queueing model in which, in contrast to previous models, the 'queues' were being associated with the nodes rather than with the links. For a network with n nodes, each link diversing from a node i is shared by some or all of the up to (n-1) types of traffic stored at time t at the node i (the term 'type' is used to indicate destination). He stated the problem as follows.

At every time t, given the knowledge of the network congestion $\{Q_i^j(t),\ i,j\in N,\ j\neq i\}$, dynamically decide what portion of each link should be used for each type of traffic, so that the total delay will be minimized. The equations describing the problem, as given by Segall, are as follows.

Dynamics

The rate of change of the queues was given by

$$\dot{Q}_i^j(t) = r_i^j(t) - \sum_{k\in E(i)} u_{ik}^j(t) + \sum_{\substack{\ell\in I(i)\\ \ell\neq i}} u_{\ell i}^j(t) \qquad i,j\in N \quad j\neq i \tag{1}$$

where $u_{ik}^j(t)$ is the portion of the capacity of the link (i,k) used at time t for traffic of type j. $r_i^j(t)$ is the instantaneous rate of type j traffic arriving at node i. In other words, the first term describes external arrival rate, and the second and the third term describe the internal flow rate of outgoing traffic and incoming traffic respectively. (for detail description of the notations, see page 16)

Constraints

Positive constraints are

$$Q_i^j(t) \geq 0$$

$$u_{ik}^j(t) \geq 0 \tag{2}$$

and capacity constraints

$$\sum_{K\in E(i)} u_{ik}^j(t) \leq C_{ik}, \qquad (i,k)\in L \quad , \quad j\in N \tag{3}$$

Cost functional

The cost function to be minimized is given by

$$J = \int_{t_0}^{t_f} \sum_i \sum_j (\alpha_i^j Q_i^j(t)) dt \tag{4}$$

where t_f is some time at which all queues are empty and α_i^j is the appropriated weights to the Q's.

Segall and Moss [38] have also applied optimal control theory to find a centralized routing policy. However the computation of the optimal control requires the knowledge of $r_i^j(t)$ over the time interval $[t_0, t_f]$. This was the central dilemma in the application of necessary conditions in the determination of a feedback solution. In order to overcome this difficulty, they considered only the situation in which all of the inputs $(r_i^j(t) \quad i,j \; N, j \neq i)$ are constant functions of time. Furthermore, they assumed that the network has enough capacity to accommodate both the inputs and the flow due to the initial congestion. In this way, they had been able to prove that for a special class of problems (single destination, no inputs, unity weights in cost function), an implementable algorithm is possible.

However, in none of the studies mentioned above, the problem of dynamic routing for the traffic, having time varying stochastic input with finite buffer, have been investigated. Since any realistic transportation or communication network must handle stochastic inputs with buffer limitation, the problem of finding an optimal dynamic feedback routing algorithm for stochastic multi-destination traffic with finite buffer is essential and has been studied in this thesis.

1.3 THE PROPOSED MODEL

The idea of modelling of queueing networks considered in this thesis is basically in line with the model introduced by Segall [37]. However, we have introduced i) buffer limitations of each node, ii) time-varying stochastic inputs and iii) channel errors. The model developed on the basis of these facts appear to be more realistic.

To compare Segall's model and the model proposed here, we illustrate some special features of the models in Table - 1

In the model proposed, buffers are provided at each node for all different types of packet as Segall considered in his model [37]. However, in contrast to Segall's model, we consider the buffer limitation and time varying stochastic inputs. Since the external arrival rates at each buffer change independently, the packets in some buffers could suffer high congestion and the input packets at the congested buffers might be rejected. On the other hand, some buffers are empty and corresponding channels may be idle. This kind of unbalanced situation can occur frequently in practical systems and this may degrade the performance of the network by increased delay and reduced throughput.

The problem considered in this thesis is to find the best routing policy so that it minimizes delay and maximizes network utilization. An exact solution of the problem is extremely difficult because of the complexity of the network and randomness of the multi-variate environment. In fact, for an exact solution, one must use Stochastic (optimal) control theory [1] [13], which makes the problem numerically intractable. However, for an approximate solution, one can use one's intuition to develop an implementable routing scheme based on immediate past state and expected external input. Then one can improve the scheme by finite dimensional optimization and verify the scheme through simulation or analysis.

The method developed is based on the following argument. Suppose we clear all the queues in the network as quickly as possible. Then, obviously, the overall waiting time at buffers will be reduced; besides, it will provide the available buffer space for input packets.

Therefore we propose a scheme for the present problem in the following way. The total size of queue in the network is minimized for each interval of time based on the immediate past state information and the expected number of input packets from external sources during the same interval.

Due to finite buffer size, transmitted packets may be rejected and will require retransmission. Therefore a constraint of the buffer limitation is incorporated in the minimization procedure to prevent the unnecessary retransmissions arising from the buffer limitation.

Also, we use weighting factors which are automatically decided according to the information such as immediate past state of queue and expected number of external input packets during the interval. In other words, we give more weight to the congested place so that we can provide better throughput of the network.

Since the procedure will be repeated for each interval of time sequentially, the solution of the minimization problem will provide a real-time adaptive control strategy. This is the idea behind the dynamic routing mechanism studied in this thesis.

1.4 ORGANIZATION OF THE THESIS

In chapter II a mathematical queueing model for a dynamic control system is presented. First we discuss external packet arrival process which was assumed to be stochastic poisson process. Also we discuss internal flow and errors. Then we provide state equations which describe the rate of variation of the queue length in each buffer. An objective function is then introduced with suitable weighting factors. Finally, constraints are described and the optimization problem is formulated at the end of the chapter.

In chapter III, we describe the numerical simulation which have been developed in this study. For the purpose of comparison, we observe three cases of routing strategies. The first is a fixed routing based on minimum hop [35], the second is the dynamic routing proposed in this thesis using proper weighting factors, and the third strategy is the same as the second with the weighting factors equal to unity. As an illustrative example, a simple network with 3 node and 6 unidirectional channel is used for the simulation.

In chapter IV, we investigate the system behaviour under three different kinds of situation such as (a) well balanced input for all the destination nodes, (b) extremely unbalanced inputs, and (c) arbitrary inputs which are neither well balanced nor extremely unbalanced.

Finally, in chapter V, the conclusions and suggestions for further studies are presented.

	Proposed Model	Segall's model
1	Finite buffer size is assumed.	Buffer limitation is not considered.
2	No transit delay and no processing delay are assumed.	No transit delay and no processing delay are assumed.
3	Knowledge of immediate past state is required.	Knowledge of input history over the time interval [t,T].
4	External arrivals are stochastic.	External arrivals are deterministic.
5	Channel errors are considered.	No channel errors are considered.
6	Numerical computation is simple.	Numerical computation is too complicated.

Table - 1 Comparison of two models.

Chapter II

A QUEUING MODEL FOR A DYNAMIC CONTROL SYSTEM AND OPTIMIZATION

2.1 NETWORK DESCRIPTION

We consider a packet-switching communication network consisting of n nodes as shown in Fig - 1.

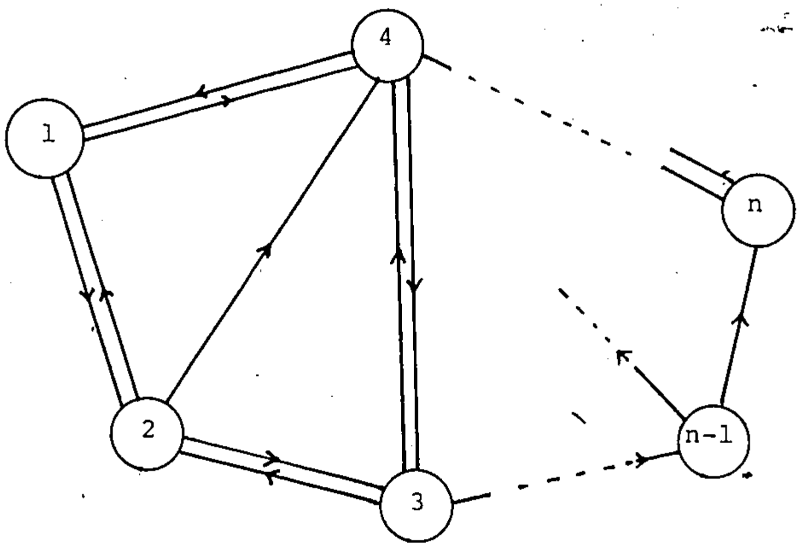


Fig-1 A packet-switching communication network.

Let 'N' denote the collection of all nodes in the network such as

$$N \equiv \{ 1, 2, \bullet \bullet \bullet , n \} . \tag{5}$$

A channel direct connecting the node i to the node k will be denoted by (i,k). The channel (i,k) has a direction from the node i to the node k.

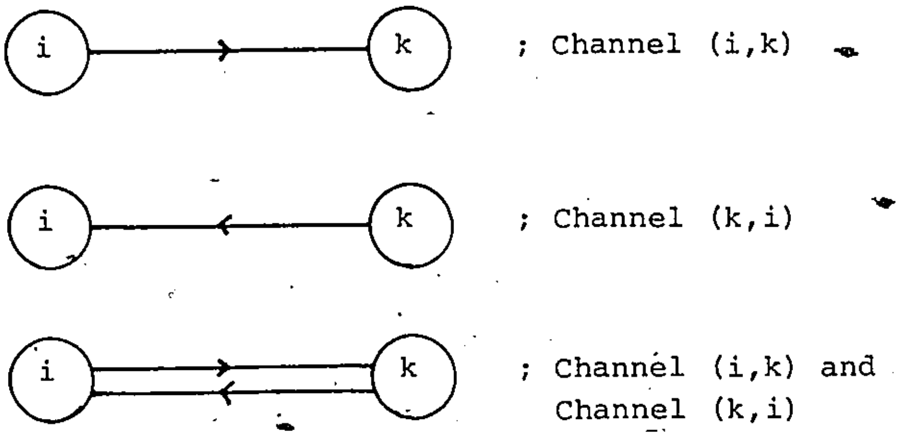


Fig - 2 — Channels

Let L denotes the collection of all channels in the network such as

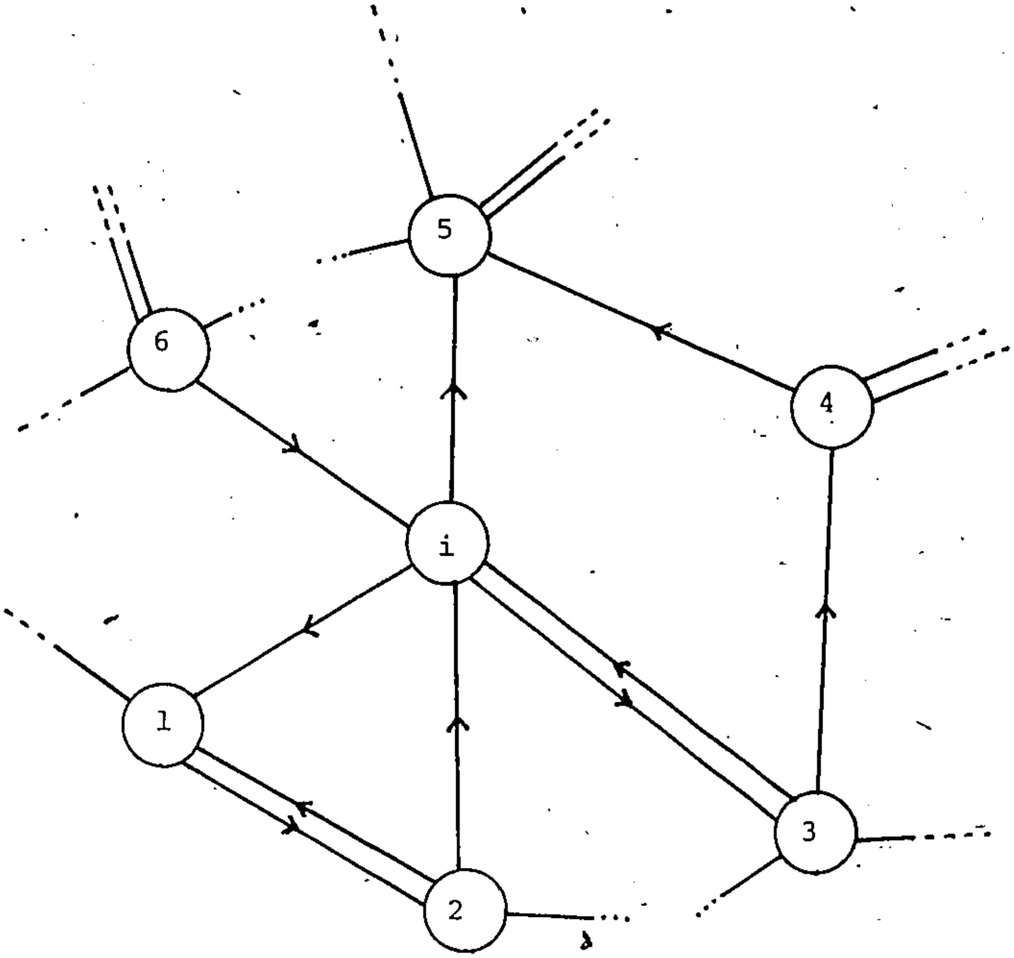
$$L \equiv \{ (i,k), \text{ where } i,k \in N \} \tag{6}$$

For every $i \in N$, E(i) denotes the set of all receiving nodes from the node i such as

$$E(i) \equiv \{ k ; k \in N, (i,k) \in L \} . \tag{7}$$

Similarly, I(i) is the set of all sending nodes to the node i such as

$$I(i) \equiv \{ \ell ; \ell \in N, (\ell,i) \in L \} . \tag{8}$$



a) Receiving nodes (radiating nodes);

$E(i) \equiv \{1,3,5\}$

b) Sending nodes (incident nodes)

$I(i) \equiv \{2,3,6\}$

Fig-3 Illustration of receiving nodes and sending nodes.

In this model, we shall assume that every packet has the same size which is small enough compared to the transmission capability of the channel. Therefore, the capacity of the channel (i,k) is denoted C_{ik} (in the unit of packets) per unit time, where $(i,k) \in L$.

Each node has (n-1) buffer which are provided for every type of packet (throughout this thesis, we use the term 'type' to indicate a packet's destination). We consider that each buffer has a finite and fixed size (storeable capacity). We use B_i^j to denote the storeable capacity of the buffer for the type j traffic at the node i. In each buffer, at each node i, we store the packets whose destination is $\{1, 2, \bullet\bullet\bullet, (i-1), (i+1), \bullet\bullet\bullet, n\}$ respectively.

< Summary of notation >

n ; number of nodes in the network

N ; set of all nodes $\{1,2,\cdots,n\}$ in the network

(i,k) ; channel direct connecting node i to node k

L ; set of all channels (i,k), such that $i,k \in N$ and there is a direct connection i to k

$E(i)$; set of all receiving nodes from the node i

$I(i)$; set of all sending nodes to the node i

C_{ik} ; capacity of channel (i,k), where $(i,k) \in L$ in units of packets/unit-time

B_i^j ; buffer size at node i for packets of type j

$a_i^j(t)$; arrival process at node i from external sources

$A_i^j(\Gamma)$; number of arrival packets of type j at node i during the interval Γ

$Q_i^j(t)$; number of waiting packets of type j at node i

$U_{ik}^j(\Gamma)$; number of packets of type j to be sent through using the channel (i,k) during the interval Γ

$\xi_{ik}^j(\Gamma)$; number of losses of packet of type j using channel (i,k) during interval Γ

$\tilde{\rho}_{ik}$; error rate of the channel (i,k)

2.2 QUEUES

Let the number of packets of type j which are waiting at node i at time t be denoted by $Q_i^j(t)$. We should note that in contrast to previous static models, the queues are being associated with the destination nodes rather than with the channels, as considered in [37]. (Fig-4)

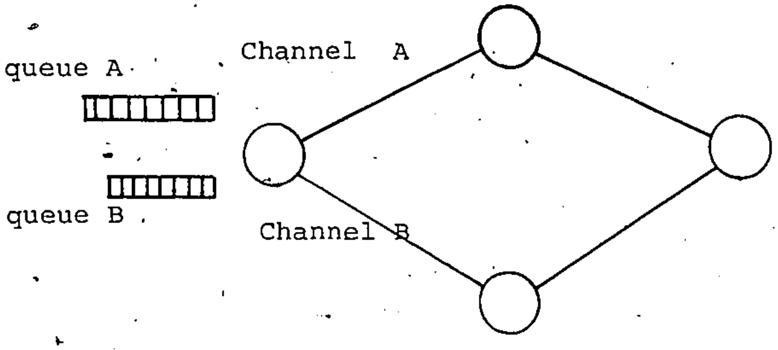


Fig-4 (a) Queues associated with channels.

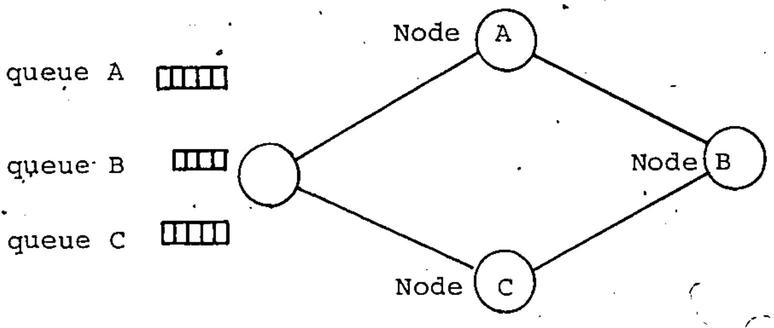


Fig-4 (b) Queues associated with destination nodes.

2.3 ARRIVAL PROCESS

For each node i, the arrival of packets of type j from the outside of the network are independent random processes. We may assume that the external arrivals are stochastic poisson processes, and it has a non decreasing path with mutually independent positive jumps. This process is denoted by $a_i^j(t)$ and described as

$$a_i^j(t) = a_i^j(0) + \int_0^t da_i^j(\theta) \tag{9}$$

Moreover $A_i^j(\Gamma)$ denotes the number of times the packets of type j arrive at node i during the time interval Γ. This random number can be described by

$$A_i^j(\Gamma) = \int_\Gamma da_i^j(t) \tag{10}$$

and the probability that $A_i^j(\Gamma)$ is equal to m is given by

$$P\{A_i^j(\Gamma)=m\} = \exp\{-\int_\Gamma \lambda_i^j(t)dt\} \ \{\int_\Gamma \lambda_i^j(t)dt\}^m/m! \tag{11}$$

where $\lambda_i^j(t)$ denotes the time dependent arrival intensity.

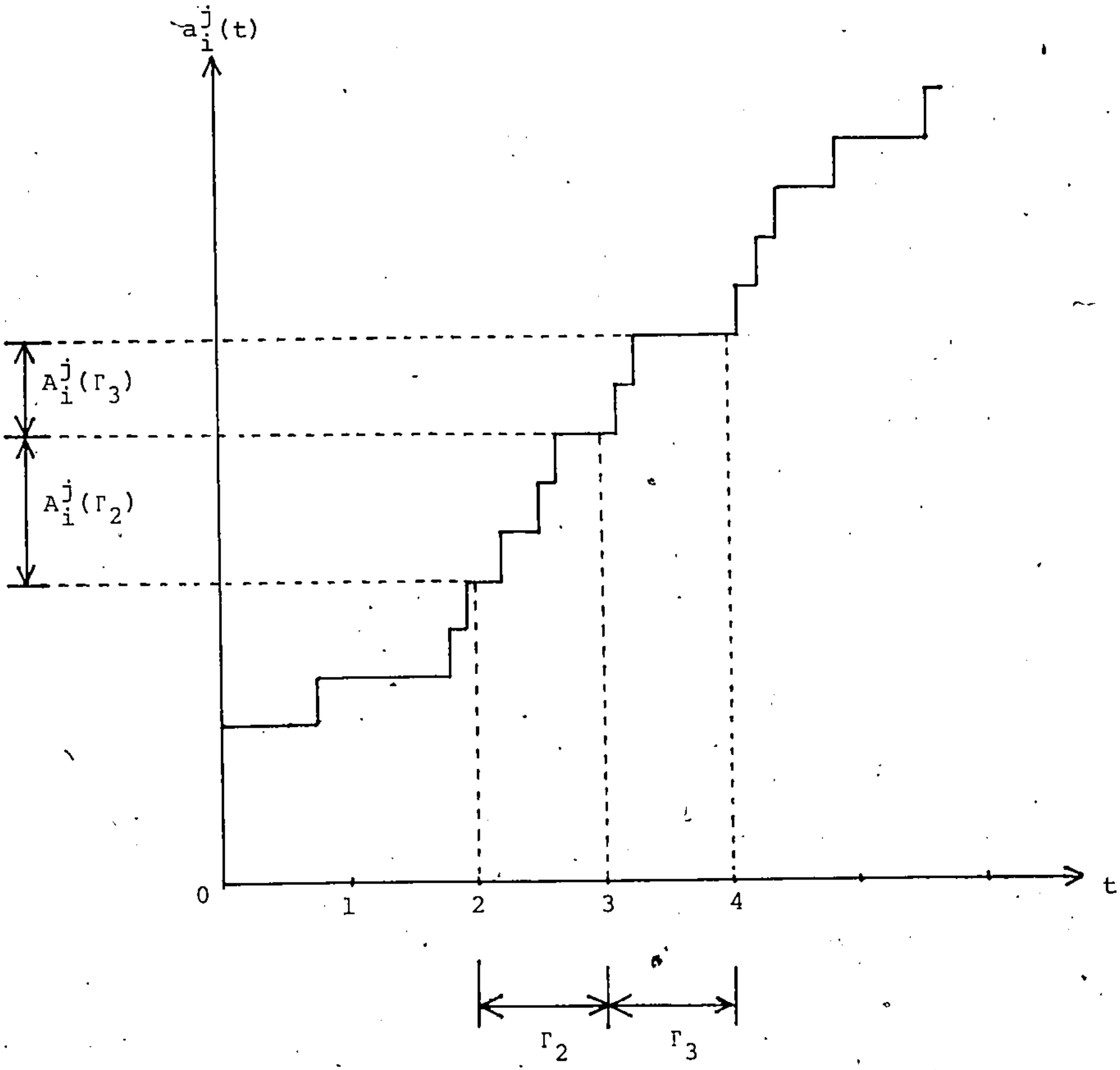


Fig-5 An external arrival process.

2.4 INTERNAL FLOW

Each channel diverging from the node i is shared by some or all of (n-1) types of the packets stored at node i. The number of packets of type j to be sent through the channel (i,k) during the time interval Γ is denoted by $U^j_{ik}(\Gamma)$, where $(i,k) \in L$. And this, $U^j_{ik}(\Gamma)$, is considered to be the control variable over the time interval Γ.

Some of the transmitted packets, $U^j_{ik}(\Gamma)$, may get lost (discarded) due to channel errors. The number of packets lost among the number of type j packets transmitted, using the channel (i,k), during the time interval, Γ, is denoted by $\xi^j_{ik}(\Gamma)$. Clearly the number of packets lost increases with the increase of the number of packets transmitted. Normally these lost packets have to be retransmitted from the node i. We may consider that the random variable $\xi^j_{ik}(\Gamma)$ arises from a doubly stochastic poisson process satisfying

$$p\{\xi^j_{ik}(\Gamma) = m \mid U^j_{ik}(\Gamma)\} = \exp(-\rho^j_{ik}(\Gamma))\,(\rho^j_{ik}(\Gamma))^m/m \qquad (12)$$

where $\rho^j_{ik}(\Gamma)$ is the expected number of packets lost given $U^j_{ik}(\Gamma)$ and the channel error rate. Therefore, we have

$$\rho^j_{ik}(\Gamma) = E\{\xi^j_{ik}(\Gamma) \mid U^j_{ik}(\Gamma), \tilde{\rho}_{ik}\} \qquad (13)$$

where $\tilde{\rho}_{ik}$ is the error rate of the channel (i,k) which is considered to be constant.

We may assume that the processing time at switching nodes, and propagation delays are negligible. Also we assume that the packets, once they reach their destination, are immediately removed from the network. Hence, the delay experienced by a packet in the network is approximated by the queueing time.

The queue length is increased by incoming flow which consists of both external and internal input. As defined earlier, the internal input for node i is coming from the neighbouring node l which is in the set I(i). Note that the internal input packets can be corrupted by an error due to channel noise. These defective packets are removed from the internal input. Therefore, the rate of incoming flow of type j packet at node i can be described by

$$R_i^j(dt) = A_i^j(dt) + \sum_{\substack{\ell \in I(i) \\ \ell \neq j}} \{ U_{\ell i}^j(dt) - \xi_{\ell i}^j(dt) \} \qquad (14)$$

where $i, j \in N$ and $i \neq j$.

Practically, available buffer space is finite so that we cannot always store up all incoming packets in a buffer. If amount of incoming packets is greater than the available buffer space, the excess will be discarded. In other words, the number of incoming packets accepted is limited by the maximum available buffer space. Therefore actual incoming flow is given by

$$R_i^j(dt) = [\, A_i^j(dt) + \sum_{\substack{\ell \in I(i) \\ \ell \neq j}} \{U_{\ell i}^j(dt) - \xi_{\ell i}^j(dt)\}] \wedge [B - Q\,(t^-)] \quad (15)$$

where $(a) \wedge (b) = \text{minimum}\{\, a\, ,\, b \,\}$

and $B_i^j - Q_i^j(t^-)$ denotes available buffer space at time t.

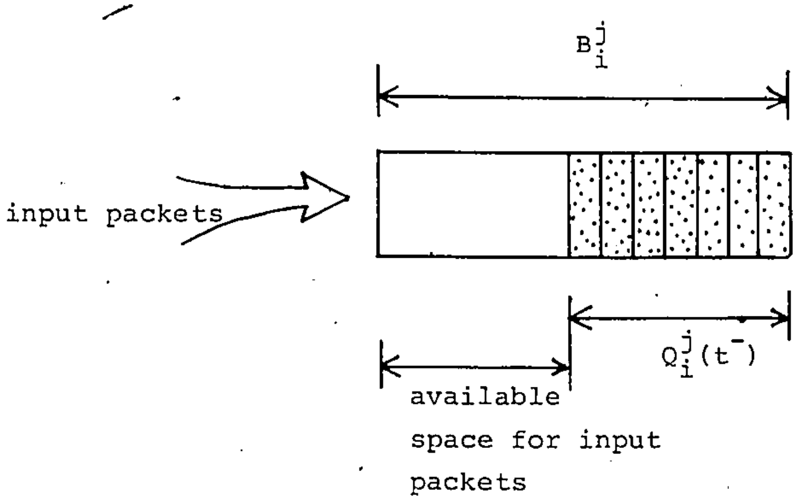


Fig-6 Available buffer space at time t

The outgoing flow is the summation of all packets transmitted from node i to node k which is in the set E(i). Here, also, the number of defective packets are removed from the outgoing flow for the same reason as before. Therefore the outgoing flow is given by

$$O_i^j(dt) = \sum_{k \in E(i)} \{U_{ik}^j(dt) - \xi_{ik}^j(dt)\} \quad . \qquad (16)$$

2.5 STATE DYNAMICS

Our state equation describes the time evolution of queue length in each buffer. This is nothing but the difference between incoming flow rate and outgoing flow rate.

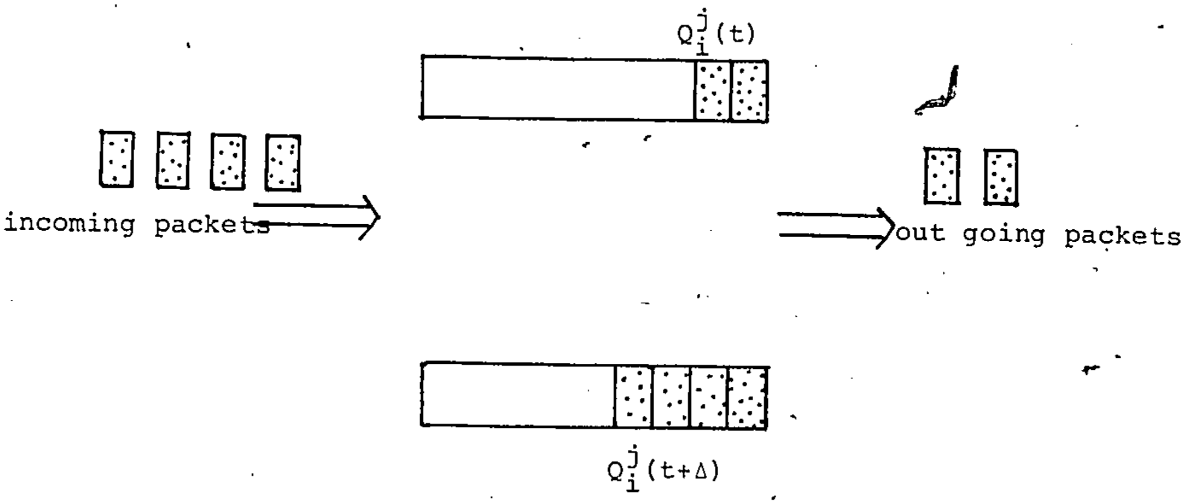


Fig-7 Formulation of a queue

Therefore the history of queues in the network is governed by the following system of (stochastic) dynamic equations.

$$\left\{\begin{aligned} dQ_i^j(t) &= R_i^j(dt) - O_i^j(dt) \\ &= \Big[A_i^j(dt) + \sum_{\substack{\ell\in I(i)\\ \ell\neq j}} \{U_{\ell i}^j(dt) - \xi_{\ell i}^j(dt)\}\Big] \wedge \big[B_i^j - Q_i^j(t^-)\big] \\ &\qquad - \sum_{k\in E(i)} \{U_{ik}^j(dt) - \xi_{ik}^j(dt)\} \\ &i,j = 1,2,- - -,n \;, \qquad i\neq j \end{aligned}\right. \tag{17}$$

A more intuitively appealing representation for the equation (17) is obtained by rewriting it as a difference equation. For a sufficiently small $\Delta>0$ we have

$$\begin{aligned} Q_i^j(t+\Delta) - Q_i^j(t) = &\Big[A_i^j(\Gamma_t) + \sum_{\substack{\ell\in I(i)\\ \ell\neq j}} \{U_{\ell i}^j(\Gamma_t) - \xi_{\ell i}^j(\Gamma_t)\}\Big] \wedge \big[B_i^j - Q_i^j(t)\big] \\ &- \sum_{k\in E(i)} \{U_{ik}^j(\Gamma_t) - \xi_{ik}^j(\Gamma_t)\} \end{aligned} \tag{18}$$

or

$$\begin{aligned} Q_i^j(t+\Delta) = Q_i^j(t) - &\Big[A_i^j(\Gamma_t) + \sum_{\substack{\ell\in I(i)\\ \ell\neq j}} \{U_{\ell i}^j(\Gamma_t) - \xi_{\ell i}^j(\Gamma_t)\}\Big] \wedge \big[B_i^j - Q_i^j(t)\big] \\ &- \sum_{k\in E(i)} \{U_{ik}^j(\Gamma_t) - \xi_{ik}^j(\Gamma_t)\} \end{aligned} \tag{19}$$

where $\Gamma_t = (t,t+\Delta]$.

The first term on the right hand side of the equation (19) represents the number of packets remaining in the network at time t. In the second term, the quantity to the left side of $\wedge$ gives the number of incoming packets during the interval $(t,t+\Delta]$, and that to the right side gives the available buffer space or the maximum number of incoming packets acceptable.

2.6 OBJECTIVE FUNCTION

There could be many ways to choose a reasonable performance function depending on one's objective. The objective of our study is to find an optimal policy which, in the long run, minimizes the average delay and maximizes the utilization of the system.

In deterministic case, an objective function of the following form has been considered in [37],

$$J = \int_0^T \sum_{i,j} \alpha_i^j Q_i^j(t)\, dt, \qquad (20)$$

where are suitable weighting factors. A solution to this optimization problem was presented by Segall and Moss in [38]. They applied the maximum principle to find the optimal control assuming continuous traffic with constant rate of input and that the network has enough capacity to accommodate both the input and the flow due to the initial congestion. Despite the simplification provided by the above assumptions, there were considerable difficulties to obtain an optimal solution because of computational complexities.

For stochastic system, an objective function of the following form has been proposed [37].

$$J = E\{\int_0^T \sum_{i,j} \alpha_i^j Q_i^j(t) dt + \phi(Q(T))\} , \qquad (21)$$

where α_i^j are weighting factors and $\phi(Q(T))$ is the final cost. In other words, equation (21) includes the expected total delay during the period of operation $(0,T)$ plus the extra delay one has to incur to get rid of the traffic left in the system at time T. This becomes a more complicated problem and the exact solution to the problem is very difficult to obtain, since a solution by numerical integration is much too complex. Further, to compute the optimal solution, one has to have all the history of external inputs for the entire interval from 0 to final time T. This is almost impossible because the system is stochastic and hence input history is not available. Since we are looking for a suboptimal solution to the problem, a more suitable and perhaps more useful objective function for this model is given by

$$\tilde{J}(t) = \sum_{i,j} \alpha_i^j \, E\{Q_i^j(t+\Delta) \mid Q_i^j(t)\} \tag{22}$$

where α_i^j are weighting factors. We consider $U_{ik}^j(\Gamma_t)$ to be dependent only on immediate past state $Q_i^j(t)$ and $E\{A_i^j(\Gamma_t)\}$. Therefore under this assumption we have

$$\begin{aligned} E\{Q_i^j(t+\Delta) \mid Q_i^j(t)\} &= Q_i^j(t) \\ &+ [E\{A_i^j(\Gamma_t)\} + \sum_{\substack{\ell \in I(i) \\ \ell \neq j}} \{U_{\ell i}^j(\Gamma_t) - E(\xi_{\ell i}^j(\Gamma_t) \mid U_{\ell i}^j(\Gamma_t))\}] \wedge [B_i^j - Q_i^j(t)] \\ &- \sum_{k \in E(i)} \{U_{ik}^i(\Gamma_t) - E(\xi_{ik}^j(\Gamma_t) \mid U_{ik}^j(\Gamma_t)) \end{aligned} \tag{23}$$

where $\Gamma_t = (t, t+\Delta]$

Using (23) into (22) we look for a solution minimizing the amount of queues in the network for every short interval of time, sequentially, given the information containing the initial state of queues and the estimated number of packets arriving from external sources.

As we described before, the number of packet lost due to channel error increases with the increase of the number of packets transmitted. If the channel has a constant error rate, we can assume that the number of packet lost due to the channel error linearly increases with the increase of the number of transmitted packets. Then we expect the number of lost packets as

$$E\{\xi_{ik}^{j}(\Gamma_t) \mid U_{ik}^{j}(\Gamma_t)\} = \tilde{\rho}_{ik} U_{ik}^{j}(\Gamma_t) , \tag{24}$$

where $0 < \tilde{\rho}_{ik} < 1$ and

$\tilde{\rho}_{ik}$ is the error rate of the channel (i,k).

Then the equation (23) can be rewritten as

$$\begin{aligned} E\{Q_i^j(t+\Delta) \mid Q_i^j(t)\} &= Q_i^j(t) \\ &+ \Big[E\{ A_i^j(\Gamma_t)\} + \sum_{\substack{\ell \in I(i) \\ \ell \neq j}} \{(1-\tilde{\rho}_{\ell i}) U_{\ell i}^j(\Gamma_t)\}\Big] \wedge \big[B_i^j - Q_i^j(t)\big] \\ &- \sum_{k \in E(i)} \{(1-\tilde{\rho}_{ik}) U_{ik}^j(\Gamma_t)\}. \end{aligned} \tag{25}$$

In fact, equation (25) can be easily linealized by using a simple constraint for the maximum available buffer space (which is the right side of $\wedge$ in the second term). Later, we will discuss the constraint for this case.

Weighting Factor

A function of the weighting factor, α_i^j, is to assign high priority to the particular queue which is more urgent than the others. The queues with high priorities are assigned larger weights α_i^j. This will clear the high priority queues faster.

The weighting factor α_i^j can be defined in several ways depending on one's purpose. For example, if we want to reduce the number of lost packets due to the shortage of available buffer space, then we may use the weighting factor α_i^j as the function of the available buffer space and the expected number of external arrivals such as

$$\alpha_i^j(\Gamma_t) = \begin{cases} 1 + Z_i^j(\Gamma_t) & \text{if } Z_i^j(\Gamma_t) > 0 \\ 1 & \text{otherwise} \end{cases} \qquad (26)$$

where

$$Z_i^j(\Gamma_t) = \frac{Q_i^j(t) + E\{A_i^j(\Gamma_t)\} - B_i^j}{B_i^j} .$$

The expression (26) means that we give more weight (or priority) to the place where the available buffer space for new incoming packets is less than at other places. In this way, we may expect to reduce the number of lost packets. In other words, this will increase the acceptance of the network there by improving throughput.

If we are interested only in minimizing delay in the network, then we could use the unit constant weight factors such as

$$\tilde{\alpha}_i^j(\Gamma_t) = 1 \qquad \text{for all} \quad i,j \in N .$$

We will see later how these weighting factors affect the system performances.

There could be many other different possibilities in the choice of the weighting factors. Therefore, it would be of interest to investigate other possibilities.

2.7 CONSTRAINTS

First of all, queue size cannot be a negative quantity so that we clearly have the following non-negative constraint;

$$Q_i^j(t) \ge 0 \quad \text{or} \quad Q_i^j(t+\Delta) \ge 0 \quad . \tag{27}$$

By definition, the traffic flow for any interval of time is also a non-negative quantity such as

$$U^j_{ik}(\Gamma_t) \geq 0 \quad . \tag{28}$$

Furthermore, total amount of traffic flow cannot exceed capacity of the physical channel. Therefore we have channel capacity constraint which can be expressed by

$$\sum_j U^j_{ik}(\Gamma_t) \leq C_{ik} \tag{29}$$

where $(i,k) \in L$ and $i \neq j$.

In addition to the above constraints, a constraint for the buffer limitation should be considered as given in the dynamic model (23). Before considering the buffer constraint, we may discuss some facts of available buffer space for input.

There could be several mechanisms to allow the input to be stored at the available buffer space. For instance, let's assume that there is available space for 10 packets at a buffer and at the same time 8 external input packets and 6 internal packets are to be stored at the buffer. Case I; if the external input packets have priority for the buffer occupation over the internal input packets, then all 8 external input packets and only 2 internal packets will be stored. Therefore, the other 4 of the internal inputs will be discarded. Case II; if both internal and external inputs have the same priority for the buffer occupation, the lost packets can include both internal packets and external packets. Finally case III; if internal input packets have the priority over external inputs, then all the lost packets will be the external ones. If some internal packets are lost, then they have to be retransmitted and, consequently, some part of the channel which could be useful to other traffic will be just wasted by the retransmission procedure.

Since the internal packet flow is controlled in this problem , it could be manipulated in the following way ; to reduce the number of the retransmission packets, for every interval of time, we give the priority of buffer occupation to the internal input packets rather than the external input packets. However, to increase the acceptance for the external input packets, we may use an additional constraint in the optimization procedure such that the amount of internal input accepted does not exceed the difference between the available space, $B_i^j - Q_i^j(t)$, and the estimated number of external input packets, $E\{A_i^j(\Gamma_t)\}$. This constraint is given by

$$\begin{cases} \sum\limits_{\substack{\ell\epsilon I(i)\\ \ell\neq j}} U_{\ell i}^j(\Gamma_t) < B_i^j - Q_i^j(t) - E\{A_i^j(\Gamma_t)\} & , \quad \text{if } B_i^j - Q_i^j(t) > E\{A_i^j(\Gamma_t)\} \\ \sum\limits_{\substack{\ell\epsilon I(i)\\ \ell\neq j}} U_{\ell i}^j(\Gamma_t) = 0 & \text{otherwise} \end{cases} \qquad (30)$$

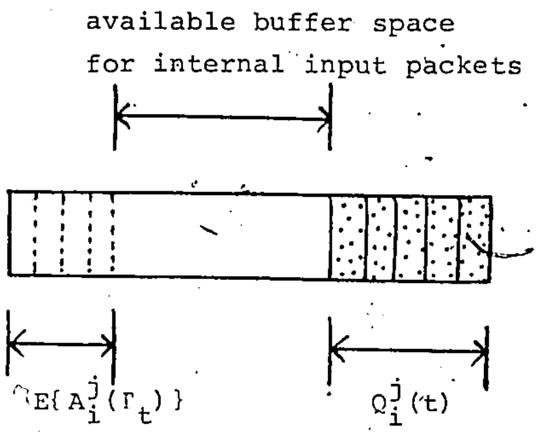


Fig-8 Available buffer space constraint.

2.8 OPTIMIZATION

Now, we can realize that the objective function (22) becomes linear function by using the constraint (30) because the control parameter $U_{ik}^j(\Gamma_t)$ will be determined in the such way that always the left side of $\wedge$ in the objective function (23) is valid. Therefore the objective function (22) can be easily implemented by a computer using a linear optimization program.

Hence the problem becomes - that every time a routing policy is to be obtained for short interval of time based on the given immediate past state information of queues at every node, the estimated number of incoming packets from external sources and the linealized objective function of (22) with the constraints (27), (28),(29), and (30).

Chapter III

NUMERICAL SIMULATION

3.1 INTRODUCTION

As an illustrative example, the proposed scheme has been tested using the simple network plotted in Fig-9.

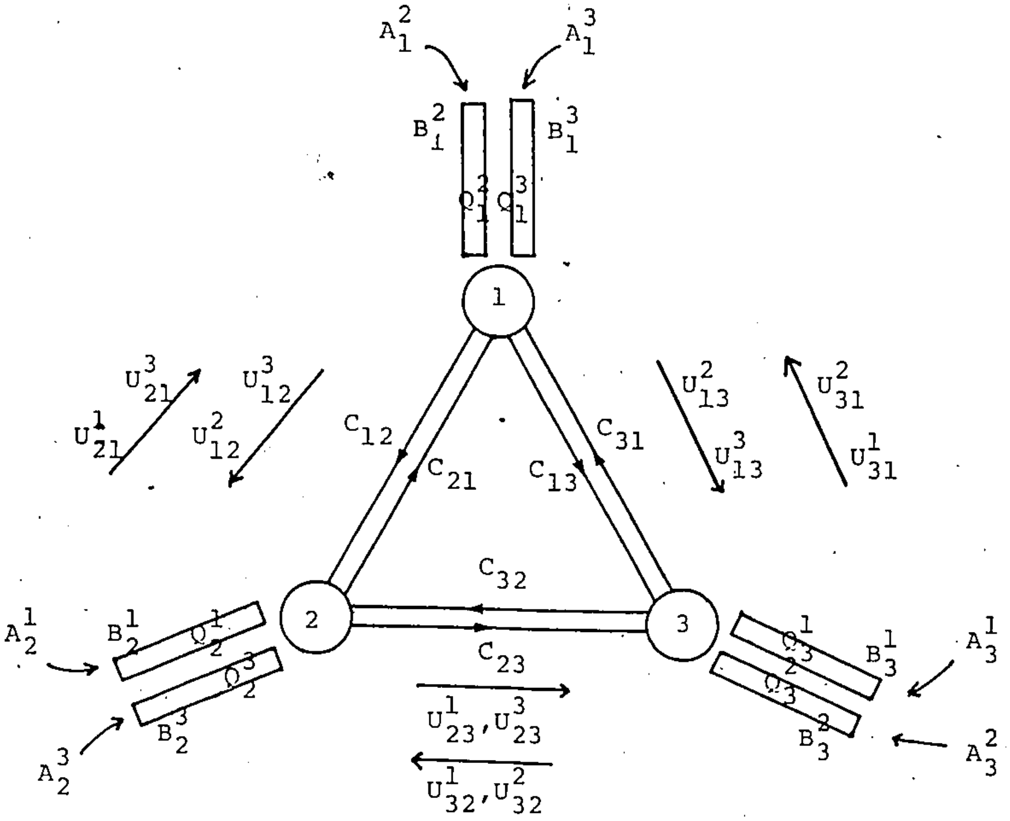


Fig-9 The example network and system parameters.

In order to study the effect of the proposed method, the scheme was compared with a method called fixed routing . In the fixed routing scheme, for each packet a route is previously chosen based on minimum hop [35] and this is fixed unless there is failure of channel or node processor. Note that in the proposed method a route for each packet is determined by solving the optimization problem as discussed in Chapter II. Also, to observe the effects of two different kinds of weighting factors were used in the objective function (22).

For the purpose of comparison, these methods are designated as follows;

Case-1 : The fixed routing based on minimum hop [35].

Case-2 : The proposed routing scheme obtained by minimizing the objective function (22) with the weighting factor defined by (26).

Case-3 : The proposed routing scheme by minimizing the objective function (22) with the unit constant weighting factor, i.e. $\tilde{\alpha}_i^j(\Gamma_t) = 1$ for all $i,j \in N$.

3.2 FLOW CHART

The simulation flow chart is illustrated in Fig-10. It is divided into five steps.

After setting the time index t to 1, in step 1, we initialize the algorithm by providing the constant network parameters such as C_{ik}, B_i^j, and $\tilde{\rho}_{ik}$ $(i,j \in N, (i,k) \in L)$.

In step - 2, the state informations $\{Q_i^j(t);\ i \neq j,\ i,j \in N\}$ are given from the network and the expected numbers of input packets from outside the network for the next interval are given.

In step - 3, new constraints for the optimization procedure are formulated based on the state information and the expected numbers of packet arrival from the external source. At the same step, new weighting factors $\{\tilde{\alpha}_i^j(\Gamma_t);\ i,j \in N\}$ are determined based on the same information as above.

At this point, all necessary data for the optimization procedure are available and therefore, in step - 4, the objective function $\tilde{J}$ is minimized subject to the given constraints (27),(28), (29) and (30) so that an updated routing policy $\{U_{ik}^j(\Gamma_t),\ i \neq j,\ i,j \in N,\ (i,k) \in L\}$ is obtained.

An integer linear programming technique [23] is used to obtain the solutions.

In step - 5, the routing policy obtained in the step - 4 is executed along with the random events $\{A_i^j(\Gamma_t), \xi_{ik}^j(\Gamma_t),\ i \neq j,\ i,j \in L,\ i,j \in N\}$ generated at the same time. Then step - 2 to step - 5 are repeated until the specified final time T.

For the fixed routing, step - 1 and step -2 are the same as in the dynamic routing. In step - 3, $U_{ik}^j(\Gamma_t)$ is assigned by following way; if $\{Q_i^j(\Gamma_t)+A_i^j(\Gamma_t)\}$ is greater than C_{ik} then $U_{ik}^j(\Gamma_t) = C_{ik}$, otherwise, $U_{ik}^j(\Gamma_t) = \{Q_i^j(\Gamma_t)+A_i^j(\Gamma_t)\}$ for all $i,j \in N$ and $(i,k) \in L$. Finally, step-4 is the same procedure as the step-5 in the dynamic routing. The flow chart for the fixed routing is illustrated in Fig-11.

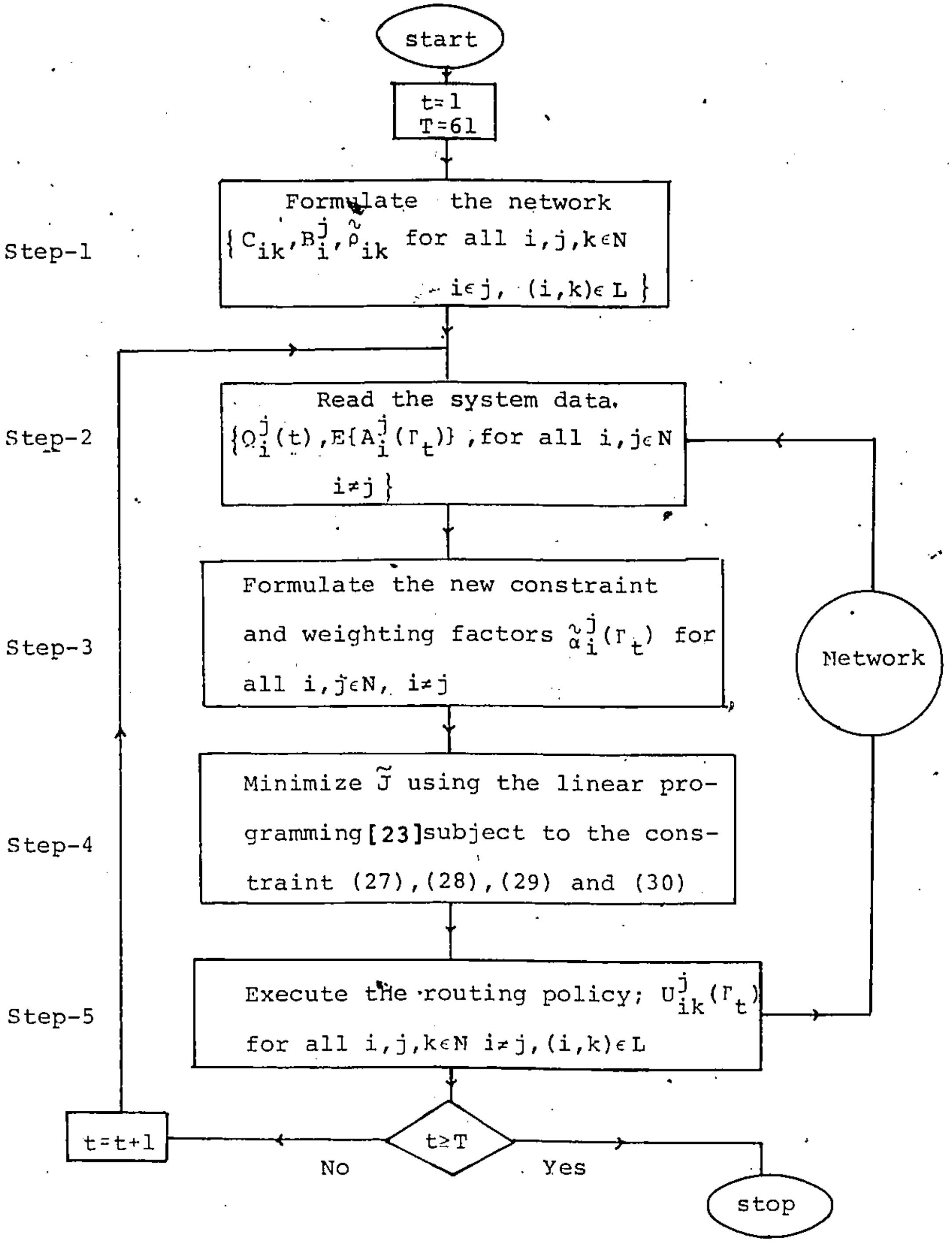


Fig-10 The flow chart for Dynamic Routing.

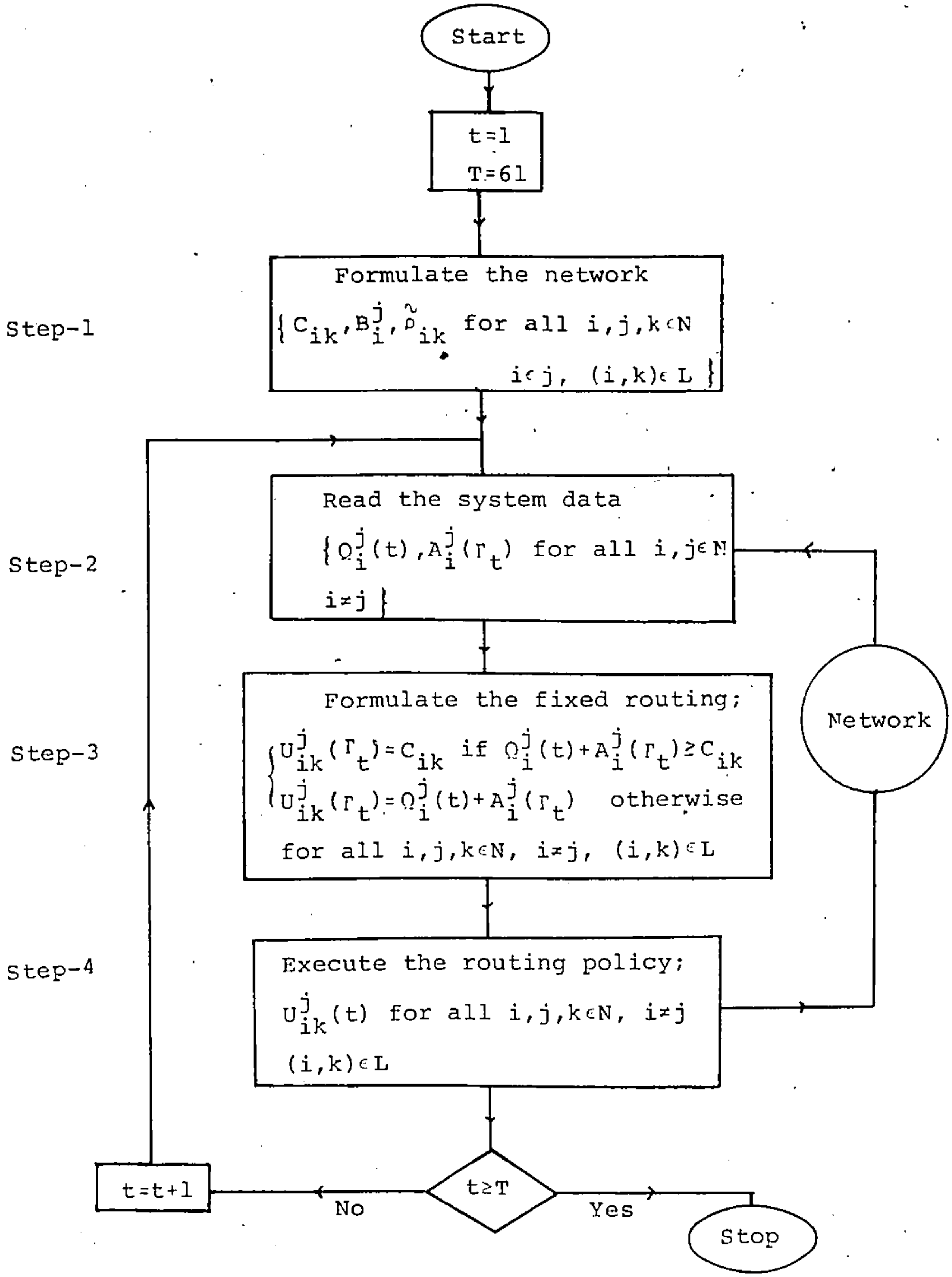


Fig-11 The flow chart for Fixed Routing

3.3 NUMERICAL EXAMPLE

In this section we use the algorithm presented in section 3.2 to compute an optimal routing policy in a simple network shown in Fig-12.

For the sake of simplicity let us assume that the channel capacities and the buffer sizes in the example network are same. We consider the following data for the buffer capacity and the channel capacity;

$B_i^j = 100$ packets

$C_{ik} = 50$ packets/unit-time

for all $i,j,k=\{1,2,3\}, i\neq j$.

The error rates of each channel are assumed to be time invariant and given by

$$\tilde{\rho}_{12} = 1 \times 10^{-2} \quad \tilde{\rho}_{21} = 5 \times 10^{-3} \quad \tilde{\rho}_{23} = 1 \times 10^{-2}$$
$$\tilde{\rho}_{32} = 7 \times 10^{-3} \quad \tilde{\rho}_{31} = 9 \times 10^{-3} \quad \tilde{\rho}_{13} = 2 \times 10^{-3}$$

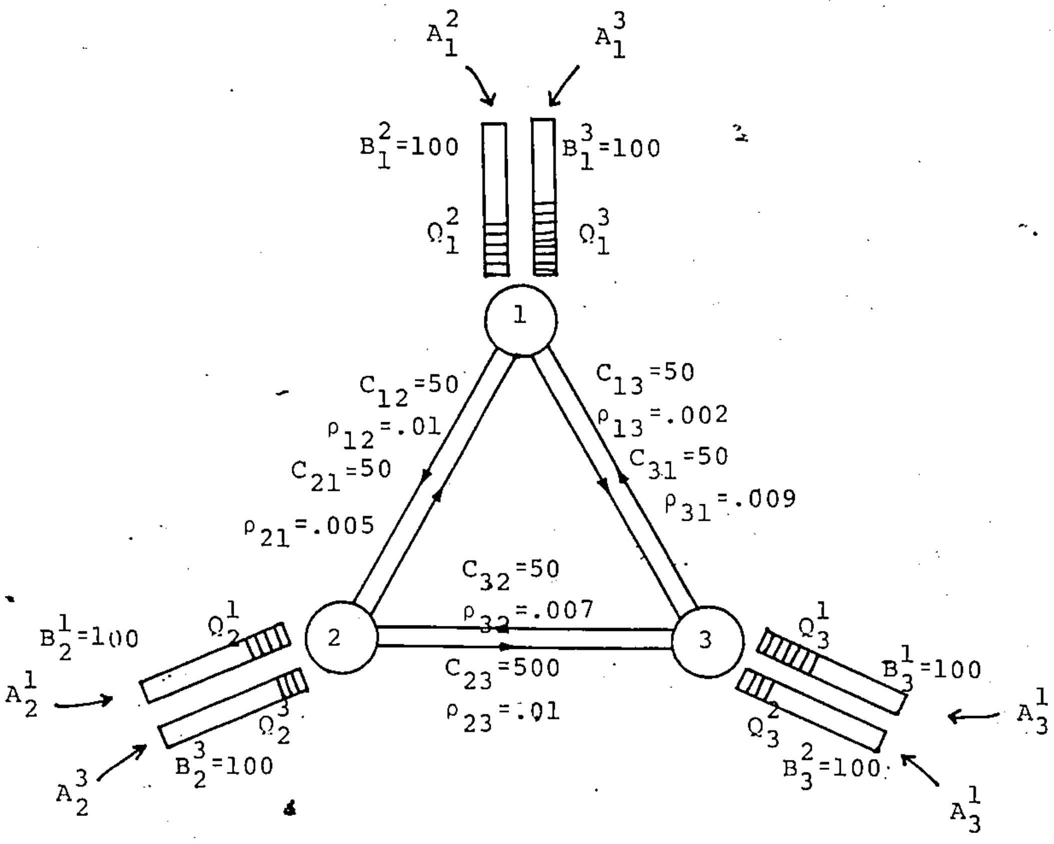


Fig-12 - The network used in the simulation.

The random events such as the number of packet arrived from external sources and transmission errors were generated using the IBM(IMSL) SUBROUTINE GGPOS.

In order to observe the effect of the proposed method on the network performance (average delay and average throughput), the simulation results have been compared with the simple fixed routing. In the fixed routing, every packet arrived at a node will be sent through the pre-fixed channel. For example, a packet destined to the node 2 arrived at the node 1 in Fig-12 is sent through the channel (1,2). This is always fixed and is the shortest path in the network. In this example, channel failure was not considered so that the pre-fixed route was not changed for the whole simulation period. Consequently, there were no alternative routes in the case of fixed routing.

On the other hand, in the case of dynamic routing proposed here, the routing is determined dynamically through a optimization procedure. In this case, alternative routing is possible if the other channels are available.

For the dynamic routing, two different weighting factors were used in the simulations to observe the effects. In the first case the weighting factor is defined by equation (26), and in the other case the weighting factor is defined as unit constant for all i and j.

Chapter IV

ANALYSIS OF THE RESULTS

4.1 INTRODUCTION

To investigate the system performance under the proposed routing scheme, we have applied three different types of external inputs to the system. The first example is with a well balanced input for all destination nodes during the operation period. The second is for an extremely unbalanced input for some destination nodes while the input for some other destination nodes are extremely light or zero during the operation period. The last example is with an arbitrary input which is neither well balanced nor extremely unbalanced.

For the first example, we may consider two types of inputs. A situation is that the average of external inputs are very high for each [illegible] and they exceed the capacity of channels. In this case, the dynamic routing may not provide a better routing policy than the fixed routing, because there is no other alternative route available. The other case is with the very low rate of external inputs and, hence there is almost no delay due to congestion. Again, it is not necessary to provide dynamic routing for this situation.

On the other hand, we expect that the fixed routing cannot provide the proper routing policy for the unbalanced external inputs which might be a more realistic situation. The dynamic routing is useful in these situations because the routing strategy is determined on the basis of the state of the network.

To analyze the results, we need to compute the average delay and the throughput of the system. The throughput of the system was computed using the following relation;

$$\text{Avg.Thr.} = \frac{1}{T}\Big[\sum_{i,j} Q_i^j(0) - \sum_{i,j} Q_i^j(T) + \sum_{t=0}^{T}\sum_{i,j} A_i^j(\Gamma_t) - \sum_{t=0}^{T}\sum_{i,j} L_i^j(\Gamma_t)\cdot 1(L_i^j(\Gamma_t))\Big] \qquad (31)$$

where

$$L_i^j(\Gamma_t) = Q_i^j(t) + A_i^j(\Gamma_t) - B_i^j$$

$$1(x) = \begin{cases} 1 & \text{if } x>0 \\ 0 & \text{otherwise} \end{cases}$$

The first term in (31) indicates the total number of packets waiting in the network at the initial time. The second term represents the total number of packets that remained at the final time T. The third term gives the total number of packets arrived from external sources to the network. Finally, the last term implies the total number of the lost packets from the external source due to shortage of the available buffer space.

Using equation (31) the average delay can be computed using the following relation;

$$\text{Avg.Delay} = \frac{\frac{1}{T} \cdot \sum_{t=0}^{T} \sum_{i,j} Q_i^j(t)}{\text{Avg.Thr}} \tag{32}$$

Note that the average delay given above represents the average number of packets waiting in the network divided by the average throughput given in the equation (31).

4.2 EXAMPLE 1

In the first example, a well balanced external input (with mean arrival rate of 50 packets per unit-time for all type of external input) has been applied to the system. These external arrival increments per unit-time for each node are shown in Fig-13. For comparison, we illustrate the changes in the queue lengths using the fixed routing policy and the dynamic routing policies. The changes of queue length in each buffer at time t using the fixed routing policy (case-1) are plotted in Fig-14. Similarly, the changes of queue length for the dynamic routing are plotted in Fig-15 (case-2) and Fig-16 (case-3). Two different weighting factors have been applied for the dynamic routing.

The straight lines shown in Fig-14, Fig-15, and Fig-16 indicate the buffer size limitations; clearly, the shaded areas in these figures indicate the number of packets lost due to the shortage of the available buffer space. Comparing the figures, we observe that there is no loss of packets for the queue in the buffer B_1^2, B_2^1, and B_3^2 for all the three cases. On the other hand, from the case-1 (Fig-14), we observe some losses of packet at B_1^3, B_2^3, and B_3^1. This is because of the fact that the arrival rates are higher than the corresponding channel capacity. This also happened in the case-2 (Fig-15) and in the case-3 (Fig-16) where dynamic routing policy was used. However, the loss of packets is more or less the same in the fixed routing or the dynamic routing.

The total number of packets waiting in the network at every time for each routing policy are plotted in Fig-18. Also the total number of packets lost at every time for each case are plotted in Fig-17 and the average delay and the average throughput are shown in the Table-2. As expected, the results of the two dynamic routing and the fixed routing are almost identical. This is quite logical because each channel is equally utilized when the external inputs are well balanced for each type of traffic. Consequently, in this case dynamic routing may not be necessary. However, in practice one may not expect this equally balanced situation.

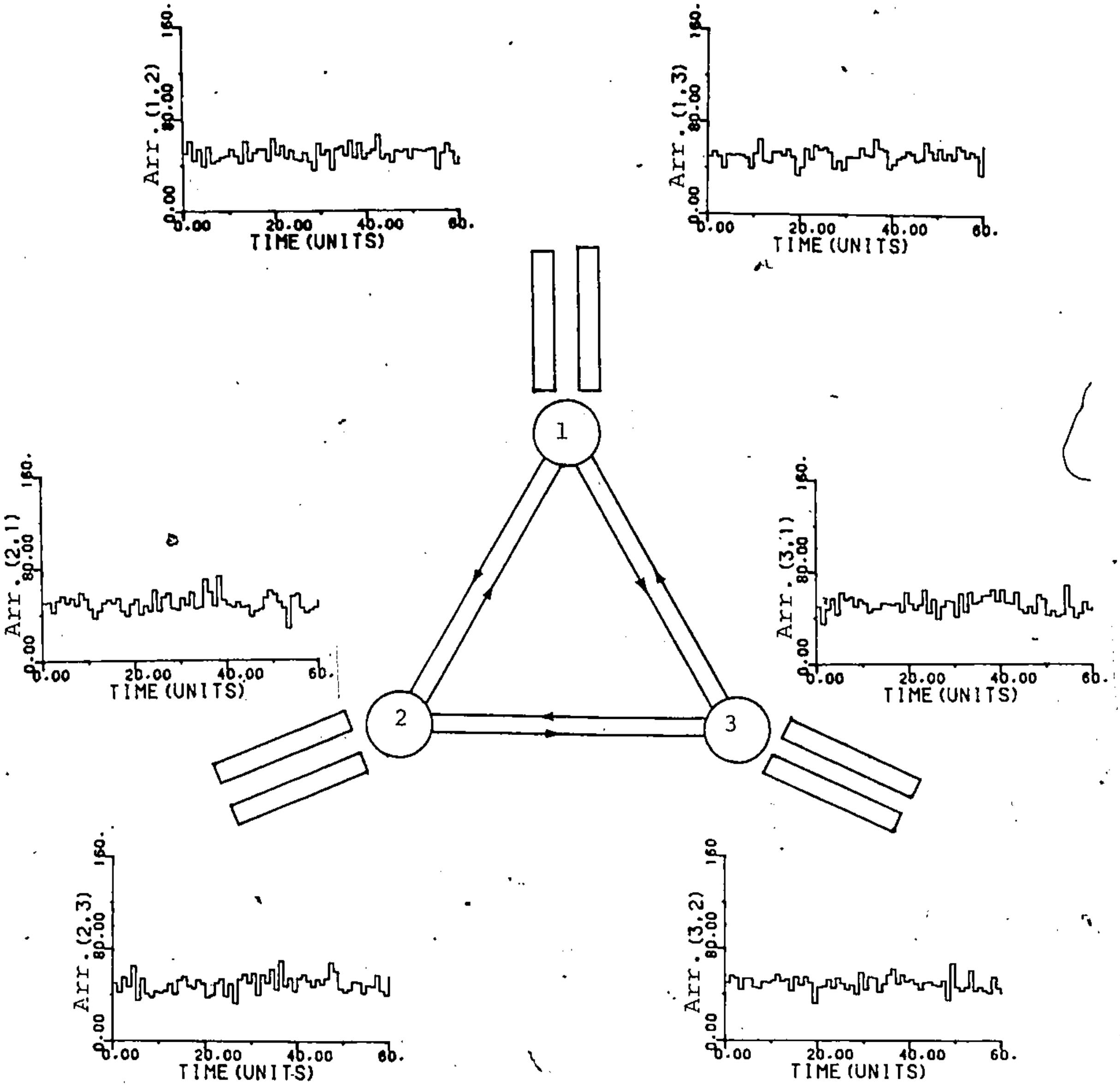


Fig-13 External arrivals at each buffer B_i^j.

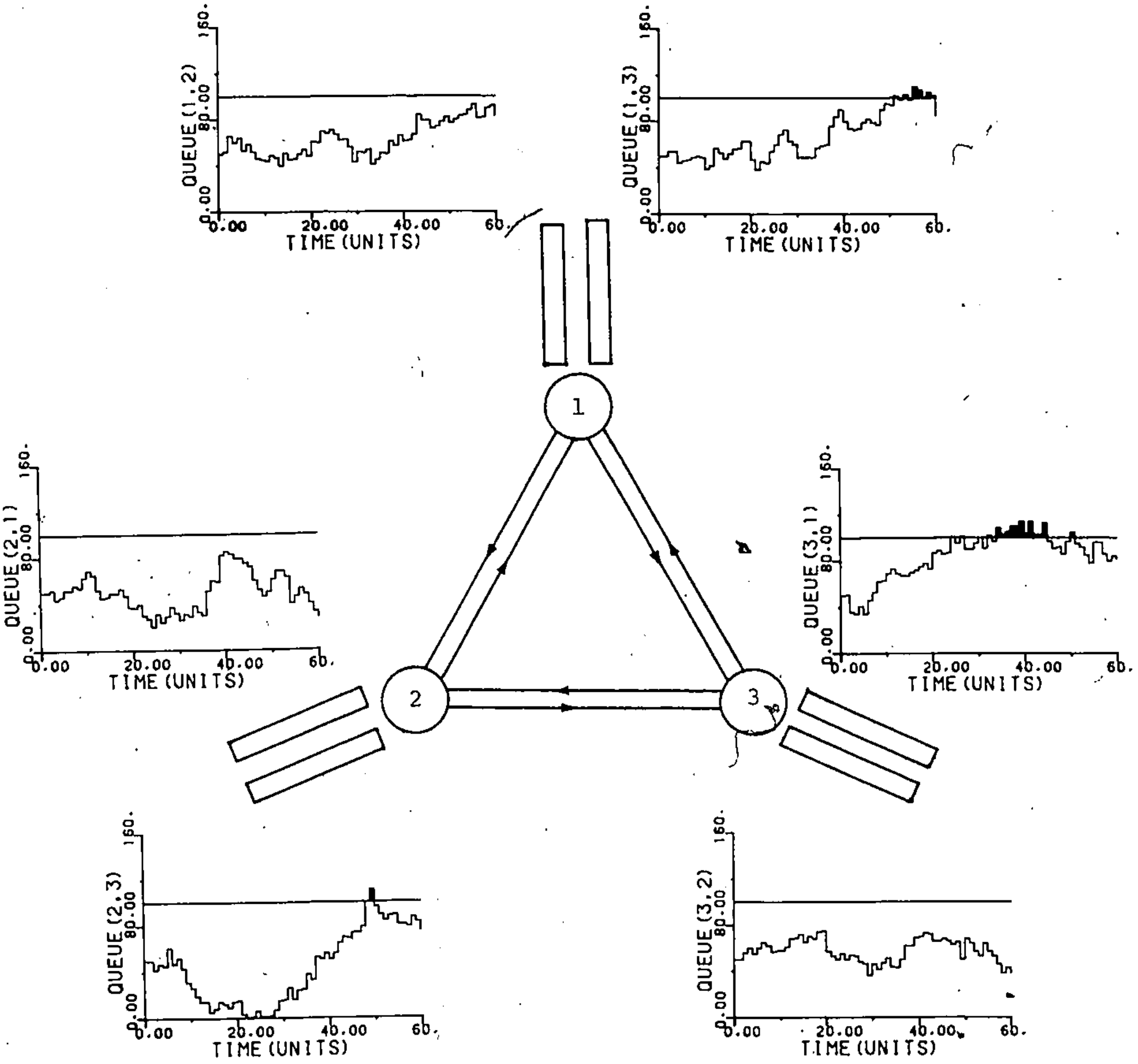


Fig-14 The state of queue at each buffer B_i^j using the fixed routing. (case-1)

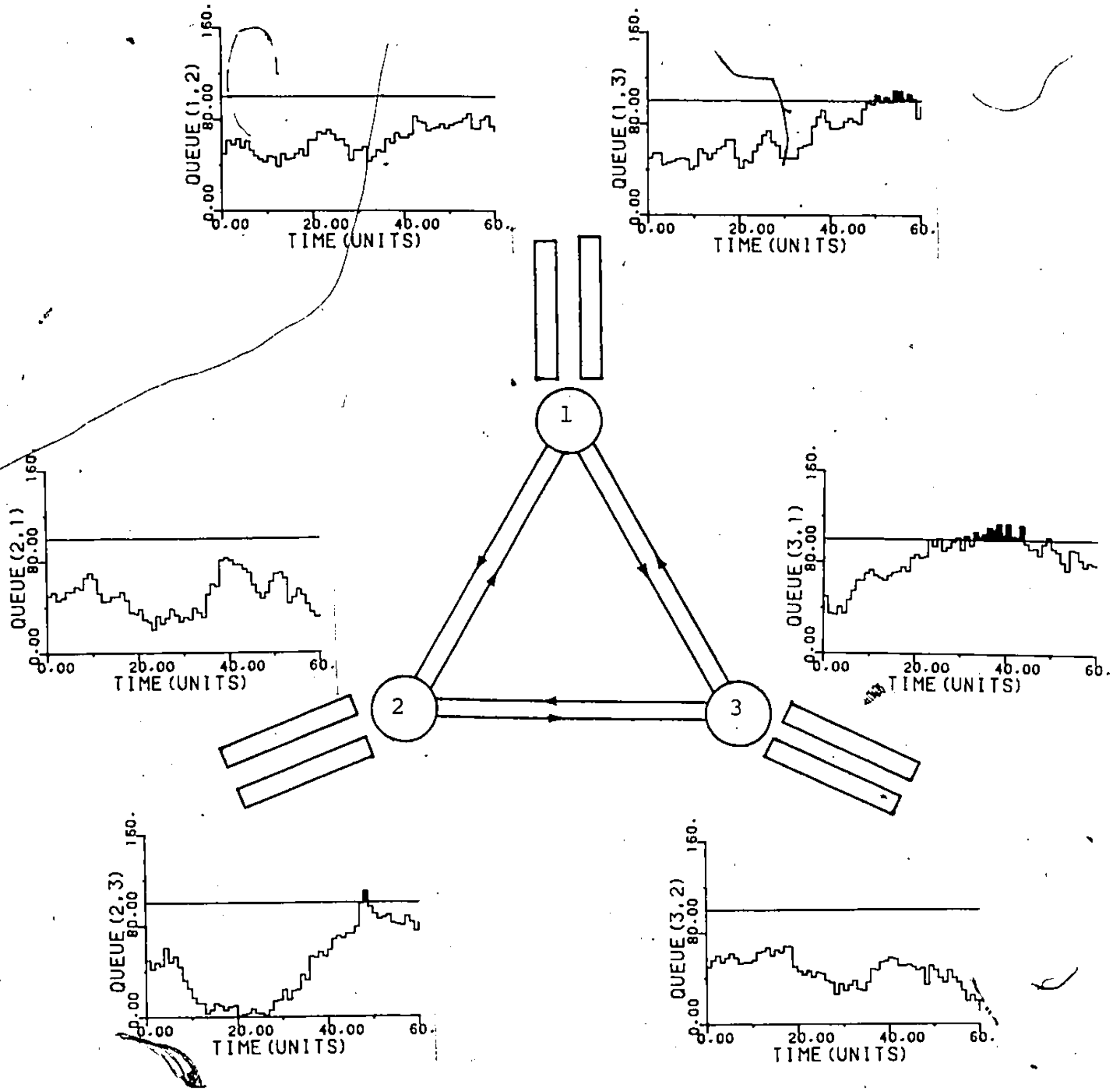


Fig-15 The state of queue at each buffer B_i^j using the dynamic routing with weighting factors (26). (case-2)

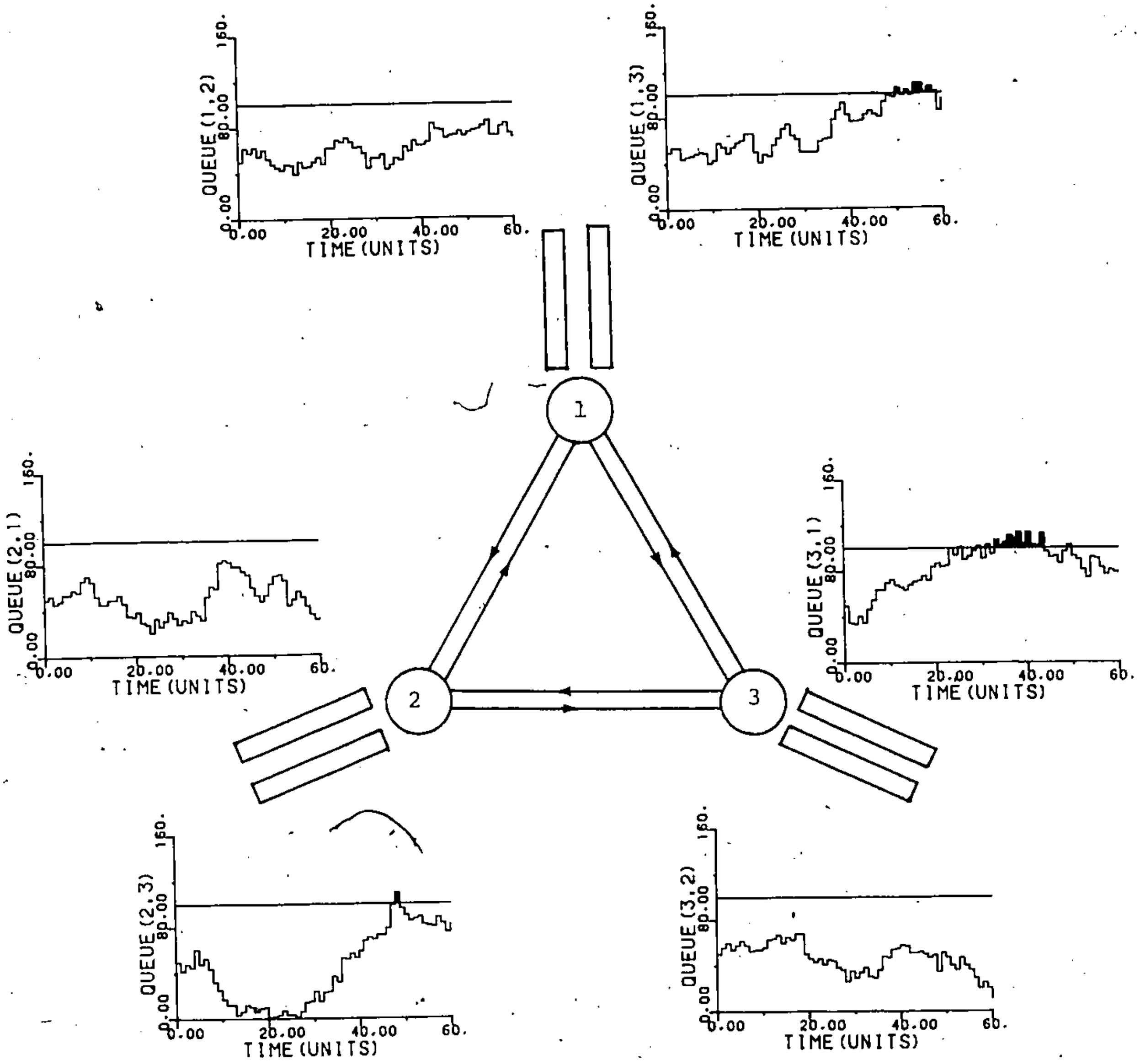
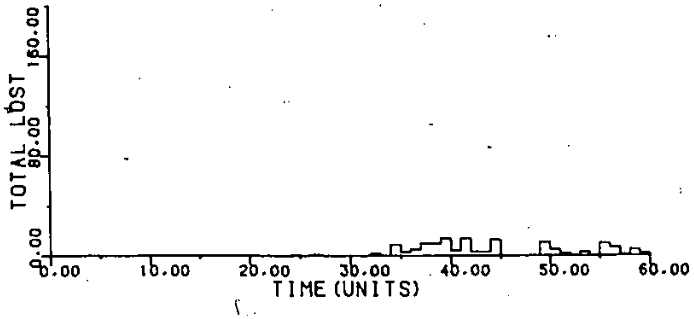
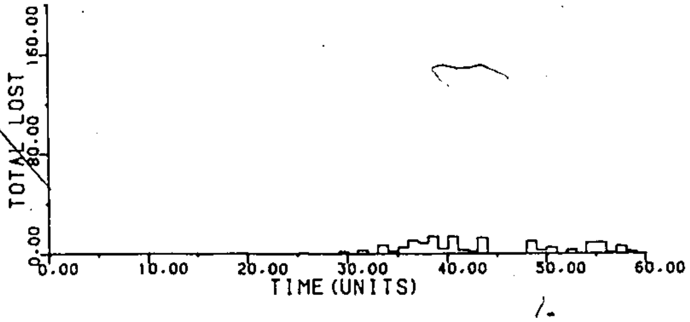


Fig-16 The state of queue at each buffer B_i^j using the dynamic routing with uniform weights. (case-3)

a) Fixed Routing



b) Dynamic Routing
with weighting
factors (26)



c) Dynamic Routing
with uniform
weights

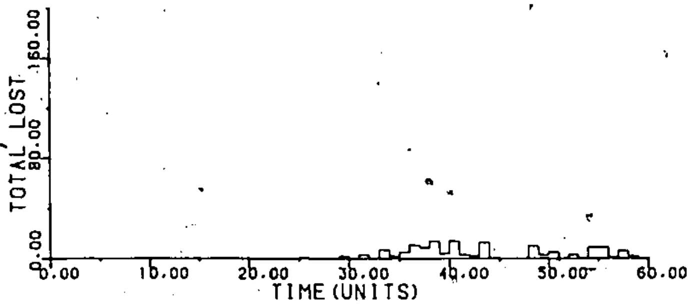
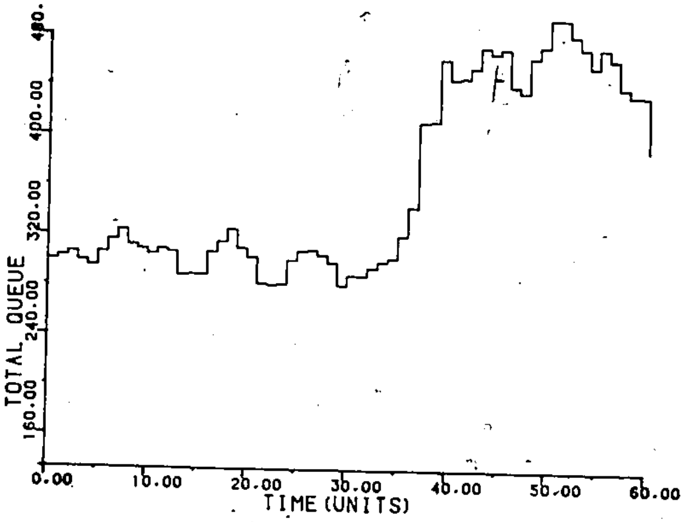
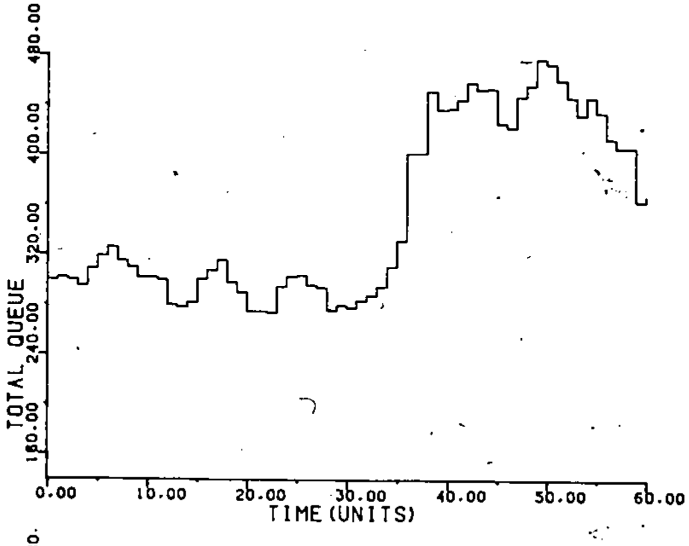


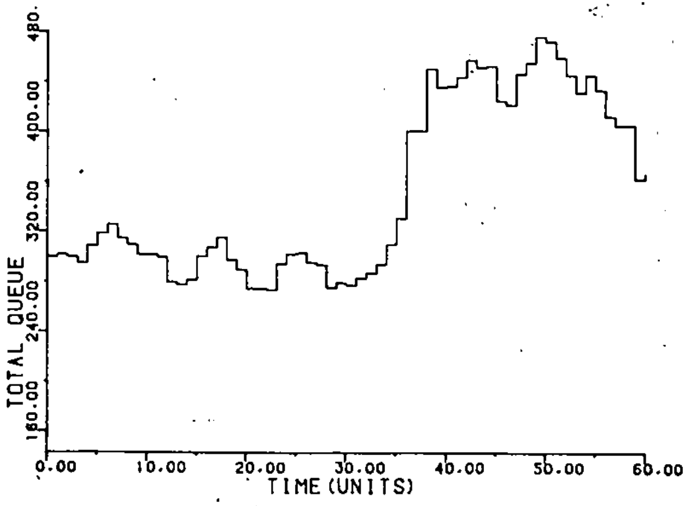
Fig-17 The number of packet losses in the network due to buffer limitations.



a) Fixed Routing



b) Dynamic Routing
with weighting
factors (26)



c) Dynamic Routing
with uniform
weights

Fig-18 The total number of packets waiting in the network.

	case 1	case 2	case 3	
average delay	1.22	1.19	1.19	unit-time/packet
average throughput	297.1	297.6	297.6	packets/unit-time
total number of packet losses	136	138	138	packets
total external input	18061			packets

Table-2

4.3 EXAMPLE 2

In the second example, we consider the other extreme case in which the external input is extremely unbalanced. The arrival increments per unit-time for each node used in the simulation are shown in Fig-19. In Fig-20, Fig-21, and Fig-22, we show the queue length in each buffer, as function of time, corresponding to the fixed and dynamic routing policies.

In case-1 i.e. fixed routing (Fig-20), there is a heavy loss of packets at B_1^3, B_2^1, and B_3^2 as shown by the shaded areas. On the other hand, the dynamic routing policy reduces this loss of packets to a minimum which could be observed from the corresponding diagrams in Fig-21 and Fig-22. This is due to the fact that dynamic routing can provide alternative paths for the packets at congested buffers as explained earlier. In this example, also, the results of the case-2 and the case-3 are almost identical. The total number of packets waiting in the network at every time for each cases are plotted in Fig-24. The total number of packets lost for each cases are plotted in Fig-23. The average delay and the average throughput for each cases are shown in the Table-3. From the simulation results it was found that in the case of fixed routing the number of packets lost, due to buffer limitation, is very large (4398 packets). On the other hand, using the dynamic routing, this number is much reduced (271 packets). As we expected, the dynamic routing can achieve much higher average throughput (224.9 packets/unit time) as compared to that for the fixed routing (154.8 packets/unit time). Further, it is also clear from the Table-3 that the average delay in the case of dynamic routing (1.17 unit time/packet) is much less than that of fixed routing (1.93 unit time/packet). These results indicate a high performance characteristic of the dynamic routing on the average delay and average throughput.

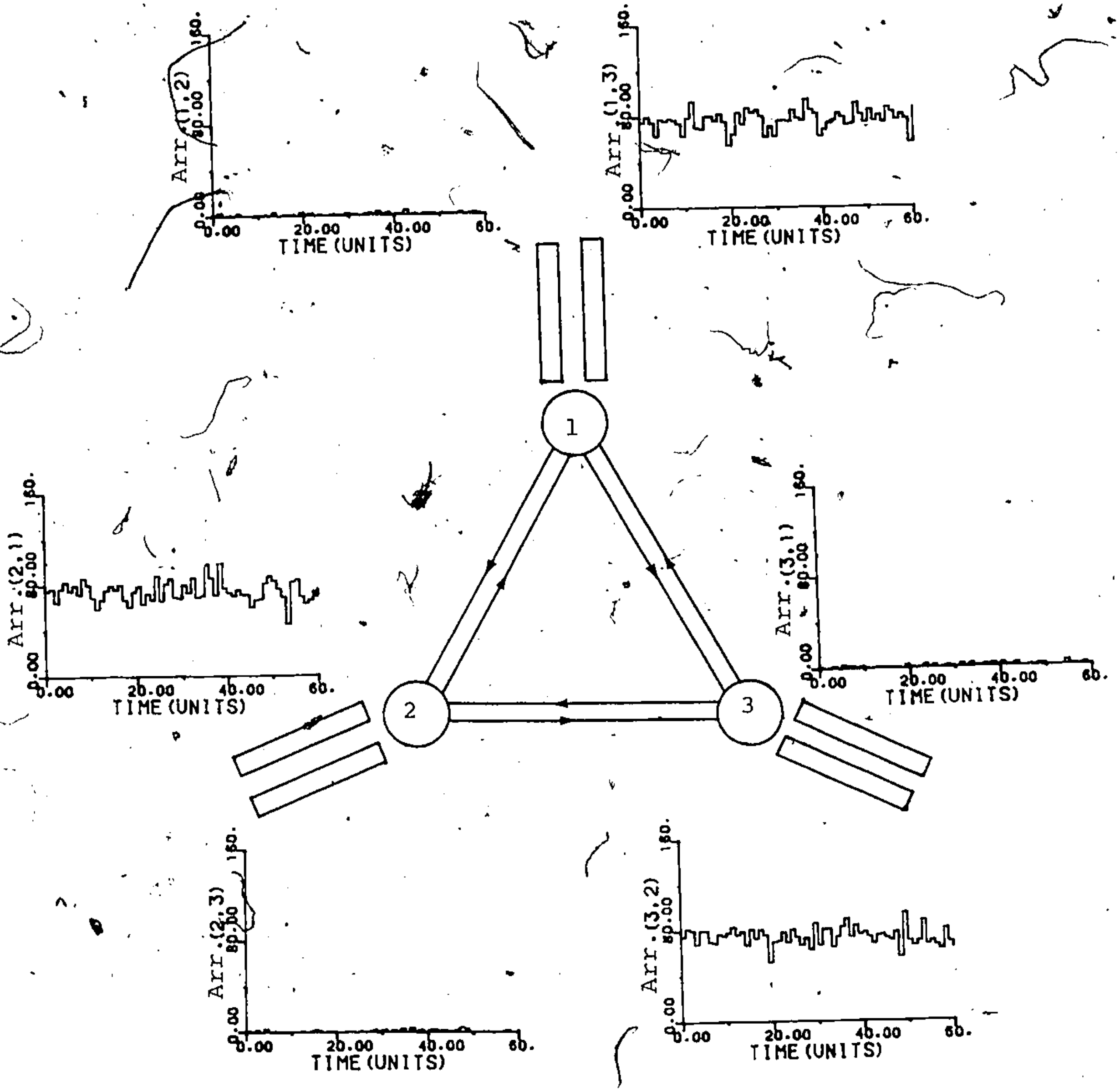


Fig-19 External arrivals at each buffer B_i^j.

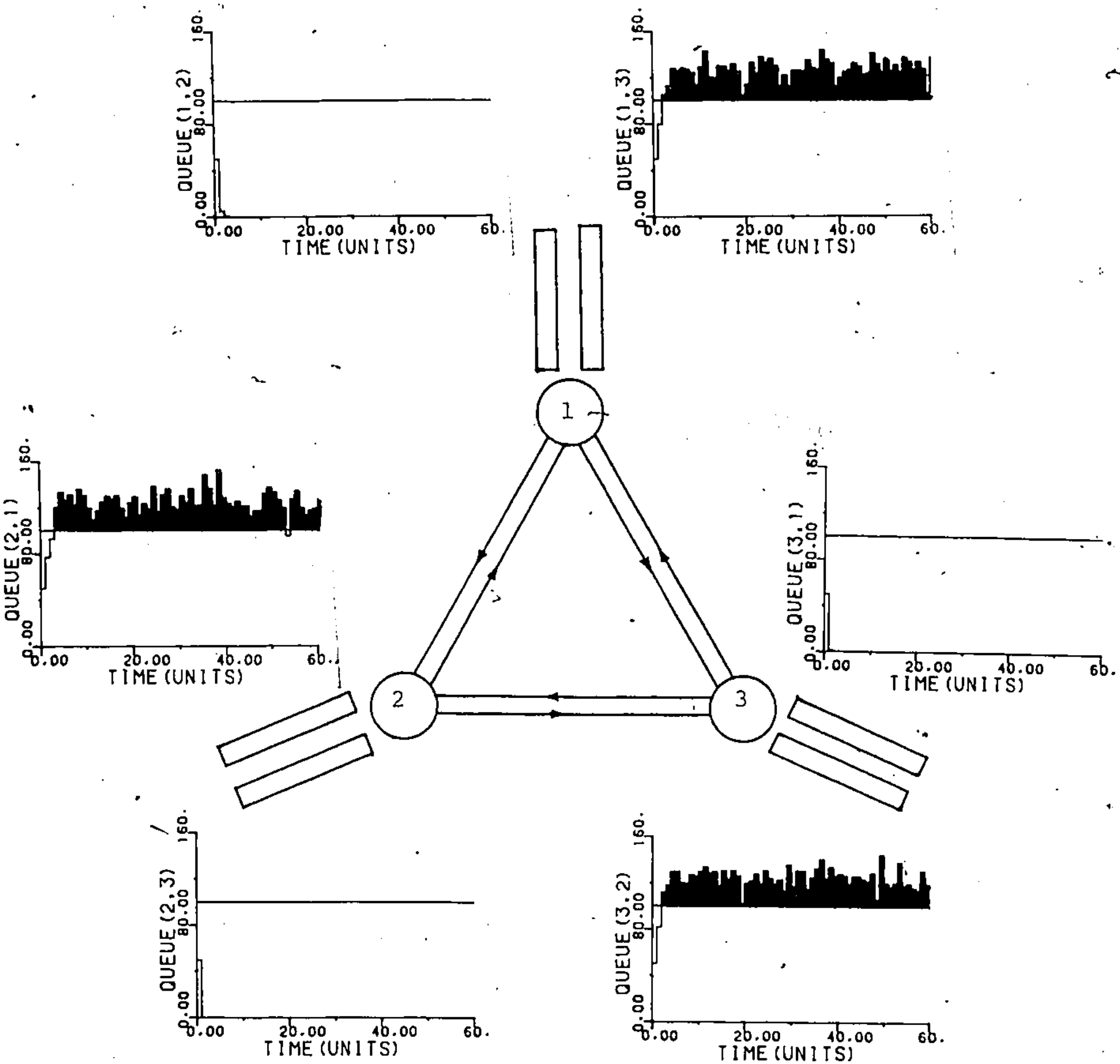


Fig-20 The state of queue at each buffer B_i^j using the fixed routing. (case-1)

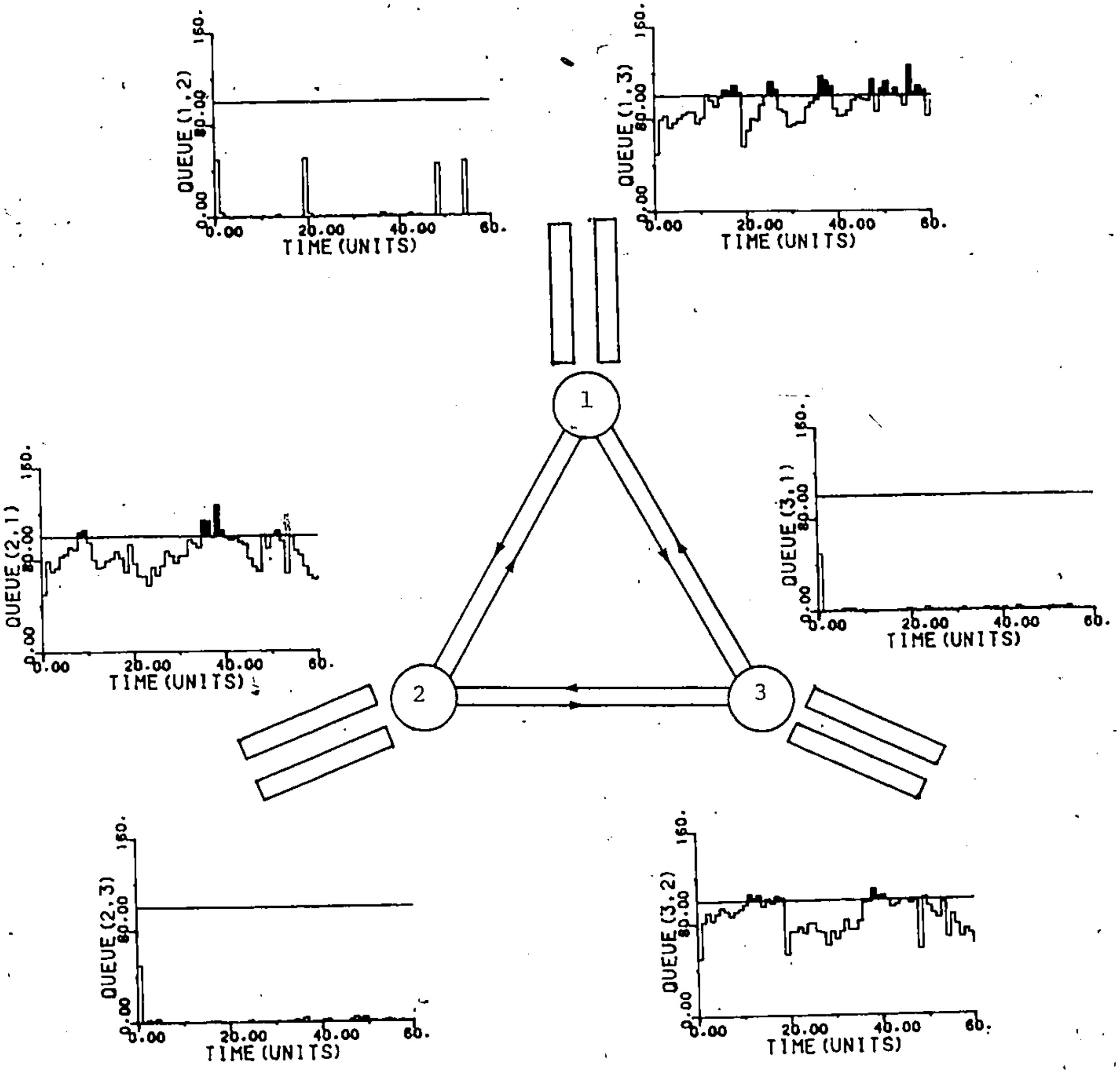


Fig-21 The state of queue at each buffer B_i^j using the dynamic routing with weighting factors (26). (case-2)

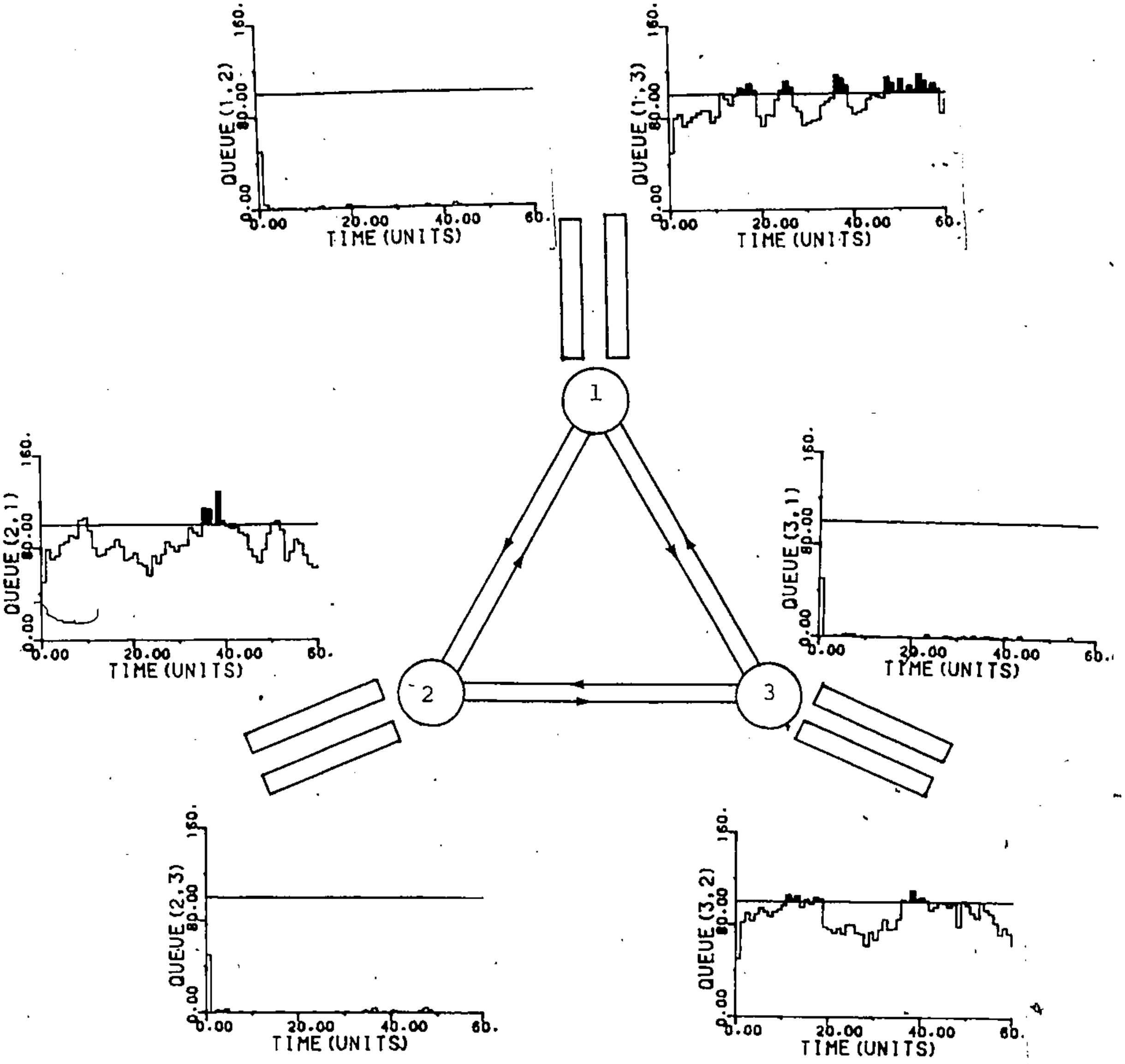
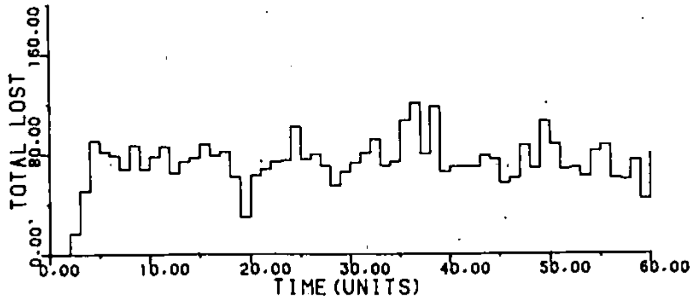
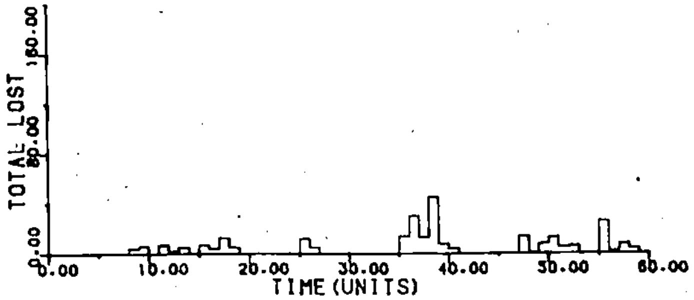


Fig-22 The state of queue at each buffer B_i^j using the dynamic routing with uniform weights. (case-3)

a) Fixed Routing



b) Dynamic Routing
with weighting
factors (26)



c) DynamicrRouting
with uniform
weights

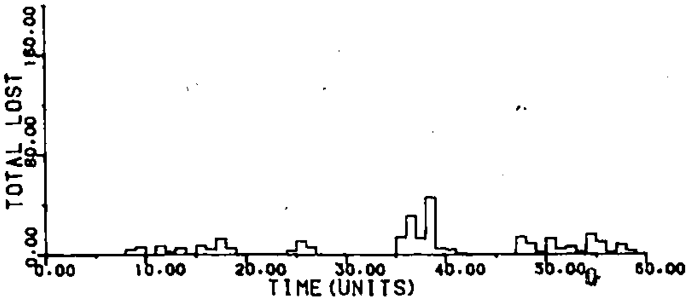
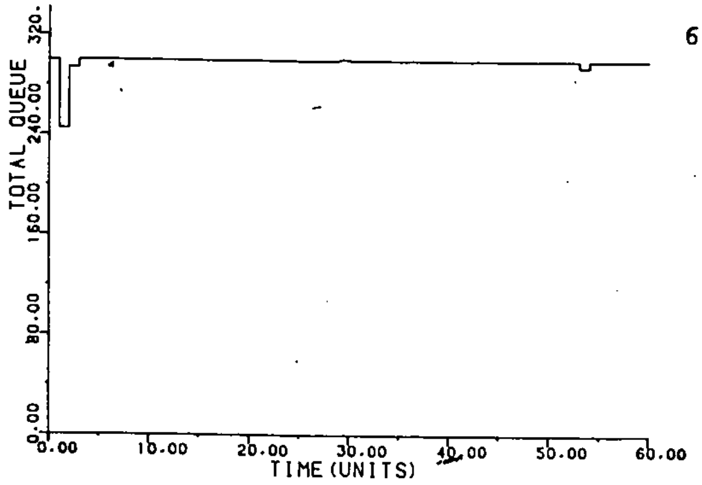
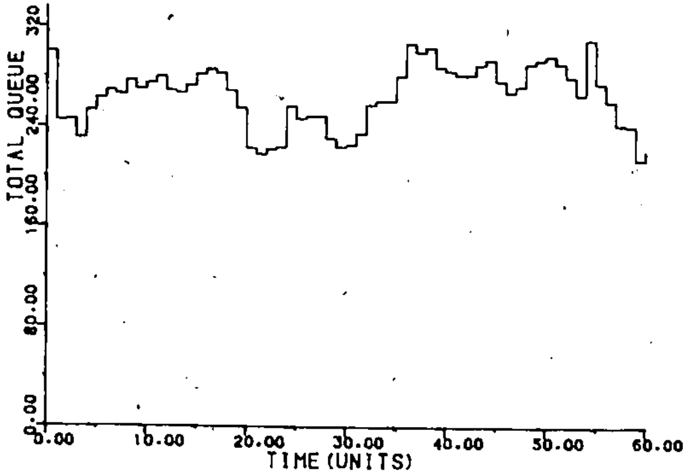


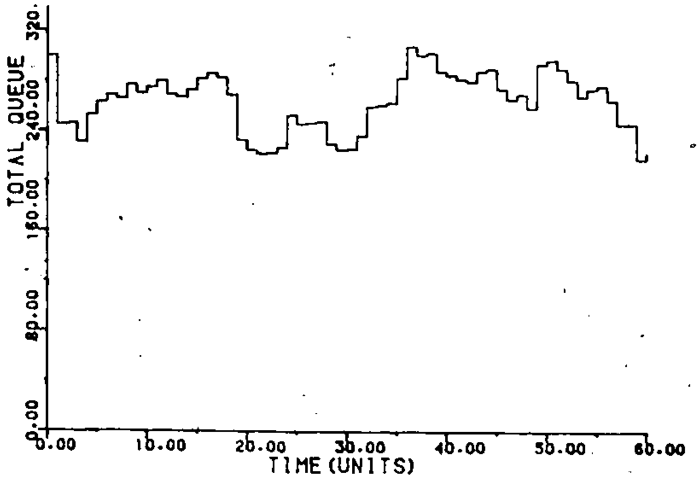
Fig-23 The number of packet losses in the network due to buffer limitations.



a) Fixed Routing



b) Dynamic Routing with weighting factors (26)



c) Dynamic Routing with uniform weights

Fig-24 The total number of packets waiting in the networks.

	case 1	case 2	case 3	
average delay	1.93	1.17	1.17	unit-time/packet
average throughput	154.8	224.9	224.8	packets/unit-time
total number of packet losses	4398	271	276	packets
total external input	13686			packets

Table-3

4.4 EXAMPLE 3

In this example, the mean of external arrival rates are changed with time and the situation is neither well balanced as in the example 1 nor extremely unbalanced as in the example 2. Therefore this example would be more close to real situation. The external arrival increments per unit-time for each node are shown in Fig-25. The changes of queue length corresponding to the fixed routing and the dynamic routing policies are plotted in Fig-26, Fig-27, and Fig-28 respectively. Finally, summary of the results are shown in Fig-29, Fig-30, and Table-4.

The results of fixed routing are shown in Fig-26, Fig-29-a, and Fig-30-a. Congestion appears at the buffer B_1^2, B_2^1, and B_2^3 in the case-1 (Fig-26), and many packets are lost due to the shortage of available buffer spaces. Every time the external arrival rates exceeded the capacity of the assigned channel, the congestion couldn't be avoided because only one outgoing channel is available for each type of traffic in the case of fixed routing. On the other hand, applying the dynamic routing for the same condition of the system, those packets in the congested buffers were distributed using the other available channels so that congestion at the buffers is minimized. This is clearly observed from Fig-27 and Fig-28. Therefore this strategy not only reduced the local congestion but also achieve the better utilization of the network (Table-4).

In the dynamic routing using the weighting factor defined in (26) i.e. case-2, some queues appeared at the buffer B_3^2 (Fig-27), although in the case of fixed routing there was no queue at this buffer. This has happened because the routing was done in order to reduce the loss of the external arrival packets. If we compare the case-2 and the case-3 (method using unit constant weight), then there are some improvements on delay (see Fig-30); but the dynamic routing with constant weights (Fig-29) resulted in larger number of lost packets. However, this is expected since, in the case-3 (i.e. constant weights), all the queues are given equal priority disregarding the possible congestion at certain buffers, while in case-2, congested buffers are cleared faster than the other non congested buffers.

We can conclude from this example that the dynamic routing can achieve much higher performance on the average delay and average throughput than the fixed routing (Table-4).

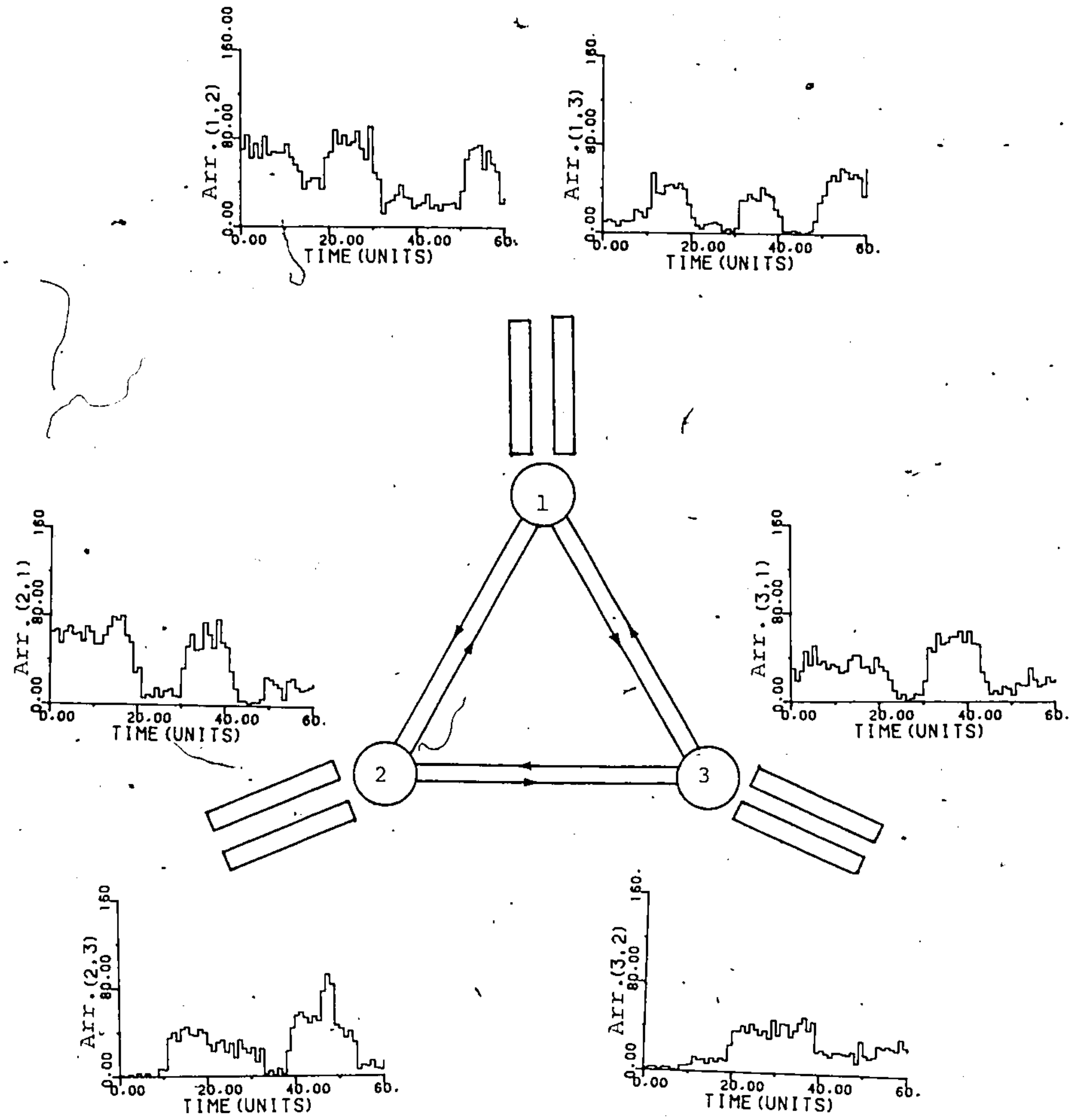


Fig-25 External arrivals at each buffer B_i^j.

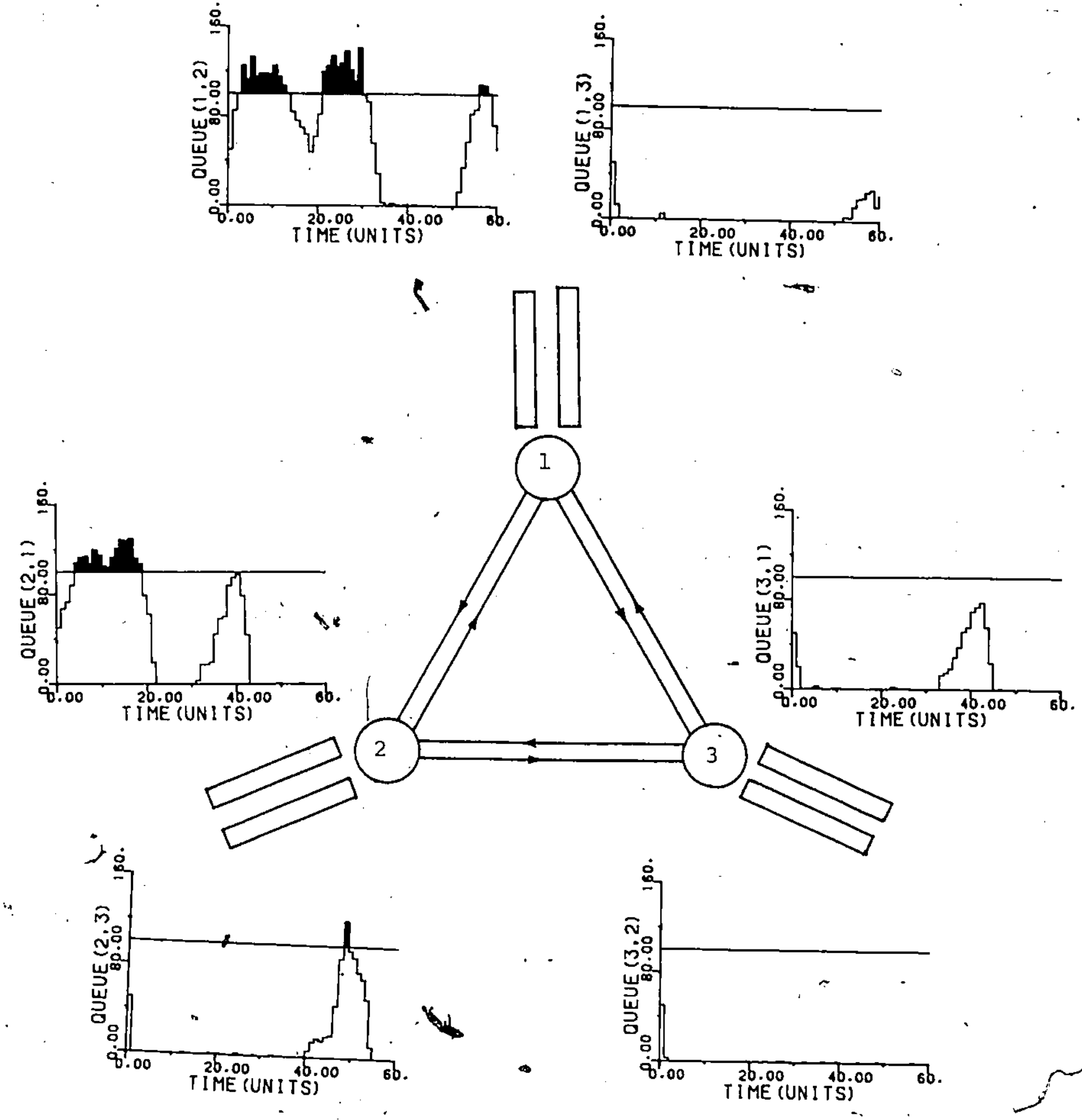


Fig-26 State of queue at each buffer B_i^j using the fixed routing.
(case-1)

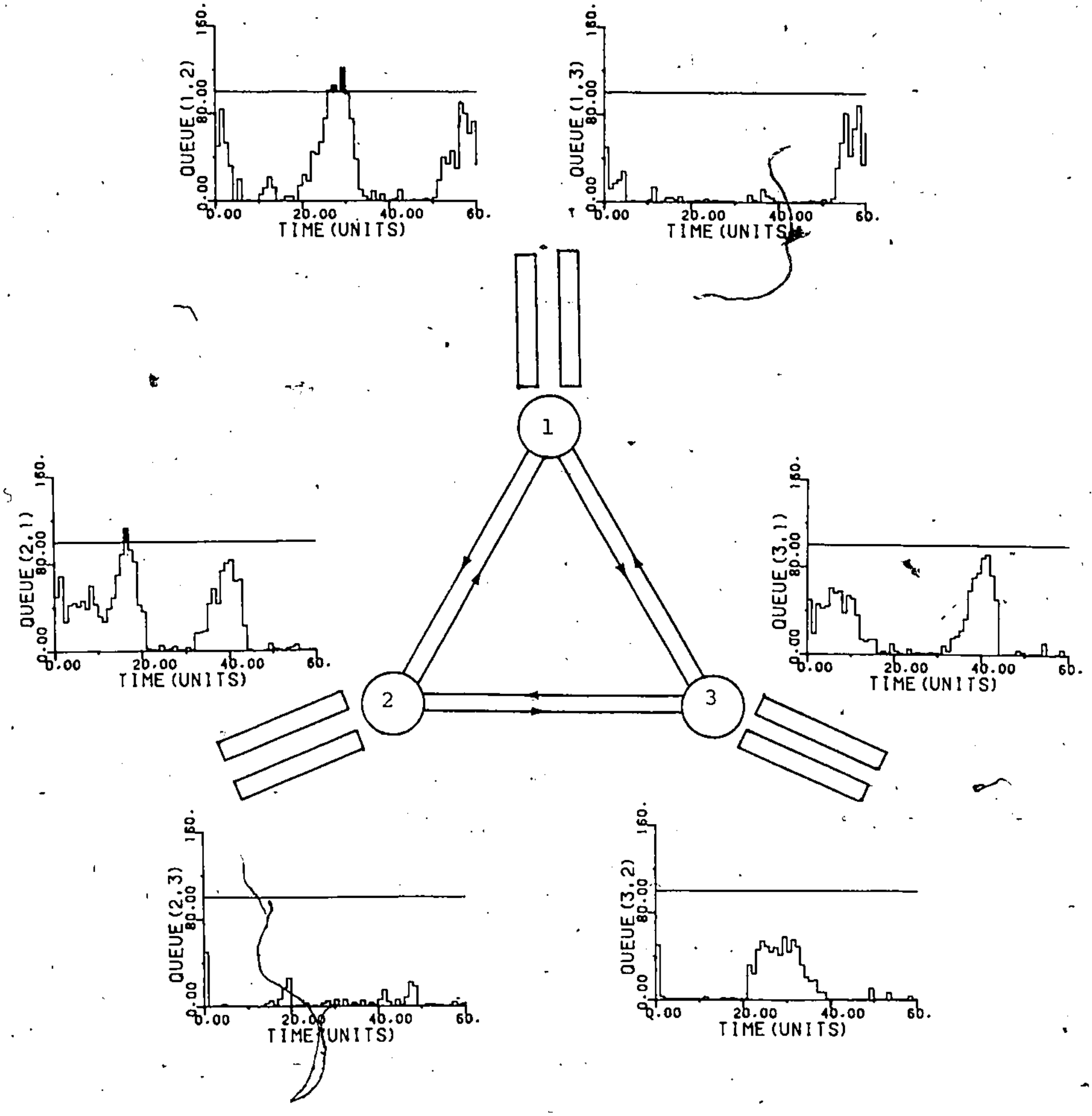


Fig-27 State of queue at each buffer B_i^j using the dynamic routing with weighting factors (26). (case-2)

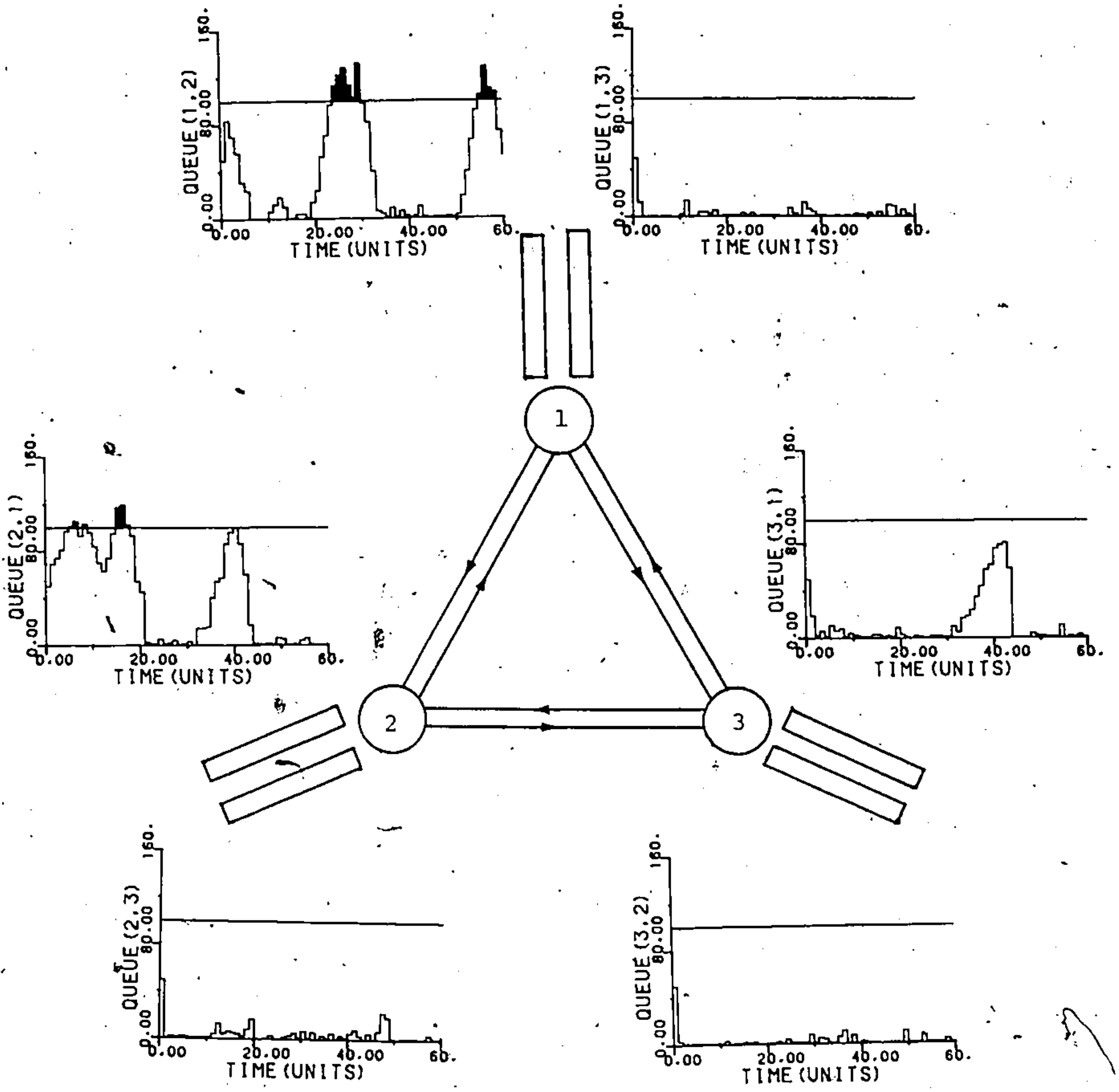
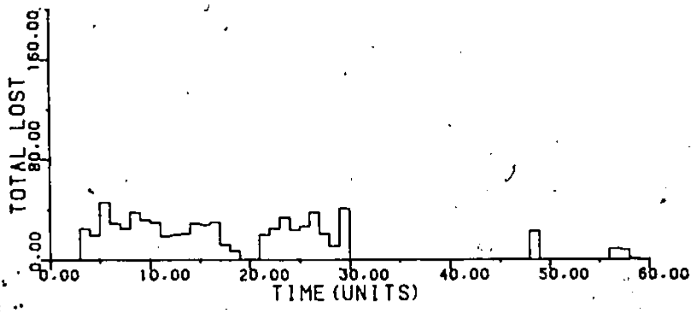
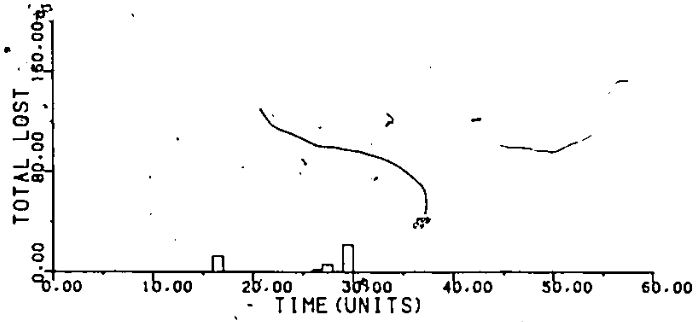


Fig-28 State of queue at each buffer B_i^j using the dynamic routing with uniform weights. (case-3)

a) Fixed Routing



b) Dynamic Routing
with weighting
factors (26)



c) Dynamic Routing
with uniform
weights

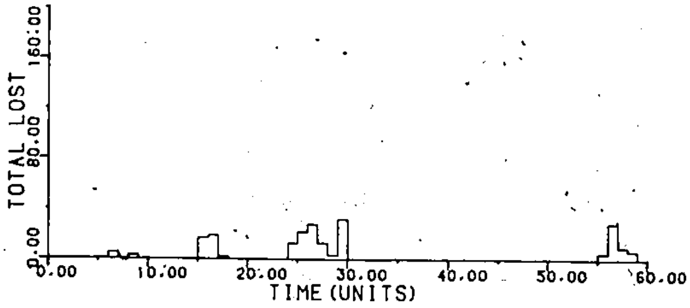
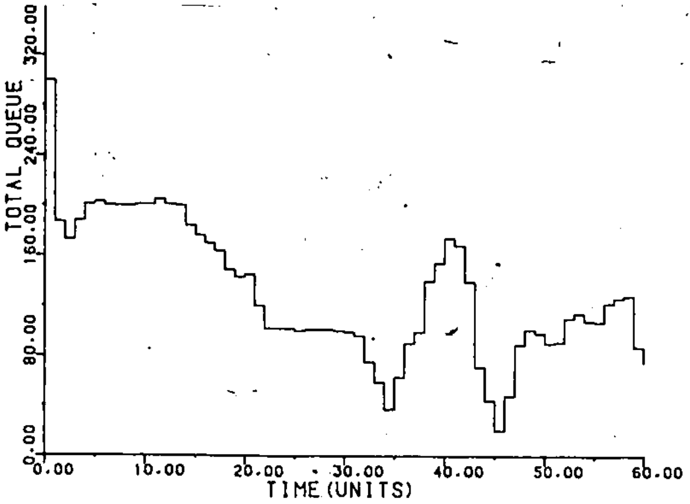
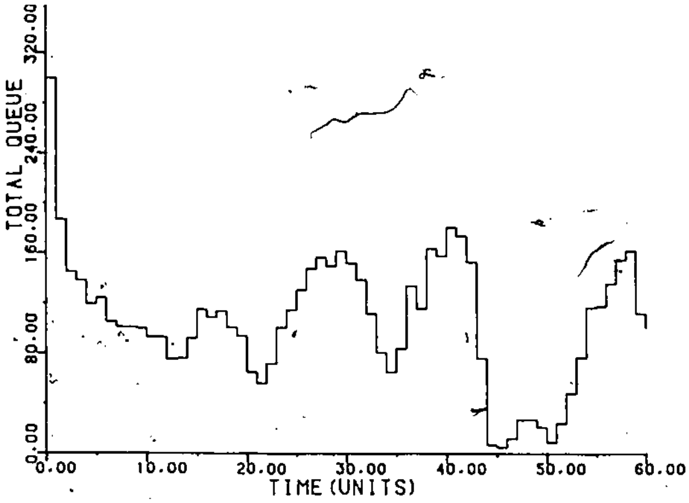


Fig-29 The number of packet losses in the network due to buffer limitation

a) Fixed Routing



b) Dynamic Routing with weighting factors (26)



c) Dynamic Routing with uniform weights

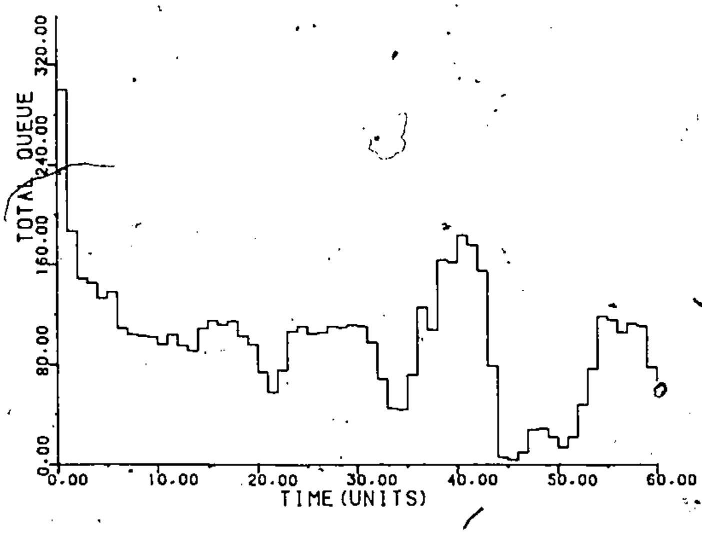


Fig-30 The total number of packets waiting in the network.

	case 1	case 2	case 3	
average delay	0.69	0.53	0.50	unit-time/packet
average throughput	183.3	193.7	191.5	packets/unit-time
total number of packet losses	693	43	209	packets
total external input	11467			packets

Table-4

Chapter V

CONCLUSIONS

It has been recognized that a model for dynamic routing is needed to accommodate a changing environment in the computer communication networks. However, not many studies concerned with the dynamic routing have been reported in the literature. Besides, to the knowledge of the author, there is no dynamic routing strategy for multi-destination traffic considering stochastic inputs, buffer limitation, and channel errors.

In this thesis, we incorporate the above situations so that our model is more realistic as compared to the previous models. Also a centralized dynamic routing strategy is proposed to minimize delay, and achieving the maximum utilization of the network. This strategy is obtained by minimizing the simple objective function proposed in this thesis. From the simulation results, it has been found that the proposed routing strategy can provide good performance in terms of minimum delay and maximum throughput.

Although we have assumed that the state information can be measured at each instant of time and used for routing control, there will be some time delay in the process of information exchange in actual communication system. Therefore, for practical application, state information should be estimated ahead of time. The estimation technique would be quite interesting for further research.

It would also be interesting to investigate distributed dynamic control strategy for stochastic systems since geographic distribution of network makes it difficult to obtain the accurate information needed. Since our dynamic model describes the individual change of the queue size in each buffer, the model can be used for minimizing the queues in each node based on the state information of the neibouring nodes. We believe that our model will also be useful for the development of the distributed dynamic control strategy.

Appendix A

PROGRAM LISTINGS

1. The program for Dynamic Routing;

pp 78-104

2. The program for Fixed Routing;

pp 105-108

```
FILE: MODEL     FORTRAN  A                     VM/SP CONVERSATIONAL MONITOR S

       DOUBLE PRECISION ATAB(21,15), UPBND(15),T(15),VAL(15)
       DOUBLE PRECISION SOLMIN,PCTTOL,ZOPT,QRP,TLRNCE
       DOUBLE PRECISION A(20,20),B(20),C(12),PSOL(18),DSOL(20),RW(458),
     $    CP(3,3),BF(3,3),ER(3,3,100),Q(3,3),ALP(3,3),BQR(3,3),QR(3,3),
     $   EQ(3,3),R(3,3),QL(3,3),QLOST,DALP(3,3),BQ(3,3),EER(3,3),TARR,
     $   QT(3,3,100),QQ(3,3,100),QLT(3,3,100),ABF,RO(3,3),AR(3,3,90),
     $   RRO(12),TE(12),RS(3,3,100),TDSE(20),TDE(20),DS(3,3),DI,DJ,TQ,P,
     $   PX(3,3),TP,AVD,THR,QQQ,QIL
       DOUBLE PRECISION DSEED
       DIMENSION IROW(21)
       COMMON /IO/ NI,NO,NM1,IROW,IPACK,ITOL
       COMMON /IYS/ ISIZE,NMRUNS,IOUT2,IOUT3,INCTR
       COMMON /SP/ SOLMIN,PCTTOL
       COMMON /CONS/ UPBND,ATAB,T,ZOPT,VAL,TLRNCE
       INTEGER S , S1,G(100),IW(58)
       DSEED=123457.D0
       NR=1
       K=61
       QLOST = 0.D0
       QQQ=0.D0
C************************
   10 FORMAT (1H0, (7D10.3))
C      UNPACKED FORMAT NO. 11
   11 FORMAT ( 7D10.3     )
   13 FORMAT (1H0,24HPRINT CONTROL PARAMETERS)
   14 FORMAT (1H0,30HUPPER BOUND ON VARIABLE 1 TO N)
   15 FORMAT (20I4)
   17 FORMAT (1H0,18HMATRIX FORMAT CODE)
   18 FORMAT (4H0I =, I4, 6I10)
   19 FORMAT (27H0STRUCTURAL VARIABLES: X(I))
   20 FORMAT (44H0ROWS X COLUMNS AND NO. OF INTEGER VARIABLES,//,
    1      I4,2H X, I3, 24X, I6)
   21 FORMAT (30H0CONSTRAINT TYPES IN ROW ORDER)
   22 FORMAT (51H0INPUT TABLEAU ECHO, CONSTRAINT VALUE LEFT, BY ROW.)
   23 FORMAT (1H0,13D10.1/(1H , 13D10.1))
   35 FORMAT (21H0OBJECTIVE FUNCTION =, F13.7,14H  AT ITERATION,I6)
   60 FORMAT(33H0                                )
       NI = 5
       NO = 6
       DO 1 I=1,3
       DO 1 J=1,3
       Q(I,J)=0.D0
    1  CONTINUE
C********** CAPASITY OF CHANNELS **********
       CP(1,2)=50.D0
       CP(2,1)=50.D0
       CP(2,3)=50.D0
       CP(3,2)=50.D0
       CP(3,1)=50.D0
       CP(1,3)=50.D0
C***************ERROR RATE OF CHANNELS***********************
       RO(1,2)=1D-2
       RO(2,1)=5D-3
       RO(2,3)=1D-2
       RO(3,2)=7D-3
```

```
      RO(3,1)=9D-3
      RO(1,3)=2D-3
C ********** BUFFER **********
      BF(1,2)=100.D0
      BF(2,1)=100.D0
      BF(2,3)=100.D0
      BF(3,2)=100.D0
      BF(3,1)=100.D0
      BF(1,3)=100.D0
      Q(1,2)=50.D0
      Q(2,1)=50.D0
      Q(2,3)=50.D0
      Q(3,2)=50.D0
      Q(3,1)=50.D0
      Q(1,3)=50.D0
      S=1
      DO 70 I=1,3
      DO 70 J=1,3
      IF(I-J)69,70,69
   69 QT(I,J,S)=Q(I,J)
      QLT(I,J,S)=0.D0
      QQ(I,J,S)=Q(I,J)
   70 CONTINUE
       DO 221 S=1,K
       READ(5,1100) ER(1,2,S),ER(2,1,S),ER(2,3,S),ER(3,2,S),
     5                   ER(3,1,S),ER(1,3,S)
  221 CONTINUE
C*******************************
C********** MEAN ARRIVAL **********
      DO 76 I=1,3
      DO 76 J=1,3
      IF(I .EQ. J) GO TO 76
      WRITE(NO,77) I,J,ER(I,J,1)
   77 FORMAT(3X,'ER(',I1,',',I1,',1)=',F10.4)
   76 CONTINUE
C*********************C
C************************ *****
C************************ *****
      TARR = 0.D0
C****************************
      RRO(1 )  =  RO(1,2)
      RRO(2 )  =  RO(2,1)
      RRO(3 )  =  RO(2,3)
      RRO(5 )  =  RO(3,1)
      RRO(6 )  =  RO(1,3)
      RRO(7 )  =  RO(1,2)
      RRO(8 )  =  RO(2,1)
      RRO(9 )  =  RO(2,3)
      RRO(10)  =  RO(3,2)
      RRO(11)  =  RO(3,1)
      RRO(12)  =  RO(1,3)
C********** CONSTRAINT MATRIX **********
      DO 110 I=1,18
      DO 110 J=1,12
      A(I,J)=0.D0
```

```
 110   CONTINUE
       DO 111 I=1,6
       A(I,I)=1.D0
       A(I,I+6)=1.D0
       A(I+12,I)=1.D0
 111   CONTINUE
       A(7,11    )= 1.D0
       A(8,10    )= 1.D0
       A(9,7     )= 1.D0
       A(10,12   )= 1.D0
       A(11,9    )= 1.D0
       A(12,8    )= 1.D0
C**************************
       A(9+6,7   )=-1.D0
       A(12+6,7  )=-A(9+6,7)
       A(9+6,8   )=1.D0
       A(12+6,8  )=-A(9+6,8)
       A(8+6,9   )=1.D0
       A(11+6,9  )=-A(8+6,9)
       A(8+6,10  )=-1.D0
       A(11+6,10)=-A(8+6,10)
       A(7+6,11  )=-1.D0
       A(10+6,11)=-A(7+6,11)
       A(7+6,12  )=1.D0
       A(10+6,12)=-A(7+6,12)
       DO 113 I=1,18
       WRITE(6,114) (A(I,J),J=1,12)
 114   FORMAT(3X,12F6.2)
 113   CONTINUE
C********** CONSTRAINT B VECTOR **********
       B(1)=CP(1,2)
       B(2)=CP(2,1)
       B(3)=CP(2,3)
       B(4)=CP(3,2)
       B(5)=CP(3,1)
       B(6)=CP(1,3)
C*********************************************
C*****************************
C     SET INITIAL DSEED
C*****************************
       DO 117 I=1,3
       DO 117 J=1,3
       IF(I.EQ.J) GO TO 117
       DI=I
       DJ=J
       DS(I,J)=123457.D0 + 54321.D0*DI + 98765.D0*DJ
  117  CONTINUE
       DO 118 I=1,12
       DI=I
       TDSE(I)=223457.D0 + 654321.D0*DI
  118  CONTINUE
C****************************************
C     GENERATE THE EXTERNAL INPUT
C****************************************
       DO 115 I=1,3
```

```
      DO 115 J=1,3
      IF(I .EQ. J) GO TO 115
      DSEED = DS(I,J)
      DO 116 S=2,61
      RLAM = ER(I,J,S)
      IF(RLAM .LE. 0.0) GO TO 161
      CALL GGPOS(RLAM,DSEED,NR,G,IER)
      RS(I,J,S)=G(1)
      GO TO 116
  161 RS(I,J,S)=0.0
  116 CONTINUE
  115 CONTINUE
      DO 119 I=1,3
      DO 119 J=1,3
      RS(I,J,1)=ER(I,J,1)
  119 CONTINUE
C***********************************************
      S=2
C
  300 ISIZE=90
      NMRUNS=1
   88 CONTINUE
C***
C***
      IOUT2 = 1
      IOUT3=1
      IPACK=0
C     WRITE(NO,13)
C     WRITE(NO,15) IOUT2,IOUT3
C***
C     SOLMIN UPPER BOUND ON OBJ. FUNCTION FOR INTEGER SOLUTION
C   PCTTOL=INPUT TOLERANCE AS FRACTION OF OBJECTIVE FUNCT. FOR
      SOLMIN=-1E5
      PCTTOL=0.0
C***
      M=19
      N=13
      NZR1VR=12
C     WRITE(NO,20) M, N, NZR1VR
      NM1=N-1
   74 IF(SOLMIN)786,787,786
C***
C     INPUT UPPER BOUND ON OBJECTIVE FUNCTION
  786 TLRNCE=SOLMIN
      PCTTOL=-1.
      GO TO 90
  787 ITOL=1
      SOLMIN = 1E35
      IF(PCTTOL)90,788,90
  788 PCTTOL=.1
C***
   90 DO 91 I=1,NM1
   91 UPBND(I)=50.D0
C     WRITE(NO,14)
C     WRITE(NO,10) (UPBND(I), I = 1,NM1)
```

```
      IROW(1)=0
      DO 93 I=2,M
 93   IROW(I)=-1
C     WRITE (NO, 21)
C     WRITE (NO, 15) (IROW(I), I = 2, M)
C**    MATRIX FORMAT: PACKED  = 1,  UNPACKED = 0.
C     WRITE(NO,17)
C     WRITE(NO,15) IPACK
C**********************************************
C*************************************************
C**** SET ALP ****
      DO 100 I=1,3
      DO 100 J=1,3
      ABF=1.0
      P=BF(I, J)-Q(I,J)-ER(I,J,S)
      IF(P.GE. 0.D0) GO TO 101
      ALP(I,J)=(-P/BF(I,J))*ABF
      OALP(I,J)=0.D0
      GO TO 100
 101  ALP(I,J)=0.D0
      OALP(I,J)=1.D0
 100  CONTINUE
C*****************
      DO 120 I=1,3
      DO 120 J=1,3
      BQ(I,J)=BF(I,J)-Q(I,J)
      IF (ER(I,J,S).GT.BQ(I,J)) GO TO 130
      EER(I,J)=ER(I,J,S)
      GO TO 131
 130  EER(I,J)=BQ(I,J)
 131  CONTINUE
C**** CONSTRAINT OF INTERNAL TRAFFIC **************************
      BQR(I,J)=(BF(I,J)-Q(I,J)-EER(I,J))*OALP(I,J)
CBQR  BQR(I,J)=(BF(I,J)-Q(I,J)            )
      QR(I,J)=Q(I,J)+ER(I,J,S)
 120  CONTINUE
      B(7)= BQR(1,2)
      B(8)= BQR(2,1)
      B(9)= BQR(2,3)
      B(10)=BQR(3,2)
      B(11)=BQR(3,1)
      B(12)=BQR(1,3)
      B(13)=QR(1,2)
      B(14)=QR(2,1)
      B(15)=QR(2,3)
      B(16)=QR(3,2)
      B(17)=QR(3,1)
      B(18)=QR(1,3)
C     DO 122 I=1,18
C     WRITE(6,121) I,B(I)
C121  FORMAT(10X,'B(',I2,') = ',F10.4)
C122  CONTINUE
C**********TEST ALP = 0.D0 ***************
C     DO 333 I=1,3
C     DO 333 J=1,3
```

```
C      RO(I,J)=0.D0
C      ALP(I,J)=0.D0
C333   CONTINUE
C**********************************
C********** OBJECT FUNT. **********
       C(1)=(-ALP(1,2)  - 1.D0)*(1.D0-RO(1,2))
       C(2)=(-ALP(2,1)  - 1.D0)*(1.D0-RO(2,1))
       C(3)=(-ALP(2,3)  - 1.D0)*(1.D0  -RO(2,3))
       C(4)=(-ALP(3,2)  - 1.D0)*(1.D0  -RO(3,2))
       C(5)=(-ALP(3,1)  - 1.D0)*(1.D0  -RO(3,1))
       C(6)=(-ALP(1,3)  - 1.D0)*(1.D0  -RO(1,3))
       C(7)=(  +ALP(2,3)  -  ALP(1,3))*(1.D0  -RO(1,2))
       C(8)=(  +ALP(1;3)  -  ALP(2,3))*(1.D0  -RO(2,1))
       C(9)=(  +ALP(3,1)  -  ALP(2,1))*(1.D0  -RO(2,3))
       C(10)=(+ALP(2,1)  -  ALP(3,1))*(1.D0  -RO(3,2))
       C(11)=(+ALP(1,2)  -  ALP(3,2))*(1.D0  -RO(3,1))
       C(12)=(+ALP(3,2)  -  ALP(1,2))*(1.D0  -RO(1,3))
C      DO 124 I=1,12
C      WRITE(6,123) I,C(I)
C123   FORMAT(10X,'C(',I2,') = ',F10.4)
C124   CONTINUE
C***************************************************
       DO 55 I=1,M
       DO 55 J=1,N
       ATAB(I,J)=0.D0
  55   CONTINUE
       DO 71 I=2,M
       ATAB(I,1)  =  B(I[illegible]1)
  71   CONTINUE
       DO 72 J=2,N
       ATAB(1,J)=   C(J-1)
  72   CONTINUE
       M1=M-1
       N1=N-1
       DO 73 I=1,M1
       DO 73 J=1,N1
       ATAB(I+1,J+1)  =  A(I,J)
  73   CONTINUE
C********************************************************
       IF ( M .LT. 2) GO TO 450
C***
C      PRINT INPUT TABLEAU FOR ERROR CHECK
C9510  WRITE(NO,22)
C      DO 80 I = 1, M
C      WRITE (NO, 23) (ATAB(I,J), J = 1, N)
C  80  CONTINUE
 9520  DO 954 I=2,M
       IF(IROW(I))953,9521,9521
 9521  DO 9523 J=2,N
 9523  ATAB(I,J)=-ATAB(I,J)
       GO TO 954
  953  ATAB(I,1)=-ATAB(I,1)
  954  CONTINUE
  450  CONTINUE
  955  DO 98 I=2,N
```

```
      IF(UPBND(I-1))96,96,98
   96 UPBND (I-1) = 1D3
   98 CONTINUE
C***********************************************
      CALL INTMIN(M,N,NZR1VR)
C***********************************************
C     WRITE(NO,35) ZOPT,INCTR
C     WRITE(NO,19)
C     WRITE(NO,18) (I, I=1,NM1)
C     WRITE(NO,10) (T(I), I=1,NM1)
      DO 701 I=1,3
      DO 701 J=1,3
      IF(I.EQ.J) GO TO 701
      WRITE(6,700) I,J,S,RS(I,J,S)
 700  FORMAT(3X,'RS(',I1,',',I1,',',I2,') =',F8.2)
      R(I,J)=RS(I,J,S)
      TARR = TARR + R(I,J)
 701  CONTINUE
C********** GENERATE CHANNEL ERROR ***********
      DO 500 I=1,12
      DSEED = TDSE(I)
      TDE(I)= TDSE(I)
      RLAM=RRO(I)*T(I)
      IF(RLAM) 557,557,555
 555  CALL GGPOS(RLAM,DSEED,NR,G,IER)
      TE(I)=G(1)
      TDSE(I) = DSEED
      GO TO 556
  557 TE(I)=0.D0
 556  T(I) = T(I) - TE(I)
  500 CONTINUE
      WRITE(NO,558) (TE(I),I=1,12)
 558  FORMAT(12F6.3)
C*********************************************
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C    CRP ; QRP= Q+R+T
C    COT ; T(1),T(2) OUT GOING TRAFFIC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IS=0
      ISS=50
  501 IS=IS+1
      IF(ISS-IS) 503,503,505
  503 WRITE(NO,504)
  504 FORMAT(1H0,22HEND OF THE COUNTER ISS)
      GO TO 3000
  505 QRP=Q(1,2)+R(1,2)+T(11)
CQRP  WRITE(NO,599) QRP
 599  FORMAT(10X,'QRP=',D23.16)
COT   WRITE(NO,598) T(1),T(12)
  598 FORMAT(3X,'T(1),T(12)=',D23.16,5X,D23.16)
      TQ=T(1)+T(12)
      QRP=QRP+0.00001D0
      IF(TQ.LE.QRP) GO TO 181
      QRP=QRP-0.00001D0
CT    T( 1)=DMIN1(T( 1),QRP)
```

```
CT    T(12)=QRP-T( 1)
C12****** PRODUCING ACTUAL ROUTING *************
      IF( (T(1).GT.0.D0).AND.(T(12).GT.0.D0)) GO TO 441
      IF( T(1). GT. 0.D0) GO TO 442
      T(12)=QRP
      GO TO 402
  442 T(1)=QRP
      GO TO 402
  441 CONTINUE
      T(1)=QRP * (T(1)/(T(1)+T(12)))
      T(12) = QRP - T(1)
      IT1=T(1)
      IT2=T(12)
      IT10 = T(1)*10.D0
      IF ( (IT10-IT1*10) .GE. 5 ) GO TO 401
      T(1)=IT1
      T(12)=IT2 + 1
      GO TO 402
  401 T(1)=IT1 + 1
      T(12)=IT2
  402 CONTINUE
C*********************************************
C*********************************************
      IX=1
      IY=12
COT   WRITE(NO,559) IX,T(IX),IY,T(IY)
 559  FORMAT(3X,'T(',I2,')=',D23.16,5X,'T(',I2,')=',D23.16)
      GO TO 502
 181  QRP=Q(2,1)+R(2,1)+T(10)
CQRP  WRITE(NO,599) QRP
COT   WRITE(NO,431) T(2),T(9)
  431 FORMAT(3X,'T(2),T(9)=',D23.16,5X,D23.16)
      TQ=T(2)+T(9)
      QRP=QRP+0.00001D0
      IF(TQ.LE.QRP) GO TO 182
      QRP=QRP-0.00001D0
CT    T( 2)=DMIN1(T( 2),QRP)
CT    T( 9)=QRP-T( 2)
C9******* PRODUCING ACTUAL ROUTING *************
      IF( (T(2).GT.0.D0).AND.(T(9).GT.0.D0)) GO TO 443
      IF( T(2). GT. 0.D0) GO TO 444
      T(9)=QRP
      GO TO 404
  444 T(2)=QRP
      GO TO 404
  443 CONTINUE
      T(2)=QRP * (T(2)/(T(2)+T(9)))
      T(9) = QRP - T(2)
      IT1=T(2)
      IT2=T(9)
      IT10 = T(2)*10.D0
      IF ( (IT10-IT1*10) .GE. 5 ) GO TO 403
      T(2)=IT1
      T(9)=IT2 + 1
      GO TO 404
```

```
  403 T(2)=IT1 + 1
      T(9)=IT2
  404 CONTINUE
C*********************************************
      IX=2
      IY=9
COT   WRITE(NO,559) IX,T(IX),IY,T(IY)
      GO TO 502
 182  QRP=Q(2,3)+R(2,3)+T( 7)
CQRP  WRITE(NO,599) QRP
COT   WRITE(NO,432) T(3),T(8)
  432 FORMAT(3X,'T(3),T(8)=',D23.16,5X,D23.16)
      TQ=T(3)+T(8)
      QRP=QRP+0.00001D0
      IF(TQ.LE.QRP) GO TO 183
      QRP=QRP-0.00001D0
CT    T( 3)=DMIN1(T( 3),QRP)
CT    T( 8)=QRP-T( 3)
C8****** PRODUCING ACTUAL ROUTING **************
      IF((T(3).GT.0.D0).AND.(T(8).GT.0.D0)) GO TO 445
      IF( T(3). GT. 0.D0) GO TO 446
      T(8)=QRP
      GO TO 406
  446 T(3)=QRP
      GO TO 406
  445 CONTINUE
      T(3)=QRP * (T(3)/(T(3)+T(8)))
      T(8) = QRP - T(3)
      IT1=T(3)
      IT2=T(8)
      IT10 = T(3)*10.D0
      IF ( (IT10-IT1*10) .GE. 5 ) GO TO 405
      T(3)=IT1
      T(8)=IT2 + 1
      GO TO 406
  405 T(3)=IT1 + 1
      T(8)=IT2
  406 CONTINUE
C*********************************************
      IX=3
      IY=8
COT   WRITE(NO,559) IX,T(IX),IY,T(IY)
      GO TO 502
 183  QRP=Q(3,2)+R(3,2)+T(12)
CQRP  WRITE(NO,599) QRP
COT   WRITE(NO,597) T(4),T(11)
  597 FORMAT(3X,'T(4),T(11)=',D23.16,5X,D23.16)
      TQ=T(4)+T(11)
      QRP=QRP+0.00001D0
      IF(TQ.LE.QRP) GO TO 184
      QRP=QRP-0.00001D0
CT    T( 4)=DMIN1(T( 4),QRP)
CT    T(11)=QRP-T( 4)
C11****** PRODUCING ACTUAL ROUTING **************
      IF((T(4).GT.0.D0).AND.(T(11).GT.0.D0)) GO TO 447
```

```
      IF( T(4). GT. 0.D0) GO TO 448
      T(11)=QRP
      GO TO 408
  448 T(4)=QRP
      GO TO 408
  447 CONTINUE
      T(4)=QRP * (T(4)/(T(4)+T(11)))
      T(11) = QRP - T(4)
      IT1=T(4)
      IT2=T(11)
      IT10 = T(4)*10.D0
      IF ( (IT10-IT1*10) .GE. 5 ) GO TO 407
      T(4)=IT1
      T(11)=IT2 + 1
      GO TO 408
  407 T(4)=IT1 + 1
      T(11)=IT2
  408 CONTINUE
C***************************************
      IX=4
      IY=11
COT   WRITE(NO,559) IX,T(IX),IY,T(IY)
      GO TO 502
 184  QRP=Q(3,1)+R(3,1)+T( 9)
CQRP  WRITE(NO,599) QRP
COT   WRITE(NO,433) T(5),T(10)
  433 FORMAT(3X,'T(5),T(10)=',D23.16,5X,D23.16)
      TQ=T(5)+T(10)
      QRP=QRP+0.00001D0
      IF(TQ.LE.QRP) GO TO 185
      QRP=QRP-0.00001D0
CT    T( 5)=DMIN1(T( 5),QRP)
CT    T(10)=QRP-T( 5)
C10****** PRODUCING ACTUAL ROUTING *************
      IF((T(5).GT.0.D0).AND.(T(10).GT.0.D0)) GO TO 449
      IF( T(5). GT. 0.D0) GO TO 453
      T(10)=QRP
      GO TO 410
  453 T(5)=QRP
      GO TO 410
  449 CONTINUE
      T(5)=QRP * (T(5)/(T(5)+T(10)))
      T(10) = QRP - T(5)
      IT1=T(5)
      IT2=T(10)
      IT10 = T(5)*10.D0
      IF ( (IT10-IT1*10). .GE. 5 ) GO TO 409
      T(5)=IT1
      T(10)=IT2 + 1
      GO TO 410
  409 T(5)=IT1 + 1
      T(10)=IT2
  410 CONTINUE
C***************************************
      IX=5
```

```
      IY=10
COT   WRITE(NO,559) IX,T(IX),IY,T(IY)
      GO TO 502
 185  QRP=Q(1,3)+R(1,3)+T( 8)
CQRP  WRITE(NO,599) QRP
COT   WRITE(NO,434) T(6),T(7)
  434 FORMAT(3X,'T(6),T(7)=',D23.16,5X,D23.16)
      TQ=T(6)+T(7)
      QRP=QRP+0.00001D0
      IF(TQ.LE.QRP) GO TO 186
      QRP=QRP-0.00001D0
CT    T( 6)=DMIN1(T( 6),QRP)
CT    T( 7)=QRP-T( 6)
C7****** PRODUCING ACTUAL ROUTING **************
      IF((T(6).GT.0.D0).AND.(T(7).GT.0.D0)) GO TO 451
      IF( T(6). GT. 0.D0) GO TO 452
      T(7)=QRP
      GO TO 412
  452 T(6)=QRP
      GO TO 412
  451 CONTINUE
      T(6)=QRP * (T(6)/(T(6)+T(7)))
      T(7) = QRP - T(6)
      IT1=T(6)
      IT2=T(7)
      IT10 = T(6)*10.D0
      IF ( (IT10-IT1*10) .GE. 5 ) GO TO 411
      T(6)=IT1
      T(7)=IT2 + 1
      GO TO 412
  411 T(6)=IT1 + 1
      T(7)=IT2
  412 CONTINUE
C************************************************
      IX=6
      IY=7
COT   WRITE(NO,559) IX,T(IX),IY,T(IY)
C*******************************************
  502 RLAM=T(IX)*RRO(IX)
      IF(RLAM) 567,567,568
  568 DSEED = TDE(IX)
      CALL GGPOS(RLAM,DSEED,NR,G,IER)
      TE(IX)=G(1)
      GO TO 569
  567 TE(IX)=0.D0
  569 T(IX)=T(IX)-TE(IX)
      RLAM=T(IY)*RRO(IY)
      IF(RLAM) 571,571,572
  572 DSEED = TDE(IY)
      CALL GGPOS(RLAM,DSEED,NR,G,IER)
      TE(IY)=G(1)
      GO TO 573
C********************************************
  571 TE(IY)=0.D0
  573 T(IY)=T(IY)-TE(IY)
```

```
      ITX=TE(IX)
      ITY=TE(IY)
      WRITE(NO,15) IX,ITX ,IY,ITY
      GO TO 501
 186  CONTINUE
      WRITE(NO,187) IS
  187 FORMAT(6H0 IS  =,I4)
      N122=T(1)
      N211=T(2)
      N233=T(3)
      N322=T(4)
      N311=T(5)
      N133=T(6)
      N123=T(7)
      N213=T(8)
      N231=T(9)
      N321=T(10)
      N312=T(11)
      N132=T(12)
      Q(1,2)=Q(1,2) + R(1,2) - N122 - N132 + N312
      Q(2,1)=Q(2,1) + R(2,1) - N211 - N231 + N321
      Q(2,3)=Q(2,3) + R(2,3) - N233 - N213 + N123
      Q(3,2)=Q(3,2) + R(3,2) - N322 - N312 + N132
      Q(3,1)=Q(3,1) + R(3,1) - N311 - N321 + N231
      Q(1,3)=Q(1,3) + R(1,3) - N133 - N123 + N213
 1000 FORMAT(10X,F15.4)
      WRITE(6,1001) N122
 1001 FORMAT(3X,'N122 =',I5)
      WRITE(6,1002) N211
 1002 FORMAT(3X,'N211 =',I5)
      WRITE(6,1003) N233
 1003 FORMAT(3X,'N233 =',I5)
      WRITE(6,1004) N322
 1004 FORMAT(3X,'N322 =',I5)
      WRITE(6,1005) N311
 1005 FORMAT(3X,'N311 =',I5)
      WRITE(6,1006) N133
 1006 FORMAT(3X,'N133 =',I5)
      WRITE(6,1007) N123
 1007 FORMAT(3X,'N123 =',I5)
      WRITE(6,1008) N213
 1008 FORMAT(3X,'N213 =',I5)
      WRITE(6,1009) N231
 1009 FORMAT(3X,'N231 =',I5)
      WRITE(6,1010) N321
 1010 FORMAT(3X,'N321 =',I5)
      WRITE(6,1011) N312
 1011 FORMAT(3X,'N312 =',I5)
      WRITE(6,1012) N132
 1012 FORMAT(3X,'N132 =',I5)
 139  DO 143 I=1,3
      DO 143 J=1,3
      IF(I.EQ.J) GO TO 143
      IF(Q(I,J).GT.BF(I,J)) GO TO 140
      QL(I,J)=0.D0
```

```
      GO TO 141
 140  QL(I,J) = Q(I,J) - BF(I,J)
 141  WRITE(6,142) I,J,QL(I,J)
 142  FORMAT(3X,'QL(',I1,',',I1,') =',F8.2)
      Q(I,J)=DMAX1(0.0D0,Q(I,J))
      QQ(I,J,S)=Q(I,J)
      Q(I,J)=Q(I,J)-QL(I,J)
      WRITE(6,170) I,J,Q(I,J)
 170  FORMAT(10X,'Q(',I1,',',I1,') =',F8.2)
      QLOST=QLOST + QL(I,J)
 143  CONTINUE
      DO 210 I=1,3
      DO 210 J=1,3
      QT(I,J,S)=Q(I,J)
      QLT(I,J,S)=QL(I,J)
      QQQ = QQQ + QT(I,J,S)
  210 CONTINUE
      S=S+1
      IF(S.LE.K) GO TO 300
C*************************
       DO 201 S=1,K
       WRITE(6,1100) QT(1,2,S),QT(2,1,S),QT(2,3,S),QT(3,2,S),
     $                QT(3,1,S),QT(1,3,S)
 1100 FORMAT(3X,6F10.4)
 201  CONTINUE
      DO 202 S=1,K
      WRITE(6,1101) QLT(1,2,S),QLT(2,1,S),QLT(2,3,S),QLT(3,2,S),
     $               QLT(3,1,S),QLT(1,3,S)
 1101 FORMAT(6X,6F10.2)
 202  CONTINUE
      DO 203 S=1,K
      WRITE(6,1102) QQ(1,2,S),QQ(2,1,S),QQ(2,3,S),QQ(3,2,S),
     $               QQ(3,1,S),QQ(1,3,S)
 1102 FORMAT(3X,6F10.4)
 203  CONTINUE
      WRITE(6,490) TARR
 490  FORMAT(10X,'TOTAL ARRIVAL =',F10.3)
      WRITE(6,150) QLOST
 150  FORMAT(10X,'QLOST = ',F15.5)
      QIL=0.D0
      DO 1104 I=1,3
      DO 1104 J=1,3
      IF(I .EQ. J) GO TO 1104
      QIL = QIL + QT(I,J,1) - QT(I,J,K)
 1104 CONTINUE
      WRITE(NO,1105) QQQ
 1105 FORMAT(3X,'SUM OF QUEUES=',F10.3)
      AVD = QQQ / (TARR-QLOST+QIL)
      WRITE(NO,1106) AVD
 1106 FORMAT(3X,'AVG.DELAY =',F10.5)
      THR=(QIL+TARR-QLOST)/60.D0
      WRITE(NO,1110) THR
 1110 FORMAT(3X,'THR. =',F10.5)
C**********************************************
        DO 204 S=1,K
```

```
        WRITE(6,1100) RS(1,2,S),RS(2,1,S),RS(2,3,S),RS(3,2,S),
     $                RS(3,1,S),RS(1,3,S)
  204 CONTINUE
      WRITE(6,1107)
 1107 FORMAT(3X,'MEAN ARRIVAL AT EACH BUFFER')
        DO 2J5  S=1,K
        WRITE(6,1100) ER(1,2,S),ER(2,1,S),ER(2,3,S),ER(3,2,S),
     $                ER(3,1,S),ER(1,3,S)
  205 CONTINUE
C********************************
C     WRITE(NO,1108)
C1108 FORMAT(3X,'MEAN ARRIVAL RATE')
C     DO 227 I=1,3
C     DO 227 J=1,3
C     IF(I.EQ.J) GO TO 227
C     DO 227 S=1,60
C     S1=S+1
C     AR(I,J,S)=ER(I,J,S1)-ER(I,J,S)
C 227 CONTINUE
C     DO 228 I=1,3
C     DO 228 J=1,3
C     IF(I.EQ.J) GO TO 228
C     AR(I,J,61)=AR(I,J,60)
C 228 CONTINUE
C     DO 229 S=1,61
C       WRITE(6,1100) AR(1,2,S),AR(2,1,S),AR(2,3,S),AR(3,2,S),
C    $                AR(3,1,S),AR(1,3,S)
C 229 CONTINUE
C***************************************
 3000 STOP
      END
C*****************************************************************
C#################################################################
C******************************************************T*************
      SUBROUTINE INTMIN(M,N,NZR1VR)
      DOUBLE PRECISION DABS
      DOUBLE PRECISION ATAB(21,15), UPBND(15), TPVAL(15), BTMVL(15),
     1VAL(15), TBSAV(21,15), SAVTAB(21,100 ), T(15)
      DOUBLE PRECISION SOLMIN, PCTTCL, TLRNCE, YVECT, ATAB11, AMAX,
     1RTIO, ALFA, ARTIO, ADELT, ZOPT, ATAB12, X1, AMAX2, AMAX3, ALW,
     2AUP, RTIO2, DIFF1, DIFF2, DIFF, SVALW, ANDCT4
      DIMENSION IROW(21), ITBROW (21)
      DIMENSION ICOL (15), ITBCOL (15), IVAR (15)
      DIMENSION            ISVROW (21,15)
      DIMENSION ISVRCL (15), ICORR (15), ISVN (15)
      DIMENSION KSVN (15)
C     ARRAY ITEMP USED FOR PACKED FORMAT DATA INPUT ONLY
      DIMENSION ITEMP (7)
      COMMON /IO/ NI,NO,NM1,IROW,IPACK,ITOL
      COMMON /IYS/ ISIZE,NMRUNS,IOUT2,IOUT3,INCTR
      COMMON /SP/ SOLMIN,PCTTOL
      COMMOM /CONS/ UPBND,ATAB,T,ZOPT,VAL,TLRNCE
      X1 = 1.0
   10 FORMAT (1H0, (7D10.3))
C     UNPACKED FORMAT NO. 11
```

```
   11 FORMAT ( 7D10.0     )
   12 FORMAT ( 1X, 8D13.7)
   13 FORMAT (1H0,24HPRINT CONTROL PARAMETERS)
   14 FORMAT (1H0,30HUPPER BOUND ON VARIABLE 1 TO N)
   15 FORMAT (20I4)
C     PACKED FORMAT NO. 16
   16 FORMAT ( 7(I3,D7.0))
   17 FORMAT (1H0,18HMATRIX FORMAT CODE)
   18 FORMAT (4H0I =, I4, 6I10)
   19 FORMAT (27H0STRUCTURAL VARIABLES: X(I))
   20 FORMAT (44H0ROWS X COLUMNS AND NO. OF INTEGER VARIABLES,//,
     1      I4,2H X, I3, 24X, I6)
   21 FORMAT (30H0CONSTRAINT TYPES IN ROW ORDER)
   22 FORMAT (51H0INPUT TABLEAU ECHO, CONSTRAINT VALUE LEFT. BY ROW.)
   23 FORMAT (1H0,10D13.3/(1H , 10D13.3))
   24 FORMAT (1H0,13HITERATION NO.,I6)
   25 FORMAT ( 1H0,8D13.5/(1H , 8D13.5))
   26 FORMAT ( 1H , I6, 7I13)
   27 FORMAT(/, 114X,I5)
   29 FORMAT(    'TOLERANCE SET AT',E15.7,' AT ITERATION',I6)
   30 FORMAT(21H PROBLEM NOT FEASIBLE)
   35 FORMAT (21H0OBJECTIVE FUNCTION =, F15.7,14H  AT ITERATION,I6)
   40 FORMAT (29H0CONTINUOUS SOLUTION COMPLETE)
   42 FORMAT (38H0FINAL TABLEAU FOR CONTINUOUS SOLUTION)
   45 FORMAT(40H0CONTINUOUS SOLUTION IS INTEGER SOLUTION)
   46 FORMAT (1H0,30HNO INTEGER VARIABLES REQUESTED)
   50 FORMAT (23H0OPTIMALITY ESTABLISTED)
   55 FORMAT(33H0PROBLEM TOO BIG FOR MACHINE SIZE)
   60 FORMAT(33H0                                )
   65 FORMAT (30H0END OF PROBLEM, ITERATION NO., I6)
   70 FORMAT('BRANCH POINT INCREASED TO',I4)
   75 FORMAT('BRANCH POINT DECREASED TO',I4)
   78 FORMAT (24H0INITIAL WORKING TABLEAU)
      INDCT7=1
      KSVN(1)=1
      INDCTR=1
      ICNTR=0
      IOUT1 = 0
      I1ROW=1000
      ADELT = 5.0E-7
   73 DO 72 I=1,N
   72 T(I)=0.
C***
C***
C     COMPUTE NO. OF Y VECTORS
  981 YVECT=UPBND(1)+1.
      IF ( NZR1VR .LT. 2) GO TO 322
      DO 982 I=2,NZR1VR
  982 YVECT=YVECT*(UPBND(I)+1.)
  322 CONTINUE
C***
C     SET SOLUTION VECTOR OF VARIABLES EQUAL TO ZERO
C     AND SAVE ORIGINAL UPPER BOUNDS
  985 DO 99 I=2,N
   99 IVAR(I-1)=0
```

```
C***
C     INITIALIZE ROW AND COLUMN IDENTIFIERS,+K=VARIABLE NO. K,
       IF ( M .LT. 2) GO TO 451
       DO 102 I=2,M
       IF(IROW(I))100,102,100
  100  IROW(I)=1-I
  102  CONTINUE
  451  CONTINUE
       ATAB11=ATAB(1,1)
       ICOL(1) = 0
       DO 103 J=2,N
       IF(ATAB(1,J))1022,1025,1025
 1022  DO 1023 I=1,M
       ATAB(I,1)=ATAB(I,1)+ATAB(I,J)*UPBND(J-1)
 1023  ATAB(I,J)=-ATAB(I,J)
       ICOL(J)=1000+J-1
       GO TO 103
 1025  ICOL(J)=J-1
  103  CONTINUE
C***
C      OUTPUT INITIAL TABLEAU
C      IF(IOUT2)104,254,104
C 104  WRITE(NO,78)
C      WRITE(NO,26)(ICOL(J),J=1,N)
C      DO 110 I=1,M
C      WRITE(NO,25)(ATAB(I,J),J=1,N)
C 110  WRITE(NO,27)IROW(I)
       GO TO 254
C***
C      CHOOSE PIVOT ROW, MAXIMUM POSITIVE VALUE IN CONSTANT COLUMN
  112  AMAX = 0.0
       IF ( M .LT. 2) GO TO 452
       DO 120 I=2,M
       IF(ATAB(I,1))120,120,115
  115  IF(ATAB(I,1)-AMAX)120,120,117
  117  AMAX=ATAB(I,1)
       IPVR=I
  120  CONTINUE
  452  CONTINUE
C***
C      IF NO POSITIVE VALUE, LP FINISHED (PRIMAL FEASIBLE)
       IF(AMAX)265,265,130
C      CHOOSE PIVOT COLUMN, ALGEBRAICALLY MAXIMUM RATIO
C      A(PIVOTROW) FOR A (PIVOTROW,J) NEGATIVE. IF NO NEGATIVE
C      A(PIVOTROW,J) PROBLEM INFEASIBLE
  130  AMAX = -1E35
       IF(N-2)143,132,132
  132  IPVC=0
       DO 140 J=2,N
       IF(ATAB(IPVR,J))133,140,140
  133  RTIO=ATAB(1,J)/ATAB(IPVR,J)
       IF(RTIO-AMAX)140,137,135
  135  AMAX=RTIO
  136  IPVC=J
       GO TO 140
```

```
  137 IF(ATAB(IPVR,J)-ATAB(IPVR,IPVC))136,140,140
  140 CONTINUE
      IF(IPVC)150,143,150
  143 GO TO (145,435,542,610,665),INDCTR
  145 WRITE(NO,30)
      GO TO 1001
C***
C     CARRY OUT PIVOT STEP
  150 ALFA=ATAB(IPVR,IPVC)
C**    UPDATE TABLEAU
      DO 160 J=1,N
      IF(ATAB(IPVR,J))152,160,152
  152 IF(J-IPVC)153,160,153
  153 ARTIO=ATAB(IPVR,J)/ALFA
      DO 175 I=1,M
      IF(ATAB(I,IPVC))157,175,157
  157 IF(I-IPVR)160,175,160
  160 ATAB(I,J)=ATAB(I,J)-ARTIO*ATAB(I,IPVC)
      IF(DABS(ATAB(I,J))-ADELT)165,165,175
  165 ATAB(I,J) = 0.0
  175 CONTINUE
  160 CONTINUE
      DO 190 J=1,N
  190 ATAB(IPVR,J)=ATAB(IPVR,J)/ALFA
C***
C     EXCHANGE ROW AND COLUMN IDENTIFIERS
      ISV=IROW(IPVR)
      IROW(IPVR)=ICOL(IPVC)
      IF(ISV)197,195,197
C***
C     IF PIVOT ROW WAS ZERO SLACK, SET MODIFIED PIVOT COLUMN ZERO.
  195 DO 196 I=1,M
  196 ATAB(I,IPVC)=ATAB(I,N)
      ICOL(IPVC)=ICOL(N)
      N=N-1
      GO TO 200
  197 DO 198 I=1,M
  198 ATAB(I,IPVC)=-ATAB(I,IPVC)/ALFA
      ICOL(IPVC)=ISV
      ATAB(IPVR,IPVC)=1./ALFA
C***
C     COUNT PIVOTS
  200 ICNTR=ICNTR+1
      IF(IROW(IPVR)+1000)210,205,210
  205 DO 207 J=1,N
  207 ATAB(IPVR,J)=ATAB(M,J)
      IROW(IPVR)=IROW(M)
      M=M-1
  210 IF(IOUT1)240,2505,240
C***
C     OUTPUT CURRENT TABLEAU
  240 WRITE (NO,24) ICNTR
      WRITE(NO,26)(ICOL(J),J=1,N)
      DO 250 K=1,M
      WRITE(NO,25)(ATAB(K,L),L=1,N)
```

```
  250 WRITE(NO,27)IROW(K)
 2505 GO TO (254,251,252,253,2535),INDCTR
C***
C       IF SEEKING INTEGER SOLUTION, TEST OBJECTIVE FUNCTION AGAINST CURRE
  251 IF(ATAB(1,1)-TLRNCE)254,435,435
  252 IF(ATAB(1,1)-TLRNCE)254,542,542
  253 IF(ATAB(1,1)-TLRNCE)254,610,610
 2535 IF(ATAB(1,1)-TLRNCE)254,665,665
C***
C       IF CONSTANT COLUMN OF ZERO SLACK ROW IS NEG., REVERSE SIGNS OF ENT
  254 IF (M .LT. 2) GO TO 453
      DO 260 K=2,M
      IF(IROW(K))260,255,260
  255 IF(ATAB(K,1))256,260,260
  256 DO 258 L=1,N
  258 ATAB(K,L)=-ATAB(K,L)
  260 CONTINUE
  453 CONTINUE
C**     GO TO NEXT PIVOT STEP
      GO TO 112
  265 CONTINUE
C***
C       IF ANY BASIS VARIBLE EXCEEDS
C
      IF ( M .LT. 2) GO TO 454
      DO 275 I=2,M
      IF(IROW(I))275,275,266
  266 J=IROW(I)
      IF(J-1000)268,268,267
  267 J=J-1000
  268 IF(UPBND(J)+ATAB(I,1))269,275,275
  269 IF(ADELT+UPBND(J)+ATAB(I,1))270,274,274
  270 ATAB(I,1)=-ATAB(I,1)-UPBND(J)
      DO 271 K=2,N
  271 ATAB(I,K)=-ATAB(I,K)
      IPVR=I
      IF(J-IROW(I))272,273,272
  272 IROW(I)=J
      GO TO 130
  273 IROW(I)=IROW(I)+1000
      GO TO 130
  274 ATAB(I,1)=-UPBND(J)
  275 CONTINUE
  454 CONTINUE
C***
C       TRUE END OF
C
      IF ( M .LT. 2) GO TO 455
      DO 280 I=2,M
      IF(IROW(I))280,280,277
  277 IF(IROW(I)-1000)279,279,278
  278 J=IROW(I)-1000
      T(J)=UPBND(J)+ATAB(I,1)
      GO TO 280
  279 J=IROW(I)
```

```
      T(J)=-ATAB(I,1)
  280 CONTINUE
  455 CONTINUE
C***
C     SET SOLUTION
      DO 285 I=2,N
      IF(ICOL(I))285,285,282
  282 IF(ICOL(I)-1000)284,284,283
  283 J=ICOL(I)-1000
      T(J)=UPBND(J)
      GO TO 285
  284 J=ICOL(I)
      T(J)=0.
  285 CONTINUE
      GO TO (286,437,548,615,670),INDCTR
C***
C     FIRST TIME
C
  286 WRITE(NO,40)
C     IF(IOUT3)287,291,287
C 287 WRITE(NO,42)
C     WRITE(NO,26)(ICOL(J),J=1,N)
C 288 DO 290 I=1,M
C     WRITE(NO,25)(ATAB(I,J),J=1,N)
C 290 WRITE(NO,27)IROW(I)
  291 ZOPT =DABS( ATAB(1,1))
      WRITE (NO, 35) ZOPT, ICNTR
      WRITE (NO, 19)
      WRITE (NO,18) (I, I = 1, NM1)
      WRITE (NO, 10) (T(I), I=1,NM1)
C***
C   COMPUTE ABSOLUTE TOLERANCE
      ATAB12=ATAB(1,1)
      ATAB11 =DABS (ATAB11 - ATAB(1,1))
      IF(PCTTOL)294,293,292
  292 TLRNCE=PCTTOL*ATAB11+ATAB12
      GO TO 294
  293 TLRNCE = 1E35
  294 CONTINUE
C***
C   DETERMINE WHETHER CONTINUOUS SOLUTION IS MIXED INTEGER SOLUTION
      IF ( M .LT. 2) GO TO 456
  301 DO 310 I=2,M
      IF(IROW(I))310,310,302
  302 IF(IROW(I)-1000)303,303,304
  303 IF(IROW(I)-NZR1VR)305,305,310
  304 IF(IROW(I)-1000-NZR1VR)305,305,310
  305 AJ01 = ATAB(I,1)
      AJ02 = ADELT
      AJ03 = X1
      IF(AMOD(-AJ01,AJ03)-AJ02) 310,310,306
  306 IF(1.0-AMOD(-AJ01,AJ03)-AJ02) 310,310,295
  310 CONTINUE
  456 CONTINUE
      IF ( NZR1VR) 307,308,307
```

```
  307 WRITE (NO,45)
      GO TO 998
  308 WRITE (NO,46)
      GO TO 998
C***
C DETERMINE WHETHER PROBLEM FITS IN MEMORY, AND IF SO WHETHER TO SAVE
C  ALL INTERMEDIATE TABLEAUS OR ONLY SOME
  295 IF(N-NZR1VR)297,297,298
  297 ISVLOC=(N*(N+1))/2
      GO TO 299
  298 ISVLOC=(NZR1VR*(2*N-NZR1VR+1))/2
  299 IF(ISIZE-ISVLOC)3001,3001,300
  300 I1ROW=0
      GO TO 315
 3001 NONBSC=0
      DO 3006 J=2,N
      IF(ICOL(J))3006,3006,3002
 3002 IF(ICOL(J)-1000)3003,3004,3004
 3003 IF(ICOL(J)-NZR1VR)3005,3005,3006
 3004 IF(ICOL(J)-1000-NZR1VR)3005,3005,3006
 3005 NONBSC=NONBSC+1
 3006 CONTINUE
      IF(N-NZR1VR)3007,3007,3008
 3007 ISVLOC=N+((N-NONBSC)*(N-NONBSC+1))/2
      GO TO 3009
 3008 ISVLOC=N+((NZR1VR-NONBSC)*(N-NONBSC+N-NZR1VR+1))/2
 3009 IF(ISIZE-ISVLOC)3010,3010,315
 3010 WRITE(NO,55)
      GO TO 998
  315 CONTINUE
C***
C BEGIN INTEGER PROGRAMMING
  400 I1=1
  402 AMAX = -X1
      KSVN(I1+1)=KSVN(I1)
C***
C
C
      DO 4085 I=2,N
      IF(ICOL(I))4085,4085,405
  405 IF(ICOL(I)-1000)406,407,407
  406 IF(ICOL(I)-NZR1VR)408,408,4085
  407 IF(ICOL(I)-1000-NZR1VR)408,408,4085
  408 IF(AMAX-ATAB(1,I))4082,4085,4085
 4082 ISVI=I
      AMAX=ATAB(1,I)
 4085 CONTINUE
C***
C.
      IF ( AMAX + X1) 4087, 420, 4087
C***
C
 4087 IVAR(I1)=ICOL(ISVI)
      BTMVL(I1)=-1.
      ISVRCL(I1)=ISVI
```

```
      ICORR(I1)=0
      VAL (I1) = 0.0
C***
C
C
      IF(ATAB(1,1)+ATAB(1,ISVI)-TLRNCE)410,409,409
  409 TPVAL(I1)=1000.
      IF(I1-1)4101,4101,4095
 4095 ISVN(I1)=0
      GO TO 4132
  410 TPVAL(I1)=1.
C***
      IF(I1-1)4100,4101,4100
C
C
 4100 IF(I1-I1ROW)4132,4101,4101
 4101 L=KSVN(I1)
      DO 412 J=1,M
      ISVROW(J,I1)=IROW(J)
      DO 411 K=1,N
      I=L+K-1
      IF(J-1)4105,4105,411
 4105 SAVTAB(M+1,I)=ICOL(K)
  411 SAVTAB(J,I)=ATAB(J,K)
  412 CONTINUE
      ISVN(I1)=N
      KSVN(I1+1)=L+N
 4132 ICOL(ISVI)=ICOL(N)
      DO 4135 J=1,M
 4135 ATAB(J,ISVI)=ATAB(J,N)
      N=N-1
      GO TO 5000
C
C
  420 CONTINUE
      IF(I1-I1ROW)4204,600,4205
 4204 I1ROW=I1
 4205 INDCT7=1
  421 AMAX = -X1
      IF ( M .LT. 2) GO TO 457
      DO 425 I2=2,M
      IF(IROW(I2))425,425,422
  422 IF(IROW(I2)-1000)423,424,424
  423 IF(IROW(I2)-NZR1VR)4241,4241,425
  424 IF(IROW(I2)-1000-NZR1VR)4241,4241,425
 4241 AMAX2 = 1.0E35
       AMAX3 = -1.0E35
      AJO = -ATAB(I2,1) + ADELT
      ALW = AINT(AJO)
      AUP=ALW+1.
      IF(N-1)426,426,4240
 4240 DO 4246 I3=2,N
      IF(ATAB(I2,I3))4244,4246,4242
 4242 RTIO=ATAB(1,I3)/ATAB(I2,I3)
      IF(RTIO-AMAX2)4243,4246,4246
```

```
 4243 AMAX2=RTIO
      GO TO 4246
 4244 RTIO2=ATAB(1,I3)/ATAB(I2,I3)
      IF(RTIO2-AMAX3)4246,4246,4245
 4245 AMAX3=RTIO2
 4246 CONTINUE
      IF ( AMAX3 + 1E34) 430,430,4247
 4247 IF (AMAX2 - 1E34) 4248,429,429
 4248 DIFF1 =DABS (AMAX2 * (ATAB(I2,1) + ALW))
      DIFF2 =DABS (AMAX3 * (ATAB(I2,1) + AUP))
      DIFF =DABS (DIFF1 - DIFF2)
      IF(DIFF-AMAX)425,425,4249
 4249 AMAX=DIFF
      SVALW=ALW
      ISVI2=I2
      IF(DIFF1-DIFF2)4251,4251,4252
 4251 ANDCT4=0.
      GO TO 425
 4252 ANDCT4=1.
  425 CONTINUE
  457 CONTINUE
      ALW=SVALW
      I2=ISVI2
      VAL(I1)=ALW+ANDCT4
      BTMVL(I1)=VAL(I1)-1.
 4255 TPVAL(I1)=VAL(I1)+1.
      GO TO 432
C***
C
  426 IF (DABS( ATAB(I2,1) + ALW) - ADELT) 427, 427, 5100
  427 BTMVL(I1)=-1.
      TPVAL(I1)=1000.
      VAL(I1)=ALW
      IVAR(I1)=IROW(I2)
      IROW(I2)=0
      GO TO 5000
C***
C
  429 BTMVL(I1)=-1.
      IF (DABS ( ATAB(I2,1) + ALW) - ADELT ) 4295, 4295, 4296
 4295 ANDCT4=0.
      VAL(I1)=ALW+ANDCT4
      GO TO 4255
 4296 TPVAL(I1)=ALW+2.
      ANDCT4=1.
      GO TO 431
C***
C
  430 TPVAL(I1)=1000.
      BTMVL(I1)=ALW-1.
      ANDCT4=0.
  431 VAL(I1)=ALW+ANDCT4
C***
C
  432 JSVN=N
```

```
      L=KSVN(I1)
  438 DO 439 I3=1,M
      ISVROW(I3,I1)=IROW(I3)
      DO 439 I4=1,N
      I6=L+I4-1
      IF(I3-1)4385,4385,439
 4385 SAVTAB(M+1,I6)=ICOL(I4)
  439 SAVTAB(I3,I6)=ATAB(I3,I4)
      ISVN(I1)=N
      KSVN(I1+1)=L+N
      ATAB(I2,1)=ATAB(I2,1)+VAL(I1)
      ISVRCL(I1)=I2
      IVAR(I1)=IROW(I2)
      ICORR(I1)=1
      IROW(I2)=0
      IF (DABS ( ATAB(I2,1)) - ADELT) 433,433,434
  433 ATAB (I2,1) = 0.0
  434 INDCTR=2
C***
C
      IF(IOUT1)240,254,240
C
  435 IF(ANDCT4)4355,4352,4355
 4352 BTMVL(I1)=-1.
      GO TO 5120
 4355 TPVAL(I1)=1000.
      GO TO 5120
C***
C
  437 GO TO 5000
C
 5000 IF(I1-NZR1VR)5050,550,550
C
 5050 I1=I1+1
      IF(IOUT1)5051,402,5051
 5051 WRITE(NO,70)I1
      GO TO 402
C***
C
C
 5100 I1=I1-1
      IF(IOUT1)5110,5115,5110
 5110 WRITE(NO,75)I1
 5115 IF(I1)995,995,5120
 5120 IF(IVAR(I1)-1000)5151,5151,5152
 5151 K=IVAR(I1)
      GO TO 5153
 5152 K=IVAR(I1)-1000
 5153 I2=ISVRCL(I1)
 5155 IF(BTMVL(I1))516,517,517
  516 IF(TPVAL(I1)-UPBND(K))518,518,5100
  517 IF(TPVAL(I1)-UPBND(K))530,530,525
C***
C
  518 INDCT5=1
```

```
 5181 IF(ICORR(I1))5198,5182,5198
 5182 IF(I1-I1ROW)5183,5198,5198
 5183 INDCT8=1
      IF(I1-1)5185,5198,5185
 5185 INDCT5=4
      ISVI1=I1-1
      I1=1
      GO TO 5198
 5190 DO 5194  I3=1,ISVI1
      I4=ISVRCL(I3)
      ICOL(I4)=ICOL(N)
      DO 5193  J=1,M
      IF(VAL(I3)-1.)5193,5191,5192
 5191 ATAB(J,1)=ATAB(J,1)+ATAB(J,I4)
      GO TO 5196
 5192 ATAB(J,1)=ATAB(J,1)+VAL(I3)*ATAB(J,I4)
 5196 INDCT8=2
 5193 ATAB(J,I4)=ATAB(J,N)
      N=N-1
 5194 CONTINUE
 5195 I1=ISVI1+1
      INDCT5=1
      GO TO 521
C***
C
 5198 N=ISVN(I1)
      L=KSVN(I1)
      DO 5199  I3=1,M
      IROW(I3)=ISVROW(I3,I1)
      DO 5199  I4=1,N
      I6=L+I4-1
      IF(I3-1)5197,5197,5199
 5197 ICOL(I4)=SAVTAB(M+1,I6)
 5199 ATAB(I3,I4)=SAVTAB(I3,I6)
 5205 GO TO  (521,526,531,5190),INDCT5
  521 VAL(I1)=TPVAL(I1)
      TPVAL(I1)=TPVAL(I1)+1.
      IF(ICORR(I1))541,522,541
  522 DO 523 I3=1,M
      ATAB(I3,1)=ATAB(I3,1)+(VAL(I1)*ATAB(I3,I2))
      IF  (DABS  (  ATAB(I3,1))  -  ADELT)  5225,5225,  523
 5225 ATAB(I3,1)=0.
  523 ATAB(I3,I2)=ATAB(I3,N)
      ICOL(I2)=ICOL(N)
      N=N-1
      IF(ATAB(1,1)-TLRNCE)5235,5100,5100
 5235 IF(I1-I1ROW)650,5415,5415
C***
C
  525 INDCT5=2
      GO TO 5198
  526 VAL(I1)=BTMVL(I1)
      BTMVL(I1)=BTMVL(I1)-1.
      GO TO 541
C***
```

```
C
  530 INDCT5=3
      GO TO 5198
  531 AMAX2 = 1.0E35
      AMAX3 = -1.0E35
      DO 536 I3=2,N
      IF(ATAB(I2,I3))534,536,532
  532 RTIO=ATAB(1,I3)/ATAB(I2,I3)
      IF(RTIO-AMAX2)533,536,536
  533 AMAX2=RTIO
      GO TO 536
  534 RTIO2=ATAB(1,I3)/ATAB(I2,I3)
      IF(RTIO2-AMAX3)536,536,535
  535 AMAX3=RTIO2
  536 CONTINUE
      IF(AMAX2-1.E35)538,537,537
C***
C
  537 BTMVL(I1)=-1.
      GO TO 521
  538 IF(AMAX3+1.E35)539,539,540
C***
C
  539 TPVAL(I1)=1000.
      GO TO 526
  540 DIFF1 =DABS ( AMAX2 * (ATAB(I2,1) + BTMVL (I1)))
      DIFF2 =DABS ( AMAX3 * (ATAB(I2,1) + TPVAL (I1)))
      IF(DIFF1-DIFF2)526,526,521
  541 ATAB(I2,1)=ATAB(I2,1)+VAL(I1)
      IROW(I2)=0
      IF (DABS ( ATAB(I2,1)) - ADELT) 5412, 5412, 5415
 5412 ATAB(I2,1)=0.
 5415 INDCTR=3
      IF(IOUT1)240,2505,240
C***
C
  542 GO TO (544,547,543),INDCT5
  543 IF(TPVAL(I1)-VAL(I1)-1.)545,544,545
  544 TPVAL(I1)=1000.
      GO TO 5120
  545 IF(VAL(I1)-BTMVL(I1)-1.)546,547,546
C***
  546 CONTINUE
  547 BTMVL(I1)=-1.
      GO TO 5120
C***
C
  548 GO TO 5000
C
  550 TLRNCE=ATAB(1,1)
      SOLMIN=1.
C***
C
      IF ( IOUT3) 552,553,552
  552 ZOPT = DABS( ATAB( 1,1))
```

```
      WRITE (NO, 35) ZOPT, ICNTR
  553 DO 560 I=1, NZR1VR
      IF(IVAR(I))554,560,554
  554 IF(IVAR(I)-1000)555,555,557
  555 J=IVAR(I)
      T(J)=VAL(I)
      GO TO 560
  557 J=IVAR(I)-1000
      T(J)=UPBND(J)-VAL(I)
  560 CONTINUE
      WRITE (NO, 19)
  565 WRITE (NO, 18) (I, I=1, NM1)
      WRITE (NO, 10) (T(I), I = 1, NM1)
      GO TO 5115
  600 GO TO (605,4205),INDCT7
  605 INDCTR=4
      IF(IOUT1)240,254,240
C***
C
  610 GO TO 5100
C***
C
  615 INDCT7=2
      GO TO 402
C***
C
C
  650 DO 655 I=1,M
      ITBROW(I)=IROW(I)
      DO 655 J=1,N
  655 TBSAV(I,J)=ATAB(I,J)
      DO 660 J=1,N
  660 ITBCOL(J)=ICOL(J)
      JSVN=N
      INDCTR=5
      IF(IOUT1)240,254,240
C***
C
  665 GO TO (544,5120),INDCT8
C
C***
C
C
  670 N=JSVN
      DO 675 I=1,M
      IROW(I)=ITBROW(I)
      DO 675 J=1,N
  675 ATAB(I,J)=TBSAV(I,J)
      DO 680 J=1,N
  680 ICOL(J)=ITBCOL(J)
      GO TO 5000
C***
C
  995 IF(ITOL)996,9976,996
  996 IF(SOLMIN-1.E35)9976,997,997
```

```
  997 ITOL=ITOL+1
      TLRNCE=FLOAT(ITOL)*PCTTOL*ATAB11+ATAB12
      N=ISVN(1)
      DO 9972 I=1,M
      IROW(I)=ISVROW(I,1)
      DO 9972 J=1,N
 9972 ATAB(I,J)=SAVTAB(I,J)
      DO 9973 K=1,N
 9973 ICOL(K)=SAVTAB(M+1,K)
      GO TO 400
  998 CONTINUE
 9976 WRITE (NO, 50)
 1001 WRITE (NO,65) ICNTR
  999 NMRUNS=NMRUNS-1
      IF(NMRUNS)1000,1000,1000
 1000 RETURN
      END
```

```
      INTEGER I,J,IA,N,M1,M2,IW(58 ),IER,T,K ,NCER(3,3)
      REAL A(20,20),B(20),C(12),PSOL(18),DSOL(20),RW(458),
     $   CP(3,3),BF(3,3),ER(3,3),Q(3,3),ALP(3,3),BQR(3,3),QR(3,3),
     $  N122,N211,N233,N322,N311,N133,N123,N213,N231,N321,N312,N132,
     $  EQ(3,3),R(3,3),L(3,3),LOST,DALP(3,3),BQ(3,3),EER(3,3),QRP,
     $  U(3,3),QT(3,3,100),QQ(3,3,100),LT(3,3,100),TARR,RO(3,3),
     $  TDSEI(90)
      INTEGER NR,B(100)
      REAL RLAM
      DOUBLE PRECISION DSEED
      DSEED=123457.D0
      NR=1
C************************
      N=12
      M1=18
      M2=0
      IA=20
      T=1
      K=61
      LOST = 0.0
      QQQ=0.0
C*********** CAPASITY OF CHANNELS **********
      CP(1,2)=50.0
      CP(2,1)=50.0
      CP(2,3)=50.0
      CP(3,2)=50.0
      CP(3,1)=50.0
C***************ERROR RATE OF CHANNELS*********************
      RO(1,2)=1E-2
      RO(2,1)=5E-3
      RO(2,3)=1E-2
      RO(3,2)=7E-3
      RO(3,1)=9E-3
      RO(1,3)=2E-3
      CP(1,3)=50.0
C ********** BUFFER **********
      BF(1,2)=100.0
      BF(2,1)=100.0
      BF(2,3)=100.0
      BF(3,2)=100.0
      BF(3,1)=100.0
      BF(1,3)=100.0
      Q(1,2)= 0.5 * BF(1,2)
      Q(2,1)= 0.5 * BF(1,2)
      Q(2,3)= 0.5 * BF(1,2)
      Q(3,2)= 0.5 * BF(1,2)
      Q(3,1)= 0.5 * BF(1,2)
      Q(1,3)= 0.5 * BF(1,2)
      DO 10 I=1,3
      DO 10 J=1,3
      QT(I,J,T)=Q(I,J)
      LT(I,J,T)=0.0
      QQ(I,J,T)=Q(I,J)
 10   CONTINUE
      T=T+1
```

```
      TARR = 0.0
C**********   ARRIVAL **********
  300 READ(5,702) R(1,2),R(2,1),R(2,3),R(3,2),R(3,1),R(1,3)
  702 FORMAT(3X,6F10.4)
      DO 701 I=1,3
      DO 701 J=1,3
      IF(I.EQ.J) GO TO 701
C     RLAM = ER(I,J)
C****************************
C     CALL GGPOS(RLAM,DSEED,NP,G,IEF)
C****************************
C     R(I,J)=G(1)
      WRITE(6,700) I,J,R(I,J)
 700  FORMAT(3X,'R(',I1,',',I1,') =',F8.2)
      TARR = TARR + R(I,J)
 701  CONTINUE
      DO 160 I=1,3
      DO 160 J=1,3
      IF(I.EQ.J) GO TO 160
      IF((Q(I,J)+R(I,J)).GT.CP(I,J)) GO TO 161
      U(I,J)=(Q(I,J)+P(I,J))
      GO TO 162
 161  U(I,J)=CP(I,J)
  162 IF(RO(I,J) .LE. 0.0) GO TO 163
      IF(U(I,J) .LE. 0.0) GO TO 163
      RLAM=U(I,J)*RO(I,J)
C**********************
      CALL GGPOS(RLAM,DSEED,NP,G,IEF)
C************************************
      NCER(I,J)=G(1)
      GO TO 164
  163 NCER(I,J)=0
  164 Q(I,J)=Q(I,J)+R(I,J)-U(I,J) + NCER(I,J)
 160  CONTINUE
      U(1,1)=0.0000000
      WRITE(6,1001) U(1,2)
 1001 FORMAT(3X,'N122 =',F10.4)
      WRITE(6,1002) U(2,1)
 1002 FORMAT(3X,'N211 =',F10.4)
      WRITE(6,1003) U(2,3)
 1003 FORMAT(3X,'N233 =',F10.4)
      WRITE(6,1004) U(3,2)
 1004 FORMAT(3X,'N322 =',F10.4)
      WRITE(6,1005) U(3,1)
 1005 FORMAT(3X,'N311 =',F10.4)
      WRITE(6,1006) U(1,3)
 1006 FORMAT(3X,'N133 =',F10.4)
      WRITE(6,1007) U(1,1)
 1007 FORMAT(3X,'N123 =',F10.4)
      WRITE(6,1008) U(1,1)
 1008 FORMAT(3X,'N213 =',F10.4)
      WRITE(6,1009) U(1,1)
 1009 FORMAT(3X,'N231 =',F10.4)
      WRITE(6,1010) U(1,1)
 1010 FORMAT(3X,'N321 =',F10.4)
```

```
      WRITE(6,1011) U(1,1)
 1011 FORMAT(3X,'N312 =',F10.4)
      WRITE(6,1012) U(1,1)
 1012 FORMAT(3X,'N132 =',F10.4)
 139  DO 143 I=1,3
      DO 143 J=1,3
      IF(I.EQ.J) GO TO 143
      IF(Q(I,J).GT.BF(I,J)) GO TO 140
      L(I,J)=0.0
      GO TO 141
 140  L(I,J) = Q(I,J) - BF(I,J)
 141  WRITE(6,142) I,J,L(I,J)
 142  FORMAT(3X,'L(',I1,',',I1,') =',F8.2)
      Q(I,J)=AMAX1(0.0,Q(I,J))
      QQ(I,J,T)=Q(I,J)
      Q(I,J)=Q(I,J)-L(I,J)
      WRITE(6,170) I,J,Q(I,J)
 170  FORMAT(10X,'Q(',I1,',',I1,') =',F8.2)
      LOST=LOST + L(I,J)
 143  CONTINUE
      DO 210 I=1,3
      DO 210 J=1,3
      QT(I,J,T)=Q(I,J)
      LT(I,J,T)=L(I,J)
      QQQ = QQQ + QT(I,J,T)
  210 CONTINUE
      T=T+1
      IF(T.LE.K) GO TO 300
C*************************
       DO 201 T=1,K
       WRITE(6,1100) QT(1,2,T),QT(2,1,T),QT(2,3,T),QT(3,2,T),
     $                 QT(3,1,T),QT(1,3,T)
 1100 FORMAT(3X,6F10.4)
 201  CONTINUE
      DO 202 T=1,K
      WRITE(6,1101) LT(1,2,T),LT(2,1,T),LT(2,3,T),LT(3,2,T),
     $                LT(3,1,T),LT(1,3,T)
 1101 FORMAT(6X,6F10.2)
 202  CONTINUE
      DO 203 T=1,K
      WRITE(6,1102) QQ(1,2,T),QQ(2,1,T),QQ(2,3,T),QQ(3,2,T),
     $                QQ(3,1,T),QQ(1,3,T)
 1102 FORMAT(3X,6F10.4)
 203  CONTINUE
      WRITE(6,490) TARR
 490  FORMAT(10X,'TOTAL ARRIVAL =',F10.3)
      WRITE(6,150) LOST
 150  FORMAT(10X,'LOST = ',F15.5)
      QIL=0.0
      DO 1104  I=1,3
      DO 1104  J=1,3
      IF(I .EQ. J) GO TO 1104
      QIL = QIL + QT(I,J,1) - QT(I,J,K)
 1104 CONTINUE
      WRITE(6 ,1105) QQQ
```

```
FILE: FIX        FORTRAN   A                   VM/SP CONVERSATIONAL MONITOR S

 1105 FORMAT(3X,'SUM OF QUEUES=',F10.3)
      AVD = QQQ / (TARR-LOST+QIL)
      WRITE(6 ,1106) AVD
 1106 FORMAT(3X,'AVG.DELAY =',F10.5)
      THR=(QIL+TARR-LOST)/60.0
      WRITE(6,1110) THR
 1110 FORMAT(3X,'THR.= ',F10.5)
      STOP
      END
```

The IMSL routines used in these programs are as follows;

\- - - - - - - - - - - - - - - - - - -

GGPOS, GGNML, GGUBS, MDNRIS,

MERFI, UERTST, UGETIO

\- - - - - - - - - - - - - - - - - - -

REFERENCES

[1] Ahmed, N.U., "Optimal Control of Stochastic Systems", Probabilistic Analysis and Related Topics, Vol 2, pp 1-68, Academic Press, 1979.

[2] Ahmed, N.U., Dabbous, T.E., Biswas, S.K.,"Modelling of Large Interconnected Power Systems", Proc. 4th International Symposium on Large Engineering Systems, Calgary, 173-178, 1982.

[3] Ahmed, N.U., Dabbous, T.E.,"Modelling and On-Line Control of Reliability Dynamics of Large Scale Interconnected power systems", Int. J. Systems Sci. Vol.14, No.12, 1321-1353, 1983.

[4] Ahmed, N.U., Lee, Y.W., "A Dynamic Routing For Computer Queueing Networks", IASTED Int. Conf. Proc. on Applied Simulation and Modelling (ASM'85), Montreal, June 1985.

[5] Athans, M., Falb, P.L., Optimal Control, McGraw-Hill, 1966.

[6] Black, U.D., Data Communications, Networks, and Distributed Processing, Reston Publishing Co., Inc. 1983.

[7] Boel, R., "Stochastic Models Of Computer Networks" Stochastic Systems: The Mathematics of Filtering and Identification and Applications, 141-167, 1981.

[8] Burke, P.J., "The Output of a Queuing System", Operations Research, Vol. 4, 699-704, 1956.

[9] Cantor, D.G., Gerla, M., "Optimal Routing in a Packet-Switched Computer Network", IEEE Trans. on Computers, Vol. C23, No. 10, Oct. 1974.

[10] Chu, K.C., "Decentralized Dynamic Allocation Scheme for Large Congested Network", I.B.M. Research Report, RC 6337 (No. 27225), Dec. 28, 1976.

[11] Chu, K.C., Gazis, D.C., "Dynamic Allocation of Parallel Congested Traffic Channels", Trans. and Traffic Theory (Proc. od 6th Int. Symp. on Trans. and Traffic Theory) D.J. Bullock, ed.; A.H. and A.W. Reed, Sydney, 1974 (Also IBM Research Report RC4545; 1973).

[12] Davies, D.W., "The Control of Congestion in Packet-

Switching Networks", IEEE Trans. on Comm., Vol. COM-20, No.3, June 1972.

[13] Fleming, W.H., Rishel, R.W., "Deterministic and Stochastic Optimal Control", Springer-Verlag, 1975.

[14] Gazis, D.C., "Modelling and Optimal Control of Congested Transportation Systems", Networks, Vol. 4, 113-124 1974.

[15] Gazis, D.C., D'Ans, G.C., "Optimal Control of Oversaturated Transportation Networks" IBM Research Report RC 4543, Sept. 1973.

[16] Georganas, N.D. "Modeling and Analysis of Message Switched Computer-Communication Networks with Multilevel Flow Control", Computer Networks, Vol. 4, No.6, 285-294 1980.

[17] Jackson, J.R., "Networks of Waiting Lines", Operations Research, Vol. 5, No.4, 518-521, 1957.

[18] Kahn, R.E., Crowther, W.R., "Flow Control in a Resource -Sharing Computer Network", IEEE Tras. on Comm. Vol.Com -20, No.3, June 1972.

[19] Kleinrock, L., Communication Nets; Stochastic Message Flow and Delay, Dover Publications, Inc. New York 1964.

[20] Kleinrock, L., Opderbeck, H., "The Influence of Control Procedures on The Performance of Packet-Switched Networks", Conf. Rec. National Telecommunication Conference, San Diego, Dec. 1974.

[21] Kleinrock, L., Kermani, P., "Dynamic Flow Control in Store-Forward Computer Networks", IEEE Trans. on Comm. Com-28, No.2, Feb 1980.

[22] Kobayashi, H., "Queueing Models for Computer Communications System Analysis", IEEE Trans. on Comm. Vol.Com-25 No.1, Jan. 1977.

[23] Kuester, J.L., Mize, J.H., "Optimization Techniques with Fortran", McGraw-Hill, 1973.

[24] Kuo, F.F., Huynh, D., "Mixed-Media Packet Networks", Protocols and Techniques for Data Communication Networks, edited by Kuo, H.H., 202-239, 1981.

[25] Lam, S.S., Reiser, M., "Congestion Control of Store-and -Forward Networks by Input Buffer Limits", Conf. Rec., National Telecommunication Conference, Los Angeles, 12-1, Dec. 1977.

[26] Lazar, A.A., "Optimal Flow Control of a Class of Queueing Networks in Equilibrium", IEEE Trans. on Auto. Cont., Vol. AC-28, No.11, Nov. 1983.

[27] McQuillan, J.M., Richer, I., Rosen, E.C., "The New Routing Algorithm for the ARPANET", IEEE Trans. on Comm., Vol. COM-28, No.5, May 1980.

[28] Pooch, U.W., Greene, W.H., Moss, G.G., Telecommunications and Networking, Little, Brown and Co. 1983.

[29] Sarachik, P.E., "Dynamic Allocation for Clearing a Congested Multi-Destination Transportation Network", IBM Research Report RC-6479 (No.27978), Apr. 1977.

[30] Sarachik, P.E., "Optimal Local Dynamic Routing for Clearing Congested Multi-Destination Networks", Dep. Elec. Eng., Polytech. Inst. New York, Jan. 1981.

[31] Sarachik, P.E., "An Effective Local Dynamic Strategy to Clear Congested Multidestination Networks", IEEE Trans. on Auto. Cont., Vol. AC-27, No.2, Apr. 1982.

[32] Schwartz, M., Pennotti, M.C., "Congestion Control in Store and Forward Tandem Link", IEEE Trans. on Comm. Vol. COM-23, No.12, Dec. 1975.

[33] Schwaltz, M., Computer-Communication Network Design and Analysis, Prentice-Hall, Inc., New Jersey, 1977.

[34] Schwartz, M., Stern, T.E., "Routing Techniques Used in Computer Communication Networks", IEEE Trans. on Comm. Vol. COM-28, No.4, Apr. 1980.

[35] Schwartz, M., "Routing and Flow Control in Data Networks", New Concepts in Multi-User Communications, J.K. Skwirzynski, Ed., NATO Advanced Study Inst. Series (Series E: Applied Science 43) 1981.

[36] Segall, A., Kailath, T., "The Modeling of Random Modulated Jump Processes", IEEE Trans. on Inform. Theory, Vol. IT-21, No.2, March 1975.

[37] Segall, A., "The Modelling of Adaptive Routing in Data-Communication Networks", IEEE Trans. on Comm. Vol. COM-25, No.1, Jan. 1977.

[38] Segall, A., Moss, F.H., "An Optimal Control Approach to Dynamic Routing in Networks", IEEE Trans. on Auto. Cont., Vol. AC-27, No.2, Apr. 1982.

[39] Tanenbaum, A.S., Computer Networks, Prentice-Hall, Inc. 1981.

[40] Tseng, C.W., "Flow Control in Store-And-Forward Computer Networks", Dep. Computer Sci. Report No. CSD-810404, Apr. 1981.

[41] Wong, J.W., "Distribution of End-to-End Delay in Message-Switched Networks", Computer Networks, Vol. 2, 44-49 1978.

VITA

Name ;   Yong Wha Lee

Born;   June 28th, 1953, Pusan, Korea

Eduacation; Yon-Sei University, Seoul, Korea

Ba.Sc. in Electrical Engineering, 1979.