CANADIAN THESES

THÈSES CANADIENNES

ANALYTICAL MODELLING OF PACKET-SWITCHING NETWORKS WITH NODAL BUFFER MANAGEMENT AND END-TO-END FLOW CONTROL

by

Andrew James Kozlowski

A thesis
presented to the University of Ottawa
in partial fulfillment of the
requirements for the degree of
M.A.Sc.
in
the Department of Electrical Engineering

OTTAWA, Ontario, 1983

ABSTRACT

Analytical models for packet-switching networks with both nodal buffer management and end-to-end flow control have no known exact queueing theory solution. An approximate model is developed to study the effects of buffer management strategies (e.g. complete sharing) on network performance measures (e.g. throughtput, delay). These measures are approximated by modifying the unblocked network measures with blocking probabilities. The model defines each node as a set of output channel queues, and separates channel traffic into 2 classes: input traffic and transit traffic. The channel joint probabilities, p(input,transit), are computed using an algorithm based on the Reiser-Kobayashi convolution algorithm. Nodal blocking probabilities are computed by summation of the joint probabilities over the subset of the state space restricted by the intersection of the nodal buffer constraints and end-to-end flow control constraints. The algorithm has been implemented in PASCAL and the results of a numerical study are presented.

Keywords: computer network modelling, queueing networks, blocking.

ACKNOWLEDGEMENTS

I wish to express my gratitude to those who helped during my work at the University of Ottawa, especially to:

Dr. Nicolas D. Georganas for his guidance throughout my thesis work.

Those in my research group who kept a lively atmosphere for working: Mr. Adrian Conway, Mr. Esper Bitar, Mr. Philip Mui, Mr. Eng Teo, Mr. Miguel Rios, Mr. Kevin Wong, and especially Mr. Pavlos Kariotellis.

Mr. Gregory Dudek for his assistance in some of the computer programming problems involved in the numerical study.

Mr. René Le Hénaff, the administrative officer, for his expeditious handling of many administrative tasks.

The bodies who provided financial support: the School of Graduate Studies and the Department of Electrical Engineering, University of Ottawa, and NSERC.

Finally, to the AMDAHL 470/V7A without whose help this thesis would have taken much longer to complete.

TABLE OF CONTENTS

APPENDICES

SYMBOLS, UNITS AND ABBREVIATIONS

Latin Symbols

$a[r], a_r$	numbers of buffer allocated to class r packets
$b[r], b_r$	maximum number of buffers of shared storage which can be occupied by class r packets at any time
B	total number of nodal buffers
c_m	capacity function for queue m in the k-domain
C	normalization constant for the queueing network
C_m	channel capacity for service center m
$C_m(\underline{Z})$	capacity function for service center m in the z-domain
D	mean network delay
$D[r], D_r$	mean delay for class r packets
e_k	unit vector in the k-th dimension
F	network state space (with only window flow control)
$g(\underline{K})$	normalization constant for a network with $\underline{K}$ customers
i	input traffic
$\underline{i}$	$= (i_1, i_2, \ldots, i_R)$ a (subnetwork) population vector by subchain
$k[m], k_m$	$= (k_{m1}, k_{m2}, \ldots, k_{mR})$ number of packets of each subchain in queue m
$\underline{K}$	population vector by subchain $(K_1, K_2, \ldots, K_R)$

$l_V(l_{V'})$ number of packets of type V(V')

L total number of subchains (open and closed)

$\underline{L}$ queue/nodal population vector by "superclass" (see appendix C) (l_V , $l_{V'}$)

m mean number of packets in the network

m_r mean number of packets in class r

M total number of service centers in the network

$\underline{n}$ $=(n_1, n_2, \ldots, n_R)$ a point in the state space

N total number of nodes in the network

$n[i], n_i$ number of class i packets in the resource, or number of input packets in a node

$n[t], n_t$ number of transit packets in a node

p(X) the probability of event X occuring

P network power

pbi vector of input traffic blocking probabilities for each node. If subscripted with a 'c', it means the computed value.

$pb[r], pb_r$ blocking probability for class r

pbn[n] blocking probability at node n for any class

pbt Same as pbi but for transit traffic.

R total number of closed subchains in the network

$R[B], R_B$ total number of buffer classes in a node

s the source node for the class under consideration

$\underline{S}$ network state vector

$S[r], S_r$ subchain r source arrival rate (pac/s)

t transit traffic

T	mean network throughput
$T[r], T_r$	mean throughput for class r packets
U	utilization
U_m	utilization for the m-th service center
v_k	visit ratio for queue k
$\underline{w}$	$=(w_1, w_2, \ldots, w_R)$ maximum number of packets in each class (window size)
$w[r], w_r$	window size for class r
$\underline{y}$	see $\underline{i}$

Non-Latin Symbols

ϵ	convergence criterion
$\lambda_i^{(r)}$	rate of class r exogenous arrivals to node i
$\underline{\rho}$	$=[\rho_1, \rho_2, \ldots, \rho_M]$ workload vector (also called $\underline{\tau}$)
$1/\mu$	mean packet length (bits)

O()	"order of" function for a number of operations count
<-	replace
=	equate
<>	not equal to
$\cup$	set union
$\circledast$	multi-dimensional convolution
[]	sometimes used as a subscript indicator

UNITS

b	bit
B	baud
bps	bits/s
kB	kilobaud
msg	message
pac	packet(s)
s	second

Abbreviations

BMS	buffer management scheme
bp	blocking probability
CCITT	Comité Consultatif Internationale de Téléphonique et Télégraphique
CCN	computer-communication network
CN	computer network
CP	complete partitioning
CS	complete sharing
DP	distributed processing
DCE	data circuit-terminating equipment
DSE	data switching equipment
DTE	data terminal equipment
FCFS	first-come first served
gf	generating function (same as pgf)
ISO	International Standards Organization
LR	Lam-Reiser BMS [LAM 79]
PE	processing element
pgf	probability generating function (same as gf)
PSN	packet-switching network
SMA	sharing with minimum allocation
SMXQ	sharing with maximum queues
SS	Saad-Schwartz BMS [SCHW 79]
VC	virtual-circuit
1-D	1-dimensional
2-D	2-dimensional

LIST OF FIGURES

LIST OF TABLES

Chapter I

INTRODUCTION

Connected networks have existed for many years in a wide variety of forms. Every day, networks ease life by facilitating the transfer of commodities through a channel or network of channels:

- transfer of vehicles through traffic networks
- transfer of voice through telephone networks
- transfer of audio/voice through television networks
- transfer of data/voice through computer-based networks.

Computer-based networks (computer networks for short) are built through the interconnection of processing elements (PE's). Such structures are currently realized in two forms, although there is no clear distinction between them in the literature:

- distribution is the interconnection of PE's designed to function as a system whose tasks can be shared
- networking is the interconnection of PE's in a random manner.

PE's communicate among each other in units of information called "messages" using a defined set of syntactic and semantic rules called a "protocol".

In either distribution or networking, the communication medium between PE's is called the communication subnetwork or subnet. If the communication is in a store-and-forward manner, and data is transfered in maximum fixed-length subparts of a message (called "packets"), then the subnet is called a packet-switching network (PSN).

Network users see the system providing certain services both at high and low levels:

1. HIGH LEVEL SERVICES
   - distributed data base, data base inquiry
   - distributed computation
2. LOW LEVEL SERVICES
   - data transmission (with or without security)
   - voice transmission
   - access to a remote computer.

The metrics used to quantify the performance are called "performance measures". From a system point of view, these are:

1. throughput
2. delay
3. blocking probabilities
4. utilization.

However, the users (often called "end-users") judge the network's performance by different measures [GRUB 77]:

1. transfer rate

2. availability
3. reliability
4. accuracy
5. channel establishment time
6. network delay
7. line turnaround time
8. transparency
9. security
10. cost.

In order to achieve the required performance, computer networks run resource management tasks. Three of the most important are [REIS 79]:

- flow control which distributes traffic so that the network runs smoothly
- routing which directs traffic through the network according to some objective criteria
- buffer management which deals with the finite number of buffers at each node.

As these tasks have conflicting requirements, they do not influence the network independently.

The purpose of this thesis is to develop an analytical model to find certain performance measures (throughput,delay, blocking probabilities) in packet-switching networks with flow control.

Such an analysis permits the study and performance evaluation of computer networks and modified configurations of computer networks. The results can be used to effectively manage a computer network through the analysis of its operations and economics.

1. Operations analysis

   (a) determination and isolation of traffic problems, e.g. bottleneck nodes

   (b) study of nodal or link failures

   (c) computing the maximum network traffic.

2. Economic analysis

   (a) feasibility of adding new communication channels and the capacity of these lines

   (b) cost of adding an extra node compared to the benefits of improved throughput or attracting new customers in a new geographic region.

Computer networks are predominantly modelled using queueing theory. However, the present results of queueing theory can assist only in a limited manner toward the solution of these problems. For example, there are very few models for networks with blocking. Hence, simplifications are often needed to permit a network model to fall into the class of tractable queueing theory problems.

This thesis is divided into three major parts.

The first part discusses the problem (flow control) and its context (packet-switching networks). Chapter 1 introduces the thesis. Chapter 2 describes computer-communication networks in more technical detail, introducing the mechanisms of flow control and buffer management.

The second part discusses the tools currently available for the modelling of packet-switching networks. Chapter 3 discusses these tools, concentrating on queueing theory approaches.

The third part discusses the analysis of the problem, describes the results of a numerical study, and draws conclusions. Chapter 4 describes the problem, and builds an analytical model for a restricted class of packet-switching networks. Chapter 5 uses the results of chapter 4 to numerically study a network. Chapter 6 summarizes the important results of this thesis.

The appendices provide information which would disturb the continuity of the thesis if imbedded in the body.

Chapter II

COMPUTER-COMMUNICATION NETWORKS

The purpose of a computer-communication network (CCN) is to provide a means of communication between data processing equipment.

CCN's can be divided into three classes based on their method of access [TOBA 80]:

1. single-access (or point-to-point),
2. multiple-access (or broadcast), and
3. a hybrid of both.

A single-access network ,e.g. a packet-switching network, permits only one user to access the communication channel at a time. Access may be allocated in time units, in frequency units or a combination of both. Examples are packet-switching networks (e.g. ARPANET, DATAPAC) and circuit-switching networks (e.g. the telephone network).

A multiple-access network permits many users to share a common transmission medium, which is allocated in time slots to individual users. Examples are ALOHA and ETHERNET.

A hybrid of these types (a store-and-forward broadcast network) is being built for packet radio in project PRNET [TOBA 80].

This thesis will discuss packet-switching networks, and those based on the pairing of network end-users into "virtual circuits" (VC) in particular. A virtual-circuit is a "bidirectional, transparent flow-controlled path between a pair of logical or physical ports" [RYBC 80b], in which all packets are guaranteed to arrive in sequence.

2.1 Packet-switching networks

Typical users of a packet-switching network (PSN) exchange messages which are routed through the network in packets, each with a destination address provided by the origin. The messages are first passed to a nodal interface whose purpose is to divide the messages into packets, and to reliably transmit the packets between nodes. The network may be "lossy", i.e. packets may be dropped (with or without proper notices) from the network if insufficient resources are available to send the packet to its destination.

Two international standards organizations, ISO and CCITT, provide a computer network architecture which defines a set of layers and protocols that provide a standard for public PSN's. In the ISO model (Fig. 1), each of the seven layers provides different services. Organizational entities (called correspondents) of the same layer exchange data in a controlled manner called a "protocol".

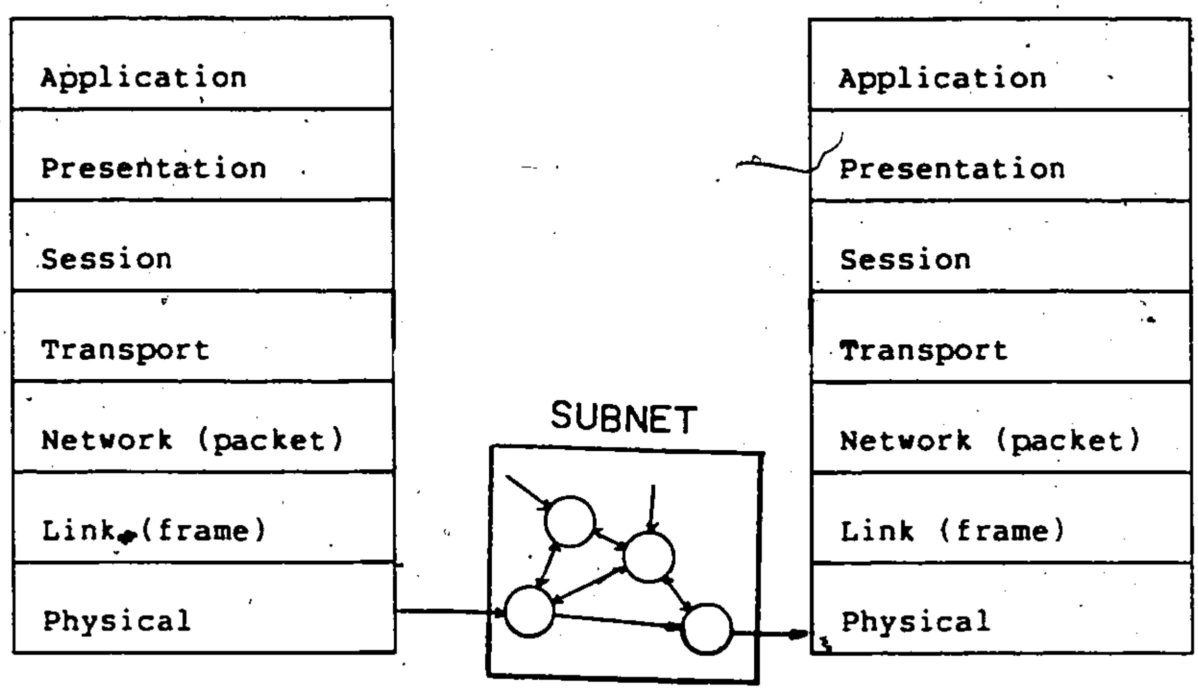


Figure 1: ISO Reference Model of Open System Interconnection [TANE 81,p.16].

The purpose of a protocol is to define the syntax and semantics so that data can be exchanged in an orderly manner. Among the problems to be solved by protocols are:

- synchronization of the activities of the correspondents
- flow control, congestion control and deadlock avoidance
- buffer management for data packets
- error recovery
- efficiency improvement, e.g. data compression.

The implementation of protocols is based on deadlock free algorithms (i.e. they do not reach states which cannot be exitted) and should be fault-tolerant (i.e. have the ability to recover from hardware and software faults to some degree).

The management of the PSN's resources (e.g. nodal processors, nodal buffers, communication channels) is done under operational strategies:

- routing
- flow control
- congestion control
- nodal buffer management
- channel scheduling.

Control of these strategies may be:

- isolated (no consultation between nodes)

- distributed (consultation between nodes)
- centralized (central control).

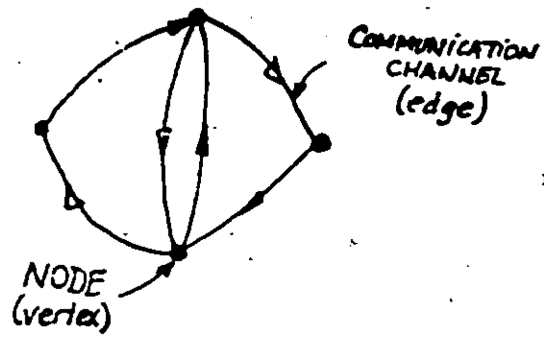
Packet-switching networks have the following features:

1. Traffic is bursty; hence, data transfer requires a low setup time.
2. Traffic is transferred through the network between nodes in a store- and-forward manner, through communication channels called "links". More generally, a link is viewed as "an abstraction of any physical route linking two separate entities" [DAVI 81,p. 86].
3. If the link capacity is exceeded, traffic is "blocked and queued" (where possible) rather than "blocked and lost". (The latter is the case in the telephone network.)

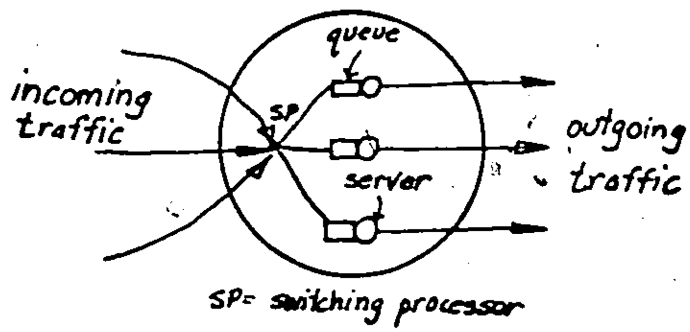
In order to investigate network performance and control mechanisms, PSN's can be modelled in a simplified abstract form with the following elements (Fig. 2):

1. Communication subnet
   - The topology defined by nodes and links.
   - Queues, which store packets in "buffers".
   - Service centers, which send data packets from queues over the links.
   - Virtual circuits, often referred to as classes.
2. terminals, devices

(a) TOPOLOGY



(b) NODE



(c) PACKET CLASSES ("virtual circuits")

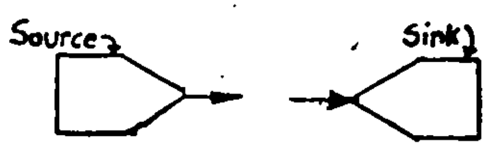


Figure 2: Abstract Description of a PSN.

3. protocols.

The four basic measures used to judge the performance of a PSN are:

1. mean throughput (T), measured in packets/second
2. mean delay (D), measured in milliseconds
3. power (P), measured in packets/second-squared
4. blocking probability (pb).

Throughput can be defined as the number of packets which pass through some exit point(s) in the network. Usually, the term includes only data traffic and not the administrative traffic (e.g. accounting and network control packets). Variations of this term may include:

1. retransmitted packets due to blocked nodes
2. overhead network traffic.

In this thesis, throughput includes only data traffic which both enters and exits the network. The term "offered load" is used to represent the amount of data traffic which is submitted to the network. Some of these packets may be blocked from entry, and therefore will not contribute to the throughput.

Delay can be defined as the time spent by the packet in the network when travelling between end-users. Packet delay is introduced by:

1. propagation ,transmission delay

2. queueing delay.

In terrestrial networks, propagation time is roughly 6 μs/km (about 45ms from St. John's to Vancouver). By comparison, satellite end-to-end transit delay is 250-300 ms.

The simplest definition of power is the ratio of throughput to delay:

$$\text{Power} \triangleq \frac{\text{Throughput (pac/s)}}{\text{Delay (s)}} \qquad (2.1A)$$

As power is not computed in this thesis, it will not be discussed further.

In real networks, users are provided with certain performance guarantees:

1. VC throughput guarantees
2. VC delay and delay variance guarantees.

Throughput guarantees give guidelines for the amount of data that a user can offer to the network with a high probability of being carried.

Delay guarantees give a response time estimate which is useful for on-line applications, e.g. airline reserveration systems. For example, ARPANET was designed for a 200ms mean roundtrip delay for single-packet messages [KLEI 75,v.II,p.462]. Delay variance gives an estimate of the predictability of the network.

Blocking probability gives the percentage of times (on the average) that a packet will arrive at a node and be rejected due to inadequate resources at that node. This is usually the result of a lack of sufficient nodal buffer space to store the packet.

Another quantifiable performance measure is service center utilization, which is the percentage of time that a server is in use. It is measured in busy-time/second.

2.1.1 Flow Control

In order to achieve "reasonable" levels of performance, mechanisms must be introduced into the network to assist in resource allocation. ("Reasonable" levels are those arrived at by a consensus of the actual network designers.) Although an optimal allocation policy is desired, the operation of present computer networks is so complex that no such solution can be obtained. Such policies are synthesized in practice through a combination of analysis and network performance monitoring.

One of the tasks of the allocation policy is flow control, which directs network operations:

1. When resources are available, flow control satisfies the needs of competing users fairly (Fig. 3) by:
   (a) speed matching between end-users, and

(b) increasing efficiency by utilizing communication capacity and storage capacity wastage.

2. When resources are scarce, by avoiding and settling contention, e.g. by preventing throughput degradation due to overload.
3. avoiding deadlock (Fig. 3).

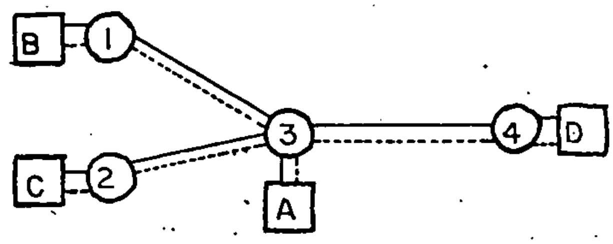
The implementation of these allocation algorithms have the additional desirable requirements in real networks:

1. failsafe operation
2. error detection and error recovery
3. fault tolerance.

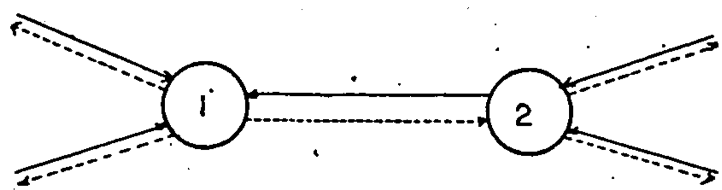
Flow control mechanisms attempt to reduce these problems through the design of good algorithms. The appearance of these problems in real networks can also occur through a faulty implementation of the algorithm. The study of such problems is in the domain of software engineering and will not be dealt with here.

The term flow control is not used consistently in the literature. However, Lemieux [1979] gives a good treatment of the concepts involved by using analogies with the telephone network. Flow control, Lemieux states, includes a set of control mechanisms:

1. traffic control



FAIRNESS - An inequitable flow-controlled network [GERL 82, p.30]. A,B and C are transmitting files to node D. If node 3 imposes input buffer limits, then A may be penalized with respect to B and C as saturation of the 3-4 link occurs.



DEADLOCK of the direct type [GERL 80].
If the network retransmits dropped packets, deadlock may occur under heavy load.
1 may fill up for packets destined for 2 and vice versa. Either node may hold all of its buffers for packets being transmitted to the other, and keep transmitting packets to the other.

Figure 3: Important Objective Criteria in Computer Networks (other than throughput and delay)

2. congestion control (Fig. 4)
3. routing control
4. delay control.

Traffic control deals with the allocation of local resources to local traffic in order to resolve local contention. The use of end-to-end windows and the queueing of blocked packets are examples.

Congestion control attempts to avoid the congestion which occurs when throughput remains constant or decreases with increased offered traffic. Overload occurs when the specified performance level is reached. Congestion is often controlled by structured buffer management policies some of which are discussed in section 2.1.2.

Routing control is used to adjust the degree of freedom of routing to maintain maximum throughput. Higher degrees of freedom increase the number of possible paths, but also increase the average amount of resources needed for each packet. In PSN's, the cost of alternate routing (a simple method of dynamic routing with only two possible paths) is of the order of 5 to 30 times that of direct routing [LEMI 79, p.12]. For this reason, alternate routing is avoided in DATAPAC, except during nodal or link failures. However, a time dependent set of fixed routes is used to improve network performance over the various time zones in Canada.

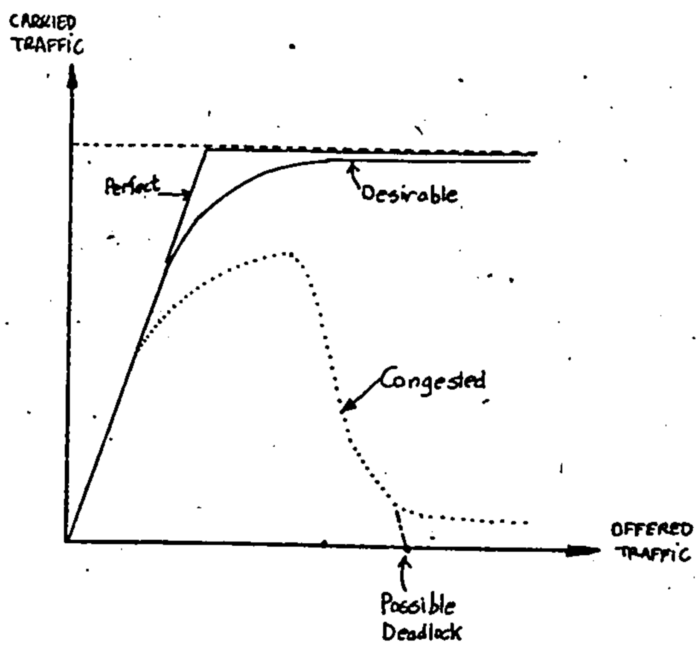


Figure 4: Congestion Control [TANE 81, p.215].

Delay control is the maximum time after which loss is declared after blocking has occurred. As blocked packets are immediately acknowledged in PSN's, they have no delay control.

2.1.1.1 Flow Control Mechanisms

Flow control mechanisms can be divided into three classes, depending on the information available to the task. (The references are papers which deal with the method analytically.)

1. end-to-end, e.g. the window mechanism [REIS 79]
2. local or hop-level, e.g. input buffer limits [LAM 79]
3. global, e.g. the isarithmic mechanism [GEOR 80b].

End-to-end flow control limits the number of packets permitted on each VC. This prevents congestion at the destination node.

Local flow control limits the packets at each node, i.e. locally, usually by a structured buffering scheme.

Global flow control limits the number of packets in the entire PSN. One such scheme is isarithmic control [DAVI 72] where tokens circulating in the network are used as permits for the nodes to send packets.

2.1.2 Buffer Management Schemes

Kamoun [1976] and Kermani [1977] discussed various schemes for sharing the B total buffers in a nodal buffer pool among R_B buffer classes. Of these, B_s buffers are sharable among all the buffer classes. They described four buffer management schemes (BMS) which can be used with any number of buffer classes:

1. CS - complete sharing
2. CP - complete partitioning
3. SMXQ - sharing with maximum queue length
4. SMA - sharing with minimum allocation.

In complete sharing, an arriving packet is accepted if any buffer space is available.

In complete partitioning, the entire buffer space is permanently partitioned among the buffer classes. In other words, no sharing is provided.

In sharing with maximum queue length, all buffers are shared but the maximum number of buffers assigned to each buffer class is limited.

In sharing with minimum allocation, each buffer class has a fixed number of buffers assigned to it. The remaining buffers are shared among all the buffer classes.

For fairly balanced systems, CS obtains a better performance (in terms of a smaller blocking probability) than CP. However, for highly asymmetrical input systems, CS favours those buffer classes associated with high input rates.

For the more restricted case of two buffer classes, two schemes have been developed:

1. LR - the Lam-Reiser "input buffer limit" scheme [LAM 79]
2. SS - Saad-Schwartz "free buffer allocation" scheme [SCHW 79].

In the Lam-Reiser scheme, a_t buffers are permanently assigned for transit traffic packets. The remaining buffers are shared.

In the Saad-Schwartz scheme, operation is separated into two regions: A and B (see Fig. 5). In region A, buffers are shared. In region B, only transit packets are permitted to enter the node.

Fig. 5 shows the state space for the above schemes for the case of nodal traffic partitioned into input traffic (n_i) and transit traffic (n_t). Input traffic is generated locally at the node. Transit traffic is brought into the node from the network. a_r (with r = i or t) represents the maximum number of buffers allocated to the packets of buffer-class r.

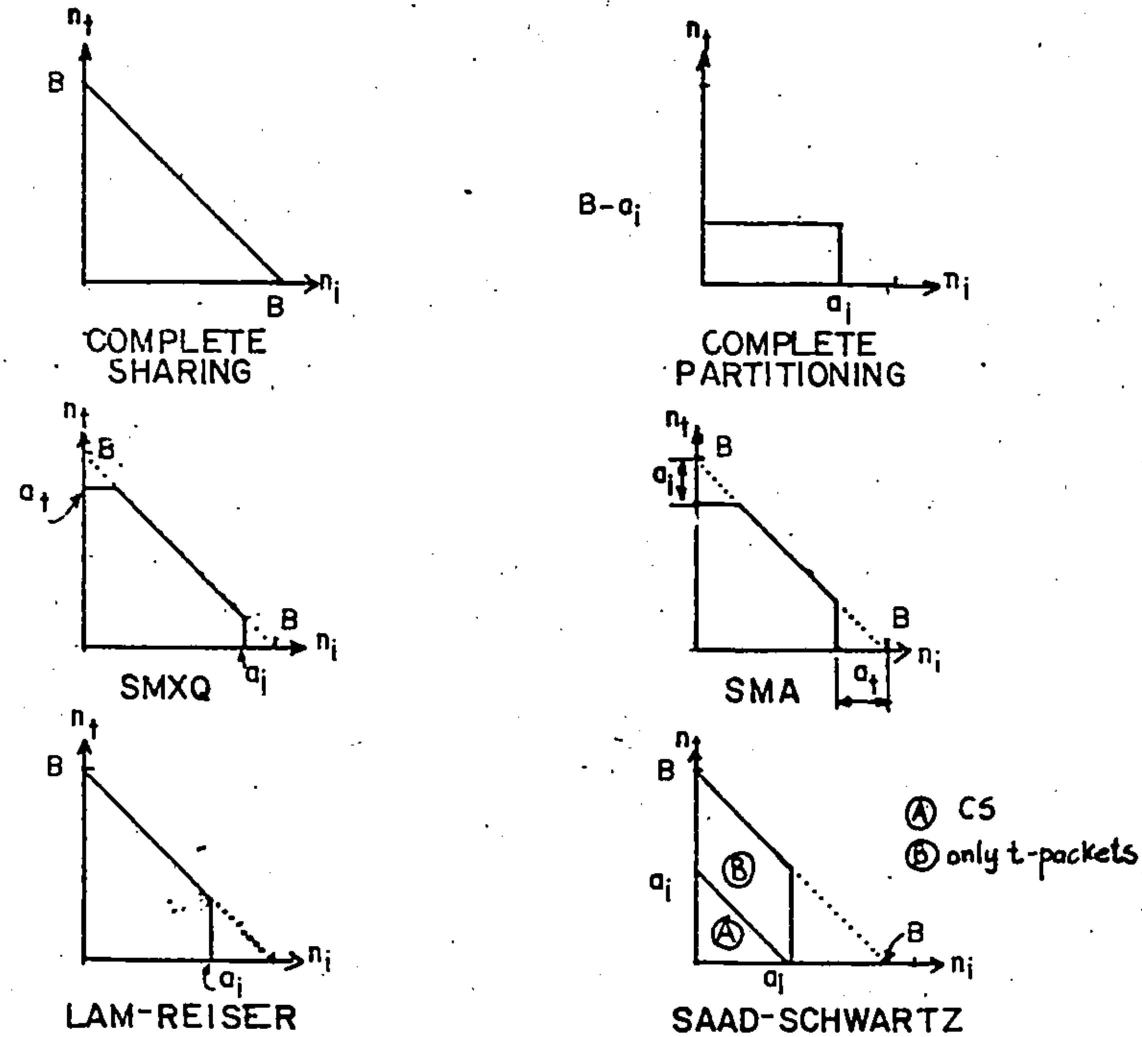


Figure 5: Common Buffer Management Schemes for 2 Buffer Classes.

Table 1 lists analytical expressions for the aforementioned schemes.

Numerical evaluation of various buffer management schemes have been reported for a single node [KAMO 76],[KERM 77]. However, there has been no reported numerical evaluation of BMS's in nodes imbedded within a network.

Complete sharing (CS)

$$F_1 = \{ \underline{n} \mid 0 \le n_r \le B, \quad r=1,2,\ldots,R_B \}$$

Complete partitioning (CP)

$$F_2 = \{ \underline{n} \mid 0 \le n_r \le a_r, \quad r=1,2,\ldots,R_B, \ \sum_{r=1}^{R_B} a_r = B \}$$

Sharing with maximum queue length (SMXQ)

$$F_3 = \{ \underline{n} \mid \sum_{r=1}^{R_B} n_r \le B; \quad 0 \le n_r \le b_r, \quad r=1,2,\ldots,R_B \}$$

Sharing with minimum allocation (SMA)

$$F_4 = \{ \underline{n} \mid \sum_{r=1}^{R} \sup(0, n_r - a_r) \le B; \ 0 \le n_r \le B_s + a_r, \quad r=1,2,\ldots,R_B \}$$

Lam-Reiser (LR)

$$F_{LR} = \{ \underline{n} \mid 0 \le n_i \le a_i; \quad 0 \le n_i + n_t \le B \}$$

Saad-Schwartz (SS)

$$F_{SS} = \{ \underline{n} \mid \begin{array}{ll} \text{complete sharing} & \text{for } 0 \le n_i + n_t \le a_i, \\ \text{transit packets only} & \text{for } a_i \le n_i + n_t \le B \end{array} \}$$

NOTES

1. $\underline{n} = (n_1, n_2, \ldots, n_{R_B})$ where n_i = number of packets in buffer-class i
2. For all these schemes, the following relation holds: $\sum_{i=1}^{R_B} n_i \le B$

TABLE 1

Analytical Expressions for Various BMS State Spaces.

Chapter III

MODELLING OF PACKET-SWITCHING NETWORKS

Performance measures of packet-switching networks can be estimated by building a model of the network and evaluating it numerically. Such models may be approached:

1. analytically, or
2. by simulation.

Analytical models are preferred because they can give insight into the workings of the network, are often much faster to evaluate and explicitly deal with a known state, i.e. the model is made under assumptions of equilibrium or non-equilibrium.

Section 3.1 deals with analytical models and with queueing theory models in particular. Section 3.2 briefly discusses simulation modelling.

Numerous analytical models for studying PSN's and their constituent parts (under certain assumptions) exist:

- networks [GEOR 80b]
- virtual circuits [PENN 75]
- nodes [KERM 77]
- protocols [BOCH 80].

This thesis deals only with network models.

After building a network model, it should be verified that the network is stable (also called "well-defined"). For a network to be stable, it must have a unique non-null and l_1-bounded equilibrium solution [REIS 76a]. A necessary condition for stability is [REIS 76b, p.20]:

$$\sum_{j=1}^{M} p_{ij} = 1 \quad \text{for all queues, } i=1,2,\ldots,M \tag{3.0A}$$

where: p_{ij} = probability of a customer leaving queue i for queue j.

Unstable networks may have a number of behaviours [REIS 76b,p.20-21]:

1. They may not attain equilibrium, e.g. a single server queue with external arrivals, infinite storage, and the arrival rate exceeding the service rate.
2. They may have complex transient behaviour before reaching equilibrium.
3. They may have several mutually exclusive equilibrium conditions, e.g. deadlocks.

3.1 <u>Analytical Models of PSN's</u>

Analytical models for PSN's have been formed with several mathematical techniques:

- queueing theory [BCMP 75]
- Petri nets [PETE 80]
- catastrophe theory [OLDE 79].

Of these, queueing theory models are the predominant means of studying PSN's.

The structure of Petri nets is described by its places, transitions, input function and output function. The Petri net automaton describes the movement of "tokens" through the net. One problem with Petri nets is that they are normally used to deal with problems of a much smaller state space size than can be handled with queueing theory. Furthermore, the numerical solution of Petri nets is not a well-developed field.

Catastrophe theory models are scarce in the literature, largely due to their mathematical sophistication and their lack of usefulness in dealing with practical problems [OLDE 79].

This thesis will concern itself only with queueing theory models. Such models describe the store-and-forward nature of packet-switching networks as a network of waiting-lines called queues. However, before examining particular models, a brief digression to the properties of algorithms which will solve these models will be made.

The algorithm used to solve the problem should be analyzed to answer questions about its properties [HOPC 79]:

1. Decidability - does the problem have answers for all instances (particular cases)?
2. Tractibility - can it be solved only in exponential time/space or is a polynomial time/space solution possible, i.e. is the problem very difficult to solve?
3. Computability - can the algorithm be evaluated on a computer?
4. Compututational complexity - how fast is the algorithm and how much space does it need?
5. If computable, numerical stability - does the algorithm behave "well" as it converges?

Only the last three properties will concern this thesis from a theoretical point of view. The computational space/time complexity of the algorithms will be discussed briefly.

3.1.1 Overview of the Theory for Queueing Networks

Solution of a queueing problem requires the computation of certain network statistics. These statistics and the performance measures can be usually determined as linear combinations of the state probabilities [GEOR 79]. The performance measures of interest to this thesis are:

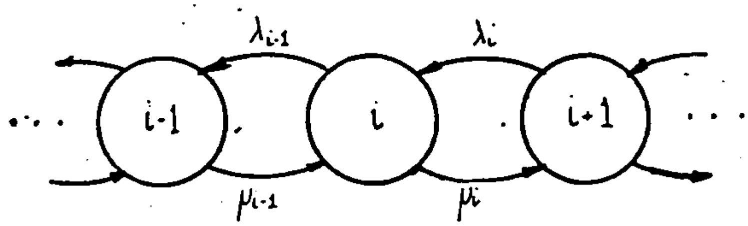
1. (mean) throughput for the network (T) and for each packet-class (T_r)
2. (mean) delay for the network (D) and for each packet-class (D_r)
3. server utilization by service center (U_m)
4. mean number of packets in the network (m) and for each packet-class (m_r)
5. packet-class blocking probabilities (pb[r])
6. nodal blocking probabilities for input (pbi[n]) and transit (pbt[n]) packets.

The random nature of the demands placed on packet-switching networks can be modelled as a stochastic process. Because the nodes in a network store and forward packets, a PSN can be considered as a network of queues which can be studied using queueing theory. However, the simplifications introduced by also assuming that the network's operation can be modelled as a discrete state-space Markov process ("Markov chain") permits a mathematically tractable solution.

The most commonly studied queueing networks are those that can be modelled as a special class of Markov chains called "birth-death" chains [KLEI 75]. Markov chain models of packet-switching networks have the following attributes:

1. Packets (customers) may enter the network through a source node and leave through a sink node. Packets are separated into a finite number ("R") of classes. A transition ("routing") matrix P = [P(i,r;j,s)] specifies the probabability of a class r packet at service center i moving to service center j and changing class membership to s. P is the underlying transition matrix of a Markov chain whose states are defined by (i,r), and which is decomposable into "m" ergodic subchains ("closed communicating classes"). Hereafter, it is assumed that m = R.
2. Queues provide storage ("buffers") for packets. Packets are selected for service from the queue by a "queueing discipline" and place work demands on the service centers ("servers"). Servers process the packets at a given service rate, μ . Each packet belongs to a single class while awaiting and receiving service at a service center.
3. A set of flow control and routing procedures regulate the flow of the packets.

The traditional solution procedure for queueing networks is to find the vector steady-state ("equilibrium") system state probability distribution ,P($\underline{S}$), by formulating a set of algebraic equations (called "balance equations") and solving them. There are two types of balance equations (Fig. 6):



$$p_{i-1}\,\mu_{i-1} + p_{i+1}\,\lambda_i = p_i(\lambda_{i-1} + \mu_i)$$

(A)

$$p_i\,\mu_i = p_{i+1}\lambda_i$$

(B)

Figure 6: Examples of Balance Equations for a Simple 1-dimensional Markov Chain.
(a) a global balance equation
(b) a local balance equation

1. global balance equations, where the rate of transitions into a state i = rate of transitions out of state i
2. local balance equations.

The global balance equations can be partitioned into a set of local balance equations [LAM 77,p.375]. Therefore, the set of solutions to the local balance equations also satisfies the global balance equations. Local balance is a sufficient but not necessary condition for global balance [BCMP 75].

More recent methods for finding the above performance measures fall into two groups:

1. direct methods, i.e. by directly computing the statistics of interest
2. indirect methods, i.e. by first computing the state probabilities and from these, the statistics of interest.

An example of a direct method is mean-value analysis (MVA) which is discussed in section 3.1.3.2. In MVA, the statistics of interest are directly computed from recursive relations.

Most methods are of the indirect type, an example being the convolution methods discussed in section 3.1.3.1. In these methods, the equilibrium system state probability distribution is computed, and from these data the desired statistics are computed. Todate, most of the work on networks has been with product-form solutions, i.e. networks with a state probability in the form [LAM 77, p.375]:

$$P(\underline{S} = (X_1, X_2, \ldots, X_M)) = C\, d(\underline{S}) \prod_{i=1}^{M} f_i(X_i) \qquad (3.1.1A)$$

where: X_i represents the conditions at service center i,
C is a normalization constant needed to make the probabilities sum to 1,
$d(\underline{S})$ is a function of the number of customers in the system,
$f_i(X_i)$ is a function that depends on the type of service center,
M is the number of service centers in the network.

Jackson [1963] first showed that queueing networks with exponential servers have a solution of the balance equations in the form of a product of simple terms. The class of networks having product-form has been greatly extended [BCMP 75], [LAM 77], [KELL 79].

The solution space of networks is often very large. Methods have been found which can reduce the size of the solution space, or at least simplify the analysis by approximation. For example,

1. aggregation of states [BCMP 75]
2. diffusion approximation [KOBA 78, p.211]
3. aggregation of variables technique for decomposable networks [COUR 77].

Aggregation of states reduces the state space by lumping together states. For example, Baskett et al. [1975] defined an aggregate state with the number of packets of each class in each service center.

The diffusion approximation is based on the theory of diffusion processes and is successful in dealing with open networks under heavy traffic. As its use in dealing with closed networks is limited, it does not have much use in the modelling of PSN's with flow control.

The aggregation of variables technique is based on the "decomposability" of the underlying transition matrix of the network. That is, networks in which strongly coupled states can be grouped together. This technique yields exact results for closed product-form networks [COUR 78].

Many important computer and computer-network performance models do not have exact solutions todate, largely due to their inability to meet the constraints of product-form solutions. Network characteristics which violate product-form include:

1. simultaneous resource possession, e.g. a job holding both processor and memory
2. certain priority disciplines, e.g. round robin
3. general service time distributions at FCFS (first-come first-served) queues
4. simultaneous job activities ("parallelism"), e.g. the inability to handle messages which are sent as a number of packets
5. bounds on queue lengths ("blocking").

Although most networks with blocking violate product-form, Pittel [1979] has proven that some networks with blocking exhibit product-form solutions. These networks must be reversible [KELL 79, p.5]. The necessary and sufficient condition for reversibility is that the probability flow between adjacent states is balanced.

Queueing networks can be separated into three groups depending on the network population constraints:

1. open networks
2. closed networks
3. semi-closed networks [LAM 77].

In open networks, customers are drawn from an infinite population (e.g. a Poisson source), travel through the network, and depart at a sink, i.e. external arrivals and departures can occur freely.

The population of customers in closed networks is finite. The number of customers in each subchain in the network is fixed.

In semi-closed networks, the source generates customers which enter the network, travel through the network, and depart at a sink (as in an open network). However at any time, the number of customers in a given subchain is bounded:

$$K_{i-} \leq K_i \leq K_{i+} \tag{3.1.1B}$$

where: K_i is the number of packets in subchain i,

K_{i-}, K_{i+} are the lower and upper bounds (respectively) of the subchain's population.

In the remainder of this chapter, networks with M queues and R subchains are considered.

3.1.2 Solution of Open Networks

Open networks are computationally simple to solve. Using simplifying assumptions (e.g. customers do not change class membership), they can be solved by a separation of variables technique [REIS 76a, p.3]:

1. Using routing and arrival rates only, solve the flow problem of determining the throughputs (T_{mr} - those of class r packets in queue m).
   This involves solving R systems , each having M linear equations in M unknowns.
2. Isolate each queue (m) from the network, and compute the equilibrium marginal queue distribution $p_m(k_m)$ subject to the throughputs above.
3. Obtain the joint distribution as a product of the M marginal distributions, $p_m(k_m)$:

$$P(\underline{S}) = \prod_{m=1}^{M} p_m(\underline{k_m}) \qquad (3.1.2A)$$

3.1.3 Solution of Closed and Semi-closed networks

Closed [REIS 76a] and semi-closed [GEOR 79] networks are computationally much more complex. Semi-closed networks are important because they can be used to model PSN's with window end-to-end flow control by setting $K_{i-} = 0$ and $K_{i+} = w_i$ for all subchains i =1,2,...,R.. Typically, their solution algorithms grow rapidly with the number of subchains, requiring [GEOR 79]:

$$O(RM(\sum_{i=1}^{R} w_i)(\prod_{i=1}^{R} w_i)) \qquad (3.1.3A)$$

operations for networks with window flow control.

Let the population vector for the R closed subchains be represented by $\underline{y} = (y_1, y_2, \ldots, y_R)$. These networks are usually solved by computing normalization constants, $g(\underline{y})$, for all networks bounded by the expression eq. (3.1.1B).

The approach is as follows [REIS 76a], [GEOR 79]:

1. Compute $g(\underline{y})$ for $K_{i-} \leq y_i \leq K_{i+}$, for i =1,2,..., R.
2. Compute the marginal queue probabilities, $p_m(\underline{y})$ from $g(\underline{y})$ (and for semi-closed networks, from the arrival rates as well).
3. Compute the network statistics from $p_m(\underline{y})$, e.g. the utilization of service center m:

$$U_m = 1 - p_m(\underline{y} = 0) \qquad (3.1.3B)$$

$g(\underline{y})$ can be obtained from its generating function (gf), $G(\underline{Z})$, [REIS 76a]

$$G(\underline{Z}) = \prod_{m=1}^{M} C_m(\underline{\rho_m} \cdot \underline{Z}) \qquad (3.1.3C)$$

where: $C_m(\rho_m \cdot \underline{Z}) = p^*(\underline{Z})$, the starred quantity referring to a quantity not divided by a normalization constant.

There exist two practical methods for computing $g(\underline{y})$ [REIS 76a]:

1. Inversion of the entire generating function $G(\underline{Z})$.
2. Convolution of the M known inversions of the capacity functions $C_m(\rho_m \cdot \underline{Z})$

   (The capacity function is a function dependent on the queueing discipline at the service center.)

Method 1 requires a complex integration and is not well suited for numerical evaluation. The general problem can by restricted in order to permit $g(\underline{y})$ to be computed through a partial fraction expansion of $G(\underline{Z})$. However, such computational schemes are numerically unstable since they involve summation of terms with alternating signs [REIS 76a, p.12].

Method 2 provides a class of algorithms known as convolution methods, the most famous being the Reiser-Kobayashi algorithm [REIS 76a], [REIS 80b] discussed in section 3.1.3.1.

The Reiser-Kobayashi convolution algorithm and Mean-value Analysis are compared in terms of computational complexity in Table 2. From this point of view, they are similar in terms of time, but the convolution algorithm is slightly superior in terms of space requirements.

	MVA [REIS 80a]	Convolution [REIS 76a]
SPACE	$M\ K_1 \ldots K_R$	$2\ K_1 \ldots K_R$
TIME (Order of operations)	$K_1 \ldots K_R\ (\ 4\ R\ M\ )$	$K_1 \ldots K_R\ (\ 4\ R\ M\ )$

R closed routing chains
M service centers
$\underline{K}$ is the population vector for the r subchains.

TABLE 2

Computational Complexity of MVA and Convolution Algorithms.

For the purpose of this thesis the most general solution for product-form queueing networks is that given by [BCMP 75], and later extended by Lam [1979] to include state-dependent lost and triggered arrivals. Georganas [1979] gave a numerical algorithm for the solution of Lam's class of networks.

Table 3 lists the constraints for a network to have a solution under the BCMP model. The four types of service centers fall into a class of queueing disciplines called "work-conserving":

1. Each packet's work demand is independent of the queueing discipline.
2. The queueing discipline assumes no knowledge about the work demands or arrival time distributions of individual packets.
3. When the service center is idle, no packet waits for service.

3.1.3.1 Convolution methods [BUZE 73],[BRUE 80]

The convolution methods permit computation of the normalization constants $g(\underline{y})$ by a recursive formula which is a convolution of functions of a discrete domain.

The capacity function for each service center, $c_m(\underline{y})$, is computed from its generating function (gf), $c_m(\rho \cdot \underline{z})$.

Structure

N service centers (queues)
R classes of customers with transition matrix
P = [P (i,r);(j,s)] defining a Markov chain.
Customers travel from service center i to j, changing class from r to s.
The chain is decomposable into "m" ergodic subchains.

The network may be open, closed or mixed.
For open networks, arrivals may be state dependent.

Service centers

Type 1(*): FCFS (first-come first served) with all customers having the same (exponential) service time distribution at the service center.
Type 2: PS (processor sharing)
Type 3: IS (infinite server i.e. no queueing delay)
Type 4: LCFSPR (last-come first served preemptive resume)

Sources for open systems

Type 1: Total arrival process to the network is Poisson with mean rate dependent on the total number of customers in the network.
Type 2(*): m Poisson streams correponding to the m ergodic chains.

TABLE 3

BCMP Model Assumptions [BCMP 75].

The symbols are those given in the BCMP paper, and not those used in the thesis. The types marked (*) are relevant to the model developed in chapter 4.

$g(\underline{y})$, the inverse of its gf, $G(\underline{Z})$, can be obtained by M convolutions (one for each service center) as:

$$g(\underline{Y}) = c_1 \circledast c_2 \circledast \ldots \circledast c_M \qquad \text{at } \underline{y} = \underline{w} \qquad (3.1.3D)$$

The convolution algorithm has the desirable property that blocking probabilities can be estimated because the individual state probabilities can easily be computed.

However, the convolution algorithm suffers from two major problems:

1. overflow/underflow problems during computation
2. exponentially increasing number of arithmetic operations.

The first problem can be reduced due to a scaling algorithm developed by Lam [1982]. Such scaling factors can be introduced into the workload vector, τ .

3.1.3.2 Mean-value Analysis [REIS 80a]

Mean value analysis is a recursive algorithm for computing:

1. mean queue sizes
2. mean waiting times
3. throughputs

in closed multi-subchain queueing networks having product-form solutions.

The concept behind MVA is the recursive relation (here for the case of a FCFS service center):

$$w_{r,l}(\underline{K}) = \tau_{r,l} \, (1 + n_l(\underline{K} - \underline{e}_r)) \qquad (3.1.3E)$$

It states that in equilibrium, a customer (packet) of subchain r, arriving at service center l with the system in state $\underline{K}$ sees a waiting time ($w_{r,l}$) of the system in state ($\underline{K} - \underline{e}_r$). $\tau_{r,l}$ is the mean service demand of a customer in subchain r at service center l.

MVA permits heuristic extensions by having a physically meaningful interpretation. These extensions allow the approximate solution of networks with a very large number of closed subchains [REIS 80a].

MVA avoids the scaling problem of the convolution algorithm by computing the statistics directly.

MVA has the disadvantage that the problem of incorporating blocking probabilities into the algorithm has no solution reported in the literature todate.

3.2 Simulation Modelling of PSN's [KOBA 78,pp.284-297]

Because simulation is not germane to this thesis, its use in modelling PSN's will be discussed only briefly.

Simulation involves abstracting the key features of the network in order to study the desired performance measures. The numerical solution involves three phases:

1. Initialization of the model to the initial network state. Usually, this is an empty network.
2. Driving the network to an equilibrium state (assuming that the steady-state statistics are desired).
3. One or more simulation runs to obtain samples in order to compute confidence intervals.

Step 2 is a difficult problem, as it is not easy to verify that an arbitrary network has reached equilibrium.

Step 3 usually requires several runs to obtain enough samples to compute estimates of the performance measures (mean and variance) within a 90 percent confidence interval. The samples are usually assumed to fall on a student-T distribution [KOBA 78].

One of the early automated methods of computing the numerical solution to queueing networks was an electromechanical analog computer called Queueiac [DUNN 56]. Dunn et al. reported that in in networks "with multiply connected queues ... the hope for solution by strictly analytical means is lost". Queueing network theory seems to have made advances from that time.

Chapter IV

PROBLEM DESCRIPTION, ASSUMPTIONS AND ANALYSIS

The analytical modelling of PSN's to study the effect of congestion schemes has no exact queueing theory solution todate. This is due to the inability of current queueing models to handle blocking, except for very small size problems. However, with some blocking approximations, models of networks with blocking can be made to fall into the class of tractable queueing problems.

The problem to be investigated is the modelling and numerical solution of a packet-switching network (Fig. 2) with two types of flow control:

1. Local flow control by imposing a buffer management scheme for two classes of packets , i.e. input traffic packets and transit traffic packets. This introduces two vectors of nodal blocking probabilities: <u>pbi</u> for input traffic packets, and <u>pbt</u> for transit traffic packets..
2. End-to-end flow control by the window mechanism.

The quantities of interest in the numerical solution are throughputs (T, T_r), delays (D, D_r) and blocking probabilities (<u>pbi</u>, <u>pbt</u>).

The amount of literature in this area is small.

Pennotti and Schwartz [1975] analyzed a tandem network with local (complete sharing of buffers) and window flow control.

The analysis of the more general "homogeneous" network (a network in which the blocking probabilities for all nodes is the same) with input buffer limits for input and transit traffic was investigated by Lam and Reiser [1979]. They gave a rule of thumb for selecting the input buffer limits for a node:

$$\frac{N_I}{N_T} < \alpha_o = \frac{\text{input throughput of the node}}{\text{total throughput of the node}} \qquad (4.0A)$$

Georganas [1980a] extended their results for homogenous networks with input buffer limits for three classes of traffic (see Fig. 7):

1. input
2. transit
3. exit.

Georganas [1980b] then found a numerical algorithm for analyzing a heterogeneous packet-switching network (a network in which the blocking probabilities for all nodes is different) with three levels of flow control:

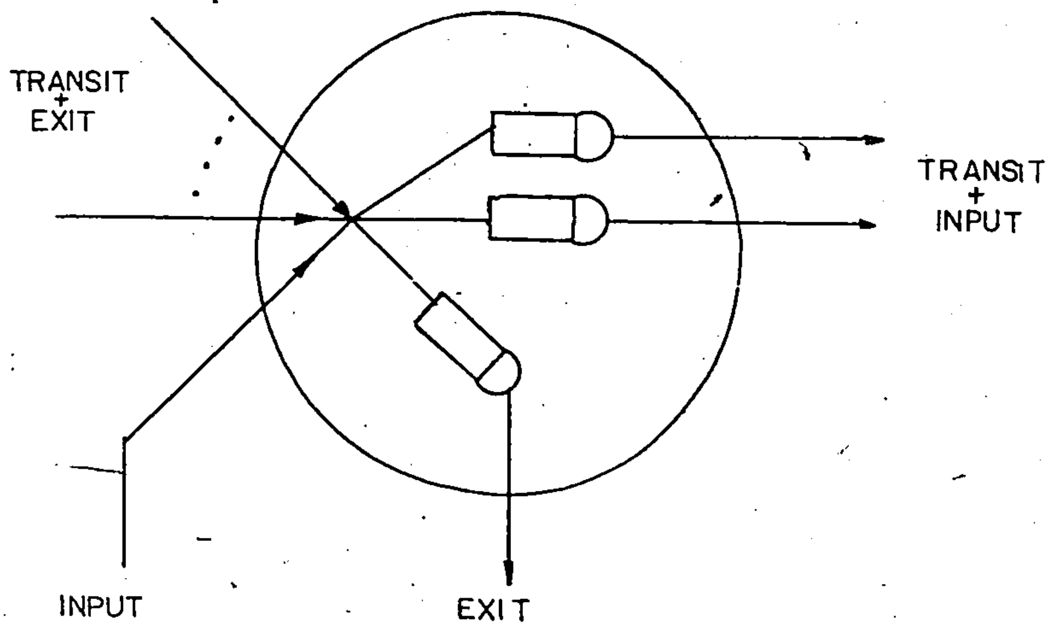


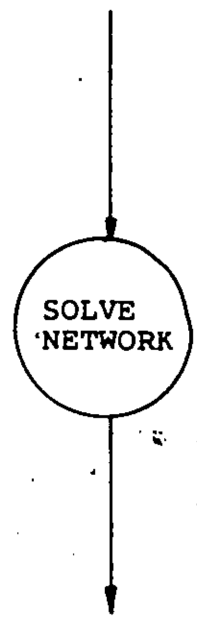
Figure 7: The Division of Nodal Traffic. Nodal traffic can be separated into 3 classes: input, transit and exit.

1. local flow control with complete sharing of buffers
2. end-to-end flow control with the window mechanism
3. global flow control with the isarithmic mechanism.

Fig. 8 shows the overall data flow diagram for the solution of the queueing network. The solution algorithm is discussed in the following sections.

NETWORK DESCRIPTION PARAMETERS

network parameters: N; R; Queues (number of queues)
mean packet length
nodal parameters: BMS; number of buffers
other BMS parameters
queue parameters: source and destination nodes
channel capacity; routing matrix
sources: source and destination nodes, window size,
packet arrival rate



T, T_r - throughputs
D, D_r - delays
m, m_r - mean number of packets
pb[r] - class-r blocking probabilities
pbi, pbt - input and transit packet blocking probabilities

Figure 8: Data Flow Diagram for Solution of the Network.

4.1 Queueing Network Assumptions

The analysis presented here is largely based on the work of Georganas [1980b] and similar assumptions will be made about the network.

The packet-switching network will be assumed a semi-closed queueing network with a product form solution. The equilibrium state probabilities are assumed to exist and to be unique. The network's "customers" will be single-packet messages (hereafter called "packets"). These packets require no modification to their structure by the network, i.e. no packet assembly/disassembly, and their size includes the extra information usually added by the link level, e.g. address information. Each source will generate a different class of packets which will be routed through a particular subchain. Hence, the terms subchain and class refer to the same entity here (hereafter called "class" or "packet class"). Packets are not permitted to change class membership while inside the network.

The following network is considered:

N switching nodes each having B total buffers shared between two buffer classes (input and transit). The buffer management schemes must be coordinate convex (see section 4.2.4) with doubly-connected adjacent states, and may differ for each node. Each node has at most one input source feeding it.

- M error-free links ("queues") served by a FCFS queueing discipline and having channel capacities of C_i bps. Each link is served by a single-server service center.

- R packet classes, each with a different source node and having window sizes of $w_r : 1 \le r \le R$.
  - each source sends packets to the network with an exponential packet interarrival time, i.e. each source is Poisson. Further, each source feeds exactly one node.
  - all classes have a probabilistic routing matrix $\{ P^{(r)}_{i,j} : 1 \le r \le R, \quad 0 \le i,j \le N \}$ (The probability of a class r packet moving from node i to node j.) It is assumed that packets originating at a node (i) will not return to that node.

- Packet lengths (v) are drawn from an exponential distribution with mean $1/\mu$ [KLEI 64, p.50]:

$$p(v) = \mu e^{-\mu v} \qquad (4.1A)$$

  each time a packet is received at a node. This is called the "independence assumption".

Packets travelling through the network can be blocked in two ways:

1. Input packets are blocked from entry into node "i" with probability pbi[i].
2. Transit packets are blocked from entry into node "i" with probability pbt[i].

The future behaviour of packets which have been refused entry to a blocked node is described in section 4.2.

The differences with the Georganas model [1980b] are:

1. Isarithmic flow control is not included.
2. Buffer management schemes other than complete sharing are investigated.

The isarithmic mechanism is not included in the analysis for two reasons:

1. Its effectiveness and reliability have not been proven, given the complexity of the mechanism involved.
2. It is not used in many current networks.

4.1.1 <u>Validity of the Assumptions</u>

In a week long measurement of ARPANET traffic in 1973, Kleinrock found that each message contained an average of 1.12 packets [KLEI 76,v.II, p.459]. Therefore, the assumption that messages and packets can be considered equivalent from a network operation point of view is plausible if an ARPANET-type network is considered.

In selecting the packet interarrival time distribution, very few functions permit a tractable model to be formulated. Of these, the Poisson (exponential) process is the limit of superimposing a large number of independent renewal point processes. This theoretical result has been shown to be valid in empirical studies of computer-communication networks with a large number of users [REIS 82], [FUCH 70].

A renewal process is a stochastic point process with the lengths of the intervals between consecutive events independent and identically distributed. A Poisson process is a particular type of renewal process where the inter-event interval lengths are exponentially distributed [REIS 82].

The independence assumption avoids the assignment of a permanent length to each packet. Such an assignment would give rise to a dependency between the interarrival times and the lengths of adjacent packets. Independence of length greatly simplifies the solution procedure for a general network, and has been found (empirically) to be a reasonable approximation for networks of a moderate connectivity [KLEI 75,v.II,p.322].

Some of the more general assumptions used are not too realistic in current computer networks, but must be made in order to find an analytical model solvable with the current results of queueing theory :

1. Typical network traffic is a complex function of the time of day (Fig. 9) and can at best be considered a quasi-stationary Poisson process, i.e. a Poisson process with the arrival rate slowly changing with time. The distribution is more complicated in networks which span several time zones because peak hours do not coincide.
2. Queueing disciplines are not always FCFS, but are normally priority based. This is necessary to include such features as equitable fairness among competing users.
3. The nodal model and network congestion control mechanism are significantly more complex in real networks , e.g. DATAPAC [SPRO 81].

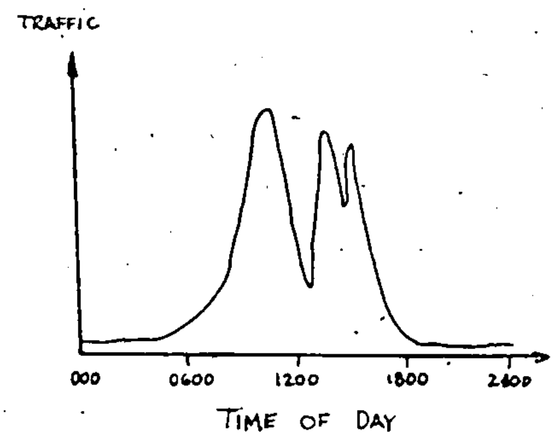


Figure 9: Traffic Function as a Time of Day. [CHOU 82]

4.2 Theoretical Analysis

Section 4.2 and its subsections develop an algorithm to solve the problem defined in section 4.1.

The computational algorithm (Fig. 10) is an iterative procedure requiring three basic steps:

1. Solution of the queueing network assuming known blocking probabilities pbi and pbt and only end-to-end window flow control.
2. Computation of pbi and pbt from the queue packet probabilities computed in step 1, by using an approximate model for the nodal buffer management scheme. The state space (F) is truncated by imposing a buffer management scheme, and the new state space (F') becomes the intersection of the state space of step 1 (F) and the space imposed by the buffer management scheme.
3. Update pbi and pbt for the next iteration according to some method of solving the system of non-linear equations.

Sections 4.2.1 - 4.2.5 describe the individual steps of the algorithm.

```
pbi = 0;  pbt = 0;     { initial values }
WHILE (pbi, pbt not converged)
        1.  Solve queueing network assuming
            (a) quasi-infinite buffers (Section 4.2.3)
            (b) given blocking probabilities:
            pbi and pbt.
            (See Fig. 16 for an expansion.)

        2.  Compute input and transit blocking probabilities
            pbi, pbt  (section 4.2.4).
            (See Fig. 17 for an expansion.)

        3.  Correct pbi, pbt for the next iteration, using
            some method of solving a system of non-linear
            algebraic equations.
            (See Appendix E.)

Compute other performance measures:
     throughput,  delay,  class blocking  probabilities
     (section 4.2.1).
```

Figure 10: Pseudo-code for the Numerical Solution of the Network.

Section 4.2.1 shows how to compute the desired performance measures: throughput, delay and class blocking probabilities.

Section 4.2.2 describes the queueing model of a node.

Section 4.2.3 discusses the model for packet retransmission at a blocked node.

Section 4.2.4 discusses the approximations used to compute nodal blocking probabilities from the state-space of the network computed assuming only window flow control.

Section 4.2.5 summarizes the entire computational algorithm.

4.2.1 Computing Performance Measures

This section derives expressions for the following quantities:

1. m- the mean number of packets in the network
2. pb[r]- the blocking probability for class r
3. T- the mean network throughput
4. D- the mean network delay

using the computed values of the input blocking probability (pbi) and the transit blocking probability (pbt). The expressions are based on the model of Georganas [GEOR 80b] for nodes with only one buffer class, and are approximations due to the modelling of the actual blocking mechanism.

The equilibrium probability of having $\underline{y}$ packets in the R classes is given by:

$$p(\underline{y}) = \begin{cases} C\, g^{*}(\underline{y}) \prod_{r=1}^{R} S_r^{y_r} & \underline{y} \in F \\ 0 & \text{otherwise} \end{cases} \qquad (4.2.1A)$$

where: y_r = the number of packets in class r,

F refers to the feasible state space for the problem

$$F = \{ \underline{y}=(y_1,y_2,\ldots,y_R) \mid 0 \le y_r \le w_r \text{ for } r=1,..,R \}$$

The mean number of class r packets is computed from the definition of the mean:

$$m_r = \sum_{\underline{y} \in F} y_r\, p(\underline{y}) \qquad (4.2.1B)$$

The mean number of packets in the network is:

$$m = \sum_{r=1}^{R} m_r \qquad (4.2.1C)$$

because the classes are independent.

The class r blocking probability (pb[r]) can be obtained by summing over the edge states of the state space, F.

$$pb[r] = \sum_{\substack{\underline{y} \in F \\ y_r = w_r}} p(\underline{y}) \tag{4.2.1D}$$

The mean throughput of class r is given by the arrival rate S_r reduced by:

1. The probability of blockage at the source node (s), pbi[s], because of the buffer management scheme.
2. The probability of class r being blocked, pb[r], because of the window end-to-end flow control constraint.

$$T_r = S_r \,(\, 1 - pbi[s] \,)(\, 1 - pb[r]) \tag{4.2.1E}$$

This assumes that BMS blocking and window blocking are independent.

The mean network throughput is simply the sum of the individual class throughputs:

$$T = \sum_{i=1}^{R} T_i \tag{4.2.1F}$$

The mean delay for class r can be computed using Little's theorem [KLEI 75, v.I, p.17]:

$$D_r = \frac{m_r}{T_r} \qquad (4.2.1G)$$

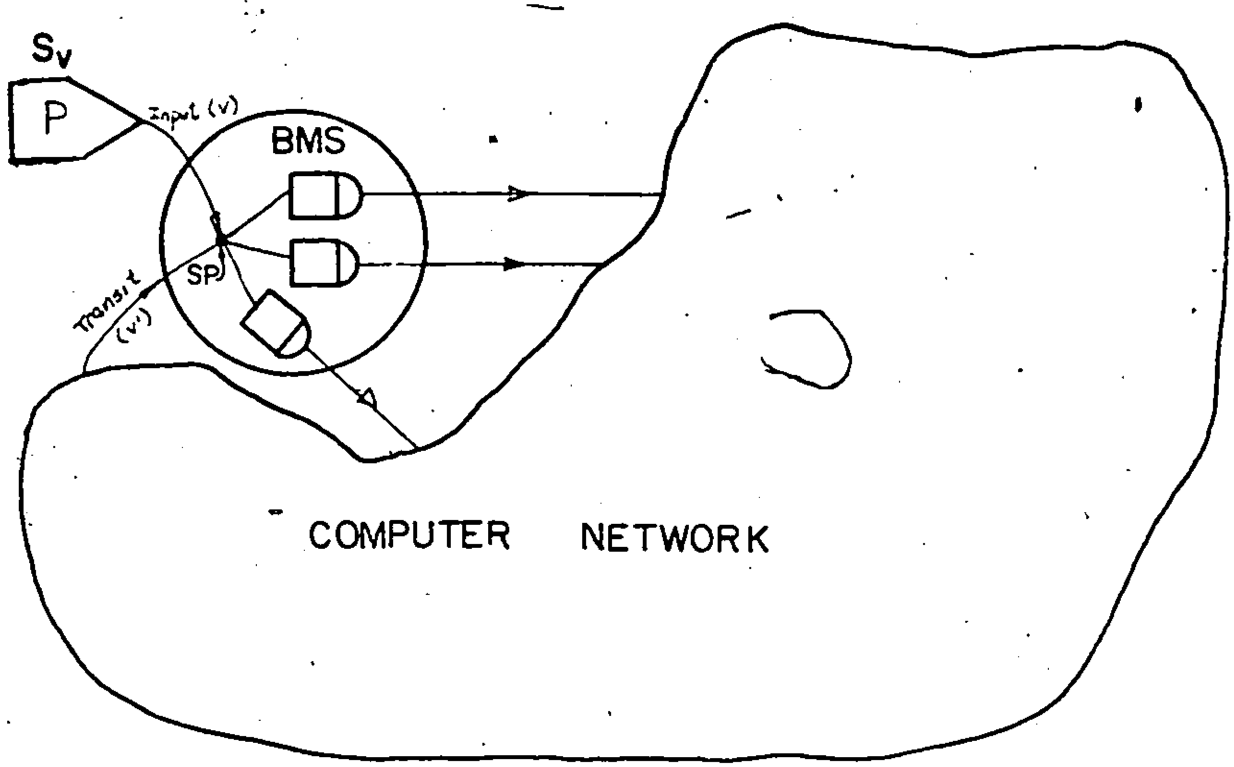
with the mean network delay expressed similarly:

$$D = \frac{m}{T} \qquad (4.2.1H)$$

4.2.2 Queueing model of a node

Each node may accept only transit traffic, only input traffic, or both input and transit traffic. In the two latter cases, a Poisson source of class V drives the node with an arrival rate of S_V pac/s (see Fig. 11).

The nodal traffic passes through a very fast switching processor (SP). Because of its high speed relative to the links, its queueing and service delays can be ignored.



SP = switching processor

NOTES:

1. For each node, the R packet classes are partitioned into 2 sets:
   (a) V - the class of packets being fed into the node by a Poisson source (if any), i.e. the "input packets" to the node.
   (b) V' - all other classes, i.e. the "transit packets" to the node.

2. The BMS is one of CS, CP, SMA, LR.

Figure 11: Queueing Model of a Node.

For each node, the packet probability distributions for each queue in the node are computed using the algorithm of Appendix C. These functions are numerically convolved to form the nodal packet probabilities for each node "n":

$$p_n(l_V, l_{V'}) = p_{q_1} * p_{q_2} * \ldots * p_{q_{q_i}} \quad (4.2.2A)$$

where: $\{ q_n , n=1,2,\ldots,q_i \}$ are the queues of node n,

l_V is the number of input packets in node n,

$l_{V'}$ is the number of transit packets in node n.

From this function, the nodal blocking probabilities are computed as described in section 4.2.4.

Each node can discriminate between the two buffer classes using the following buffer management schemes:

1. CS- complete sharing
2. CP- complete partitioning
3. SMA (same as SMXQ for two buffer classes) - sharing with minimum allocation
4. LR- Lam-Reiser.

Node 0 is a special node (called the "sink node") introduced to accept all traffic which has passed through the source-destination virtual circuit pairs. Its use is to model the window end-to-end flow control so that the semi-closed network can be analyzed as a closed network.

Therefore, pbi[0] = 0, pbt[0] = 0 (i.e. no packets are blocked from node 0) and $P_{i,0}^{(r)} = 1$ for i = all queues which feed the sink node (i.e. sink queues only send packets to the sink).

4.2.3 The Blocking Approximation and Visit Ratios

This section discusses the mechanism by which packets are blocked from nodes. Section 4.2.4 discusses blocking from a computational standpoint.

Because nodal blocking can not be incorporated into the current results of queueing theory, certain assumptions must be introduced in order to find approximations which would make the resulting model fit into the class of tractable queueing networks:

1. The finite buffers are approximated by quasi-infinite ones, which are able to to accept the entire packet population from all R classes.
2. Packets are considered blocked at a node if the buffers occupied exceed the constraints imposed by the nodal buffer management scheme. Blocked packets are retransmitted from the sender ad infinitum (see Fig. 12). In real networks, packets would be retransmitted only a fixed number of times, before being dropped from the network.

Approximation 2 is modelled by assuming fixed nodal blocking probabilities, whose values are given by either:

1. the initial guess, or
2. the computed blocking probabilities from the previous iteration (see section 4.2.4).

Because of blocking, and the resulting retransmission traffic to node j, the mean service time for channel i is increased from $1/(\mu C_i)$ to

$$\frac{1}{\mu_i} \triangleq \frac{1}{\mu C_i\ (1 - pbt[j])} \qquad (4.2.3A)$$

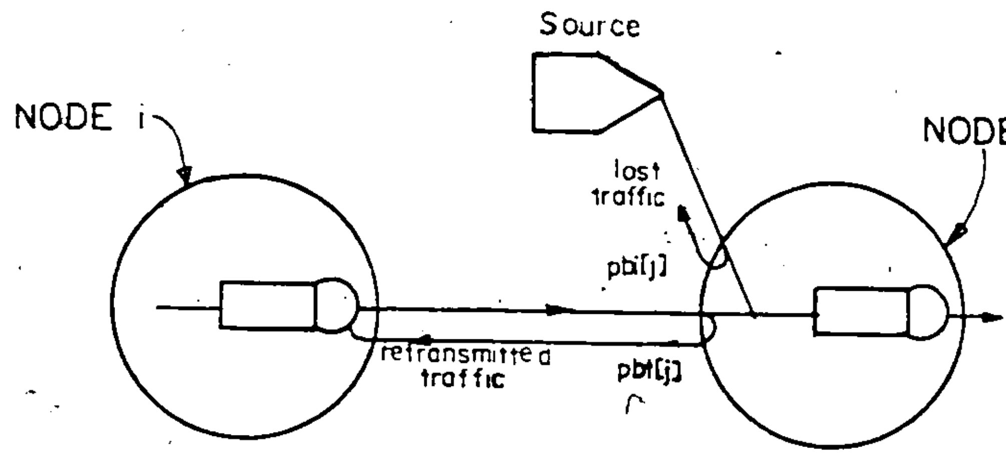
The packet arrival rate from the source to the entry node (s) for class i decreases to :

$$S_i(1 - pbi[s]) \qquad (4.2.3B)$$

4.2.3.1 <u>Visit ratios under blocking</u>

(a) Nodal visit ratios with no blocking

Let $v'^{(r)}_i$ be the number of times that class r packets visit node "i" while travelling through virtual circuit r in the network with no blocking ("unblocked visit ratios").



NOTES

1. Node j has input-packet blocking probability pbi[j] and transit-packet blocking probability pbt[j].

2. If x is the transmission time for a single packet of the mean packet length, then the mean transmission time for a packet (initial transmission + retransmission(s))
$= x + x(\ pbt[j]\ ) + x(\ pbt[j]\ )^2 + \ldots$
$= \dfrac{x}{1 - pbt[j]}$ s/pac

Figure 12: The Nodal Blocking Approximation for Internodal Traffic.

If there is no blocking, the computation of visit ratios is a forced-flow problem, requiring the solution of R systems of linear equations (one system for each class) [GEOR 80b].

$$v'^{(r)}_i = \lambda^{(r)}_i + \sum_{j=0}^{N} p^{(r)}_{j,i} \, v'^{(r)}_j$$

$$\text{where:} \quad \lambda^{(r)}_i = \begin{cases} 1 & \text{if } S^{(r)}_i \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.2.3C}$$

For each class, r, this is an (N + 1) by (N + 1) system of linear equations:

$$\begin{bmatrix} p^{(r)}_{0,0} - 1 & & \uparrow p^{(r)}_{i,j} \downarrow & \\ & p^{(r)}_{1,1} - 1 & & \\ \uparrow & & \ddots & \\ p^{(r)}_{i,j} \rightarrow & & & p^{(r)}_{n,n} - 1 \end{bmatrix} \begin{bmatrix} v'^{(r)}_0 \\ v'^{(r)}_1 \\ \vdots \\ v'^{(r)}_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdots \\ -1 \\ \cdots \\ 0 \end{bmatrix} \tag{4.2.3D}$$

where: -1 in right column is in the row corresponding to the source node for class r.

(b) Nodal visit ratios with blocking

The unblocked visit ratios $v'^{(r)}_i$ must be modified to include two effects in a network with blocking. These factors lead to an expression for the nodal visit ratios with blocking (eq. 4.2.3E):

1. loss of network packets due to blockage at the virtual circuit source node because of the buffer management scheme (numerator term)
2. retransmission of packets due to blockage at that node (i) (denominator term).

Therefore, for each class r the visit ratios with blocking are :

$$v_i^{(r)} = \begin{cases} 1 & \text{for i=source node for class-r} \\ v'^{(r)}_i \dfrac{1 - pbi[s]}{1 - pbt[i]} & \text{otherwise} \end{cases} \qquad (4.2.3E)$$

where: s is the source node for class r.

Once a packet enters a node, it is routed to an output queue according to the routing matrix, $P^{(r)}_{i,j}$:

$$v_k^{(r)} = v_i^{(r)} \, p_k^{(r)} \qquad (4.2.3F)$$

where : k represents the queue from node i to node j.

The workload vector can now be computed:

$$\tau_{mr} = \frac{v_k^{(r)}}{\mu_i} = \frac{v'^{(r)}_i \; \dfrac{1 - pbi[s]}{1 - pbt[i]} \; P_k^{(r)}}{C_i \; (1 - pbt[j])} \qquad (4.2.3G)$$

where: τ_{mr} is the workload associated with queue m and class r in a network with blocking.

4.2.4 Computing Nodal blocking probabilities

In this section, approximations for the blocking probabilities pbi and pbt are derived from the solution of the network assuming quasi-infinite buffers.

The following assumptions are made:

1. blocking probability for all packets (input and transit) pbn << 1
2. blocking probability for input packets pbi << 1
3. blocking probability for transit packets pbt << 1.

However, if these assumptions do not hold, some qualitative statements about the network can be made, if the numerical results are interpreted carefully. For example, bottleneck nodes will have a high blocking probability, and can thus be identified.

This analysis deals only with buffer management schemes which have a "coordinate convex" solution space (F), because networks with such schemes have product-form solutions [KELL 79], [KAUF 81]. If $\underline{n}$ is a point in F, then:

1. $\underline{n} \in F \rightarrow n_i \geq 0$ for $i=1,2,\ldots,R$

2. $(\underline{n} \in F)$ and $(n_i > 0) \rightarrow \underline{n_i}^- \in F$

(4.2.4A)

where: F is the set of allowed states for node n, and

$$\underline{n_i}^- = (n_1, \ldots, n_{i-1}, n_i-1, n_{i+1}, \ldots, n_R)$$

$$\underline{n_i}^+ = (n_1, \ldots, n_{i-1}, n_i+1, n_{i+1}, \ldots, n_R)$$

(4.2.4B)

Fig. 14 (panel A) shows the state space at a node assuming both window flow control and a buffer management scheme. The contour C shows the BMS constraint to the solution. (Only three line segments define the BMS in this example; however, any number is possible provided the space remains coordinate convex.)

The term "n-dimensional" buffer scheme refers to a BMS with n degrees of freedom in specifying its state space. For example, complete sharing is a one-dimensional scheme because the blocking probability depends on one variable - the total number of packets in the node. However, complete partitioning is a two-dimensional scheme (in the problem considered) because blocking must be considered in two variables - the number of input packets and the number of transit packets.

4.2.4.1 Approximating nodal blocking probabilities

The following approximation is used, based on the one-dimensional buffer scheme of Fig. 13.

In the this case, the probability that a node is blocked can be approximated by [GEOR 80b]:

1. computing the packet-probability distribution assuming quasi-infinite buffers, $p(\ell)$
2. replacing: $p(\ell = B) \leftarrow p(\ell \geq B)$.

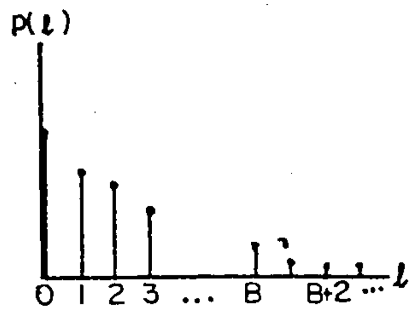
Fig. 13 shows this "mapping" of probabilities graphically.

In a two-dimensional buffer scheme, the state space is as shown in Fig. 14. Panel B is the state space required to compute blocking probabilities.

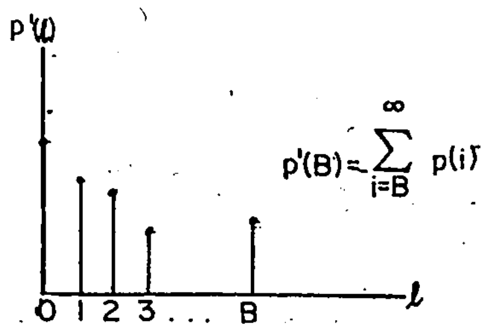
For this figure, the nodal blocking probability for all packets, pbn[n], can be computed by summing along the outer boundary of the state space:

$$pbn[n] = \sum_{\underline{n} \varepsilon B_i^+} p(\underline{n}) \qquad (4.2.4C)$$

where: $B_i^+ = \{ \underline{n} \varepsilon F' : \underline{n}_i^+ \notin F' \}$



(A) Infinite buffer



(B) Approximation of finite buffer packet-probability distribution.

Figure 13: Approximation of a Finite Buffer of Length B by an Infinite Buffer in a Node.

where: $B_i^+ = \{ \underline{n} \in F' : \underline{n}^+ \notin F' \}$

F' = the state space corresponding to the intersection of both window flow control and buffer management constraints.

A desirable mapping of the blocked states (region D in panel A) into the area A + C must have the following properties:

1. It gives the same distribution as the exact solution.
2. It is computationally fast.

In order to obtain an approximation, the following assumptions are made in the formation of an intuitive model:

1. The probability weight of the blocked area (D) is small with respect to that of the unblocked area A + C.
2. All the blocked states of D will be mapped onto the edge states, i.e. onto the contour C.

Assumption 1 states that the blocked states do not influence the unblocked state distribution much. This is equivalent to the pbn << 1 assumption that Lam [1979] and Georganas [1980b] used in their one-dimensional models.

Assumption 2 states that the blocked states will be distributed only on the edge states rather than throughout the area A + C. If the probability weight of area A is large with respect to that of D + C, this seems intuitively to be a good assumption. Georganas [1980b] made an equivalent assumption for the one-dimensional case.

Numerically, the mapping of region D into the unblocked area is as follows.

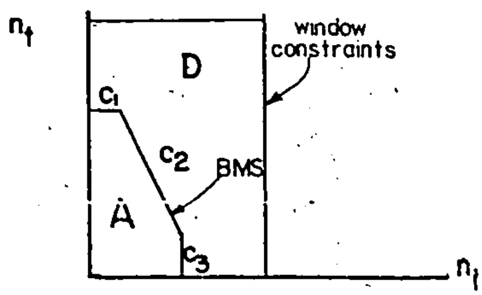
Let P_D = probability of being in the blocked area (region D of Fig. 14) found by summing the probability weights of the points in region D. Each point on C has its probability modified:

$$p(i,t) \leftarrow p(i,t) + \frac{P_D \cdot p(i,t)}{\sum_{\text{along } C} p(i,t)} = p(i,t)\left(1 + \frac{P_D}{\sum_{\text{along } C} p(i,t)}\right) \qquad (4.2.4D)$$

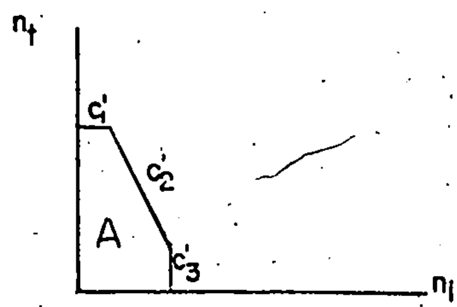
where: p(i,t) is the probability of having "i" input and "t" transit packets in node with $(i,t) \in C$.

This approximation has the undesirable property that it is not self-correcting, i.e. that probabilities will not distribute themselves properly along the contour (C) as the number of iterations in the algorithm increases.

Fig. 15 gives the edges over which to sum in order to compute the blocking probabilities for various buffer management schemes.



(A)



(B)

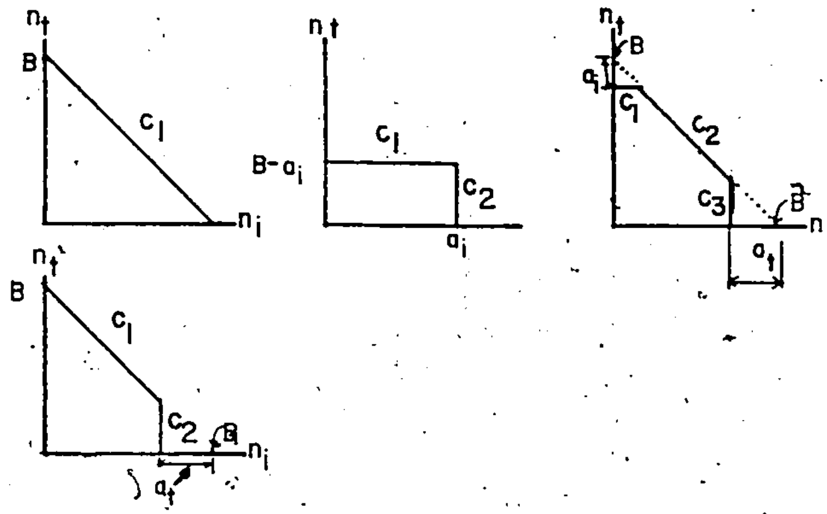
NOTES

1. The contour $C = c_1 - c_2 - c_3$ in panel A, represents the constraint on the nodal state space by the buffer management scheme.

2. The contour $C' = c'_1 - c'_2 - c'_3$ in panel B, represents the intesection of the buffer management constraint with the window flow control constraint.

3. "A" represents the area of the space restricted by the buffer management scheme.
   "D" represents the area in which both input and transit packets are blocked.

Figure 14: State Space for a Given Node.



BMS	pbi	pbt
CS	C_1	C_1
CP	C_2	C_1
SMA, SMXQ	$C_2 + C_3$	$C_1 + C_2$
LR	$C_1 + C_2$	C_1

<u>Notes</u>

1. The table gives the edges over which the probability weights are to be summed.
2. The exact state space for the node is assumed.

Figure 15: Computations for pbi,pbt for Various BMS's. The exact solution space for the node is assumed correct.

4.2.5 The Complete Algorithm

The results of the previous sections can now be incorporated into the algorithm of Fig. 10.

Note that the iteration of the WHILE loop is actually the solution of a system of implicit non-linear algebraic equations. Step 1 assumes a given set of blocking probabilities at the start of the iteration, i, i.e. $\underline{pbi}^{(i)}$ and $\underline{pbt}^{(i)}$. Step 2 computes new blocking probabilities (based on the computations of step 1), i.e. $\underline{pbi}^{(i+1)}$ and $\underline{pbt}^{(i+1)}$. Therefore the system is of the form:

$$\underline{pbi}^{(i+1)}[j] = \underline{f}_j\left(\underline{pbi}^{(i)}, \underline{pbt}^{(i)}\right)$$

$$\underline{pbt}^{(i+1)}[j] = \underline{g}_j\left(\underline{pbi}^{(i)}, \underline{pbt}^{(i)}\right) \qquad (4.2.5A)$$

where: i - is the iteration number,

f,g - are the implicit non-linear algebraic equations.

This system of equations can be solved by any number of algorithms for the numerical solution of non-linear algebraic equations, e.g.

1. the method of successive substitutions, and
2. Broyden's method (see Appendix E).

Convergence of the solution vector occurs when the l_2-norm is less than a given value ε :

$$|| \underline{pb}^{(i+1)} - \underline{pb}^{(i)} ||_2 \leq \varepsilon \qquad (4.2.5B)$$

where: $\underline{pb}^{(i)} = (\underline{pbi}^{(i)}, \underline{pbt}^{(i)})$

First, the normalization constant of the network is computed assuming a given set of blocking probabilities (Fig. 16). Next, the nodal packet-probabilities are computed using the packet-probabilities for each queue in the node, which are convolved together (Fig. 17).

Fig. 18 shows the flow chart for a single iteration of the algorithm.

INPUT

- pbi, pbt
- network description (Fig. 32)

OUTPUT

- C^{-1} } normalization constants
- $g^*(\underline{y})$ }

ALGORITHM

Compute visit ratios (eq. 4.2.3G);

Compute normalization constants (Appendix B);

Figure 16: Pseudo-code Expansion of Fig. 10 , step 1.

INPUT

- C^{-1}
- $g^*(\dot{y})$
- network description (Fig. 32)
- $\underline{pbi}^{(n)}$, $\underline{pbt}^{(n)}$

OUTPUT

- $\underline{pbi}^{(n+1)}$, $\underline{pbt}^{(n+1)}$

ALGORITHM

```
FOR n := 1 to N do begin { for all nodes }

   Compute marginal probabilities , p (l[V], l[V']),
   for the 1st queue
   in node n with the algorithm in Fig. 30.

   FOR j := other queues in node n do begin

      Compute marginal probabilities of queue j
      p_j ( l[V], l[V'] ),
      with the algorithm in Fig. 30.

      Perform the 2-dimensional convolution:

      p (l[V],l[V']) <- p (l[V],l[V']) * p_j(l[V],l[V']).

      end;

   Map p (l[V],l[V']) into the area restricted
   by the BMS (eq. 4.2.4D).

   Compute pbi, pbt by summing p ( l[V], l[V'] )  over
   the appropriate states as given by Fig. 15.

   end;
```

Figure 17: Pseudo-code Expansion of Fig. 10, step 2.

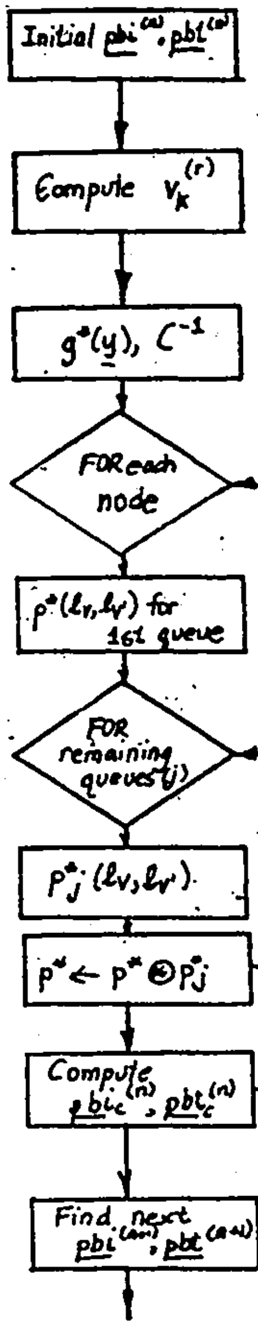


Figure 18: Flow Chart for a Single Iteration (n) of the Solution Algorithm.

Chapter V

NUMERICAL RESULTS

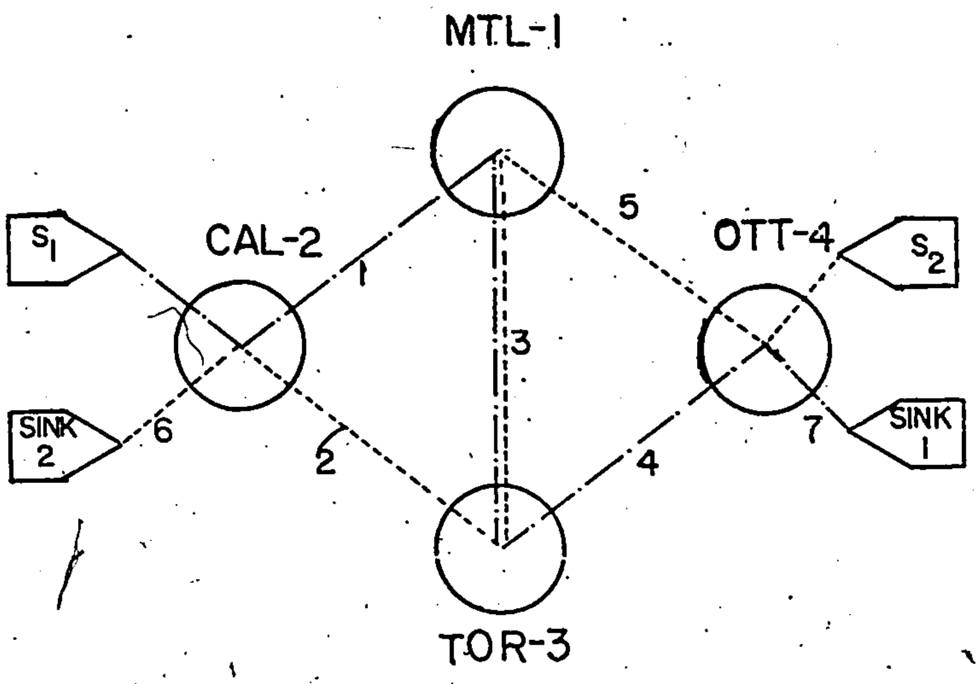
A PASCAL program called NAP1 (see Appendix F) was developed for applying the analysis of chapter 4 to study the computer network of Fig. 19. The network has the same structure as that in [GEOR 80b], but certain parameters have been changed:

1. Window sizes increased from 2 to 8.
2. Nodal buffer sizes increased from 2 to at least 6.

The network was studied to evaluate:

1. the use of Broyden's method in solving the implicit system of non-linear equations (section 5.1),
2. the effect of several buffer management schemes on nodes 4 (the source node for class 2) and 2 (the destination node for class 2) (section 5.2).

The network of Fig. 19 contains a subset of the backbone of the major Canadian X.25 network called DATAPAC [GEOR 80b]. DATAPAC links are 56kB and the sinks are assumed to accept data at about half that rate (25 kB). A fixed routing scheme is used. There are two packet classes with windows of 8 each.



N = 4 nodes

Queues = 7 queues
R = 2 classes

$1/\mu$ = mean packetlength
= 1000 bits

Node	BMS	Number buffers
1	CS	6
2	CS *	8
3	CS	6
4	CS *	6

Channels 1-5 are 56kB.
Channels 6-7 are 25kB.

Source	Window	Arrival rate (pac/s)
1	8	5
2	8	5,10,15,...,40

<u>Notes</u>

1. * - BMS of nodes 2 and 4 are varied during the study of section 5.2.

Figure 19: Network "NDG80" (a crude DATAPAC model).

BMS	a_i	a_t
CS	-	-
CP	3	-
SMA	2	2
LR	2	-

Notes

1. - means not applicable.
2. See section 2.1.2 for an explanation of the parameters.

TABLE 4

BMS Parameters for Nodes of Network NDG80.

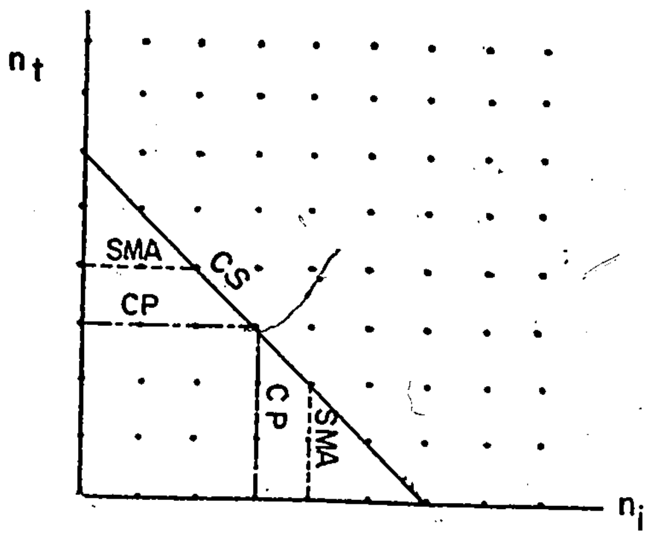


Figure 20: State Space for a Node with 6 Buffers. The line segments mark the various BMS constraints.

The source rates are chosen to permit certain effects to be demonstrated, e.g. channel saturation.

The nodal buffer sizes (6 and 8) are chosen so that the various BMS can be discriminated. With B < 6, the BMS constraints tend to be similar (see Fig. 20).

5.1 Study of Broyden's method [BROY 65] to solve the system of non-linear equations

Georganas [1980b] used the method of successive substitutions to solve the system of non-linear equations for the blocking probabilities. All nodes had a complete sharing BMS, i.e. had only one buffer class.

This numerical method proved acceptable in the one-dimensional case of a single buffer class, because the amount of computation to find pbi was small compared to the amount considered here (for the two-dimensional case).

Because the number of computations increased about an order of magnitude per iteration (due to the more complicated algorithm to compute queue distributions and the two-dimensional convolution to find the nodal distribution), a more efficient method for solving the system of non-linear equations was desired. Broyden's method [BROY 65] was chosen because it requires a smaller number of functional evaluations compared to other methods, and it is relatively easy to implement (see Appendix E).

The l_2-norm was reduced from 10^{-3} (in [GEOR 80b]) to 10^{-4} and the two methods of solution were compared (Fig. 21). Broyden's method showed the following improvements:

1. a fairly consistent number of iterations to convergence (about 15),
2. a large improvement in the maximum number of iterations to convergence (16 instead of 47).

One disadvantage of Broyden's method is that it requires 8 iterations (for the 8 blocking probabilities of this problem) to estimate the Jacobian numerically. Therefore, when convergence is "easy" the method of successive substitutions is much faster. However, such cases can not be determined a priori.

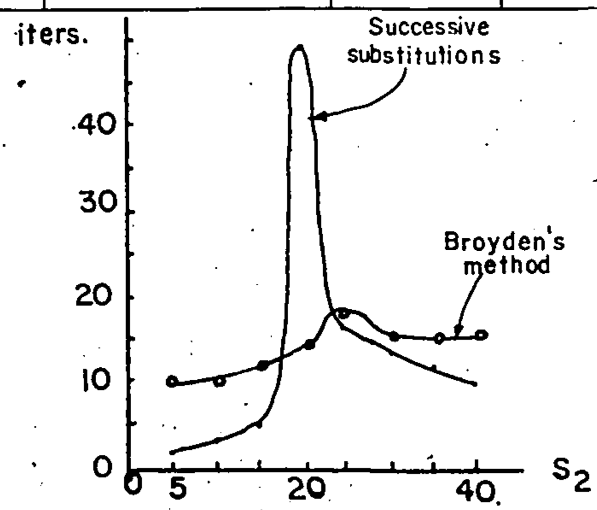
Broyden's method was thus found to be a superior method for the thesis problem.

5.2 Study of varying buffer management schemes

Fig. 22 shows some of the graphs which are useful in the performance evaluation of packet-switching networks.

Network information gives information about the general performance of the network, e.g. total throughput.

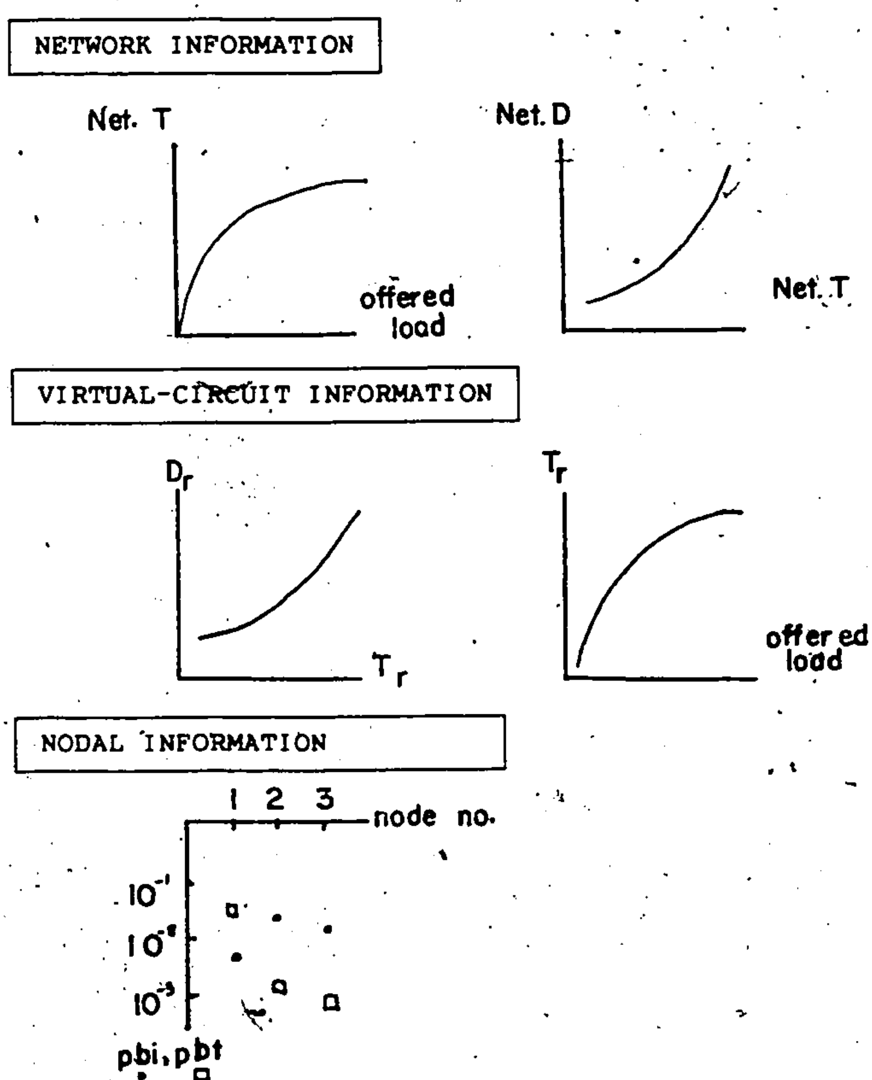
Arrival rate S[2]	Successive substitutions (1)	Broyden's method (2)	(3)
5	2	10	2
10	3	10	2
15	5	11	3
20	47	14	6
25	14	16	8
30	11	14	6
35	10	15	7
40	9	15	7
Mean CPU time per iteration	3.9s	3.9s	



NOTES

1. Column (2) includes the initial iterations required to compute the Jacobian. Column (3) excludes these iterations.

Figure 21: Comparison of Two Methods of Solving Non-linear Equations. The l_2-norm in both cases is 10^{-4}.



Notes

1. Throughput is the carried load.
2. Arrival rate(s) is(are) the offered load(s).

Figure 22: Important Performance Graphs in Evaluating PSN's.

Virtual-circuit information gives end-users estimates of the performance that they should expect.

Nodal blocking information gives an indication of how well the buffer management scheme in the node is performing.

The network of Fig. 19 will be studied by keeping the class 1 arrival rate fixed at 5 pac/s. Class 2 arrivals will be varied from 5 to 40 pac/s in order to evaluate the effects of increased load on the system.

Before numerical analysis of the network begins some important values will be computed.

The ideal throughput (Ideal T) is simply the offered load, i.e. all packets to the network are transferred through the network with no blockage. For this network,

$$\text{Ideal T} = S_1 + S_2 = 5 + S_2 \tag{5.2A}$$

The maximum carrying capacity (Max T) can be approximated by locating bottlenecks. For class 2, the slowest link is the exit link (number 6) at 25 kB. Hence,

$$\begin{aligned} \text{Max T} &= S_{1(max)} + \frac{25 \text{ kB/s}}{1 \text{ kB/pac}} \\ &= 30 \text{ pac/s} \end{aligned} \tag{5.2B}$$

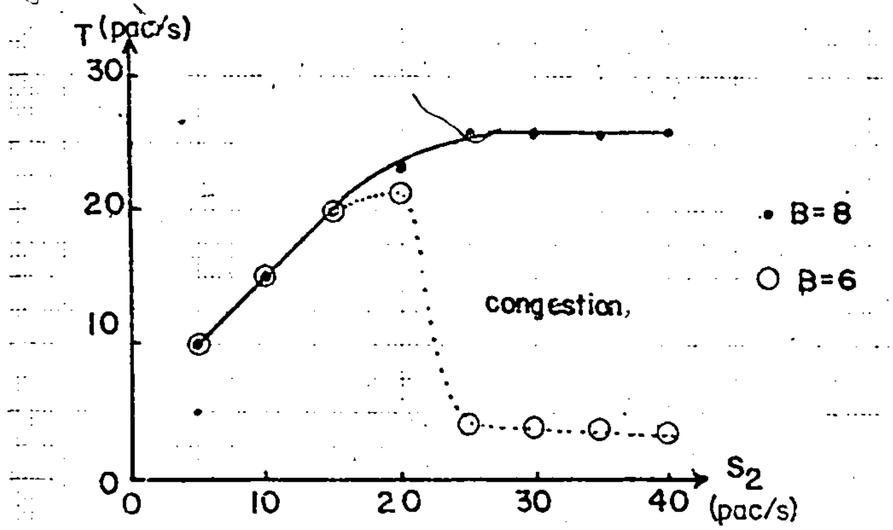
This approximation can be considered an upper bound, because it disregards any window or buffer management scheme constraints.

The ideal delay is the packet transmission time through the network with no queueing delay. That is, packets are served immediately upon entering a queue. For the network of Fig. 19, packets must hop through four servers:

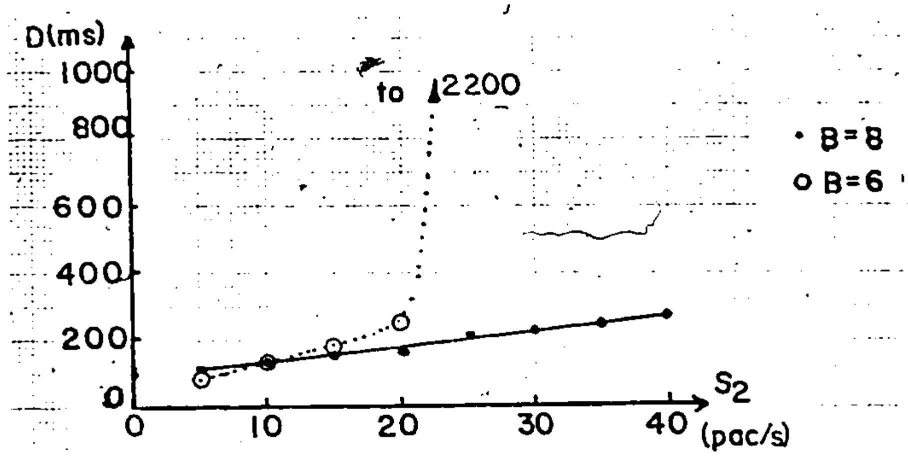
3 servers at 56kB	= 3(1kb/pac / 56kb/s)	= 53.6 ms
1 server at 25kB	= 1/25	= 40.0 ms
Ideal delay =		93.6 ms.

Initially, the network was evaluated with 6 buffers in node 2 (CAL). However, with B=6 the network graph shows the classical congested condition (Fig. 23). In order to solve this problem, the bottleneck node was found by locating the node with the highest pbt and increasing its buffer size to 8. In the congested region, pbt[2] was 0.99 - an area where the model is not accurate; however, the actual value was not important in locating the congested node.

In the delay curve of Fig. 23, the delay rises rapidly at the onset of congestion (from 0.2s to 2.2s).



(A) Network throughput vs class-2 offered load



(B) Network delay vs class-2 offered load

Figure 23: Effects of Increased Buffer Size at Node 2.

5.2.1 Network NDG80 with node 2 having B=8

In this section, the BMS of node 4 is varied with the parameters shown in Table 4. Node 4 is the source node for class 2.

Fig. 24 shows the class 2 throughput against the offered load. Upto about 20 pac/s, the offered packets are easily carried by the network. With more than 20 pac/s, the VC begins to saturate, nearing its maximum carrying capacity (25 pac/s). There is little difference between the curves for the various BMS's. Since the bottleneck is a slow link (which saturates the network quickly after 20 pac/s), the various BMS's have little effect provided that the network can carry the load. (At S_2 = 40 pac/s, NAP1 calculates that there are about 8 packets in the network in total : 1 from class 1, and 7 from class 2. With these packets evenly distributed among the four nodes, each node has about 8/4 = 2 packets.)

Fig. 25 illustrates the network throughput and delay curves. The throughput curves do not show significant variation among the four BMS's. However, CP shows slightly increased delays. The other three schemes have similar delay characateristics.

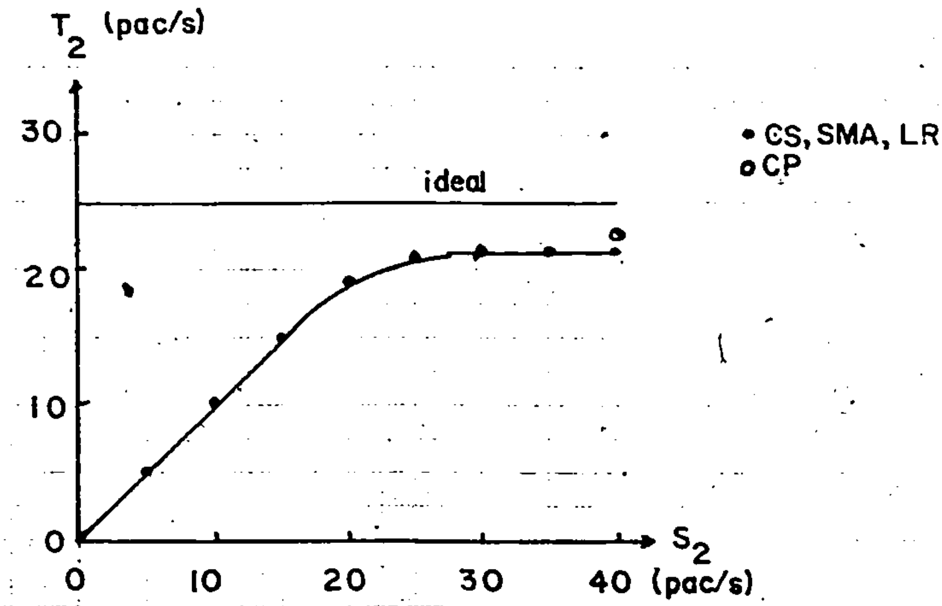
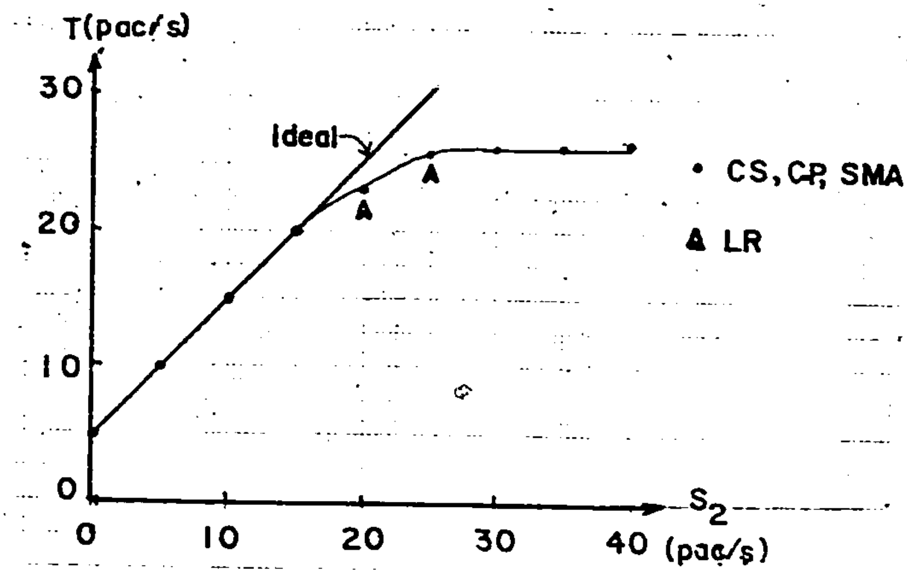
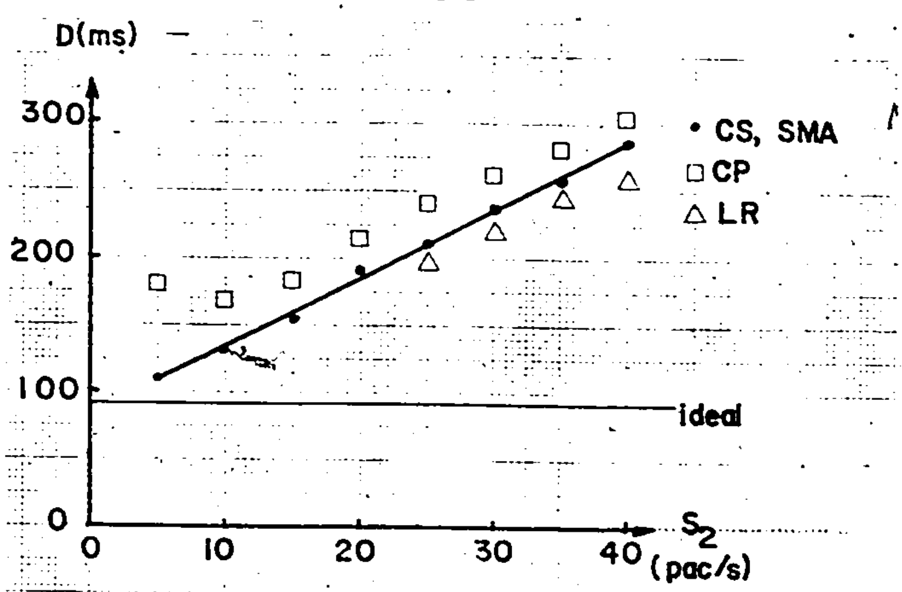


Figure 24: Class 2 Throughput vs Offered Load.



(A) Network Throughput



(B) Network Delay

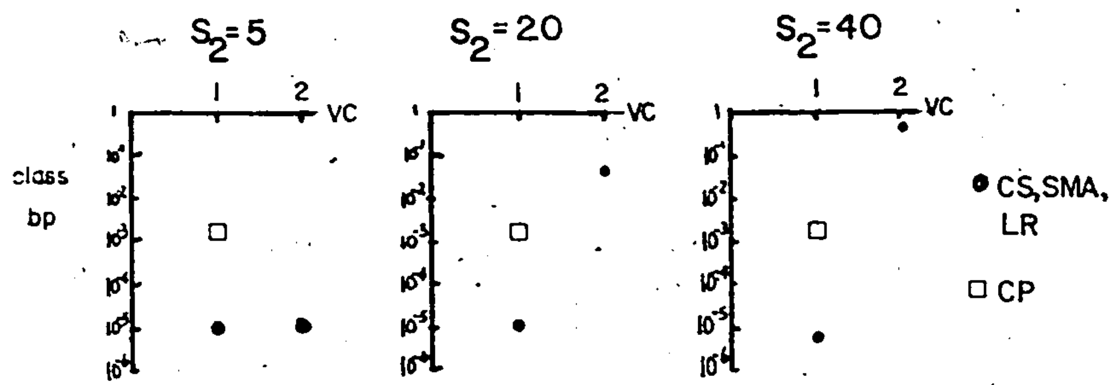
Figure 25: Network Throughput and Delay Curves.

Figs. 26 and 27 show the computed blocking probabilities for the network computed by NAP1. The class blocking probabilities in Fig. 26 show the blocking due to the window constraints only. At $S_2 = 40$, about 50 per cent of the class 2 packets are blocked at the source. Only the CP scheme varies from the others.

Fig. 27 shows that, as expected, the nodal blocking probabilities increase from $S_2 = 5$ to $S_2 = 40$. Node 4 experiences the greatest variation because its buffer management scheme is changing.

The case $S_2 = 20$ pac/s is the most important as it corresponds to onset of network congestion and can give information about network degradation. CS gives blocking probabilities of about 10^{-3}.

The Lam-Reiser scheme gives slightly lower pbt, at the expense of larger pbi. This result is qualitatively expected. CP gives the highest blocking probabilities because of its partitioning of buffers equally between the two classes, even though class 2 has much higher traffic.

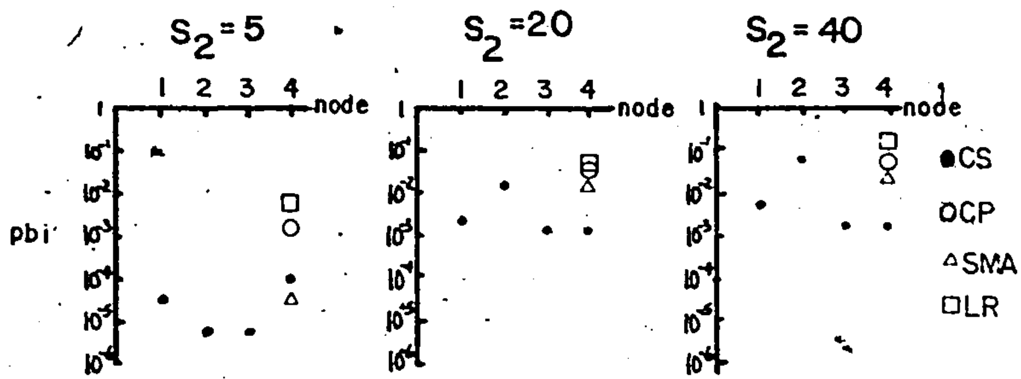


NOTES

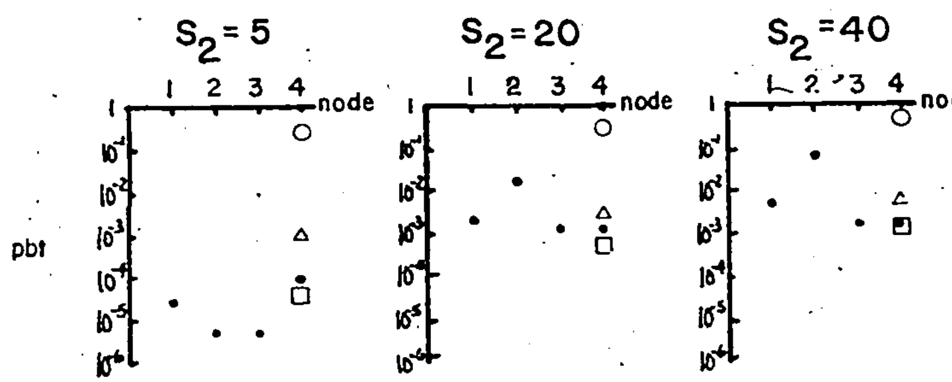
1. All blocking probabilities are drawn on a logarithmic axis.

2. Only points which have significant variations from the CS scheme are plotted as separate points.

Figure 26: Computed Class Blocking Probabilities.



(A) pbi - blocking probabilities for input packets



(B) pbt - blocking probabilities for transit packets

NOTES

1. All blocking probabilities are drawn on a logarithmic axis.

2. Only points which have significant variations from the CS scheme are plotted as separate points.

Figure 27: Computed Nodal Blocking Probabilities.

5.2.2 Network NDG80 with equal channel capacities

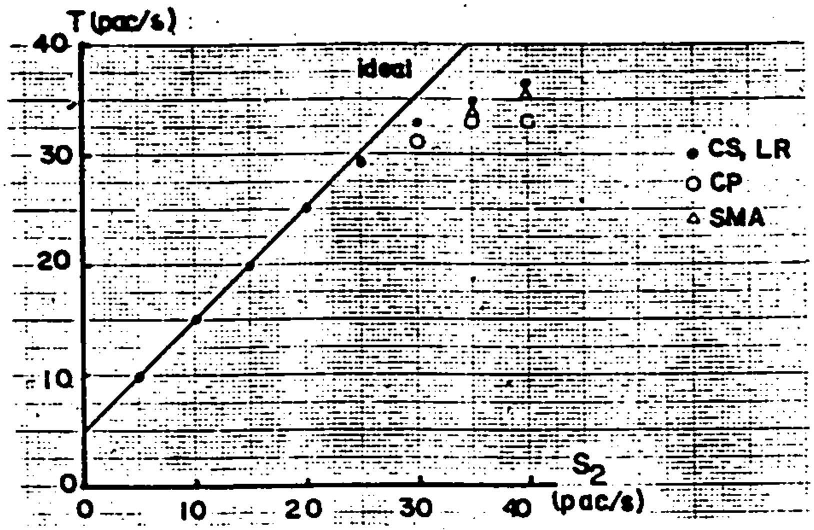
To remove bottlenecks caused by channel links saturating, all channel cacpacities were set to 56kB (Fig. 28). The BMS of node 2 (the exit node for class 2) was varied instead of node 4 in order to block class 1 packets at their source node. (The parameters values of Table 4 were used for node 2.)

Panel A shows a small variation in network throughput characteristics for the BMS's. CS and LR are the best, and CP the worst. The same holds true for the network delay characteristics.

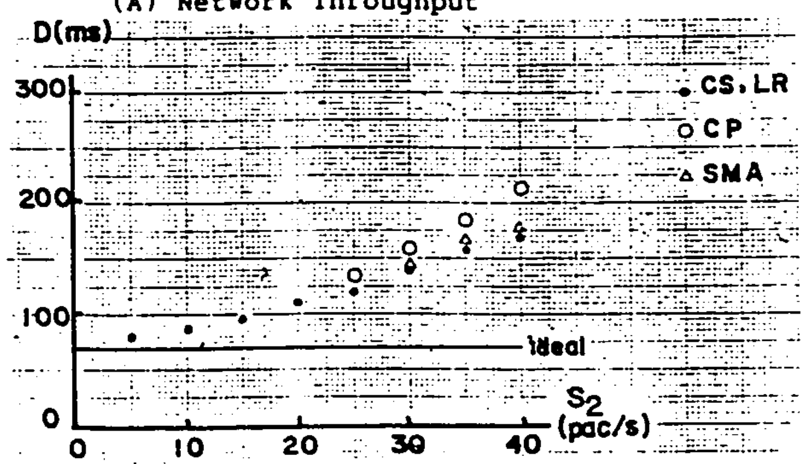
5.3 Conclusions to the numerical results

The results of the previous section lead to improvements over previous results in the numerical solution of computer network performance measures [GEOR 80b] as follows.

Broyden's method was found to be superior to the method of successive substitutions in solving the system of non-linear algebraic equations for the input and transit blocking probabilities.



(A) Network Throughput



(B) Network Delay

NOTES

1. Mean number of packets (m) in the network at S_2 = 40 is about 8.

2. The BMS of node 2 is varied.

Figure 28: Performance of Networks with All Links at 56kB.

The separation of nodal traffic into input traffic and transit traffic by an analytical method permitted a number of buffer management schemes to be studied numerically.

The numerical results are expected qualitatively, but have not been checked by simulation.

Chapter VI
CONCLUSIONS

The work involved in this thesis produced the following original results:

1. ANALYTICAL EXPRESSION

   A recursive formula (see Appendix C) for finding the probabilities of two different sets of classes of packets in a queue. This formula was derived from a queueing theory model of the network.

2. ANALYTICAL MODEL

   An analytical model (see Chapter IV) for studying nodal blocking in a network under various buffer management schemes and end-to-end flow control. The model is based on the assumption that the blocking probabilities are small, and that the blocked states do not greatly influence the queue packet-probabilities of the quasi-infinite buffer case.

3. SOFTWARE

   The computer program NAP1 (written in PASCAL) which is a valuable research tool for studying the performance of virtual-circuit based packet-switching networks with nodal buffer management and end-to-end flow control.

A number of ideas for further investigation surfaced during this work:

1. <u>SIMULATION</u>

   Checking of the blocking assumptions used through simulation of the network.

2. <u>SCALING ALGORITHM</u>

   Incorporation of Lam's [1982] scaling algorithm to prevent underflow or overflow problems. (These did not occur in this work because of the small dimensionality of the problems studied.)

3. <u>NAP1 EXTENSIONS</u>

   Integration of a simulation program into NAP1 would provide a valuable research tool into the study of computer networks. The provision for a common input syntax for both the analytical and simulation models would be desirable.

Because the time computational complexity of the proposed algorithm to solve for the blocking probabilities grows exponentially with the number of classes, its use in real networks is limited . However, the time complexity grows linearly with the number of queues. This linearity permits networks with many queues to be handled.

The lack of analytical methods to handle blocking in queueing networks forces the use of approximations. These approximations are often based on intuitive arguments and can not be justified in general.

However, the corner stone of the tractability of queueing network problems, Kleinrock's "independence assumption", is itself an approximation which is known to be reasonable only in some network topologies. Choosing between results based on a crude approximation and no results, the former is the more desirable. Hopefully, a more appealing approach to queueing network problems will be found, and perhaps it will permit blocking to be handled more rigourously.

Appendix A

MATHEMATICAL PRELIMINARIES

This appendix summarizes the key mathematical definitions and properties used in appendix C [KLEI 76, pp.330-331], [OPPE 75].

Probability generating function (pgf)

The pgf is the z-transform of the probability distribution with the substitution $z^{-1} \rightarrow z$.
For a one-dimensional distribution, $g(k)$:

$$G(z) = E[z^k] = \sum_k g(k)\, z^k$$

For an R-dimensional distribution, $g(k_1, \ldots, k_R)$:

$$G(z_1, z_2, \ldots, z_R) = E\left[\prod_{i=1}^{R} z_i^{k_i}\right] = \sum_{k_1} \cdots \sum_{k_R} g(k_1, k_2, \ldots, k_R)\, z_1^{k_1} \cdots z_R^{k_R}$$

Basic function definitions

$$\delta_n = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

Kronecker delta function

$$u_n = \begin{cases} 1 & n \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Unit step function

Convolution

1-D $y(n) = f(n) * g(n) = \sum_k f(k)\, g(n-k)$

2-D $y(m,n) = f(m,n)*g(m,n) = \sum_{k_1} \sum_{k_2} f(k_1,k_2)\, g(m-k_1, n-k_2)$

Transform Properties and Pairs

Definition: $\{ f(k),\ k=0,1,\ldots \} \quad \longleftrightarrow \quad F(z) = \sum_{k=0}^{\infty} f(k)\, z^k$

SEQUENCE	Z-TRANSFORM
δ_n	1
δ_{n-k}	z^k
u_n	$\frac{1}{1-z}$
u_{n-k}	$\frac{z^k}{1-z}$
$x(k_1,k_2)$	$X(z_1,z_2)$
$x(k_1-a_1,k_2-a_2)$	$z_1^{a_1} z_2^{a_2} X(z_1,z_2)$
$x(k_1,k_2)*y(k_1,k_2)$	$X(z_1,z_2)\ Y(z_1,z_2)$

Appendix B

COMPUTATION OF THE NORMALIZATION CONSTANT

Algorithm 1 (described in this appendix) uses the Reiser-Kobayashi convolution algorithm to compute the normalization constants $C(y_1, y_2, \ldots, y_R)$ for all subnetworks within the feasible state space.

$$C^{-1}(\underline{y}) = g^*(\underline{y}) = \sum_{\underline{y} \in F} p(\underline{y}) \tag{B-1}$$

where:

$$F = \{ \underline{y} = (y_1, y_2, \ldots, y_R) \mid 0 \le y_r \le w_r \text{ for } r=1,..,R \}$$

The inverse of the normalization constant (C^{-1}) for the network under study is computed by:

$$C^{-1} = C^{-1}(w_1, w_2, \ldots, w_R) = \sum_{\underline{y} \in F} C^{-1}(y_1, \ldots, y_R) \left(\prod_{i=1}^{R} S_i^{y_i} \right)$$

$$= \sum_{\underline{y} \in F} g^*(\underline{y}) \prod_{i=1}^{R} S_i^{y_i} \tag{B-2}$$

The inputs to algorithm 1 are:

1. M - the number of queues
2. $\underline{\tau}$ - the workload vector for each queue

The outputs are:

1. $g^*(\underline{y})$ - the normalization constant of all smaller networks within F.
2. C^{-1} - the inverse of the normalization constant for the network.

The pseudo-code for algorithm 1 is given in Fig. 29.

Algorithm 1 requires $O(RM \prod_{i=1}^{R} w_i)$ operations, and has storage requirements of $O(\prod_{i=1}^{R} w_i)$.

```
Initialization:
     g*(0,0,...,0) = 1; { g* is of dimension R }
     C^-1 = 0;

FOR j = 1 to M do
  FOR i[1] = 1 to w[1] do
    FOR i[2] = 1 to w[2] do
      ...
        FOR i[R] = 1 to w[R] do begin
```

$$g_j^*(\underline{i}) = g_j^*(\underline{i}) + \sum_{r=1}^{R} \tau_{jr}\, g_j^*(\underline{i} - \underline{e}_r)$$

```
        IF j = M then { compute C^-1 }
```

$$C^{-1} = C^{-1} + g_j^*(\underline{i}) \prod_{k=1}^{R} S_k^{i[k]} ;$$

```
        end;
```

Notes

1. $\underline{i}' = (i_1, i_2, \ldots, i_R)$

Figure 29: Pseudo-code for Algorithm 1.

Appendix C

COMPUTATION OF P_m ($\underline{K}$, $\underline{L}$)

This appendix computes the probability of a queue having two sets of packet classes V and V', where V is a set with either 0 or 1 classes and V' = {1,2, ..., R} - V. That is, the network classes are partitioned into two disjoint sets. The name "superclass" will be used to describe V or V', because the symbols refer to sets of packet classes. (Although V is a set with only 0 or 1 elements, the method used in this section can be modified to include more than one class in the set V.)

Consider a queue ($\hat{M}$) having arrivals from a Poisson source of superclass V, and arrivals from the network of superclass V'. V and V' are mutually exclusive because a packet submitted to a node will not return to the same node (by assumption).

The following analysis is largely based on [REIS 76a], and proceeds as follows:

1. The generating function of the desired probability function, $p(1_V, 1_{\widehat{V'}})$ is a special case of [REIS 76a, eq.(21)].

2. Certain simplifying substitutions relevant to the particular problem are introduced into 1.
3. Further simplifying assumptions are made, and the generating function is derived.
4. The generating function is inverted to find p(l_V , $l_{V'}$).

1. Special case of a General Formula

Consider a queueing network with R closed subchains and having L subchains in total, i.e. L - R open subchains. Recall, that chapter 4 assumes that subchains are synonymous with classes in this problem.

Let $\underline{Z} = (z_1, z_2, \ldots, z_L)$ be the transform variable associated with the population vector with the L subchains ($\underline{K}$).

Let $= (z_{11}, z_{12}, \ldots, z_{1L}, \ldots, z_{M1}, \ldots, z_{ML})$ be the transform variable associated with the number of packets in each queue , separated by the L subchains ($\underline{k}$).

The generating function for the network is [REIS 76a]:

$$P^*(\underline{Z}, \underline{z}) = \sum_{\underline{k} \geq 0} \sum_{\underline{K} \in F} z_{11}^{k_{11}} \ldots z_{1L}^{k_{1L}} \ldots z_{M1}^{k_{M1}} \ldots z_{ML}^{k_{ML}} \quad z_1^{K_1} z_2^{K_2} \ldots z_L^{K_L} \tag{C-1}$$

where: $\underline{K} = (K_1 , K_2, \ldots, K_L)$

$\underline{\chi} = (k_{11}, k_{12}, \ldots, k_{1L}, \ldots, k_{ML})$

In order to isolate the $\hat{M}$-th queue, and partition the L classes in that queue into two sets (V and V'), the following substitutions are made:

$$\text{Let } \hat{\underline{Z}} = \underline{Z} \text{ with } \begin{cases} z_{Mr} = z_V & \text{for } r \in V \text{ in queue } \hat{M} \\ z_{Mr} = z_{V'} & \text{for } r \in V' \text{ in queue } \hat{M} \\ z_{ml} = 1 & \text{for the other queues, } m \neq \hat{M},\ l \in \{1,2,\ldots,R\} \end{cases} \tag{C-2}$$

following the approach of [REIS 75].

2. Simplifying Assumptions

Using [REIS 76a, eq.(21)]:

$$p^*(\underline{Z}, \underline{z}) = \prod_{m=1}^{M} C_m\Big(\sum_{l=R+1}^{L} \rho_{ml}\, z_{ml} + \sum_{l=1}^{R} \rho_{ml}\, z_{ml}\, z_l \Big) \tag{C-3}$$

and substituting

$$\mathcal{P}^*(\underline{Z}, \underline{z}) = \prod_{m=1}^{M} \begin{cases} C_m\left(\sum_{l=R+1}^{L} \rho_{ml} + \sum_{l=1}^{R} \rho_{ml}\, z_l \right) & \text{for } m \neq \hat{M} \\ C_m \begin{cases} \rho_{\hat{M}V} z_{\hat{M}V} + z_{\hat{M}V'} \sum_{\substack{l=R+1 \\ l \neq V}}^{L} \rho_{\hat{M}l} + \sum_{l=1}^{R} \rho_{\hat{M}l} z_{\hat{M}l} Z_l & \text{for } V \in \{R+1, \ldots, L\} \\ \sum_{l=R+1}^{L} \rho_{\hat{M}l} z_{\hat{M}l} + z_{\hat{M}V'} \sum_{\substack{l=1 \\ l \neq V}}^{L} \rho_{\hat{M}l} Z_l + \rho_{\hat{M}V} z_{\hat{M}V} Z_V & \text{for } V \in \{1, 2, \ldots, R\} \end{cases} & \text{for } m = \hat{M} \end{cases} \tag{C-4}$$

Simplifying notation can be introduced [REIS 76a, eq.(25)]:

$$G^*_{(\hat{M}-)}(\underline{Z}) = \prod_{\substack{m=1 \\ m \neq \hat{M}}}^{M} C_m \left(\sum_{l=R+1}^{L} \rho_{ml} + \sum_{l=1}^{R} \rho_{ml} \, Z_l \right)$$

(C-5)

3. Additional Simplifications

Additional simplifications can be introduced for the network model presented in chapter 4. Because a virtual-circuit based PSN is being modelled, all the classes are closed subchains. Therefore,

$L = R$

$V, V' \subset \{1,2,\ldots,R\}$

and

$$G^*_{(\hat{M}-)}(\underline{Z}) = \prod_{\substack{m=1 \\ m \neq \hat{M}}}^{M} C_m \left(\sum_{l=1}^{R} \rho_{ml} \, Z_l \right)$$

(C-6)

Equation (C-4) can be simplified:

$$p^{*}(z_{\hat{M}V}, z_{\hat{M}V'}, \underline{Z})$$

$$= C_{\hat{M}}\Big(z_V \rho_{\hat{M}V} Z_V + z_{V'} \sum_{\substack{l=1 \\ l \neq V}}^{R} \rho_{\hat{M}l} \, Z_l \Big) \quad G^{*}_{(\hat{M}-)}(\underline{Z}) -$$

$$= C_{\hat{M}}\big(z_V \rho_{\hat{M}V} Z_V + z_{V'} (\underline{\rho_{\hat{M}}} \cdot \underline{Z} - \rho_{\hat{M}V} Z_V) \big) \quad G^{*}_{(\hat{M}-)}(\underline{Z})$$

$$= \frac{C_{\hat{M}}\big(z_V \rho_{\hat{M}V} Z_V + z_{V'} (\underline{\rho_{\hat{M}}} \cdot \underline{Z} - \rho_{\hat{M}V} Z_V) \big)}{C_{\hat{M}}(\underline{\rho_{\hat{M}}} \, \underline{Z})} \qquad \text{(C-7)}$$

Furthermore, all service centers are FCFS. Hence, we can substitute the FCFS capacity function [REIS 76a]:

$$C(x) = C_{FCFS}(x) = \frac{1}{1 - x} \qquad \text{(C-8)}$$

Eq. (C-7) now becomes,

$$p^{*} \triangleq p^{*}_{M}(z_{\hat{M}V}, z_{\hat{M}V'}, \underline{Z}) = \frac{(1 - \underline{\rho_{\hat{M}}} \cdot \underline{Z}) \qquad G^{*}(\underline{Z})}{1 - z_V \rho_{\hat{M}V} Z_V - z_{V'} (\underline{\rho_{\hat{M}}} \cdot \underline{Z} - \rho_{\hat{M}V} Z_V)} \qquad \text{(C-9)}$$

4. Inversion of the Generating Function

Before inverting the generating function, the space of the solution will be discussed and some notation introduced.

The transformed space vector, $\underline{Y}$, and its k-domain vector, $\underline{X}$, have the following components in the R+2 dimensional space.

$$\begin{array}{ccl} \underline{Y} & = & (z_1, z_2, \ldots, z_R, z_V, z_{V'}) \\ \updownarrow & & \qquad\qquad \updownarrow \\ \underline{X} & = & (k_1, k_2, \ldots, k_R, l_V, l_{V'}) \end{array} \qquad \text{(C-10)}$$

The following transform pairs for impulses are needed for notational simplicity:

$$z_i \longleftrightarrow \Delta_i^{(\underline{X})} = \begin{cases} 1 & \text{if } (X_i = 1) \text{ and } (X_j \neq 0) \text{ for all } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_{a,0} = \begin{cases} 1 & \text{if } a = 0 \\ 0 & \text{otherwise} \end{cases} \qquad \text{(C-11)}$$

Rearranging eq. (C-9),

$p^*(z_V, z_{V'}, \underline{Z})$ =

$$\underbrace{\rho_{\hat{M}V} z_V z_{V'} p^*}_{a} + \underbrace{z_{V'} \Big(\sum_{\substack{l=1 \\ l \neq V}}^{R} \rho_{\hat{M}l} z_l \Big) p^*}_{b} + \underbrace{G^*(\underline{Z})}_{c} - \underbrace{\Big(\sum_{l=1}^{R} \rho_{\hat{M}l} z_l \Big) G^*(\underline{Z})}_{d} \qquad \text{(C-12)}$$

The terms of (C-12) will be individually inverted in the order c, d, a and b:

(c) $G^*(Z_1, Z_2, \ldots, Z_R) = \sum_{\underline{\chi} \geq \underline{0}} \Delta_{R+1}^{(\chi)} \Delta_{R+2}^{(\chi)} g(\underline{X} - \underline{\chi})$

$= g^*(k_1, \ldots, k_R) \, \delta_{l_V, 0} \, \delta_{l_{V'}, 0}$

(C-13)

(d) $\sum_{l=1}^{R} \rho_{Ml} Z_l) \; G^*(\underline{Z}) = \rho_{M1} Z_1 G^*(\underline{Z}) + \ldots + \tilde{\rho}_{MR} Z_R G^*(\underline{Z})$

$\longleftrightarrow \rho_{M1} (\Delta_1^{(\underline{X})} * (g^*(k_1, \ldots, k_R) \, \delta_{l_V, 0} \delta_{l_{V'}, 0})) + \ldots$

$= \sum_{l=1}^{R} \rho_{Ml} (\sum_{\underline{\chi} \geq 0} \Delta_1^{(\chi)} g^*(\underline{X} - \underline{\chi}) \, \delta_{l_V, 0} \delta_{l_{V'}, 0})$

$= \sum_{l=1}^{R} \rho_{Ml} \; p^*(k_1, \ldots, k_l - 1, \ldots, k_R) \, \delta_{l_V, 0} \, \delta_{l_{V'}, 0}$

(C-14)

(a) $\rho_{\hat{M}V} z_V Z_V p^* \longrightarrow \rho_{\hat{M}V} (\underbrace{\Delta_{R+1}^{(\underline{X})} \circledast (\underbrace{\Delta_V^{(\underline{X})} \circledast p^*(\underline{X})}_{a1})}_{a2})$

-(a1) $\sum_{\underline{\chi} \geq 0} \Delta_V^{(\chi)} \; p^*(\underline{X} - \underline{\chi})$

$= p^*(k_1, k_2, \ldots, k_V - 1, \ldots, k_R, l_V, l_{V'})$

(C-15)

(a2) $$\sum_{\underline{X}\geq 0} \Delta_{R+1}^{(\underline{X})} p^*(\underline{X}_{k_V+k_V-1} - \underline{X})$$

$$= p^*(k_1,k_2,\ldots,k_V-1,\ldots,k_R,l_V-1,l_{V'})$$

(C-16)

(b) $$\sum_{\substack{l=1\\ l\neq V}}^{R} \rho_{\hat{M}l}(z_V, z_l\, p^*) \rightarrow \sum_{\substack{l=1\\ l\neq V}}^{R} \rho_{\hat{M}l}(\Delta_{R+2}^{(\underline{X})} \circledast (\Delta_1^{(\underline{X})} \circledast p^*(\underline{X})))$$

$$= \sum_{\substack{l=1\\ l\neq V}}^{R} \rho_{\hat{M}l}\; p^*(k_1,k_2,\ldots,k_l-1,\ldots,k_R,l_V,l_{V'}-1)$$

(C-17)

Hence p* can be computed recursively:

$$p_{\hat{M}}(k_1,k_2,\ldots,k_R,l_V,l_{V'}) = p^* =$$

$$= \rho_{\hat{M}V} p^*(k_1,k_2,\ldots,k_V-1,\ldots,k_R,l_V-1,l_{V'})$$

$$+ \sum_{\substack{l=1\\ l\neq V}}^{R} \rho_{\hat{M}l}\, p^*(k_1,\ldots,k_l-1,\ldots,k_R,l_V,l_{V'}-1)$$

$$+ \left(g^*(k_1,\ldots,k_R) - \sum_{l=1}^{R} \rho_{Ml}\, g^*(k_1,\ldots,k_l-1,\ldots,k_R)\right) \delta_{l_V,0}\; \delta_{l_{V'},0}$$

(C-18)

with the initial condition:

$$P^{*}_{\hat{M}}(0,0,\ldots,0) = 1 \tag{C-19}$$

For queues having only one buffer class, V is the empty set. In other words, the set V' contains all the classes. Hence, Georganas's result [1980b, eq.(A11)] is a degenerate case of eq. (C-18).

Appendix D

COMPUTATION OF QUEUE MARGINAL PROBABILITIES WITH TWO BUFFER CLASSES

This appendix describes algorithm 2 which uses the inversion formula derived in Appendix C to compute the marginal queue probabilities for queue m ($p_m(l_V, l_{V'})$) from the conditioned queue probabilities, $p_m(\underline{i}, l_V, l_{V'})$. The approach is based on [GEOR 80b].

The marginal queue probability is obtained as a convex combination of $p^*_m(\underline{i}, l_V, l_{V'})$ [GEOR 79]:

$$p_m(l_V, l_{V'}) = C \sum_{\underline{i} \in F} p^*_m(\underline{i}, l_V, l_{V'}) \prod_{j=1}^{R} S_j^{i_j} \qquad \text{(D-1)}$$

where: F is the set of feasible states defined in appendix B.

The inputs to algorithm 1 is:

1. M - the number of queues in the network
2. m - the queue under study
3. $\underline{I}$ - the workload vector for each queue
4. $\underline{w}$ - the vector of window sizes
5. V - the class of the source feeding the queue (if any)

   V' - all other classes.

The outputs are:

1. p (l_V, $l_{V'}$) the probabilities of having l_V packets of class V and $l_{V'}$ packets of other classes in queue m.

The pseudo-code for algorithm 2 is give in Fig. 30.

The algorithm computes both the conditioned queue probabilities and the marginal queue probabilities (from eq. (D-1)) in the body of the loop.

Algorithm 2 requires O($RM \sum_{i=1}^{R} w_i \prod_{i=1}^{R} w_i$) operations, and has storage requirements of O($\max_i(w_i) \sum_{i=1}^{R} w_i \prod_{i=1}^{R} w_i$).

```
Initialization:
     p  (0,  0,0) = 1.

FOR l[V] = 0 to w[V] do
          { convolve over all packets in class V }

  FOR l[V'] = 0 to (maxnetpop - l[V]) do
            { convolve over all packets of remaining classes }

    p  (l[V], l[V']) = 0;

    FOR i[1] = 0 to w[1] do
      FOR i[2] = 0 to w[2] do
        ...
          FOR i[R] = 0 to w[R] do begin
            IF l[V'] >  Σ_{j=1}^{R} i[j] for {j | τ[m][j] <> 0 and j <> V }
              p  (i, l[V], l[V'] ) = 0
            else
              p  (i, l[V], l[V'] ) = as in eq. (C-18);

            p(l[V],l[V']) = p(l[V],l[V'])
                 +  p(i, l[V],l[V'])( Π_{j=1}^{R} s_j^{i_j} )

            end;
p( l[V], l[V'] ) = C p( l[V],l[V'] )
          { normalize the probability }
```

<u>Assumptions</u>

1. Only source V feeds this queue. If no source feeds this queue directly, then some steps are inoperative.
2. maxnetpop = total network population from all classes

$$= \sum_{i=1}^{R} w_i$$

Figure 30: Pseudo-code for Algorithm 2.

Appendix E

SOLUTION OF NON-LINEAR EQUATIONS BY BROYDEN'S METHOD

Given a system of "n" non-linear equations, $\underline{f}(\underline{x}) = 0$, and an initial guess, $\underline{x}^{(0)}$, Broyden's method is a quasi-Newton approach for finding a numerical solution to the system [BURD 81],[BROY 65].

Table 5 compares Newton's and Broyden's methods. Broyden's method requires fewer functional evaluations per iteration and fewer arithmetic operations per iteration. However, it is not self-correcting in the sense that a bad initial guess may not lead to convergence. The convergence is super-linear instead of quadratic; however, this is usually compensated for by the reduction in operations per iteration.

Broyden's method finds a sequence of vectors, $\underline{x}^{(i)}$, in a manner similar to Newton's method, but uses a matrix 'A' with special properties [BURD 81]. The iteration formula is:

$$\underline{x}^{(i+1)} = \underline{x}^{(i)} - A_{(i)}^{-1} f(\underline{x}^{(i)}) \tag{E-1}$$

	NEWTON	BROYDEN
Operations per iteration:		
- function evaluations	$n^2 + n$	n
- arithmetic	$O(n^3)$	$O(n^2)$
- total	$O(n^3 + n^2 + n)$	$O(n^2 + n)$
Convergence	quadratic	super-linear

TABLE 5

Comparison of Newton's and Broyden's methods.

where $A^{-1}_{(i)} = J(\underline{x}^{(i)})$, i.e. the Jacobian evaluated at $\underline{x}^{(i)}$.

$A^{-1}_{(0)}$ is initially computed by approximating the partial derivatives of the Jacobian with finite differences, followed by matrix inversion:

$$J \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \qquad \text{(E-2)}$$

The finite difference approximation for the partial derivatives is given by:

$$\frac{\partial f_j}{\partial x_k}(\underline{x}^{(i)}) = \frac{f_j(\underline{x}^{(i)} + \underline{e}_k h) - f_j(\underline{x}^{(i)})}{h} \qquad \text{(E-3)}$$

where: $\underline{e}_k$ is a unit vector in the k-th dimension,

h is the error order.

Rather than recomputing J after each iteration (i), the A-matrix can be updated with fewer operations:

$$A_i^{-1} = A_{i-1}^{-1} + \frac{(\underline{s}_i - A_{i-1}^{-1}\, \underline{y}_i)\, \underline{s}_i^t\, A_{i-1}^{-1}}{\underline{s}_i^t\, A_{i-1}^{-1}\, \underline{y}_i} \qquad \text{(E-4)}$$

where: $\underline{s}_i = \underline{x}^{(i)} - \underline{x}^{(i-1)}$

$\underline{y}_i = \underline{f}(\underline{x}^{(i)}) - \underline{f}(\underline{x}^{(i-1)})$.

The iterations continue until the l_2-norm is less than a given value, ϵ :

$$\|\underline{x}^{(i)} - \underline{x}^{(i-1)}\|_2 = \sum_{j=1}^{n} (x^{(i)}[j] - x^{(i-1)}[j])^2 \leq \epsilon \qquad \text{(E-5)}$$

Appendix F

NAP1 - NETWORK ANALYSIS PROGRAM 1

F.1 Design

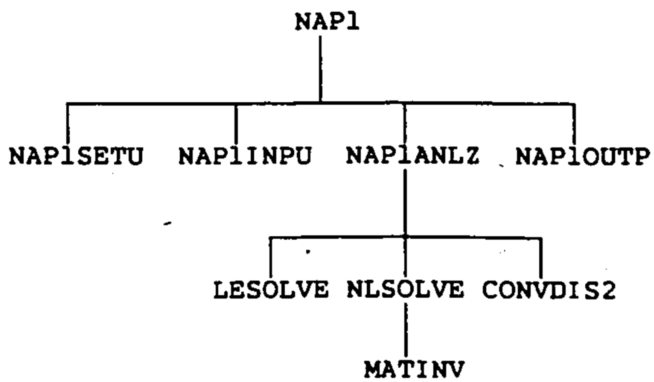
NAP1 is a set of PASCAL procedures which computes performance measures for a network of the type described in chapter 4. The network model is solved analytically using a method based on the Reiser-Kobayashi convolution algorithm.

The PASCAL procedures were written at the University of Ottawa for an AMDAHL 470/V7A computer running CMS under the IBM VM/SP operating system. The PASCAL compiler was the AAEC version 2.0 (Austrailian Atomic Energy Commission) PASCAL 8000 dated July 27, 1980.

The design philosophy is as follows (Figs. 31 ,32, and 33). NAP1DBDEF is a file holding the global data structure, e.g. number of nodes, connection between nodes, blocking probabilities, compile-time constants. It must be included into certain files before compilation.

NAP1 is the main program which calls:

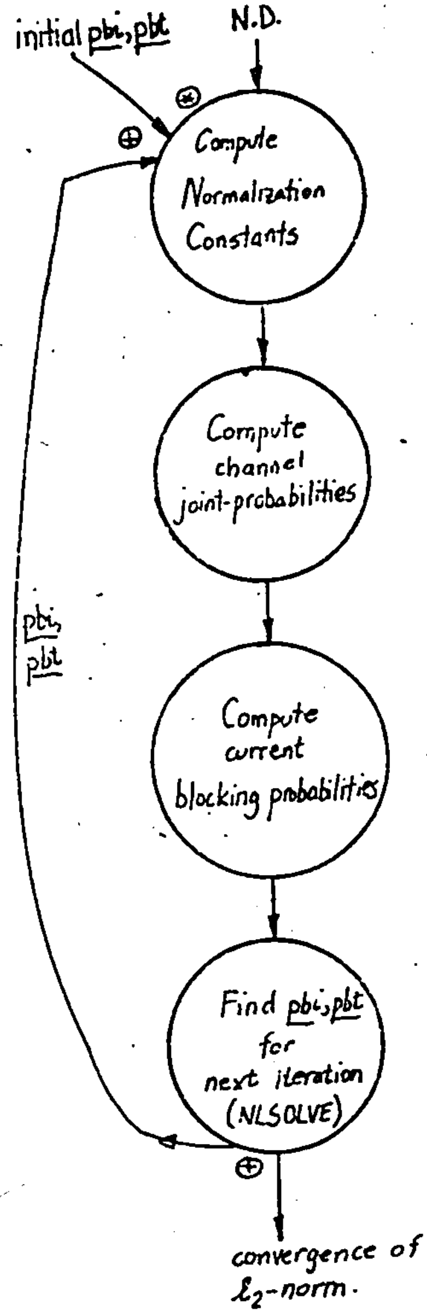
1. NAP1SETU for initialization and file assignments
2. NAP1INPU to read the network structure data into the global data base (from either disk or terminal).



Notes

1. NAP1 is the main program. The remaining modules are external procedures.
2. Only the modules relevant to key computations are shown.
3. Modules communicate through a global data structure (NAP1DBDEF) which holds data such as number of nodes, channel capacities, classes.

Figure 31: Basic Structure of NAP1.



N.D. = network description
= { N, Queues, R, packetlength, nodal BMS and parameters, node-to-queue mapping, channel capacities, routing matrix }

p.b. = blocking probabilities
= { pbi, pbt }

g* = normalization vector

C = normalization constant

Figure 32: Main Data Flow for NAP1.

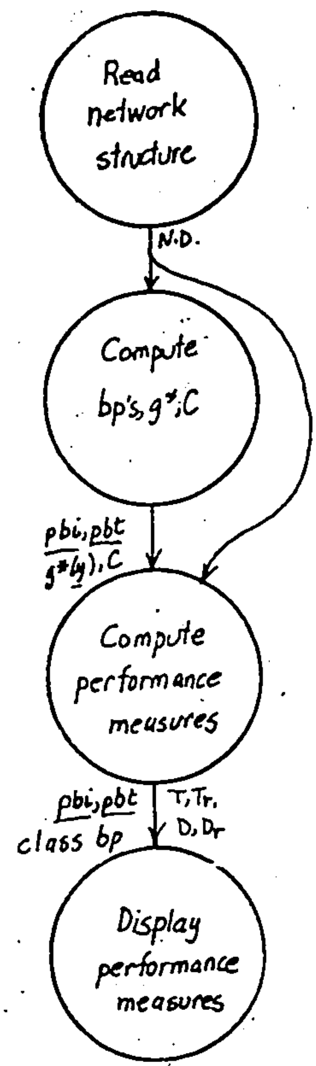


Figure 33: Data Flow Diagram for Computation of pb's, g and C.

4. NAP1ANLZ computes the nodal blocking probabilities using NLSOLVE to solve the system of non-linear equations by Broyden's method. (MATINV inverts a square matrix.) CONVDIS2 performs a two-dimensional convolution of functions with a discrete domain to compute nodal blocking probabilities from queue probabilities.
5. NAP1OUTP calculates the remaining performance measures (throughput, delay) and displays the output. Optionally, the output can be stored on disk.

Sample input, output and a terminal session for NAP1 are shown in Figs. 34 and 35.

F.2 Testing

NAP1 was tested by comparing its results with those shown in [GEOR 80b]. The results were the same except for minute differences (less than 1 percent) due to :

1. Solving the system of non-linear equations by Broyden's method which leads to a slightly different set of solutions.
2. Using the smaller norm of 10^{-4} instead of 10^{-3}.

(A) Sample disk input (file: EX12 NAP1)

```
NETWORK FROM NDG H/L FC WITH W=8.
           4 nodes.
           7 queues.
           2 classes.
        1000 bits/pac.
  6   1   3  1   0   0   0
  8   2   1   6  1   0   0   0
  6   2   2   4  1   0   C   C
  6   2   5   7  2   3   C   C
   2    1 5.600E+04   1  1.E+00
   3    2 5.600E+04   2  1.E+0C
   1    3 5.600E+04   1  1.E+00   2  1.E+00
   3    4 5.600E+04   1  1.E+00
   4    1 5.600E+04   2  1.E+00
   2    0 2.500E+04   2  1.E+00
   4    0 2.500E+04   1  1.E+00
    8   2   4  5.000E+00
    8   4   2  5.000E+00

R;
```

(b) Sample disk output (file: EX12 NAP1OU)

```
NETWORK FROM NDG H/L FC WITH W=8.  on 07/04/83
 Parameter :Arr.rate[  2] = 5.000E+00
 Converge in   13 iter(s) with norm= 4.630E-05 using  4.596E+01 CPU s.
   1  4.992E+00  2.467E-01  1.232E+00  1.5E-03
   2  4.994E+00  1.119E-01  5.542E-01  9.6E-06
   0  9.987E+00  1.788E-01  1.786E+00
     4 nodes have input blocking prob's of :
  3.157E-05  3.096E-06  9.155E-05  1.105E-03
     4 nodes have transit blocking prob's of :
  3.157E-05  3.096E-06  9.155E-05  5.536E-01
NETWORK FROM NDG H/L FC WITH W=8.  on 07/04/83
 Parameter :Arr.rate[  2] = 1.000E+01
 Converge in   13 iter(s) with norm= 4.607E-05 using  4.593E+01 CPU s.
   1  4.991E+00  2.498E-01  1.246E+00  1.6E-03
   2  9.910E+00  1.338E-01  1.326E+00  1.3E-03
   0  1.490E+01  1.726E-01  2.572E+00
     4 nodes have input blocking prob's of :
  3.386E-04  2.965E-04  2.461E-04  7.765E-03
     4 nodes have transit blocking prob's of :
  3.386E-04  2.965E-04  2.461E-04  5.542E-01
NETWORK FROM NDG H/L FC WITH W=8.  on 07/04/83
 Parameter :Arr.rate[  2] = 1.500E+01
 Converge in   13 iter(s) with norm= 4.396E-05 using  4.599E+01 CPU s.
   1  4.976E+00  2.536E-01  1.262E+00  1.6E-03
   2  1.446E+01  1.653E-01  2.390E+00  1.5E-02
   0  1.943E+01  1.879E-01  3.652E+00
     4 nodes have input blocking prob's of :
  1.478E-03  3.115E-03  7.425E-04  2.178E-02
     4 nodes have transit blockin!
?
  1.478E-03  3.115E-03  7.425E-04  5.556E-01
NETWORK FROM NDG H/L FC WITH W=8.  on 07/04/83
 Parameter :Arr.rate[  2] = 2.000E+01
 Converge in   13 iter(s) with norm= 4.146E-05 using  4.599E+01 CPU s.
CMS
```

Figure 34: Sample Input/Output for NAP1.

```
GLOBAL TXTLIB PASCAL
ACC 200 F
R;
?
LOAD NAP1 (START
EXECUTION BEGINS...
Network Analysis Program 2. 0
Specify input stream for data:
<> for terminal
<filename> for disk file (ft/fm=NAP1/A1)
?
?------------------------------------------- EX12
Specify output stream for results:
<> for terminal
<filename> for disk file (ft/fm=NAP1OU/A1)
?------------------------------------------- EX12
Ready to accept user input.
NETWORK FROM NDG M/L FC WITH W=8.
The network has     4 nodes.
    7 queues.
    2 classes.
The packetlength is  1000 bits.
--------------------------------------------------------------
| Node | Buffers |  Queues               |   BMS+p1,p2,p3    |
--------------------------------------------------------------
|    1 |    6    |  3                    | 1 :  0   0   0 |
|    2 |    8    |  1   6                | 1 :  0   0   0 |
|    3 |    6    |  2   4                | 1 :  0   0   0 |
|    4 |    6    |  5   7                | 2 :  3   0   0 |
--------------------------------------------------------------

---------------------------------------------------------------------------
| Queue | src | des |  car( bps) | < class,Prob >                         |
---------------------------------------------------------------------------
|     1 |   2 |   1 | 5.6000E+04 |( 1, 1.E+00)                            |
|     2 |   3 |   2 | 5.6000E+04 |( 2, 1.E+00)                            |
|     3 |   1 |   3 | 5.6000E+04 |( 1, 1.E+00)( 2, 1.E+00)                |
|     4 |   3 |   4 | 5.6000E+04 |( 1, 1.E+00)                            |
|     5 |   4 |   1 | 5.6000E+04 |( 2, 1.E+00)                            |
|     6 |   2 |   0 | 2.5000E+04 |( 2, 1.E+00)                            |
|     7 |   4 |   0 | 2.5000E+04 |( 1, 1.E+00)                            |
---------------------------------------------------------------------------

----------------------------------------------------
| VC | Window | Arr.Node | Des.Node | Arr.Rate |
|    | (pacs) |          |          | (pacs/s) |
----------------------------------------------------
|  1 |      8 |        2 |        4 | 5.00E+00 |
|  2 |      8 |        4 |        2 | 5.00E+00 |
----------------------------------------------------
Modify? N -no   S<n><x,y>-source rate   W<n><x>-window
 B<n,max,sch,p1,p2,p3>-bufs
?------------------------------------------- N
Which inparms do you want varied ?
Enter <symbol><specifier><min><max><inc>.
< > for no parameter variation.
<W> for window
<S> for source rate
<I> for input nodal buffers
<T> for transit nodal buffers
<E> to exit
?------------------------------------------- [ ]
Enter the mode: D<level><als> for debug, N<als> for normal
?------------------------------------------- N 1
Entering normal mode.
User input accepted.
*** Analysis beginning.
!
!
CP
```

Figure 35: Terminal Session using NAP1.

REFERENCES

[BCMP 75] F.Baskett,K.M.Chandy,R.R.Muntz,F.G.Palacios, "Open, Closed and Mixed Networks with Different Classes of Customers", JACM,v.22, n.2(Apr 1975) 248-260.

[BOCH 80] G.Bochmann,C.Sunshine, "Formal Methods in Communication Protocol Design", IEEE Trans.-Communications, v.COM-20, n.3 (Apr 1980) 553-574.

[BROY 65] C.G.Broyden "A Class of methods for solving non-linear simultaneous Equations", Mathematics of Computation,v.19(Oct 1965)577-593.

[BRUE 80] S.C.Bruell,G.Balbo, "Computational Algorithms for Closed Queueing Networks", North-Holland, U.S.A. (1980).

[BURD 81] R.L.Burden,J.D. Faires,A.C.Reynolds, "Numerical Analysis",2nd ed., Prindle,Weber and Schmidt, U.S.A. (1981)chapter 9.3.

[BUZE 73] J.P.Buzen, "Computational Algorithms for Closed Queueing Networks with Exponential Servers", Comm. of the ACM,v.9(Sep 1973)527-531.

[CHOU 82] W.Chou,A.A. Nilsson, A.W.Bragg, "The need for Dynamic Routing in a Network spanning several time Zones", IEEE Communications Mag.(July 1982)13-16.

[COUR 77] P.J.Courtois, "Decomposability,Queueing and Computer Applications", Academic Press, U.S.A. (1977).

[COUR 78] P.J. Courtois, "Exact Aggregation in Queueing Networks", Proc. First Meeting AFCET-SMF, Paris, France (Sep. 1978).

[DAVI 72] D.Davies, "The Control of Congestion in Packet-Switching Networks", IEEE Trans.-Communications, v.COM-20, n.3 (June 1972) 539-546.

[DAVI 81] D.W.Davies,et al., "Distributed Systems-Architecture and Implementation", Lecture notes in Computer Science, vol. 105, Springer-Verlag, 1981.

[DUNN 56] P.F.Dunn, C.D. Flagle, P.A. Hick, "The Queueiac: an Electromechanical Analog for the simulation of waiting-line Problems", Operations Research, vol.4, no.6 (Dec. 1956) 648-667.

[FUCH 70] E.Fuchs, P.E. Jackson, "Estimates of Distributions of Random Variables for Certain Communication Traffic Models", Communication of the ACM, v.13, n.12 (Dec 1970) 752-757.

[GERL 80] M.Gerla,L.Kleinrock, "Flow Control: A Comparative Survey", IEEE Trans.-Communications, v.COM-20, no.4 (Apr 80) 553-574.

[GERL 82] M.Gerla,M.Staskauskas, "Fairness in Flow Controlled Networks", Jour. of Telecommunication Networks(1982)29-38.

[GEOR 79] N.D.Georganas, "Numerical Solution of queueing networks with multiple semiclosed chains", Proc. IEE,v.126,n.3(Mar 1979)229-231.

[GEOR 80a] N.D.Georganas, "Congestion Control of packet-switched networks with three types of buffer limits", Computer Communications(Dec 1980)270-277.

[GEOR 80b] N.D.Georganas, "Modeling and Analysis of Message Switched Computer-Communication Networks with Multilevel Flow Control", Computer Networks 4(1980)285-294.

[GRUB 77] D.S. Grubb, I.W. Cotton, "Criteria for Evaluation of Data Communications Services", Computer Networks, v.1, no.6, (1977) 325-340.

[HOPC 79] J.E. Hopcroft,J.D. Ullman, "Introduction to Automata Theory, Languages, and Computation", Addison-Wesley(1979).

[JACK 63] J.R.Jackson, "Jobshop-like Queueing Systems", Management Science,v.10(Oct 1963)131-142.

[KAMO 76] F.Kamoun,L.Kleinrock, "Analysis of Shared Storage in a Computer Network Environment", Proc. 9th Hawaii Inter. Conf. on System Sciences(Jan 1976) 89-92.

[KAUF 81] J.S.Kaufman, "Blocking in a Shared Resource Environment", IEEE Trans.-Communication, v.COM-29, n.10 (Oct 1981) 1474-1481.

[KELL 79] F.P. Kelly, "Reversibility and Stochastic Networks", Wiley and Sons, 1979.

[KERM 77] P.Kermani,L.Kleinrock, "Analysis of Buffer Allocation Schemes in a Multiplexing Node", Proc.ICC,v.2(1977)30.4:269-275.

[KLEI 64] L. Kleinrock, "Communication Nets: Stochastic Message Flow and Delay", Dover, (1964).

[KLEI 75] L.Kleinrock, "Queueing Systems: Volume I - Theory", "Queueing Systems: Volume II - Computer Applications", Wiley & Sons(1975).

[KOBA 78] H.Kobayashi, "Modeling and Analysis: An Introduction to System Performance Evaluation Methodology", Addison-Wesley(1978).

[LAM 77] S.S.Lam, "Queueing Networks with Population Size Constraints", IBM J R&D(Jul 1977)370-378.

[LAM 79] S.S.Lam,M.Reiser, "Congestion Control of Store-and-Forward Networks by Input Buffer Limits - An Analysis", IEEE Trans.-Communications, v.COM-27, n.1 (Jan 1979) 127-133.

[LAM 82] S.S.Lam, "Dynamic Scaling and Growth Behaviour of Queuing Network Normalization Constants", JACM, v.29,n.2(Apr 1982)492-513.

[LEMI 79] C.Lemieux, "Flow Control in Switched Telephone Networks: Theory and experience", Flow Control in Computer Networks, J.L.Grangé, M.Gien, eds., North-Holland (1979)1-16.

[OLDE 79] W.Older, "Quantative Analysis of Congestion-Sensitive Routing", Flow Control in Computer Networks, J.L.Grangé, M.Gien, eds., North-Holland (1979)131-154.

[OPPE 75] A.V.Oppenheim,P.W. Schafer, "Digital Signal Processing", Prentice-Hall,U.S.A.(1975).

[PENN 75] M.C.Pennotti,M.Schwartz, "Congestion Control in Store and Forward Tandem Links", IEEE Trans.-Communications,V.COM-23,n.12(Dec 1975) 1434-1443.

[PETE 80] J.L. Peterson, "Petri Net Theory and the Modelling of Systems", Prentice-Hall, 1980.

[PITT 79] B.Pittel, "Closed Exponential Networks of Queues with Saturation: The Jackson-type stationary distribution and its Asymptotic results", Mathematics of Operations Research, vol. 4, no.4 (Nov. 1979) 357-360+.

[REIS 75] M.Reiser,H.Kobayashi, "Queueing Networks with Multiple Closed Chains: Theory and Computational Algorithms", IBM J R&D(May 1975)283-294.

[REIS 76a] M. Reiser, "Numerical Methods in Separable Queueing Networks", IBM Watson Research Center Report, RC5842 (Feb. 1976).

[REIS 76b] M.Reiser,C.H.Sauer, "Queueing Network Models: Methods of Solution and their Program Implementation", IBM Watson Research Center Report,RC 6109(July 1976).

[REIS 79] M.Reiser, "A Queueing Network Analysis of Computer-Communication Networks with Window Flow

Control", IEEE Trans.-Communications, v.COM-27, n.8 (Aug 1979) 1199-1209.

[REIS 80a] M.Reiser,S.S.Lavenberg, "Mean-Value Analysis of Closed Multi-chain Queueing Networks", JACM, v.27, n.2(Apr 1980)313-322.

[REIS 82] M.Reiser, "Performance Evaluation of Data Communication Systems", Proc. IEEE, v.70, n.2, (Feb. 1982) 171-196.

[RYBC 80a] A.Rybczynski, "X.25 Interface and End-to-End Virtual Circuit Service Characteristics", IEEE Trans.-Communications,v.COM-28,n.4(Apr 1980) 500-510.

[RYBC 80b] A.Rybczynski, J.B. Palframan, "A Common X.25 Interface to Public Data Networks", Computer Networks 4(1980)97-110.

[SCHW 79] M.Schwartz,S.Saad, "Analysis of Congestion Control Techniques in Computer Communication Networks", Flow Control in Computer Networks, J.L.Grangé,M.Gien,eds., North-Holland (1979) 113-130.

[TANE 81] A.S.Tanenbaum, "Computer Networks", Prentice-Hall, U.S.A.(1981).

[TOBA 80] F.A.Tobagi, "Multiaccess Protocols in Packet Communication Systems", IEEE Trans.-Communications, v.COM-28, n.4 (Apr 1980).

[WOLF 78] B.Wolfinger,D.Drobnik, "Simulation of Protocol Layers of Communication in Computer Networks", Computer Networks and Simulation, North-Holland (1978)119-139.