Imperial College of Science and Technology

(University of London)

Department of Computing and Control

REPRESENTATIVE QUEUEING NETWORK MODELS OF COMPUTER SYSTEMS IN TERMS OF TIME DELAY PROBABILITY DISTRIBUTIONS

by

Peter George Harrison

A Thesis submitted for the Degree of Doctor of Philosophy

September, 1979

ABSTRACT

In order to obtain a good representation of Computer Systems for performance evaluation, conventional analytic models require improvement from two points of view.

First there has been a tendency to concentrate on known analytic results and their extensions, obtaining representation of a specific system by choice of model parameter values. It is argued here that a *truly representative* model is best achieved by studying the properties of the real system first, and then determining the appropriate model type and structure from them.

Secondly, the most crucial performance measures for both management and users, are the time delays that relate to the rate at which individual tasks are being processed. Conventional models predict only overall resource utilisations and queue lengths.

Much of this thesis is concerned with distributions of time delays in queueing networks. An approximate method for their determination is presented which is applicable to a very general class of networks and gives an efficient implementation. Exact results are then derived for cycle time distribution, first in cyclic and then in more general tree-like networks. Validation of both methods is by comparison with simulated results, sufficiently detailed data from real systems being unavailable.

Subject to adequate precision, approximate methods are, in general, more feasible as tools because of their greater generality and superior efficiency. We view and apply the exact method as a standard by which to assess the accuracy of various approximations whilst also recognising its potential as a practical tool for simple cases.

Finally, the thesis addresses the almost universal assumption of "equilibrium", that is the assumption that the state space probability distribution is time independent. The time periods over which this assumption can or should not be made are quantified via time-dependent analysis that is applicable to a very general class of networks and relevant in many transient situations.

LIST OF ABBREVIATIONS

ACF	Autocorrelation function
APL	A Programming Language
BCMP	Reference to the principal result of [BASK75]
C-network	Defined on p.97
CPU	Central processing unit
FCFS	First come first served
FPI	Future path independence
I-O	Input-output
IS	Infinite server
KS	Kolmogorov Smirnov
LCFS	Last come first served
L.H.S.	Left hand side
p.d.f.	Probability density function
PS	Processor sharing
PSA	Permanent stationarity assumption
QNA	Queueing network analysis
R	The set of all real numbers
R^+	The set of all positive real numbers
R.H.S.	Right hand side
s.t.	Such that
w.r.t.	With respect to
Z	The set of all integers
Z^+	The set of all positive integers
$\exists$	There exists
$\nexists$	There does not exist

ACKNOWLEDGEMENTS

First, I would like to thank Professor Manny Lehman, the "boss" and my supervisor, for his help, advice and encouragement in relation to this research in particular and generally at work in the college.

I am grateful to Murray Woodside for his helpful comments on an early draft of this thesis and to the research group of Professor Heinz Beilner at the University of Dortmund, West Germany, for much interesting discussion and assistance in developing the simulator using their generator program. I am indebted to Professor Schassberger and to Dr. Rodney Coleman for their advice on the mathematical theory of stochastic processes. The APL software package of Appendix 7 and tables of Appendix 9 were developed and produced at the NUMAC IBM 370 installation, University of Newcastle. The plots of Appendix 10 were provided by Raj Chauhan of IP Sharp Associates, London, this company also providing some resources for program development.

I would also like to thank my friends and colleagues Justin Newland, Graham Benyon-Tinker and especially Art Mecklenburg for many interesting discussions on this subject, as well as many others. Finally, I am very grateful to "Tids" for her patient devotion to the typing of this thesis.

CONTENTS

Chapter 8. (cont.)

Appendix 8 (cont.)

§1. Introduction

Modelling of computer systems as an aid to performance evaluation has been undertaken in various forms for many years. Such models provide the ability to predict system behaviour in a variety of environments. Prediction is important since the performance of a computer installation, quantified according to some objective measure such as response time (interactive system) or throughput, is frequently highly sensitive to small alterations in system characteristics or the behaviour of the user community. Thus optimum tuning of the system parameters determining (as far as possible) these characteristics is highly desirable and, indeed, essential in heavily utilised installations.

The value of model based predictions is that a model should be far more flexible than the real system to work with. It can represent real system behaviour in a small fraction of the corresponding time required by the system itself and allows a wide range of experiments to be performed which may not even be practicable at all on the real system. For example, the effect of introducing new hardware such as an extra channel or more storage may be studied simply by altering appropriate model parameters. Furthermore, even to run possible experiments on the real system may well require that system to be dedicated throughout. This may be very costly, particularly if several experiments are to be performed. However, any model is worthless if it is not *representative* of the actual system for which it makes predictions; i.e. models must be adequately *validated* so that the accuracy of their predictions might be expected to be good.

The processing capabilities of computer systems consisting of hardware and operating systems (referred to henceforth as just "computer systems") have been represented by various types of model. These can be classified broadly as statistical, simulation or analytic, together with hybrids. For the past decade, considerable interest has been shown in models based on results of queueing network analysis

[JACK63, GORD67]; particularly since the publication by J.P. Buzen of an efficient algorithm for computing the associated marginal state space probabilities, [BUZE73].This computation was previously impracticable for even quite simple networks due to the sheer size of the state space. Such a modelling approach is in the analytic category and many fine papers, reporting both theoretical and practical research have appeared, for example [KELL75, DENN77, FAYO79]. In particular, the recent Queueing Networks edition of ACM Computing Surveys, [ACM78], gives an excellent review of the current state of the art.

However, the now traditional modelling approach using queueing network analysis (abbreviated to QNA henceforth) has perhaps not achieved as much as might have been hoped for from the point of view of providing a good representation. This is because there has been a tendency to apply existing analytic results to represent the system under investigation by suitable assignment of values to the model parameters. In contrast to this, to achieve a good representative model, it is argued here that the properties of the real system should be studied first, [LEHM79b], the most appropriate model type and structure subsequently being developed from or fitted to these properties.

Nevertheless, despite this criticism, it is not felt that QNA is a poor basis for computer system modelling; on the contrary, queueing network models are considered by the author to be excellent for this purpose, for the following reasons:

(i) Their structure matches very closely that of the operating systems of multiprogramming computer systems which allocate resources to tasks according to some queueing discipline. Thus one would expect such models to be representative, (relatively) easy to understand,

interpret and maintain;

(ii) Predicted performance measures, such as resource utilisations and queue length probability distributions, can be derived directly from the analytic solution for the state space (marginal) probabilities; see, for example, [BUZE73, DENN78].

(iii) Their parameterisation is simple: a queueing network model, under its necessary set of assumptions, is totally defined by the mean service times and queueing disciplines of its centres, the routing probabilities and the total number of customers in a closed network or the arrival rate in an open one;

(iv) They are fast in execution as a result of their analytic nature.

Simulation models also possess many of the advantages given above and can represent explicitly events at any level of detail. However, in view of their algorithmic mode of operation, they can be very slow (and expensive) in execution.

Typically, QNA has been used to predict the utilisations of and queue length probability distributions at the resources in the modelled system. Calibration and validation has been accomplished by matching such predictions with the corresponding values obtained by monitoring the real system. Valuable though these performance measures may be, the most crucial quantities to users and management alike are *time delays* which directly relate to the rate at which individual tasks are being processed by the system. More detailed discussion of the importance of such time delays and applications of time delay analysis is given in chapters 2 and 8. Suffice it to say here that such analysis can lead to prediction of response time distribution in an interactive

environment, of great value to the user in planning his work schedule as well as to management in organising installation protocol, and to prediction of cycle time distribution, of prime importance in any polling environment such as real time process control, multiplexor handling etc.

In this thesis, the approach advocated is to validate queueing network models in terms of the distributions of time delays as well as via resource utilisations and queue lengths, so inspiring confidence in the prediction of such time delays in different environments. Very little work has been published on the *distributions* of time delays although, of course, the ubiquitous Little's Law has been applied to obtain mean values via model throughput, e.g. [REIS79]. The reasons for this are in part a result of the approach of using off-the-shelf results and applying these to the system requiring to be modelled, so that the real need for such an analysis has been obscured somewhat. The derivation of the distributions of time delays is a difficult theoretical problem. Even when soluble it requires considerable computing power to obtain numerical results in even simple cases. The author's approach has been first to develop a relatively efficient, but approximate method to compute the distributions of time delays in a very general class of networks. Then an exact method, applicable to a rather more restricted class of networks, is defined as a standard for validation purposes as well as a practical method in simple cases; the exact algorithm is somewhat inefficient in execution.

A tacit assumption invariably made by the QNA modeller is that the system under investigation has reached a state of stochastic equilibrium - in other words, the joint probability distribution of its queue lengths, represented by state space probabilities in the queueing network model, is time independent.

Intuitively one would expect such an assumption to be valid for the majority of time periods modelled, excepting of course running up and running down times of the systems when the variations caused by edge effects are significant. However, little quantitative work has been carried out in this area. One chapter of this thesis is devoted to an analysis of the transient characteristics of the state space probabilities for Jackson type queueing networks, [JACK63], so that time intervals over which the equilibrium assumption should and should not be made may be identified. In fact the method generalises well to the more general BCMP, [BASK75], network. The relevance of such an analysis to a study of time delays is clear, and in fact an improvement to the accuracy of the approximate method referred to above was derived for a simple cyclic network by precisely this means, [HARR78a]. Furthermore, it is also apparent, for precisely the reasons given above, that transient analysis may be applied with great benefit in a wide variety of situations in QNA.

Following this overview of the subject matter of this thesis, some more detailed background information is given in the next chapter wherein the importance of time delays is emphasised, particularly cycle times.* Following that chapter the fundamental theoretical results of the research into the distribution of time delays in queueing networks are presented. First, in chapter 3, by making the so-called permanent stationarity assumption, an approximate result of very general application is derived. Exact results for the cycle times in cyclic networks and the more general "tree-like"† networks under more restrictive

---

* Cycle time is formally defined in chapter 3.

† Tree-like networks are formally defined in chapter 5.

assumptions, are then derived in chapters 4 and 5 respectively.

In chapter 6, validation with respect to results of simulation experiments is discussed for the approximate method and the (assumptions underlying the) exact method. Obviously, one would have liked to have performed validation with respect to observations monitored on one or more actual computer systems, but the system event level of detail required for such data collection makes data of this kind extremely difficult and expensive to obtain. Thus this validation has to remain an area for future effort.

A convergent, iterative technique is presented in the following chapter for solving the Kolmogorov differential-difference equations for Jackson type queueing networks in an analysis of the transients discussed above.

In chapter 8, possible applications for the results of the research presented in this thesis are discussed - indeed in many cases it was interest in the application which initiated the research - and future research directions are identified.

Throughout the thesis, all of the results and ideas presented are the original work of the author unless otherwise stated. In particular, the theoretical results for cycle time distribution, both approximate (chapter 3) and exact (chapters 4 and 5), and the transient analysis of chapter 7 constitute the author's main achievements over the past two years.

§ 2. Queueing Network Modelling of Computer Systems and the Importance of Time Delays

2.1 Shortcomings of contemporary methods

In the Introduction, the use of QNA as a method of modelling computer systems was strongly supported. Advantageous though this approach is, however, it does also possess certain disadvantages, with respect to the underlying theoretical analysis as well as the manner in which it has been applied.

The first disadvantage, common to some degree to all analytic modelling methods, is that in order to obtain an analytic solution certain assumptions,which may not be valid in practice, must be made about the characteristics of the components of the model. Assumptions typically made in QNA and which frequently do not hold in the actual system modelled are:

(i) The queueing discipline of the servers must be FCFS.[1] This assumption was relaxed to include servers of PS,[2] LCFS[3] and IS[4] disciplines in [BASK75] by the celebrated 'BCMP' result, but *priority* disciplines can still not be represented.

(ii) The service time distribution of each centre must be negative exponential. Again this restriction was relaxed by the BCMP result to allow any distribution with rational Laplace transform, and so for all practical purposes, a general distribution, [COX55]. However, the relaxation

---

1. First Come First Served.
2. Processor Sharing.
3. Last Come First Served.
4. Infinite Server.

applies only in the cases of PS, LCFS and IS disciplines. For FCFS discipline the server must still provide negative exponentially distributed service times.

(iii) The service time distribution of each centre must be independent of the queue lengths existing at all other centres - i.e. only local state dependence is allowed. In particular a solution cannot be found for blocking situations in which the service rate of one centre is reduced to zero when the queue length at a different centre reaches some value (e.g. finite waiting room example).

(iv) The routing probabilities between centres must be constant. This is another limitation on state dependence.

(v) The state space probabilities of the network are time independent - i.e. the network is assumed to be in a state of (stochastic) equilibrium. As discussed in the Introduction, this assumption is not unduly restrictive; it is acceptable both intuitively and in practice. However, very little work has appeared to indicate quantitatively over what time intervals it is valid.

As a result of these restrictions, the set of soluble networks is rather limited and many practical situations exist for which the corresponding networks are at present insoluble.

A second disadvantage, in the *use* of QNA, has already been discussed in the Introduction. It concerns the modelling approach in which there has been a tendency to apply QNA models without first making a phenomenological study of the actual system first. Care is also necessary in the *validation* of queueing network models. They are relatively easy to validate with respect

to readily available analytic predictions (concerning resource characteristics), but other measures, for example time delays, are of more interest, and validation should always be performed with respect to the measures of interest as specified *before* commencement of the modelling process. Now, from a study of the real system information would emerge, not only about its structure but also about the measures requiring prediction, which would almost invariably include time delays experienced by individual tasks.

One of the most serious disadvantages inherent in contemporary QNA modelling is, in the author's opinion, the inability to model the progress of an individual task through a network. Conventional analysis is oriented towards resources or servers as opposed to tasks or customers in that it is overall service centre utilisations and queue length probability distributions which can be predicted from the state space (marginal) probability distribution. This disadvantage was referred to implicitly in the discussion of the first in that priority queueing disciplines cannot, at the present time, be represented.

In the following section, efforts which have been made in attempts to overcome some of the disadvantages listed above are described. There follow two sections in which is discussed the evolution of the fundamental research reported in this thesis. This addresses the largely unstudied problems of the analysis of the time delays incurred by individual customers and of the transient properties in queueing networks.

2.2 Research addressing these problems

Most of the research undertaken up to the present time has been concerned with the problem of the limitation to local state dependence as described in the previous section. New or

extended methods in QNA, both approximate and exact, have been developed in order to find solutions for the state space probabilities in previously insoluble networks.

An approach frequently taken to solve networks for which the BCMP assumptions (the most significant of which were listed in the previous section) do not hold is to use approximate methods requiring fewer assumptions. Such methods that have been developed include the representation of sub-networks by equivalent single, locally state dependent servers, [CHAN75b], the use of network decomposition techniques, [COUR75, COUR77], and the diffusion approximation for the "heavy traffic" case of many customers, [KOBA74a, KOBA74b, GELE75]. The approach can be applied quite generally with various degrees of approximation and results in simpler, approximately equivalent networks to analyse. Thus computation becomes more efficient (important typically in communication network modelling where there may be very many service centres, see [REIS79] for example) and *avoids* the problems of violating assumptions rather than finding new solutions. An approximation frequently made is blatantly to violate certain assumptions, which although somewhat crude, usually gives predictions in good agreement with real world measurements. This is a manifestation of the so-called "robustness" (empirical) property of queueing networks which effectively states that a QNA model's predictions are stable in that they do not vary significantly when perturbations are made to its defining characteristics. The robustness property has been widely exploited of late [BARD79, BOUH79, PUJO79] in an attitude of "if it works do it" - i.e. validation is purely on the basis of experiment. The closeness between the structures of computer systems and queueing network models gives an intuitive explanation for robustness, but a formal analysis is really required.

The exact methods have in the main been derivations of solutions for specific cases with less restrictive assumptions. Of course, in theory *any* problem which can be represented by a Markov process can be solved exactly under the equilibrium assumption. The solution is quite simply the solution to the linear equations $\underline{P}'\underline{\underline{Q}} = 0$ where $\underline{\underline{Q}}$ is the instantaneous transition rate matrix for the process and $\underline{P}$ the equilibrium state space probabilities of the embedded Markov chain. However, this method is of no use in practice since for any non-trivial problem the number of states is excessively large, increasing combinatorially with the number of customers and number of centres in the network. Thus closed form solutions, e.g. the product form solution of [BASK75], have been sought and it is shown in [CHAN77] that the existence of a *product* form solution is equivalent to a network possessing the property of local balance.

A detailed study of state dependencies in queueing networks with particular reference to blocking is given by Mecklenburg, [MECK78]. Here solutions are derived for networks with non-locally state dependent service rates and routing probabilities, subject to certain constraints on the dependencies; in particular solutions are valid for reversible networks, defined in [KING69]. A fully general solution based on complex variable theory is given in [FAYO79] for the case of two-centre networks.

Valuable though research of this kind has been, little work has been reported addressing the other disadvantages discussed in section 2.1. Restrictions still exist on the type of queueing discipline, for example any priority discipline other than FCFS being barred, and on the form of service time distributions. In the latter case, of course, the only problem in practice arises with FCFS discipline which requires exponential service times for any solution to be possible in a Markovian framework.

With regard to the modelling methods used in currently reported research, the phenomenological approach advocated here has, typically, not been used [KRZE77a, SHUM77, SAUE75]. It was, however, adopted by the author in [HARR78b] and is supported by the modelling methodology described in [KIEN79]. The recent upsurge of interest in the operational analysis of queueing networks, [DENN78, BUZE78, BARD79], is also consistent with the approach in that the resulting models are actually defined in terms of measurements made on real systems.

2.3 Representation of time delays incurred by individual tasks

2.3.1 Their importance

As already discussed, conventional methods of QNA are essentially server oriented and little research has been carried out into the behaviour of individual customers in queueing networks. In section 2.1 an example of this was seen in the inability of current techniques to represent priority queueing disciplines. But the implications are far more extensive than this.

The importance of analysis of time delays incurred by tasks passing through the various components constituting a computer system is clear. Optimisation of response time (interactive system) and turnaround time at a computer installation is a major requirement of the user community. Indeed, predictability is essential if a user is to successfully integrate his computer usage into his work schedule. For example, even if the response time has a rather large expected value, it may still be tolerable if it is fairly consistent; that is if the standard deviation is small and there is little chance of response times considerably greater than the expected value. In fact predictability is often more important than magnitude. Thus, such considerations become an important

objective to the management for whom cost effectiveness (exemplified by throughput typically) is the main concern, subject to provision of a certain predefined minimum service quality for the user community.

Delays are in general composed of a sum of sub-delays incurred by passing through a sequence of components, e.g. CPU, various I-O activities and back to CPU; the number of terms in the sum depends on the size characteristics of the individual tasks in question, and in the case mentioned would be the number of I-O transfers required. Thus the analysis required may be divided into two areas:

(i) The time delay incurred from a single linear sequence of components;

(ii) Aggregation of successive such time delays (loops).

The prediction of response time comes in the second category. An application for time delay prediction in the first category occurs in communication network modelling where the probability of a message transmission taking longer than some specified time may be required.

A second, and very significant, application in the first category arises in the modelling of any system involving polling to permit the prediction of the probability of system failure. For example,

(i) In a multiplexor system it would be possible to predict the probability of data loss through failure of the polling routine to sample a data line frequently enough;

(ii) In a process control or machine tool control system one could predict the probability of a system fault caused by

the failure of a scheduler to test sensor inputs at some minimum specified rate. This sort of prediction is obviously of great value in view of the possibly catastrophic consequences caused by failure - say in nuclear reactor control.

More details as to the actual construction of models such as these are given in chapter 8.

The two problem areas mentioned above have been considered with respect to *mean values* of the time delays, when Little's Law (see for example [KLEI75]) may be used under suitable independence assumptions. Thus, for example, if successive cycles* of a task in an interactive system (e.g. sequences of service requests between successive requests for the CPU) are assumed independent, the mean response time is simply the mean cycle time (derivable through Little's Law) multiplied by the mean number of cycles. Recent research into mean value analysis of time delays, applicable as an approximation to complex networks, is presented in [REIS79] and [BARD79].

However, the mean value of a time delay alone is frequently insufficient. In the examples given above, for example, higher moments are required to give the standard deviation for response time and in polling systems percentiles are also required.

2.3.2 The author's approach

The author's work on the distribution of time delays began with an approximate study of cycle times in cyclic queueing networks and a generalisation of these, common server networks†, [HARR78a]. The method presented was based on the assumption of

---

* The term cycle is defined in chapter 3.

† Common server networks are similar to central server networks and defined formally in [HARR78a].

permanent stationarity which is defined formally in the following chapter and basically assumes that a queueing network is such that the equilibrium, time independent, state space probability distribution is valid at all times. For each of the possible sequences of successive queue lengths faced by some specific customer, the cycle time distribution is evaluated (approximately) as the convolution of the distributions of the sojourn times spent in each individual queue, taken in isolation. These results are then weighted according to the joint probability distribution of the queue lengths faced, as given by the permanent stationarity assumption. An improvement in this approximation is also developed via an exact algorithm for calculating this joint probability distribution.

This work has been extended to apply to networks of very general characteristics, with the time delay in question no longer restricted to cycle time, as described in chapter 3.

This leads naturally to the exact methods of chapters 4 and 5. These are based on a study of the discrete state transitions in networks as opposed to an analysis in continuous time. The class of networks analysed is chosen so that at all stages in the computations involved, whatever the state of a network, the position of some test customer is known. Expansion of the state space is consequently unnecessary in contrast to the method of [YU77] discussed below. Thus the results give relatively efficient implementations and may be used to provide standards against which to compare approximate methods (see chapter 6).

2.3.3 Other work

An approximate study of *response* time in queueing network models has been made by Lazowska and Sevcik [LAZO77a, LAZO77b, LAZO

78] in which response time is defined as the sum of successive cycle times for a particular task. The approximation arises in that it is assumed that successive cycles are independent and the distribution of the number of cycles required by a task is geometric. The resulting response time distribution is shown to be asymptotically exponential and, despite the approximations, gives results which compare quite favourably with actual observations.

Exact results have been derived for certain cases:

(a) Chow derives the *cycle time* distribution in cyclic networks of two centres with FCFS queueing discipline and exponential service times, [CHOW77a], which is extended to the central server case in [CHOW77b]. The approach taken (in the former case) is to observe that the behaviour of the second centre in the cycle, given the queue length there on arrival of the customer in question, is that of the centre taken in isolation. The probability distribution is derived for the queue length faced on arrival at the second centre conditional on that existing initially at the first, as a function of the sojourn time of the customer at the first centre. From this analysis in continuous time, the cycle time distribution follows as a complex result requiring numerical integration.

(b) Wong derives the Laplace transform of the time delay distribution for messages in *open* networks encountered in communication system modelling, [WONG78], by the use of probability generating functions and the properties of the Poisson arrival process.

(c) An exact solution for the passage time distribution of a network with a special "tagged" customer between predefined states, subject to routing constraints, is given for BCMP networks in the form of recursion equations for the Laplace transform in [YU77].

His approach is to apply results from the general theory of stochastic processes to queueing networks. Naturally, his results may be shown to be equivalent, for the appropriate class of networks, to those derived in chapters 4 and 5. However, the method is of limited *practical* use in that the recursion equations span the whole state space which is extended very considerably to include the information for tagging (so that the position of the tagged customer in the network is known in any state) and routing constraints. In fact the method has a close analogy with that of deriving the state space probabilities by solving the complete set of balance equations for a network explicitly (c.f. section 2.2) - a simple solution on paper but not in practice!

Although not an analytic method, another approach to the analysis of time delays is given in [SHED79] where simulation methods are applied in the Markovian framework used in (c). In this way, numerical results can be obtained for a much larger class of networks than in the case of numerical computation based on the corresponding analytic method.

2.4 The equilibrium assumption and transient analysis

The final line of research pursued and described here concerns a quantitative assessment of the equilibrium assumption made almost universally in QNA. The approach taken is to solve by an iterative method the Kolmogorov differential-difference equations for networks, so yielding the time dependent state space probabilities. In fact, this method was originally developed by the author for the simple case of a cyclic network with two servers, [HARR78a], as a refinement of the approximate analysis (assuming permanent stationarity) of cycle time distribution. A transient analysis in continuous time was performed to derive the probability

distribution of the second queue length faced conditional on the first, the arrival at the first centre effectively setting a time origin. In this way the joint probability distribution of the two queue lengths faced could be computed more accurately. It will be noticed that this application of the transient analysis is not dissimilar to the (independent) approach of [CHOW77a] discussed above.

Very little published work exists in the area of transient analysis in queueing networks, although in [GRAS77a,b] is presented a method adopting the approach of numerical solution of the Kolmogorov equations by means of the Runge-Kutta technique. This method is also used as a means for deriving time delay distributions.

The transient analysis developed in chapter 7 can be applied to networks of the Jackson type and the possibility of extension to the more general BCMP case is immediately apparent, although this is not done here. The method results in a simple iterative scheme, which is shown to be convergent, and is suitable for implementation by computer. Other applications of this research, in addition to the equilibrium assumption assessment, are concerned with the study of the immediate effects of *disturbances* in queueing networks. These are discussed in some detail in chapter 8.

§3. Time Delay Distributions under the Permanent Stationarity Assumption.

3.1 Model Specification

The queueing network time delays considered in this chapter are defined to be the timeselapsed between a customer arriving at some pre-defined service centre, α say, and his departure from some pre-defined service centre, β say. Under the usual assumption made in QNA that transitions between servers by customers are instantaneous, in a closed network the time of departure of the customer from centre β will be that of arrival at the successor centre in the customer's (infinite) path. This thesis is primarily concerned with cycle time distributions in closed queueing networks. In these, cycle time may be defined to be the time elapsed between successive occurrences of a particular customer's arrival at some specific service centre, subject to certain constraints on the centres entered in the path taken. In this case the successor centre of β will be α and the constraints restrict valid cycles to certain paths. For example, in the network shown in fig. 3.1

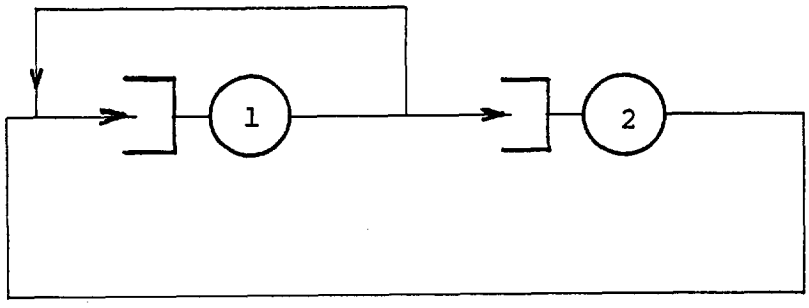


fig. 3.1 Network with constraints on valid cycles

a customer could re-enter centre 1 after leaving centre 1 without first entering centre 2 and such a path may well be considered

illegal as a cycle. However, for the networks considered in the exact analysis presented in chapters 4 and 5, by choosing centre α as the head of a tree-like network in the latter, the constraints become null in view of the order invariance or non-overtaking property required and fully defined therein.

In this chapter, first the moments of the time delay and then its distribution are derived for the case of a single path or sequence of centres entered between α and β under the assumption of permanent stationarity of the network. The distribution is first obtained in the form of its Laplace transform and then formulated as a recurrence relation derived by inversion thereof. A discrete form of the distribution is also derived and is easily seen to be convergent as the discrete time step decreases towards zero by precisely the same argument as is applied formally in the following chapters in the derivation of exact results.

The permanent stationarity assumption, abbreviated to PSA, states that a queueing network is in its stationary (equilibrium) state at *any* time, unconditionally on its state at all other times, and that its servers operate independently so that they can be considered in *isolation*. Thus, if the network has M servers with state space S; $1 \leq i,j \leq M$; $\underline{k},\underline{k}' \in S$ and the random variable $\underline{K} \in S$, the probability

$$P(\underline{K} = \underline{k} \mid t = t_i)$$

where a special customer (henceforth referred to as the "test" customer) arrives at centre i at time t_i, and the probability

$$P(\underline{K} = \underline{k}' \mid t = t_j)$$

are independent of $\underline{k}'$ and $\underline{k}$ respectively. Furthermore each has the equilibrium distribution, e.g. that of [JACK63] under appropriate additional assumptions.

This assumption is intuitively reasonable when an overall, averaged view of the network is acceptable after each service completion of the test customer, for example in the following cases:-

(i) In an open network - the length of the queue at centre j arrived at after a transition from centre i is unlikely to be strongly dependent on the queue length at centre i. This is in contrast to the case of a closed network in which the total number of customers is fixed. For example, for a closed cyclic network with 2 service centres, 1 and 2, and N customers, if there are k customers at centre 1 there are N-k at centre 2 with probability 1, $(0 \leq k \leq N)$. Thus if on arrival of the test customer at centre 1, all the other N-1 customers are at centre 2 $(k=1)$, then in the case of comparable service rates of the two centres, one would expect considerably more customers at centre 2 on arrival of the test customer than would be predicted by the steady state solution of the network.

(ii) To a lesser extent, in closed networks in which there are several service centres, in particular when centres have multiple arrival streams from other centres. The reasoning for this is similar to that given in (i).

(iii) When the queue length at each centre in the sequence of centres considered for the test customer's path is large;e.g.the heavy traffic situation. This is the case for open, closed and cyclic networks (to decreasing degrees) and follows since after a longer waiting time at any centre, the network will have undergone more transitions and more nearly approached its steady state.

When several paths are valid in the passage from centre α to centre β, as will be the case in general for cycle times in non-cyclic networks for example, the final result, whether for moments, distribution or Laplace transform of the distribution is easily obtained by weighting the results for each valid path

according to the probability that that path is followed.

I.e.

$$\text{Time Delay Distribution} = \sum_{\text{paths } p} P(p \mid p \text{ is valid})\{\text{Time delay distribution for path } p\}$$

Section 3.4 describes how the path probabilities may be obtained from the specifications of a queueing network.

Having derived the theoretical results for this approximate method, the chapter closes with a discussion of the (very general) applicability of the method in practice and validation methods.

3.2 Moments of time delays

The moments of the time delay distribution for some path of length M servers numbered 1,2,....,M for a given set of queue lengths, $\{n_i \mid 1 \leq i \leq M\}$, at times of arrival of the test customer, may be derived in terms of the moments of the service time distributions of the individual service centres in the path. These moments may then be weighted according to the corresponding queue lengths joint probability distribution to give the moments of the time delay.

In this section, as in section 3.3 also, the results pertain only to a single path, the general result being simply derived as a weighted average over all possible paths as described in section 3.4.

The time delay distribution for any customer is the convolution of the waiting time distributions at each of the M service centres if independence of these distributions is assumed as in the PSA case. The waiting time distribution at a FCFS type of centre, i say, again assuming independence of centres, is the convolution of n_i service time distributions, where n_i is the number

of customers at centre i on arrival of the test customer. Thus it is necessary to derive an equation for the p'th moment, M_p, of a convolution of n random variables, $Y_1,\ldots,Y_n$ say.

Now,
$$M_p = E(Y_1+\ldots+Y_n)^p$$
$$= E\left\{ \sum_{\substack{\sum_{i=1}^{n} r_i = p}} p! \prod_{i=1}^{n} \frac{Y_i^{r_i}}{r_i!} \right\}$$

where $r_i \geq 0, \quad 1 \leq i \leq n$

$$\therefore \quad M_p = p! \sum_{\Sigma r_i = p} \prod_{i=1}^{n} \frac{M_{r_i}^{(i)}}{r_i!} \qquad \ldots\ldots\ldots\ldots (E3.1)$$

where $M_{r_i}^{(i)}$ is the r_i'th moment of the random variable Y_i.

Thus the p'th moment, $M_p(\underline{n})$, for the time delay distribution, given the set of queue lengths $\underline{n}$ at the said arrival times, is given by equation (E3.1) with n=M in which, for a FCFS centre, i say

$$M_t^{(i)} = t! \sum_{\substack{\sum_{j=1}^{n_i} k_j = t \\ k_j \geq 0}} \prod_{j=1}^{n_i} \frac{S_{k_j}^{(i)}(q_i)}{k_j!} \qquad \ldots\ldots\ldots\ldots (E3.2)$$

$(1 \leq i \leq M)$

where $S_{k_j}^{(i)}(q)$ is the k_j'th moment of the service time distribution of centre i for a queue length q. At this stage a further approximation is introduced for paths containing servers with non-constant service rates. In this approximate analysis, the

queue length existing at any centre holding the test customer is only considered on arrival of this customer. Thus, the queue length at centre i ($1 \leq i \leq M$) is unknown throughout the sojourn time of the test customer at centre i, except initially. Therefore any choice of q_i in equation (E3.2) is bound to lead to further approximation. To avoid this would involve a much more complicated analysis of the probability distribution of the queue length existing on any service completion at centre i conditional on the initial queue length. Such additional work is not considered worthwhile for the following reasons:-

(i) The PSA method was designed to be simple to apply as a practical tool. The introduction of this new complication would severely limit the domain of network structures for which the PSA method is practicable in view of the vastly increased computing resources required;

(ii) By suitable choice of q_i, the decrease in accuracy introduced ought to be negligible compared with that arising from PSA. The actual quantitive difference is not analysed here, the important validation being between the results of the adopted PSA method and real data, simulated data and exact results, as described in chapter 6.

The choice of q_i ($1 \leq i \leq M$) would typically be the mean queue length at centre i or that giving a throughput (conditional on non zero queue length) equal to that achieved in the equilibrium situation, a simple measure to compute [BUZE73]. In section 3.5 it is shown how this choice results in an exact computation of the mean time delay. In the rest of this chapter it will be assumed that centre service rates are state independent so that the problem does not arise; for the PS and IS type of servers considered below,

values for q_i may be chosen in exactly the same way.

For the case of a PS server i,

$$M_p^{(i)} = S_p\left[\frac{1}{q_i}\ \phi_i\left(\frac{t}{q_i}\right)\right]$$

where: q_i is some form of average for the queue length at centre i, chosen for example as described above for servers with state dependent service rates, and so introducing further approximation;

$\phi_i(t)$ is the service time p.d.f.† of centre i for a queue length of 1 and

$S_p(\phi)$ is the p'th moment of the p.d.f. ϕ.

This is so since for a PS service centre with queue length n>0 and service time p.d.f. f(t), the (cumulative) distribution function of the waiting time, T, for each customer is given by

$$\begin{aligned}\Pr(T\le t \mid \text{queue length } n) &= \Pr(T\le t/n \mid \text{queue length } 1)\\ &= \int_0^{t/n} f(u)\,du\\ &= \int_0^{t} \frac{1}{n} f(v/n)\,dv\end{aligned}$$

using the change of variable u=v/n.

For an IS server i, $M_p^{(i)} = S_p(\phi_i)$ for all queue lengths greater than zero trivially. For LCFS queueing discipline the method is not really applicable for the reason given in the discussion of FCFS centres; the queue length at any centre holding the test customer is unknown after the instant of arrival. Some estimate, as a function of the queue length faced on arrival, for the expected

---

† probability density function.

number of service completions at centre i required before departure of the test customer could be made to yield a value for the number of convolutions necessary. However, the overall effect could be a considerable decrease in accuracy, particularly if there are several centres with LCFS queueing discipline. The reason is that in this case it is in the fundamental principle of the method rather than in the assignment of parameter values that the approximation would be made.

If centre i has exponential service time distribution with mean $\mu_i(n_i)$ for a queue length of n_i, then the n'th moment is well known to be

$$\frac{n!}{\{\mu_i(n_i)\}^n}$$

The expression for the p'th moment of the time delay distribution, $M_p(\underline{n})$ given by (E3.1) is dependent on the queue lengths encountered on arrival at each service centre via the direct dependence on $M_t^{(i)}$ $(1\leq i\leq M)$ which is dependent on n_i. Now suppose the time delay p.d.f. is $\psi(\underline{r},t)$ where r_i $(1\leq i\leq M)$ is the number of customers at service centre i at the time of arrival of the test customer. Then, assuming permanent stationarity and a closed queueing network of M´ centres and population N, the overall time delay distribution, $\Psi(t)$ is given by

$$\Psi(t) = \sum_{r_1=1}^{N}\sum_{r_2=1}^{N}\cdots\sum_{r_M=1}^{N} P(n_i=r_i \mid n_i>0;1\leq i\leq M)\psi(\underline{r},t)$$

$$= \sum_{r_1=1}^{N}\cdots\sum_{r_M=1}^{N}\prod_{i=1}^{M}\frac{P(n_i=r_i)}{P(n_i>0)}\psi(\underline{r},t)$$

assuming without loss of generality that the centres in the chosen path are enumerated $\{1,2,\ldots,M\}$.

$$\text{Thus,} \quad \Psi(t) = \sum_{r_1=1}^{N} \cdots \sum_{r_M=1}^{N} \prod_{i=1}^{M} P'(n_i=r_i)\psi(\underline{r},t) \quad \ldots\ldots(E3.3)$$

$$\text{where} \quad P'(n_i=r_i) = \frac{P(n_i=r_i)}{P(n_i>0)} \qquad (1 \le i \le M)$$

is the re-normalised queue length probability for the case $n_i \neq 0$. This may be evaluated using the expression for the normalising constant G(N) given in [BUZE73] as:

$$P'(n_i=r_i) = \frac{X_i^{r_i-1}}{G(N-1)} \{G(N-r_i)-X_i G(N-r_i-1)\}^{\dagger}$$

This result is in fact a special case (because of the PSA) of that of Mitrani and Sevcik [MITR79]:

At the instant of arrival at a centre in a closed queueing network in a state of equilibrium, a customer sees the equilibrium state space probability distribution for that network with himself removed.

Multiplying equation (E3.3) by t^p and integrating w.r.t. t over the interval $[0,\infty)$ the p'th moment of the overall cycle time distribution is

$$Y_p = \sum_{r_1=1}^{N} \cdots \sum_{r_M=1}^{N} \left\{ \prod_{i=1}^{M} P'(n_i=r_i) \right\} M_p(\underline{r}) \quad \ldots\ldots(E3.4)$$

where P′ is defined as above.

In the APL package of Appendix 7 the first two moments of cycle time distribution are computed by the method described in this section by the function PSM.

---

† $X_i = e_i/\mu_i$ where e_i is the visitation rate and μ_i the service rate of centre i $(1 \le i \le M')$.

3.3 Time Delay Distribution

3.3.1 Its Laplace Transform

The time delay distribution for a path consisting of M centres in a closed queueing network of N customers, $\Psi(t)$, is given, under the assumption of permanent stationarity, by equation (E3.3) in terms of the $\psi(\underline{r},t)$ where $1 \le r_i \le N, 1 \le i \le M$.

In this section, it therefore remains to derive an expression for $\psi(\underline{r},t)$, the p.d.f. of the time delay for a queueing network path in which the number of customers present at its centre i is r_i at the arrival time of the test customer. The weighting of the results for all valid paths is discussed in section 3.4.

Now assuming independence, $\psi(\underline{r},t)$ is the convolution of the waiting time distributions for each centre i in the path taken at which there are r_i customers. Let $F(\underline{r},p) = L(\psi(\underline{r},t))$, the Laplace transform of $\psi(\underline{r},t)$. Then

$$F(\underline{r},p) = \prod_{i=1}^{M} L(\Phi_i(r_i,t))$$

where $\Phi_i(r_i,t)$ is the waiting time distribution at centre i when the queue length there is $r_i > 0$ at the time of arrival of the test customer. That is, for a FCFS queueing discipline

$$\Phi_i(r_i,t) = \overset{r_i}{\underset{j=1}{*}} \phi_i(t)$$

and for a PS discipline

$$\Phi_i(r_i,t) = \frac{1}{q_i} \phi_i(t/q_i)$$

where $\overset{r_i}{\underset{j=1}{*}}$ denotes r_i convolutions, $\phi_i(t)$ is the service time

distribution of centre i, assumed state independent,and q_i is some averaged queue length for centre i, as discussed in the previous section. For an IS centre, q_i is simply set to 1 in the PS case.

For computation of the Laplace transform further use may be made of the independence property of the PSA. The Laplace transform of $\Psi(t)$,

$$L(\Psi(t)) = \sum_{\substack{r_i=1 \\ 1\le i\le M}}^{N} \prod_{j=1}^{M} P'(n_j=r_j)\ L(\Phi_j(r_j,t))$$

$$= \prod_{i=1}^{M} \sum_{r_i=1}^{N} P'(n_i=r_i)\ L(\Phi_i(r_i,t))$$

by the dependence of the factors on only a single value of the subscript i which follows from the assumed independent behaviour of the service centres. The second form of the result obviously provides a far more efficient computation and is performed by the function PSA in the APL package of Appendix 7 for the case of cycle times in tree-like queueing networks.

In section 3.3.2 the case of exponential servers with (the more complex) FCFS queueing discipline is considered, and $\psi(\underline{r},t)$is obtained by inverting the Laplace transform $F(\underline{r},p)$ given in equation (E3.5). In section 3.3.3, a discrete approximation for $\psi(\underline{r},t)$ is derived by direct convolution of the discrete form approximations of its constituent centre service time distributions. The result is convergent as the discrete time step approaches zero, the proof being the same as that for the analogous formulae presented as part of the exact analysis of cycle time distribution in chapters 4 and 5, and not given here.

3.3.2 Inversion of the Laplace Transform

In the case of exponential service time distributions, for example as would be required for the stationary state space probability distribution used in section 3.2,

$$F(\underline{r},p) = \prod_{i=1}^{M} \left\{ \frac{\mu_i}{p+\mu_i} \right\}^{r_i} \qquad \text{.....(E3.5)}$$

for FCFS queueing discipline

and $$F(\underline{r},p) = \prod_{i=1}^{M} \left\{ \frac{\mu_i/q_i}{p + \mu_i/q_i} \right\} \qquad \text{.....(E3.6)}$$

for PS queueing discipline

where μ_i is the service rate of service centre i (when the queue length is 1 for the case of a PS server).

For FCFS servers, the expression for $F(\underline{r},p)$ may be written

$$F(\underline{r},p) = \prod_{i=1}^{L} \left(\frac{\mu_i}{p+\mu_i} \right)^{k_i} \qquad \text{.....(E3.7)}$$

where $\{ \mu_k \mid 1 \le k \le L \} = \{ \mu_i \mid 1 \le i \le M \}$,

L is the number of unique elements in this set and

$$k_i = \sum_{\substack{j=1 \\ r_j = k_i}}^{M} r_j$$

For example, in the case of a network of M centres with non-degenerate (i.e. unique) service rates independent of

queue lengths, L=M and $k_i=r_i$ $(1\le i\le M)$ so that

$$F(\underline{r},p) = \prod_{i=1}^{M} \left(\frac{\mu_i}{p+\mu_i} \right)^{r_i} \quad \text{, equation (E3.5).}$$

The expression on the right hand side of equation (E3.7) may be inverted to give $\psi(\underline{r},t)$ which is rewritten as $\psi(\underline{k},t)$ by evaluation of the Bromwich (contour) integral [SPAI70].

This is carried out in Appendix 1, yielding the result

$$\psi(\underline{k},t) = \prod_{i=1}^{L} \mu_i^{k_i} \sum_{j=1}^{L} \psi_j(\underline{k},t) \quad \text{.....(E3.8)}$$

where
$$\psi_j(\underline{k},t) = \frac{e^{-\mu_j t}}{\prod_{i=1}^{L}(k_i-1)!} \sum_{\substack{\sum_{i=1}^{L} \ell_i = k_j-1 \\ \ell_i \ge 0}} \left\{ \frac{(k_j-1)!\, t^{\ell_j} (-)^{k_j-1-\ell_j}}{\prod_{i=1}^{L} \ell_i!} \prod_{1\le i\ne j\le L} \frac{(k_i+\ell_i-1)!}{(\mu_i-\mu_j)^{k_i+\ell_i}} \right\}$$

.....(E3.9)

A recurrence relation is also derived in Appendix 1 for this result as

$$\psi_j(\underline{k},t) = \frac{Q_j(\underline{k},t)}{\prod_{i=1}^{L}(k_i-1)!} \quad \text{.....(E3.10)}$$

where $$Q_j(\underline{k},t) = tQ_j(\underline{k}^{j-},t) - \sum_{\ell\ne j} Q_j(\underline{k}^{j-,\ell+},t) \quad \text{.....(E3.11)}$$

$(k_j\ge 2 \text{ and } k_i\ge 1 \ , \ 1\le i\ne j\le L)$

with boundary condition

$$Q_j(\underline{k},t) = \left\{ \prod_{1\le i\ne j\le L} \frac{(k_i-1)!}{(\mu_i-\mu_j)^{k_i}} \right\} e^{-\mu_j t} \quad \text{.....(E3.12)}$$

$$(k_j=1 \text{ and } k_i\ge 1 \ , \ 1\le i\ne j\le L)$$

in which $\underline{k}^{j-} = (k_1,\ldots,k_j-1,\ldots,k_L)$

and $\underline{k}^{j-,\ell+} = (k_1,\ldots k_j-1,\ldots,k_\ell+1,\ldots k_L)$

The equivalence of the two forms of this result is also shown.

As an example, consider the case of a single server with L=1 so that, from equation (E3.8),

$$\psi(\underline{k},t) = \psi(k_1,t) = \mu_1^{k_1}\psi_1(k_1,t)$$

Using (E3.11), $Q_1(k_1,t) = tQ_1(k_1-1,t)$

where $Q_1(k_1,t) = (k_1-1)!\psi_1(k_1,t)$

Using (E3.12), $Q_1(1,t) = e^{-\mu_1 t}$

$$\therefore \quad Q_1(k_1,t) = t^{k_1-1}Q_1(1,t) = t^{k_1-1}e^{-\mu_1 t}$$

$$\therefore \quad \psi_1(k_1,t) = \frac{t^{k_1-1}}{(k_1-1)!}e^{-\mu_1 t}$$

which is the familiar Erlang - k_1 distribution. This result could have been obtained immediately from (E3.9), there being only one term in the summation for L=1.

Although the solution for $\psi_j(\underline{k},t)$ is given by equation

(E3.9), the number of terms involved in the summation increases combinatorially with k_j. For numerical computation, the recurrence relation (E3.11) and boundary conditions (E3.12) may well provide a better approach although the obvious recursive solution involves many function calls with consequent large storage and execution time requirements. This difficulty can be alleviated by saving certain intermediate values of the recursive function to avoid later unnecessary recomputation, for example as in [MICH67, HARR74].

An alternative approach to this method of inverting the Laplace transform is to perform the operation numerically using the values of $L(\Psi(t))$ derived in section 3.3.1 corresponding to a suitable set of values for the parameter p. However, in view of the averaging nature of the Laplace transform operation, such inversion is rather difficult though not impossible. This topic is discussed further in chapters 4 and 5.

3.3.3 Discrete approximation for time delay distribution

An alternative approach to deriving the distribution of the time delay by inversion of its Laplace transform is to perform the necessary convolutions directly; this may be achieved numerically by first computing discrete forms of the constituent service time distributions and then performing simple summations. The presentation given here is not fully rigorous, this being left to the parallel development of chapters 4 and 5.

Now,

$$\Psi(t) = \sum_{\substack{r_i=1 \\ 1\le i\le M}}^{N} \prod_{j=1}^{M} {}^{*} P'(n_j=r_j)\Phi_j(r_j,t)$$

is the notation of 3.3.1 where

$$\Phi_j(r_j,t) = \overset{r_j}{\underset{i=1}{*}} \phi_j(t)$$

and $\phi_j(t)$ is the service time distribution of the j'th server, assumed here FCFS.

Thus,

$$\Psi(t) = \overset{M}{\underset{i=1}{*}} \sum_{r_i=1}^{N} P'(n_i=r_i)\Phi_i(r_i,t)$$

by the independence assumption, where a closed network of N customers is under analysis.

Now let $\phi_i(t)$ be represented in discrete form by the probability distribution $H_i(j), j=0,1,2,\ldots,$ then

$$H_i(j) = \Pr\{(j-1)\Delta < t \leq j\Delta\}$$

$$= \begin{cases} \int_{(j-1)\Delta}^{j\Delta} \phi_i(t)\, dt & (j \geq 1) \\ 0 & (j=0) \end{cases}$$

for some time step $\Delta \in R^+$. *

For example, if centre i has negative exponential service time distribution then, as shown in chapter 4, H_i is geometric and the convolutions may be performed via a simple recurrence relation given in Appendix 2.

Let the corresponding discrete forms for the sojourn time distribution of the test customer at centre i, $\Phi_i(r_i,t)$, be $J_i(r_i,j)$ and for the time delay, $\Psi(t)$ be $K(j)$; $j=0,1,2,\ldots,1 \leq i \leq M$. Then

$$J_i(r_i, \;) \simeq \overset{r_i}{\underset{n=1}{*}} H_i(\;)$$

---

* R^+ denotes the set of all positive real numbers.

and $$K(\ ) \simeq \overset{M}{\underset{i=1}{*}} \sum_{r_i=1}^{N} P'(n_i=r_i) J_i(r_i,)$$

Thus K(j) may be computed numerically via the following relationships :

$$K(j) = K_M(j) \qquad (j=0,1,....)$$

where $$K_i(j) = \sum_{k=0}^{j} K_{i-1}(k) \left\{ \sum_{r_i=1}^{N} P'(n_i=r_i) J_i(r_i, j-k) \right\}$$

$$K_0(j) = \begin{cases} 1 & (j=0) \\ 0 & (\text{otherwise}) \end{cases}$$

$$J_i(r_i,j) = \sum_{k=0}^{j} J_i(r_i-1,k)\ H_i(j-k)$$

$$J_i(0,j) = \begin{cases} 1 & (j=0) \\ 0 & (\text{otherwise}) \end{cases}$$

Such a computation requires no assumption about the specific form of the service time distribution and is performed by the function PSD in the APL package of Appendix 7 for the case of tree-like networks.

An approximation has been introduced at an early stage in the analysis and manipulations (viz. convolutions) made on the resulting inexact values. Thus care should be taken to ensure that the error does not grow unacceptably, and it is clear on comparison with the parallel, rigorous analysis in chapters 4 and 5 that the method converges to the exact result as $\Delta \to 0$. The proof need not be repeated here.

3.4 Computation of path selection probabilities

In order to derive the distribution, its Laplace transform or its moments, of a time delay in a queueing network, unconditional on the path chosen, the results for each possible individual path must be weighted according to the probability of choosing the path.

Let P_n be the probability of choosing the valid path n, $\Psi_n(t)$ the time delay distribution conditional on path choice n, $L_n(s)$ its Laplace transform and Y_{np} its p'th moment. Expressions for $\Psi_n(t)$, $L_n(s)$ and Y_{np} have been derived in previous sections of this chapter.

Thus the unconditional time delay distribution, Laplace transform and moments are, with a slight change of notation

$$\Psi(t) = \sum_{\substack{\text{valid paths} \\ n}} P_n \Psi_n(t)$$

$$L(s) = \sum_{\substack{\text{valid paths} \\ n}} P_n L_n(s)$$

$$Y_p = \sum_{\substack{\text{valid paths} \\ n}} P_n Y_{np}$$

by simple laws of conditional probability.

It thus remains to determine $\{P_n | n \text{ a valid path}\}$. This is a trivial problem for networks in which no path includes the same centre more than once, i.e. no path contains a loop such as would be possible in passing from centre α to centre β in the network shown in fig. 3.2 below.

For such cases, P_n is simply derived from the routing probability matrix for the network. Suppose path n consists of centres $c_1, c_2, \ldots . c_\ell$ where $c_1 = \alpha$ and $c_\ell = \beta$. Then

$$P_n = \Pr(\text{Choice of path } n \mid n \text{ is valid})$$

$$= \frac{\Pr(\text{Choice of path } n \;\&\; n \text{ is valid})}{\Pr(n \text{ is valid})}$$

$$= \frac{P_n'}{\sum_{\text{valid } n} P_n'}$$

where $$P_n' = \prod_{i=1}^{\ell-1} p_{c_i c_{i+1}}$$

In particular, the tree-like networks discussed in chapter 5 possess this property so that the APL functions referred to previously, PSM, PSA and PSD, compute the path probabilities quite simply. In fact it is a property of tree-like networks that any path from the "root" centre to a "leaf" centre, i.e. a cycle in a closed tree-like network, is uniquely determined by the identity of the leaf centre. Thus the path choice probabilities are proportional to the visitation rates of the corresponding leaf centres. Values for these are readily available in view of their necessity in the evaluation of state space marginal probabilities, and this is the method used in the package.

In the event that a network contains valid paths which include loops, a problem arises in that the number of valid paths is no longer finite and various approaches to the computation of $\{P_n | n \text{ a valid path}\}$ may be adopted.

The most general and practicable from the programming point of view is iterative. Consider paths from centre α to centre β which may include a loop starting and ending at centre γ as shown in fig. 3.2.

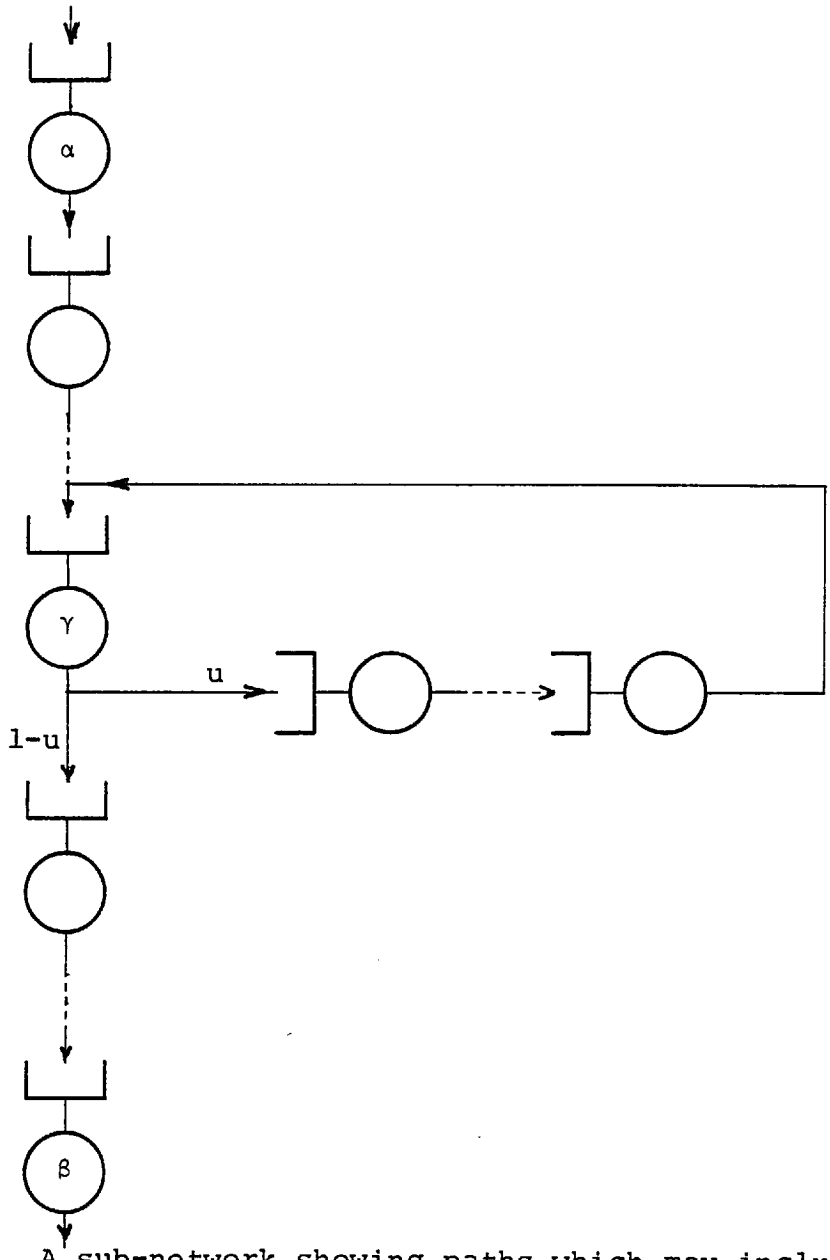


fig. 3.2 A sub-network showing paths which may include a loop.

In the figure the probability of entering the loop on departure from centre γ is u. Clearly more complex cases with nested loops are possible but the method described could be extended to cope with these. It is assumed for the sake of clarity that all paths from α to β must include centre γ, the modification of the method being obvious if this is not the case.

Denote the distribution of the time delay between arrivals at centres λ and μ by $\Psi_{\lambda\mu}(t)$ and the path from α to β which includes i circuits round the loop by $n(i)$, $i \geq 0$.

Now, the probability of i circuits round the loop is

geometric, viz. $(1-u)u^i$, so that

$$P_{n(i)} = P_{n(0)}\, u^i$$

and so

$$\Psi_{\alpha\beta^+} = P_{n(0)} \sum_{i=0}^{\infty} u^i\, \Psi_{\alpha\gamma} * \Psi_{\gamma\gamma}^i * \Psi_{\gamma\beta^+}$$

where β^+ denotes the successor centre to β (possibly exit from the network), $\Psi_{\gamma\gamma}^i$ denotes i convolutions of $\Psi_{\gamma\gamma}$. The summation is clearly convergent since $0<u<1$ and the probability distribution $\Psi_{\gamma\gamma}^i$ is bounded. Thus the computation may proceed iteratively and will terminate when the specified precision is achieved.

Expressions for the Laplace transform and moments of $\Psi_{\alpha\beta^+}$ follow trivially.

An alternative approach in simple cases is to proceed via the Laplace transform from which the moments follow and which may be inverted to give the distribution itself.

Suppose the loop in fig. 3.2 consists of the single centre γ, i.e. the routing probability $p_{\gamma\gamma}=u$. Then if $L_{\lambda\mu}$ is the Laplace transform of the $\Psi_{\lambda\mu}$ defined above,

$$L_{\alpha\beta^+} = P_{n(0)} \sum_{i=0}^{\infty} u^i\, L_{\alpha\gamma}\, (L_{\gamma\gamma})^i L_{\gamma\beta^+}$$

$$= P_{n(0)}\, L_{\alpha\gamma}\, \frac{1}{1-uL_{\gamma\gamma}}\, L_{\gamma\beta^+}$$

The convergence of the series is again clear and the resulting expression for $L_{\alpha\beta^+}(s)$ may be inverted in simple cases by the method described in section 3.3.2 or numerically as discussed later.

3.5 Summary

3.5.1 General remarks

The most significant property of the so-called PSA method for deriving approximate time delay distributions in queueing networks is that it can be applied in a wide range of practical situations to give useful numeric results. This is due to the generality of the network classes which conform to the assumptions underlying the method as well as to the efficiency, with respect to computing resources, arising from the relative simplicity of the calculations.

The only fundamental assumptions of the PSA method are that service centres behave completely independently and that for each centre successive service times are also independent; none of the results presented in this chapter require any assumptions about the form of the service time distributions for each centre as is the case in the traditional derivation of the state space probabilities, [JACK63, GORD67, BASK75]. Hence it would be possible to apply the method to networks in which the service time distributions and/or the queue length probability distributions (marginal state space probabilities) of the servers were empirical. I.e. network specification could be based purely on observations made on the actual system being modelled. This is of great appeal as regards achievement of a representative model in that several intermediate steps are absent compared with the analytic approach. The philosophy is the same as that in the operational approach to queueing network analysis, [BUZE78a, DENN78].

Thus, the PSA method frequently applies to networks of even greater generality than those in the BCMP derivation of state space probabilities [BASK75], although certain network properties cannot be represented easily. In particular we have

already discussed the LCFS queueing discipline and also networks containing loops in a path for which the time delay distribution is required.

In order to achieve such generality and computational efficiency, some fairly strong approximations have been made, recall, for example, the handling of state dependent service rates. However, the method is intuitively sound, reflecting well the flow of customers through a network. Furthermore, some of the results presented are exact in some or all cases:

(i) The mean of the time delay distribution derived by the PSA method is always exact by the following argument.

For any path in a queueing network, the associated time delay is the sum of the sojourn times of the test customer at each centre in the path. It is assumed that these sojourn times are independent as is usual in all queueing network analysis, for example via the Markov property in the Jackson case, [JACK63]. Thus the mean time delay is equal to the sum of the mean sojourn times for each centre in the path.

Now, by Little's Law applied to centre i say,

$$M_i = \frac{Q_i}{T_i}$$

where M_i is the mean sojourn time,
Q_i is the mean queue length,
T_i is the throughput,
} for centre i.

But Q_i and T_i depend only on the centre i queue length probability distribution (marginal state space probability distribution) and its service rate, for which no approximation is made under PSA, by suitable choice of q_i in equation (E3.2)

(section 3.2) if the service rate is not constant.

The actual choice is that giving the equilibrium throughput, conditional on non zero queue length. Then, in equation (E3.2),

$$S_{k_j}^{(i)}(q_i) = \frac{P(n_i>0)}{T_i}$$

where n_i is the queue length random variable for centre i. Then equation (E3.2) gives, for t=1,

$$M_1(i) = \frac{n_i\, P(n_i>0)}{T_i}$$

and equation (E3.1) yields

$$M_1(\underline{r}) = \sum_{i=1}^{M} \frac{r_i P(r_i>0)}{T_i}$$

Thus, equation (E3.4) may be written

$$Y_1 = \sum_{i=1}^{M} \sum_{r_i=1}^{N} \frac{r_i P(r_i>0) P'(n_i=r_i)}{T_i} \sum_{\substack{r_j=1 \\ 1\le j\ne i\le M}}^{N} \prod_{\substack{k=1 \\ k\ne i}}^{M} P'(n_k=r_k)$$

$$= \sum_{i=1}^{M} \frac{Q_i}{T_i} \prod_{\substack{k=1 \\ k\ne i}}^{M} \sum_{r_k=1}^{N} P'(n_k=r_k)$$

$$= \sum_{i=1}^{M} \frac{Q_i}{T_i} \quad \text{as required.}$$

(ii) For networks consisting of only one centre, all the results are exact since the independence assumption of PSA is satisfied trivially.

(iii) For networks containing only one customer, the independence assumption is again satisfied in that only one centre can be busy at any time - that serving the test customer. In this case the next *state* transition is uniquely determined to be that considered in the PSA method, viz. that of departure of the test customer from the centre. Thus the analysis satisfies precisely the same assumptions as those applying to the exact analysis of chapters 4 and 5.

3.5.2 Validation

The accuracy of the approximation is discussed in Appendix 8 where comparisons are made with the exact results for specific cases of cycle times in tree-like networks and validation is performed with respect to simulated results. Of course the ultimate test is validation based on observations made on at least one actual computer system, but as pointed out in the Introduction and chapter 6 such data is exceedingly difficult to obtain and absolute validation remains an area for future investigation.

Furthermore, rather than attempting a limited validation by comparison with results, real or simulated, corresponding to a few specific network structures, ideally a formal error analysis should be made. This could provide simply computable bounds on the error of the approximate PSA method as a function of the parameters of the network under analysis. Such an analysis is proposed in chapter 8, but is expected to pose many problems. Consequently for the time being we adopt the approach taken by many others, e.g. [BARD79, PUJO79], of making intuitively good approximations and validating as discussed in the previous paragraph via numerical comparisons.

3.5.3 Cycle and Response times

Perhaps the most typical time delay requiring analysis is the cycle time in a closed network, or the sum of (successive)

such cycle times which can represent response time. The former is easily computed directly since it is a time delay of the type discussed in this chapter. The distribution of the sum of cycle times may have its moments computed via equation (E3.1), successive cycles being independent under PSA. Furthermore, because of the same independence assumption, the Central Limit Theorem may be applied giving an asymptotic Normal distribution for the convolution representing response time, as the number of cycles involved becomes sufficiently large. The use of the Central Limit Theorem for this purpose is discussed in more detail in chapters 4 and 8.

3.5.4 Conclusion

In summary, in contrast to chapters 4 and 5, this chapter is not intended to provide a formal (approximate) solution to a precise class of problems. Rather a methodology has been described for giving approximate results, in a wide range of modelling situations, to be validated by comparison with empirical data based on observations taken from the actual computer system being studied. The approach taken has been to make approximations, sometimes rather coarse, in order to provide a feasible method for producing quantitive estimates of time delay distributions in queueing networks. Thus the main value of the work is its ease of implementation on a computer so providing a practical and efficient tool for the performance analyst - contrasting with the exact methods described in the following chapters, see Appendix 8. Applications for such a tool have been given in chapter 2 and further details are discussed in chapter 8.

§4. Exact cycle time distribution for cyclic queueing networks

4.1 Introduction

In chapter 2 the importance of studying cycle time as a particular case of a time delay in queueing networks was emphasised; both in its own right and as the constituent of the response time of a network, i.e. the time delay which results from several (successive) cycles of the test customer. In this chapter a method is described for deriving the cycle time distribution for customers in cyclic networks of servers with constant service rates and FCFS queueing disciplines.

In section 4.2 the approach taken is described and the resulting solution is derived as an analytic expression for the Laplace transform in section 4.3. In section 4.4 it is shown how to compute the moments of the distribution via numerical differentiation of its Laplace transform and in the following section an expression for the moments is derived via analytic differentiation. In section 4.6 a discrete form of the distribution itself is derived and is shown to be convergent as the discrete time step approaches zero.

In section 4.7 some computational techniques for performing the operations required on the large data structures resulting from the preceding analysis are described. In the following two sections discussion is presented on the desirability of and difficulties in inversion of the Laplace transform of the distribution, and on the use of decomposition techniques to link together sub-cycles as an aid to efficiency of computation. In section 4.10 application of the theory to response time prediction is discussed and the chapter closes with a short summary of the chief relevance of the analysis.

4.2 Approach taken

The cyclic networks analysed in this chapter are of the Jackson type, [JACK63], following a Markov process, consisting of M servers, with FCFS queueing discipline and negative exponential service time distributions, and containing a population of N customers.

The first step in the analysis is to consider the corresponding tandem network consisting of the service centres in the same sequence, but with the last centre no longer connected to the first. There are no external arrivals and departures from the network occur at the last centre.

Formally, for a cyclic network of M centres numbered 1,2,...,M, let the routing probability matrix, p, be given by

$$p_{i,i+1} = 1 \qquad 1 \le i \le M-1$$

$$p_{M1} = 1$$

$$p_{ij} = 0 \qquad \text{otherwise}$$

Then in the corresponding tandem network, the new routing probability matrix, p', is given by

$$p'_{i,i+1} = 1 \qquad 1 \le i \le M-1$$

$$p'_{ij} = 0 \qquad \text{otherwise}$$

Of course the corresponding tandem network is not unique since the first centre could be any of the M centres.

The method consists of the following steps:

(i) On arrival of a test customer at centre 1 in the closed network, the steady state probability distribution for the state space of the network is assumed. Thus the result

presented by Mitrani & Sevcik [MITR79] can be applied. This states that the state space probability distribution seen by the arriving customer is the same as the equilibrium distribution for the same network with itself removed. The same result is obtained, for the classes of network considered here, by renormalising the state space probabilities and excluding states in which there are no customers at centre 1. This is the method used in chapter 3.

(ii) The corresponding open network is now considered. The cycle time in the closed network is the same as the time taken for the test customer to depart from the open network if the assumption is made that returning customers joining queues behind the test customer can have no effect on the rate of progress of the test customer through the network, i.e. that departed customers can be disregarded. Hence it must not be possible for customers to be overtaken by other customers, i.e. the cyclic ordering of customers must be invariant, and the service rates of the servers must be unaffected by the addition of new customers to their queues which is equivalent to demanding constant service rates. Order invariance is ensured by the requirement of a FCFS queueing discipline at all centres together with the existence of only one path in the network. Note that PS discipline is precluded by both the order invariance and service rates requirements.

The invariance of order in both the closed and open networks implicitly tags the test customer in the open one in that it is always the leftmost (or furthest from departure) and its position is therefore always known uniquely. Such implicit

tagging, although not possible for networks of the most general type, results in a much smaller state space than would be required in the analysis of the Markov process with an additional state space dimension included for the "tagging" information, as in [YU77] for example. Thus it improves computational efficiency.

With the assumptions listed above, the cycle time for the test customer is identical to the time taken for the open network to empty, i.e. the time taken for the open network to enter the state with zero customers at all centres. Now, the network can empty by passing through any of a (finite) number of (finite) sequences of state transitions. Thus for any particular sequence, the conditional cycle time distribution is the convolution of the distributions of the sojourn times for each state in the sequence, by the Markov property. The unconditional cycle time distribution is therefore a weighted sum of convolutions of state sojourn time distributions, the weights being the probabilities of occurrence of the corresponding sequences of states. The following formal analysis proceeds on this basis.

4.3 Analytic solution for the Laplace Transform

First, we define some notation:

Consider a cyclic network of M centres and N customers and corresponding open, tandem network with state space

$$S_{NM} = \{\underline{n} \mid n_i \geq 0,\ 1 \leq i \leq M;\ \sum_{i=1}^{M} n_i \leq N\}$$

and state transition matrix T defined by

$$T_{\alpha\beta} = \Pr(\text{Next state is } \beta \mid \text{present state in } \alpha)$$

$$(\alpha, \beta \in S_{NM})$$

The state transitions constitute the embedded Markov chain in the continuous time Markov process assumed for the network.

Define $$R_{st} = \{(i_1, i_2, \ldots, i_n) \mid n \in Z^+; i_j \in S_{NM},\ 1 \le j \le n;$$

$$i_1 = s;\ i_n = t;\ T_{i_k i_{k+1}} \neq 0,\ 1 \le k < n\}$$ *

i.e. the set of all sequences of states entered, or *routes*, from state s to state t.

If $$\underline{i} = (i_1, i_2, \ldots, i_n) \in R_{st}$$

then let $|\underline{i}| = n$, the number of steps in the route $\underline{i}$.

Lemma L4.1

For all $s,t \in S_{NM}$, if $\underline{r}_1, \underline{r}_2 \in R_{st}$ then $|\underline{r}_1| = |\underline{r}_2|$ and the number of departures from each *centre* is identical for $\underline{r}_1$ and $\underline{r}_2$.

In other words, all routes from state s to state t have the same length and contain the same transitions between centres (as opposed to states).

Proof

If $R_{st} = \phi$, the result is trivial.

For $R_{st} \neq \phi$, consider the leftmost centre with at least one customer in state s, ℓ say. Since there are no external arrivals to the network,

---

* Z^+ is the set of positive integers.

(a) $t_i = 0$ for $i<\ell$

(b) For all $r \in R_{st}$, total no. of arrivals to centre $\ell = 0$

(c) Thus the no. of departures from centre $\ell = s_\ell - t_\ell$ for all $r \in R_{st}$

(d) Thus the total no. of arrivals to centre $\ell+1 = s_\ell - t_\ell$ for all $r \in R_{st}$

Now consider centre k, $\ell \le k \le M$ and assume that for all $r \in R_{st}$ the number of arrivals to centre k is identical; a_k say.

Then a_{k+1} = no. of departures from centre k

$= a_k + s_k - t_k$ for all $r \in R_{st}$

$\therefore$ a_{k+1} is identical for all $r \in R_{st}$

The result was shown to be true for $k=\ell$, $\ell+1$ and is trivially so for $k<\ell$.

Therefore by induction, for all $r \in R_{st}$, the number of arrivals to each centre is identical, which proves the lemma.

Corollary CL4.1

$\{T^n | n = 1,2,....\}$ are disjoint

Proof

$(T^n)_{ij}$ is the probability that n transitions after being in state i, the network is in state j. But n is uniquely determined by i,j so that

$(T^n)_{ij} \neq 0$ for at most one n.

Corollary CCL4.1

$r_i \neq r_j$ for $1 \le i \neq j \le |r|$, $r \in R_{st}$

Proof

If not, let $r_i = r_j = u$ for $i < j$. Then $\{(u),(r_i,r_{i+1},\ldots,r_j)\} \subset R_{uu}$ contradicting the Lemma.

For $\alpha,\beta \in S_{NM}$, $R_{\alpha\beta} \neq \phi$, let the state transitions for route $r \in R_{\alpha\beta}$ occur at times $\tau_i^{(r)}$ $(1 \leq i \leq |r|-1)$ and let the state of the network at time t be $X^{(r)}(t)$. Define $\tau_0^{(r)} = 0$ and

$$\delta_i^{(r)} = \tau_i^{(r)} - \tau_{i-1}^{(r)} \quad (r \in R_{\alpha\beta} \ , \ 1 \leq i \leq |r|-1)$$

Now let $R'_{\alpha\beta} = \{r | r \in R_{\alpha\beta}; r_i \neq \beta, 1 \leq i < |r|\} = R_{\alpha\beta}$ here by Corollary CCL4.1. $R'_{\alpha\beta}$ is used for increased generality, see below.

The cumulative distribution function of the time delay, C, for the network to pass from state α to state β, $(\alpha,\beta \in S_{NM})$, is

$$G_{\alpha\beta}(t) = \sum_{r \in R'_{\alpha\beta}} \Pr(r|\alpha,\beta) \Pr(C \leq t | r)$$

since the end states α,β are implied by the route r, and where $\Pr(r|\alpha,\beta) = \Pr\{r | r_1 = \alpha, r_{|r|} = \beta\}$. Now,

$$\Pr(C \leq t|r) = G(t|r) = \Pr\left(\sum_{i=1}^{|r|-1} \delta_i^{(r)} \leq t \right)$$

For simplicity, the superscript r will be omitted from the variables X, τ_i and δ_i, its presence being implicit. Thus,

$$\begin{aligned} G(t|r) &= \int_0^t \Pr(\delta_{|r|-1} \leq t-u \mid \tau_{|r|-2} = u) \, d\Pr(\tau_{|r|-2} \leq u) \\ &= \int_0^t \Pr(\delta_{|r|-1} \leq t-u) \, d\Pr(\tau_{|r|-2} \leq u) \end{aligned}$$

since $\delta_{|r|-1}$ is determined solely via X(t) for $t \geq \tau_{|r|-2}$, $\tau_{|r|-2}$ being a Markov time.

Let $d_\nu(t)$ be the distribution function for the time spent in state ν, so that

$$d_{X(\tau_{i-1}^+)}(t) = \Pr(\delta_i \leq t)$$

where τ_{i-1}^+ denotes a time τ such that $\tau_{i-1} < \tau < \tau_i$

Thus, by induction and since convolution is commutative (Appendix 3),

$$G(t|r) = \mathop{*}_{i=1}^{|r|-1} d_{X(\tau_{i-1}^+)}(t)$$

which is in fact a well-known property of Markov processes.

Now, $X(\tau_{i-1}^{+}) \equiv r_i$, the i'th element of route r, so that

$$G(t|r) = \overset{|r|-1}{\underset{i=1}{*}} d_{r_i}(t) \quad \text{and so}$$

$$G_{\alpha\beta} = \sum_{r \in R'_{\alpha\beta}} Pr(r|\alpha,\beta) \overset{|r|-1}{\underset{i=1}{*}} d_{r_i}$$

$$\text{But} \quad Pr(r|\alpha,\beta) = \prod_{i=1}^{|r|-1} T_{r_i r_{i+1}}$$

$$\therefore \text{ for } \alpha \neq \beta, \quad G_{\alpha\beta} = \sum_{\gamma \in S_{NM}} T_{\alpha\gamma}\, d_\alpha * G_{\gamma\beta} \qquad \ldots\ldots\ldots (E4.1)$$

where $G_{\gamma\gamma}(t) = 1$ for all $t \geq 0$, $\gamma \in S_{NM}$. This is so since if $\nexists\, r \in R_{\alpha\beta}$ with $r_2 = \gamma$ then either $T_{\alpha\gamma} = 0$ or $R_{\gamma\beta} = \phi$ so that $G_{\gamma\beta} \equiv 0$.

$$\text{Let} \quad L(s) = \int_0^\infty e^{-st}\, dG(t) \quad ,$$

the Laplace transform of G(t) and let $D_\nu(s)$ be the Laplace transform of $d_\nu(t)$, $\nu \in S_{NM}$. Then, in similar notation, for $\alpha \neq \beta$,

$$\left.\begin{aligned} L_{\alpha\beta}(s) &= \sum_{r \in R'_{\alpha\beta}} \prod_{i=1}^{|r|-1} T_{r_i r_{i+1}}\, D_{r_i}(s) \\ \text{or} \quad &\sum_{\gamma \in S_{NM}} T_{\alpha\gamma}\, D_\alpha(s)\, L_{\gamma\beta}(s) \end{aligned}\right\} \ldots\ldots (E4.2)$$

where $L_{\gamma\gamma}(s) \equiv 1$, for all $s \geq 0$, $\gamma \in S_{NM}$.

This result applies to networks in general - in fact to any such Markov processes. Here it may be written, for all $\alpha,\beta \in S_{NM}$, $\delta_{\alpha\gamma} L_{\gamma\beta} = D_\alpha T_{\alpha\gamma} L_{\gamma\beta} + \delta_{\alpha\beta}$ (with summation convention on $\gamma \in S_{NM}$) since for $T_{\alpha\gamma} \neq 0$, $R_{\gamma\alpha} = \phi$ by Corollary CCL4.1 and $L_{\gamma\alpha} = 0$.

Under the FCFS and Markovian assumptions, the service time distribution for each server must be exponential, for example see [FELL62]. Let the service rate of centre j be μ_j $(1 \leq j \leq M)$ and the total service rate in state $\underline{u} \in S_{NM}$ be $\lambda_{\underline{u}}$, so that

$$\lambda_{\underline{u}} = \sum_{\substack{u_j>0 \\ 1\le j\le M}} \mu_j$$

Define the function θ by

$\theta(u,v)$ = number of the centre from which a departure causes a state transition $u \to v$ $(u,v \in S_{NM})$ which is undefined if a one-step transition $u \to v$ is not possible.

Then, by inspection of the balance equations for the network or the instantaneous transition rate matrix for the underlying Markov process,

$$T_{uv} = \begin{cases} \dfrac{\mu_{\theta(u,v)}}{\lambda_u} & \text{if a one-step transition } u \to v \text{ is valid} \\ 0 & \text{otherwise} \end{cases}$$

By the exponential assumption, the time spent in state u is exponentially distributed with mean λ_u^{-1} (see Appendix 4) and so

$$D_u(s) = \frac{\lambda_u}{s + \lambda_u}$$

The modified transition matrix, T^*, is defined by

$$T^*_{uv} = \begin{cases} \dfrac{\mu_{\theta(u,v)}}{s + \lambda_u} & \text{if a one-step transition } u \to v \text{ is valid} \\ 0 & \text{otherwise} \end{cases}$$

Thus, from equation (E4.2), dropping the prime from $R_{\alpha\beta}$,

$$L_{\alpha\beta}(s) = \sum_{r \in R_{\alpha\beta}} \prod_{i=1}^{|r|-1} T^*_{r_i r_{i+1}} \qquad (\alpha,\beta \in S_{NM})$$

where for $\alpha = \beta$, $R_{\alpha\beta} = \{\alpha\}$ and the empty product gives $L_{\alpha\alpha}(s) = 1$ as required.

$$\text{Thus, } L_{\alpha\beta}(s) = \sum_{k=1}^{\infty} \sum_{\substack{r\in R_{\alpha\beta} \\ |r|=k}} \prod_{i=1}^{k-1} T^*_{r_i r_{i+1}}$$

For cyclic networks, by Lemma L4.1, only one term in the first summation is non-zero, but the result still holds in general networks and is used in section 5.4. Now, for $k \geq 3$,

$$\sum_{\substack{r\in R_{\alpha\beta} \\ |r|=k}} \prod_{i=1}^{k-1} T^*_{r_i r_{i+1}} = \sum_{\gamma_{k-1}\in S_{NM}} \left\{ \sum_{\substack{r'\in R_{\alpha\gamma_{k-1}} \\ |r'|=k-1}} \prod_{i=1}^{k-2} T^*_{r'_i r'_{i+1}} \right\} T^*_{\gamma_{k-1},\beta}$$

for if $\nexists\, r \in R_{\alpha\beta}$ with $r_{k-1} = \gamma_{k-1}$ then either no one-step transition $\gamma_{k-1} \to \beta$ exists so that $T^*_{\gamma_{k-1},\beta} = 0$, or $R_{\alpha\gamma_{k-1}} = \phi$.

By a simple inductive argument, equation (E4.2) yields the result

$$\sum_{\gamma_2\in S_{NM}} \cdots \sum_{\gamma_{k-2}\in S_{NM}} \sum_{\gamma_{k-1}\in S_{NM}} T^*_{\alpha\gamma_2} T^*_{\gamma_2\gamma_3} \cdots T^*_{\gamma_{k-2}\gamma_{k-1}} T^*_{\gamma_{k-1},\beta} = \left[\{T^*\}^{k-1} \right]_{\alpha\beta}$$

$$\text{and so } L_{\alpha\beta}(s) = \sum_{k=1}^{\infty} \{T^*\}^{k-1}_{\alpha\beta}$$

Note that for cyclic networks the single term on the r.h.s. is trivially obtained from the second form of equation (E4.2). Thus,

$$L_{\alpha\beta}(s) = (I - T^*)^{-1}_{\alpha\beta} \qquad \ldots\ldots\ldots (E4.3)^{\dagger}$$

since T is a stochastic matrix and for $s>0$, $D_u(s)<1$ for all $u\in S_{NM}$, so the series converges. In fact here $\exists k\in Z^+$ s.t. $(T^*)^k = 0$ since ultimately the network has no customers and can have no transitions.

Define $S_I = \{ \underline{n} \mid \underline{n}\in S_{NM}, \Sigma n_i = N, n_1 > 0 \}$, the set of initial states with the test customer at centre 1. In what follows, the state labelled "o" is that in which the open network contains no customers and state $\alpha \in S_I$. Let the cycle time distribution conditional on start state α be denoted by $G(t|\alpha)$ with Laplace transform $L(s|\alpha)$ so that $G(t|\alpha) = G_{\alpha o}(t)$ and $L(s|\alpha) = L_{\alpha o}(s)$.

---

† This clearly satisfies equations (E4.2) which may be written $(I-T^*)L = I$ in matrix form by Corollary CCL4.1.

The unconditional cycle time distribution,

$$G(t) = \sum_{\alpha \varepsilon S_I} Z(\alpha)\ G(t|\alpha)$$

where $Z(\alpha)$ is the equilibrium probability for state α in the closed network. This has Laplace transform

$$L(s) = \sum_{\alpha \varepsilon S_I} Z(\alpha)\ (I - T^*)^{-1}_{\alpha o} \qquad \ldots\ldots\ldots\ldots (E4.4)$$

Now, using the result in [MITR79], for $\underline{n} \varepsilon S_I$

$$Z(\underline{n}) = \frac{\mu_1 \prod_{i=1}^{M} \mu_i^{-n_i}}{G(N-1)}$$

where $G(N-1)$ is the normalising constant for the closed network with one customer removed. The visitation rates are all taken to be 1 since the network is cyclic.

Computational techniques for efficient evaluation of the Laplace transform, $L(s)$, are discussed in section 4.7.3.

4.4 Numerical evaluation of the moments of cycle time distribution

The p'th moment of the cycle time distribution is well known to be $p!M_p$ where

$$M_p = \frac{(-)^p}{p!} \left[\frac{d^p}{ds^p} L(s)\right]_{s=0}$$

so that any moment may be evaluated by numerical evaluation of the derivatives of $L(s)$ at the origin. This is of course the most difficult area in which to perform the calculation, but the simple linear method has been found to converge perfectly adequately for calculation of the first two moments. Thus

$$M_1 = - \lim_{\Delta \to 0} \frac{L(\Delta) - 1}{\Delta}$$

$$M_2 = \tfrac{1}{2} \lim_{\Delta \to 0} \frac{L(2\Delta) - 2L(\Delta) + 1}{\Delta^2}$$

The method used in the APL function NM (Appendix 7) evaluates M_1 and M_2 with successively smaller values for Δ until the user defined precision is achieved, whereupon the mean and standard deviation are produced.

Clearly more efficient and reliable numerical techniques exist for computation of the moments, and may be implemented, especially for the higher derivatives. However, discussion of such techniques is not appropriate here.

4.5 Analytic result for cycle time distribution moments

From the expression for the Laplace transform of the cycle time distribution (E4.4), a formula for the p'th moment of the distribution may be derived in terms of the derivatives of T^* with respect to s. The result is given by propositions P4.1 and P4.2 below, using Lemma L4.2 which precedes them.

Lemma L4.2

Let A(s) be an (n x n) matrix with elements dependent on the variable s. Assuming A is differentiable w.r.t. s,

$$\frac{d}{ds}(A^{-1}) = -A^{-1} A' A^{-1}$$

where $$A' = \frac{dA}{ds}$$

Proof

Let $B = A^{-1}$

Then $\sum_{j=1}^{n} B_{ij} A_{jk} = \sum_{j=1}^{n} A_{ij} B_{jk} = \delta_{ik} \quad (1 \le i,k \le n)$

where δ_{ik} is the Kronecker delta.

$\therefore \quad A_{ij} B'_{jk} = - A'_{ij} B_{jk}$

where the summation convention is applied for doubly occurring suffices.

$\therefore \quad B_{\ell i} A_{ij} B'_{jk} = - B_{\ell i} A'_{ij} B_{jk}$

$\therefore \quad B'_{\ell k} = \delta_{\ell j} B'_{jk} = -(BA'B)_{\ell k}$ as required.

Proposition P4.1

Let $F(s) = (I - A(s))^{-1}$

Then

$$F^{(p)}(s) = \frac{d^p}{ds^p} F(s) = p! \sum_{\substack{\sum_{i=1}^{|\underline{m}|} m_i = p \\ m_j > 0 \\ 1 \le j \le |\underline{m}|}} \left\{ \prod_{i=1}^{|\underline{m}|} \frac{(I-A)^{-1} A^{(m_i)}}{m_i !} \right\} (I-A)^{-1}$$

where $A^{(m)} = \frac{d^m A}{ds^m}$ and $|\underline{m}|$ is the number of components in $\underline{m}$ so that $|\underline{m}| \le p$.

Proof

The proof is by induction on p.

For the case p=1, the formula gives

$$F^{(1)}(s) = \sum_{m_1=1} 1! \prod_{1}^{1} (I-A)^{-1} A^{(1)} (I-A)^{-1}$$

$$= (I-A)^{-1} A^{(1)} (I-A)^{-1}$$

which is true by Lemma L4.2

Now assume the result is true for derivatives up to the p'th.

$$F^{(p+1)}(s) = \frac{d}{ds} F^{(p)}(s)$$

$$= X + Y \quad \text{say}$$

where

$$X = \sum_{\substack{\Sigma m_i = p \\ m_i > 0}} p! \sum_{j=1}^{|\underline{m}|} \left\{ \prod_{i=1}^{j-1} \frac{(I-A)^{-1} A^{(m_i)}}{m_i!} \right\} \frac{(I-A)^{-1} A^{(m_j+1)}}{m_j!} \left\{ \prod_{i=j+1}^{|\underline{m}|} \frac{(I-A)^{-1} A^{(m_i)}}{m_i!} \right\} (I-A)^{-1}$$

and

$$Y = \sum_{\substack{\Sigma m_i = p \\ m_i > 0}} p! \sum_{j=1}^{|\underline{m}|+1} \left\{ \prod_{i=1}^{j-1} \frac{(I-A)^{-1} A^{(m_i)}}{m_i!} \right\} \frac{(I-A)^{-1} A^{(1)}}{1!} \left\{ \prod_{i=j}^{|\underline{m}|} \frac{(I-A)^{-1} A^{(m_i)}}{m_i!} \right\} (I-A)^{-1}$$

with slightly abbreviated notation.

First consider X and for each $\underline{m}, j$ define $\underline{n}$ by

$$n_i = m_i \qquad (1 \le i \ne j \le |\underline{m}|)$$

$$n_j = m_j + 1$$

Then,

$$X = \sum_{\substack{\Sigma n_i = p+1 \\ n_i > 0}} p! \sum_{\substack{j=1 \\ n_j > 1}}^{|\underline{n}|} \left\{ n_j \prod_{i=1}^{|\underline{n}|} \frac{(I-A)^{-1} A^{(n_i)}}{n_i!} \right\} (I-A)^{-1}$$

Now define $\underline{k}$ for each $\underline{m}, j$ by

$$k_i^{(j)} = m_i \qquad (1 \le i < j)$$

$$k_j^{(j)} = 1$$

$$k_i^{(j)} = m_{i-1} \qquad (j < i \le |\underline{m}|+1)$$

Then,

$$Y = \sum_{\substack{\Sigma k_i = p+1 \\ k_i > 0}} p! \sum_{\substack{j=1 \\ k_j=1}}^{|\underline{k}|} \left\{ k_j \prod_{i=1}^{|\underline{k}|} \frac{(I-A)^{-1}A^{(k_i)}}{k_i!} \right\} (I-A)^{-1}$$

Thus, re-labelling k_i by n_i in the expression for Y,

$$X+Y = \sum_{\substack{\Sigma n_i = p+1 \\ n_i > 0}} p! \sum_{j=1}^{|\underline{n}|} \left\{ n_j \prod_{i=1}^{|\underline{n}|} \frac{(I-A)^{-1}A^{(n_i)}}{n_i!} \right\} (I-A)^{-1}$$

but $\sum_{j=1}^{|\underline{n}|} n_j = p+1$, so

$$F^{(p+1)}(s) = X + Y = \sum (p+1)! \left\{ \prod_{i=1}^{|\underline{n}|} \frac{(I-A)^{-1}A^{(n_i)}}{n_i!} \right\} (I-A)^{-1}$$

which proves the proposition.

Proposition P4.2

The p'th moment ($p \ge 1$) of cycle time distribution, $p!M_p$, is given by

$$M_p = \sum_{\underline{n}\in S_I} Z(\underline{n}) \left[\sum_{\substack{\sum_{i=1}^{|\underline{m}|} m_i = p \\ m_i > 0}} \left\{ \prod_{i=1}^{|\underline{m}|} (I-T)^{-1} T'(m_i) \right\} (I-T)^{-1} \right]_{\underline{n}\, o}$$

in the notation of section 4.3 wherein $Z(\underline{n})$ is given and where

$$T'_{uv}(m_i) = \frac{T_{uv}}{\lambda_u^{m_i}}$$

Proof

Setting $A = T^*$ and $s=0$ in P4.1,

$$A(0) = T \quad \text{and}$$

$$A^{(m_i)}(0) = (-)^{m_i} m_i ! \frac{T_{uv}}{\lambda_u^{m_i}}$$

so $$F^{(p)}(0) = p! \sum_{\substack{\Sigma m_i = p \\ m_i > 0}} \left\{ \prod_i (I-T)^{-1} (-)^{m_i} T'(m_i) \right\} (I-T)^{-1}$$

$$= (-)^p p! \sum_{\substack{\Sigma m_i = p \\ m_i > 0}} \left\{ \prod_i (I-T)^{-1} T'(m_i) \right\} (I-T)^{-1}$$

and $$M_p = \frac{(-)^p}{p!} \sum_{\underline{n}\in S_I} Z(\underline{n}) \left[F^{(p)}(0) \right]_{\underline{n}\, o}$$ as required.

Computational techniques for the evaluation of the moments of cycle time distribution are discussed in section 4.7.4, with particular attention paid to the first two moments. Frequently these moments will be the only ones required, in particular if independence of successive cycles is assumed and the Central Limit Theorem applied to give predictions about response time distribution.

4.6 Discrete form of cycle time distribution

4.6.1 Introduction

So far the cycle time distribution for cyclic queueing networks, under certain assumptions, has been derived exactly in the form of its Laplace transform, section 4.3, and its set of moments, sections 4.4 and 4.5. In this section a technique is described by which the distribution itself can be derived approximately in discrete form. The approach taken is to express the constituent, negative exponential, distributions involved in the weighted convolutions of section 4.3 in discrete form so that the convolutions can be computed directly, without resorting to the method of multiplication of Laplace transforms and subsequent inversion by some means. Of course this introduces an approximation in the method at the very first step, but by choosing a sufficiently fine mesh on the time axis, a good representation of the exponential distributions can be achieved. The numerical results obtained for cycle time distribution show good agreement with corresponding simulated results as well as for the mean and standard deviation computed exactly as described in the preceding sections (chapter 6). Furthermore, it is shown in this section that the result of the method converges to the exact distribution as the mesh size approaches zero.

In any case, it must be remembered that any queueing network analysis representation of computer systems is inevitably approximate anyway. There is no reason to presume that the discrete form of the negative exponential distribution (step function in continuous time) is any worse an approximation than the more commonly used continuous form, even though the latter form is a necessary assumption for Markovian analysis. Thus, even without the convergence property referred to above, the discrete analysis would be worthwhile in its own right.

4.6.2 Deviation of the approximate result

Recall from section 4.3 that the cycle time distribution for a cyclic network is denoted by $G(t)$ and that of time spent in state ν by $d_\nu(t)$, $\nu \varepsilon S_{NM}$, where $d_\nu(t)$ is negative exponential with mean λ_ν^{-1}.

Now, the discrete form of a continuous (cumulative) probability distribution $F(t)$, $t \geq 0$, with mesh $t_j = j\Delta$ of size Δ, $j=1,2,\ldots$. defined on the t-axis, may be defined approximately for a random variable $J \varepsilon Z^+$ by

$$Pr(J \leq j) = P(j) = F(t_j)$$

so that the (non-cumulative) discrete distribution

$$Pr(J = j) = p(j) = F(t_j) - F(t_{j-1}) \qquad (t \geq 2)$$

and $$p(1) = F(t_1) = F(\Delta)$$

Thus, in the case of a negative exponential distribution with mean λ^{-1} for F,

$$p(j) = e^{-\lambda\Delta(j-1)} - e^{-\lambda\Delta j} \qquad (j \geq 1)$$

i.e. $$p(1) = 1 - e^{-\lambda\Delta}$$

$$p(j) = e^{-\lambda\Delta}\, p(j-1) \qquad (j \geq 2)$$

which is geometric, where an integer random variable J corresponds to continuous time $T = J\Delta$.

Let $x_\nu = e^{-\lambda_\nu \Delta}$ and p_ν be the discrete approximation for d_ν, the distribution of the sojourn time in state ν of the network.

Then, $$p_\nu(j) = (1 - x_\nu)\, x_\nu^{j-1} \qquad (j \geq 1)$$

and $p_\nu(0)$ is defined to be 0.

Denote the approximate discrete time delay distribution from state s to state t≠s by Π_{st} ; $s,t \in S_{NM}$.

Now, in the discrete domain the Z-transform or probability generating function is analogous to the Laplace transform in the continuous domain in that the Z-transform of the convolution of (discrete) probability distributions is equal to the product of the Z-transforms of the individual constituents.

Let the Z-transform of Π_{st} be denoted by $H_{st}(z)$ defined as

$$H_{st}(z) = \sum_{i=0}^{\infty} \Pi_{st}(i)\, z^i \qquad (s,t \in S_{NM})$$

where $\Pi_{st}(0) = \delta_{st}$

so that the probability of passing from state t to itself in time zero is unity.

Then, by a derivation identical to that given in section 4.3 for the Laplace transform in the continuous time domain, the result analogous to equation (E4.3) is

$$H_{st}(z) = \{I - T^{\dagger}(z)\}^{-1}_{st} \qquad \text{.....(E4.5)}$$

where $$T^{\dagger}_{uv}(z) = T_{uv} \sum_{i=1}^{\infty} z^i\, x_u^{\,i-1}\,(1 - x_u) \qquad (u,v \in S_{NM})$$

the summation being the Z-transform of p_u.

Thus, $$T^{\dagger}_{uv}(z) = \frac{T_{uv}(1 - x_u)z}{1 - x_u z}$$

The fact that state t is not, in general, o does not affect the analysis.

Now, from equation (E4.5),

$$H - T^{\dagger} H = I$$

in matrix notation, pre-multiplying by the right-hand side.

$$\therefore \quad H_{st} = \delta_{st} + \sum_{u \varepsilon S_{NM}} T^{\dagger}_{su} H_{ut}$$

so that

$$(1 - x_s z) H_{st} = (1 - x_s z) \delta_{st} + (1 - x_s) z \sum_{u \varepsilon S_{NM}} T_{su} H_{ut}$$

Then, by comparing coefficients of the powers of z,

z^0 : $\quad \Pi_{st}(0) = \delta_{st}$ as defined

z^1 : $\quad \Pi_{st}(1) = (1-x_s) \sum_{u \varepsilon S_{NM}} T_{su} \Pi_{ut}(0)$

$z^j, j \geq 2$: $\quad \Pi_{st}(j) = x_s \Pi_{st}(j-1) + (1-x_s) \sum_{u \varepsilon S_{NM}} T_{su} \Pi_{ut}(j-1)$

Now define $x_t = 0$ so that the transition time from state t is zero.

A derivation of the same result from first principles, using the geometric property of $\{p_\nu | \nu \varepsilon S_{NM}\}$ given in Appendix 2, is given in the author's paper, [HARR79a].

Thus,

$$\Pi_{st}(j) = \begin{cases} x_s \Pi_{st}(j-1) + (1-x_s) \sum_{u \varepsilon S_{NM}} T_{su} \Pi_{ut}(j-1) & (j \geq 1) \\ \delta_{st} & (j=0) \end{cases} \quad \ldots\ldots (E4.6)$$

which enables $\Pi_{st}(j)$ to be computed by a simple iterative procedure. The same recurrence relation and initial condition still hold whether or not Π_{st} is cumulative, the cumulative result being given by choosing

$$T_{tt} = 1$$

and the non-cumulative result by choosing

$$T_{tt} = 0$$

That this is so may be seen as follows:

$$\Pi_{tt}(j) = T_{tt}\,\Pi_{tt}(j-1) \qquad (j \geq 1)$$

by setting s=t, so that

$$\{\Pi_{tt}(j) \mid j = 1, 2, \ldots\} \Leftrightarrow T_{tt}$$

Now, for $j \geq 1$,

$$\Pi_{st}(j) = x_s\,\Pi_{st}(j-1) + (1-x_s) \sum_{u \neq t} T_{su}\,\Pi_{ut}(j-1)$$

$$+ (1-x_s)\,T_{st}\,\Pi_{tt}(j-1)$$

so unique solutions for $\Pi_{st}(j)$ are given by the initial condition $\Pi_{st}(0) = \delta_{st}$ together with

$$\{\Pi_{tt}(k) \mid k > 0\} \Leftrightarrow T_{tt}$$

Finally, if $T_{tt} = 0$, $j = 0, 1, 2, \ldots$.

$$\Pi_{tt}(j) = \delta_{j0}$$

giving a non-cumulative discrete distribution, and if $T_{tt} = 1$

$$\Pi_{tt}(j) = 1$$

giving the cumulative version.

The computational techniques for the evaluation of the discrete form of the cycle time distribution are fairly straightforward, the most significant points being discussed in section 4.7.5.

Although exponential service time distributions have been assumed in the derivation of equations (E4.6), the argument of section 4.3 gives the following equation analogous to equation (E4.1):

For $s,t \in S_{NM}$; $s \neq t$

$$\Pi_{st} = \sum_{u \in S_{NM}} T_{su} p_s * \Pi_{ut}$$

and $\Pi_{ss}(\ell) \equiv 1$ for all $\ell \geq 0$.

This is, of course, satisfied by the equations (E4.6) but it is also clear that the convolutions could be performed numerically and directly, although somewhat less efficiently, regardless of the form of the distributions, p_s. Nevertheless, it must be remembered that the exponential assumption is required in other parts of the theory.

4.6.3 Error bounds and convergence properties

The accuracy of the results will obviously vary according to the choice of Δ and may be assessed by comparing them with corresponding simulated results or the associated moments with their exact counterparts, derivable as described in previous sections. The results for the networks analysed may be found in Appendix 8 and show good agreement.

Bounds on the exact discrete form of G(t) are given by the following Lemma and Proposition, from which it is then shown, in proposition P4.4, that the method converges to the

exact result as Δ approaches zero.

Lemma L4.3

Given continuous (cumulative) probability distributions $F_P(t)$ and $F_Q(t)$ $(t\geq 0,\ t\in R)^{\dagger}$ represented in discrete form by $P(i)$ and $Q(i)$ $(i\geq 0,\ i\in Z)^{\dagger}$, $n \in Z^+$, $\Delta \in R^+$ s.t.

$$P(i) = F_P(t_i), \quad P(0) = F_P(0) = 0,$$

$$Q(i) \leq F_Q(t_i) \leq Q(i+n)$$

where $t_i = i\Delta$;

let $F_R = F_P * F_Q$ and $R = P * Q$

so that
$$R(i) = \sum_{j=0}^{i} p(i-j)Q(j) = \sum_{j=0}^{i-1} Q(i-j-1)\ p(j+1)$$

in which
$$p(j) = P(j) - P(j-1) \qquad (j\geq 1)$$

$$p(0) = P(0) = 0$$

then
$$R(i) \leq F_R(t_i) \leq R(i+n+1) - \sum_{j=i}^{i+n} Q(i+n-j)\ p(j+1) \qquad (i\geq 0)$$

Proof

$$F_R(t) = \int_{u=0}^{t} F_Q(t-u)\ dF_P(u) \qquad \text{and so}$$

$$\sum_{j=0}^{i-1} \int_{u=j\Delta}^{(j+1)\Delta} F_Q(t_i-t_{j+1})\,dF_P(u) \leq F_R(t_i) \leq \sum_{j=0}^{i-1} \int_{j\Delta}^{(j+1)\Delta} F_Q(t_i-t_j)\,dF_P(u)$$

since F_Q is an increasing function.

$$\therefore \quad \sum_{j=0}^{i-1} Q(i-j-1)p(j+1) \leq F_R(t_i) \leq \sum_{j=0}^{i-1} Q(i+n-j)p(j+1)$$

---

† R is the set of all reals, Z is the set of all integers.

using the given inequality, which proves the Lemma.

Corollary CL4.3

$$R(i) \leq F_R(t_i) \leq R(i+n+1) \qquad (i \geq 0)$$

Proposition P4.3

Let the probability distribution of the time to pass from state α to state β ($\alpha, \beta \in S_{NM}$), denoted by $G_{\alpha\beta}(t)$, have exact discrete form

$$\Psi_{\alpha\beta}(j) = G_{\alpha\beta}(t_j) \qquad (j \geq 0)$$

where $t_j = j\Delta$.

Then $$\Pi_{\alpha\beta}(j) \leq \Psi_{\alpha\beta}(j) \leq \Pi_{\alpha\beta}(j+\ell_{\alpha\beta})$$

where $\ell_{\alpha\beta} = |r|$ for $r \in R_{\alpha\beta}$ which is well defined by Lemma L4.1.

Proof

For $\ell_{\alpha\beta} = 0$, $\alpha = \beta$ and $G_{\alpha\beta}(t) = 1$ for all $t \geq 0$

$$\Psi_{\alpha\beta}(j) = 1 \quad \text{for all } j \geq 0$$

But $\Pi_{\alpha\beta}(j) = 1$ for all $j \geq 0$ by definition.

Suppose inductively that the result is true for all $\alpha, \beta \in S_{NM}$ s.t. $\ell_{\alpha\beta} \leq n$ and consider $\alpha', \beta' \in S_{NM}$ s.t. $\ell_{\alpha'\beta'} = n+1$.

Now, by equation (E4.1) section 4.3,

$$G_{\alpha'\beta'} = \begin{cases} \sum_{\gamma \in S_{NM}} d_{\alpha'} * G_{\gamma\beta'} T_{\alpha'\gamma} & (\alpha' \neq \beta') \\ 1 & (\alpha' = \beta') \end{cases}$$

By Corollary CL4.3, for $j \geq 0$,

$$\left(p_{\alpha'} * \Psi_{\gamma\beta'} T_{\alpha'\gamma}\right)(j) \leq \left(d_{\alpha'} * G_{\gamma\beta'} T_{\alpha'\gamma}\right)(t_j) \leq \left(p_{\alpha'} * \Psi_{\gamma\beta'} T_{\alpha'\gamma}\right)(j+1)$$

since p_γ is the exact discrete probability distribution for the time spent in state γ, with cumulative distribution $P_\gamma(j) = d_\gamma(t_j)$.

Thus, by summing over $\gamma \in S_{NM}$,

$$\left(\sum_{\gamma \in S_{NM}} p_{\alpha'} * \Psi_{\gamma\beta'} T_{\alpha'\gamma}\right)(j) \leq \Psi_{\alpha'\beta'}(t_j) \leq \left(\sum_{\gamma \in S_{NM}} p_{\alpha'} * \Psi_{\gamma\beta'} T_{\alpha'\gamma}\right)(j+1) \quad \ldots\ldots(*)$$

For all $\gamma \in S_{NM}$ s.t. $T_{\alpha'\gamma} \neq 0$, $\ell_{\gamma\beta'} = n$ by Lemma L4.1, so that, by the inductive hypothesis,

$$\Pi_{\gamma\beta'}(j) \leq \Psi_{\gamma\beta'}(j) \leq \Pi_{\gamma\beta'}(j+n)$$

and so the result follows by linearity on substitution into (*).

This proposition supplies fairly coarse bounds for the exact discrete distribution, $\Psi_{\alpha\beta}$, particularly for routes $r \in R_{\alpha\beta}$ with large path length $\ell_{\alpha\beta}$. Tighter bounds could quite simply be applied in practice by successively applying the more precise result of Lemma L4.3, but this is considered unnecessary in view of the following proposition and its corollary.

Proposition P4.4

As $\Delta \to 0$, $\Pi_{\alpha\beta}(j) \to \Psi_{\alpha\beta}(j)$ for all $\alpha,\beta \in S_{NM}$, $j \geq 0 \in Z$

Proof

Since the network considered is open with no arrivals, or alternatively because the transition matrix T is lower triangular, $\exists\ n \in Z^+$ s.t. $T^n = 0$.

$\therefore$ For all $\alpha,\beta \in S_{NM}$, $\ell_{\alpha\beta} < n$

$\therefore$ $\Psi_{\alpha\beta}(j) \leq \Pi_{\alpha\beta}(j+n) \leq \Psi_{\alpha\beta}(j+n)$ $(j \geq 0)$

by Proposition P4.3.

Now, $G_{\alpha\beta}$ is continuous since it is a weighted sum of convolutions of exponential distributions (or alternatively since it has rational Laplace transform as derived in section 4.3) so that

for all $\varepsilon > 0$, $\exists\ \delta_{\alpha\beta} \in R$ s.t.

$| G_{\alpha\beta}(t_j + n\delta) - G_{\alpha\beta}(t_j) | < \varepsilon$ for all $\delta < \delta_{\alpha\beta}$

i.e. $| \Psi_{\alpha\beta}(j + n) - \Psi_{\alpha\beta}(j) | < \varepsilon$

if Δ is chosen to be less than $\delta_{\alpha\beta}$.

Thus the proposition is proved for $\Pi_{\alpha\beta}(j)$ if $j \geq n$. For $j < n$ a similar argument is applied using a lower bound of 0 rather than $\Psi_{\alpha\beta}(j)$ in the double inequality and replacing $G_{\alpha\beta}(t_j)$ by 0.

Corollary CP4.4

Let the exact discrete form of cycle time distribution,

$G(t)$, for times $t = t_0, t_1, t_2, \ldots, (t_0 = 0)$, be denoted by $\Psi(j)$ $(j \geq 0)$, so that $\Psi(j) = G(t_j)$.

Then as $\Delta \to 0$, the unconditional discrete distribution approximation,

$$\Pi(j) = \sum_{\alpha \varepsilon S_I} Z(\alpha)\, \Pi_{\alpha o}(j) \to \Psi(j)$$

Proof

Trivial by the simple linear relationship of Π in terms of $\Pi_{\alpha o}$ $(\alpha, o \varepsilon S_{NM})$

4.7 Computational techniques

4.7.1 Outline of the section

This section presents the methods used to compute efficiently the various quantities related to the cycle time distribution. One of the first problems encountered is the representation of the transition matrix T and its modified version $T^*(s)$, which requires a mapping from the state space, S_{NM} to the positive integers, Z^+. This problem is addressed in section 4.7.2. The following sections 4.7.3, 4.7.4 and 4.7.5 describe the non-straightforward techniques used in the numerical evaluations of the Laplace transform, $L(s)$, the moments (computed via the analytic method) and the discrete form of the cycle time distribution respectively.

4.7.2 Representation of the transition matrix

Given any state $\underline{n} \varepsilon S_{NM}$ it is a simple matter to list the possible states accessible from $\underline{n}$ via one-step transitions, $A(\underline{n})$, and derive the transition probabilities:

For $\underline{n}' \in A(\underline{n})$

$$\left.\begin{array}{ll} \text{(a)} & n'_{i+1} = n_{i+1} + 1 \\ \text{(b)} & n'_{i} = n_{i} - 1 \\ \text{(c)} & n'_{j} = n_{j} \qquad (j \neq i,\ i+1) \end{array}\right\} \; n_i > 0 \quad 1 \leq i \leq M$$

where (a) is null for the case $i = M$.

In the notation of section 4.3,

$$T_{\underline{n}\,\underline{n}'} = \frac{\mu_\theta(\underline{n},\underline{n}')}{\lambda_{\underline{n}}}$$

a result which is immediate from the instantaneous transition rate matrix.

In order to construct T for computer representation, some ordering on the state space, S_{NM}, is required, i.e. a mapping : $S_{NM} \to Z^+$ and its inverse.

Definition D4.1

The mapping $f_{NM} : Z^+ \to S_{NM}$ is defined as follows:

(i) $f_{NM}(1) = \underline{0}$ where $\underline{n} = \underline{0}$ if $n_i = 0$, $1 \leq i \leq M$

(ii) Given $f_{NM}(i) = \underline{n}$

$f_{NM}(i+1) = \underline{n}'$

$$\left.\begin{array}{ll} \text{where} & n'_M = n_M + 1 \\ & n'_j = n_j \quad (j < M) \end{array}\right\} \quad \text{if} \quad \sum_{j=1}^{M} n_j < N$$

Otherwise, let k be the maximum integer such that $n_{k+1} \neq 0$, $1 \leq k \leq M - 1$. Then

$$\begin{array}{lll} n'_k = n_k + 1 & & \\ n'_j = 0 & & (j>k) \\ n'_j = n_j & & (j<k) \end{array}$$

This mapping means that the states are numbered consecutively according to a counting system modulo N with M digits subject to the constraint that the sum of the digits cannot exceed N; i.e. the infeasible states are omitted.

Proposition P4.5

The mapping f_{NM} is a 1-1 correspondence.

Proof

The proof is by induction on the number of centres in the network.

(i) For a network of one centre the result is trivial:

$$f_{N1}(i) = (i - 1) \qquad (1 \leq i \leq N+1)$$

(ii) Suppose the result is true for networks of less than m centres and N customers, for all $N \in Z^+$, and consider state $\underline{n} \in S_{Nm}$. If $n_1 = 0$ the result is true by the inductive hypothesis applied to centres 2,3,....,m since by Definition D4.1

$$f_{Nm}(i) = 0, f_{N,m-1}(i) \qquad \text{for } 1 \leq i \leq k_0$$

where $\underline{a} = x, \underline{b}$ is such that

$$a_1 = x$$

$$a_j = b_{j-1} \qquad (2 \leq j \leq |\underline{b}| + 1)$$

and $f_{N,m-1}(k_0) = (N,0,0,\ldots,0)$

By Definition D4.1,

$$f_{Nm}(k_0 + 1) = (1,0,0,\ldots,0)$$

and for $k_1 \geq i \geq k_0+1$

$$f_{Nm}(i) = 1, f_{N-1,m-1}(i-k_0)$$

where $f_{N-1,m-1}(k_1) = (N-1,0,0,\ldots,0)$

Thus the result is true for $n_1 = 1$ by the same argument.
By applying a simple induction argument to n_1 the proposition is proved.

Proposition P4.6

The representation of the transition matrix with states numbered according to the function f is lower triangular.

Proof

The proof is by induction on the number of centres in the network.

(i) Trivial for networks of one centre.

(ii) Suppose the proposition is true for networks of less than m centres and N customers, for all $N \in Z^+$, and consider state $\underline{n} \in S_{Nm}$.
If $n_1 = 0$, the proposition is true by the inductive hypothesis.
Otherwise, consider a transition from state $\underline{n}$ to state $\underline{n}'$ $(\underline{n},\underline{n}' \in S_{Nm})$.

Suppose this transition is caused by departure of a customer from a centre other than centre 1. Then by definition D4.1 the inductive hypothesis may be applied to centres 2, 3,....,m with $N - n_1$ customers, i.e. to $S_{N-n_1,m-1}$, and the proposition is true.

Otherwise, the transition is caused by a departure from centre 1 which results in a state with lower valued numbering

(by Definition D4.1) which is unique by Proposition P4.5.

Now let $g_{NM} = f^{-1}_{NM}$ for the network S_{NM}.

Proposition P4.7

$$g_{NM}(\underline{k}) = 1 + \sum_{m=1}^{M} \sum_{n=1+N-\sum_{i=1}^{m-1} k_i}^{N-\sum_{i=1}^{m-1} k_i} \binom{M+n-m}{n}$$

where $\binom{r}{s} = \dfrac{r!}{s!\,(r-s)!}$ $\quad (r,s \in Z^+)$

and $\underline{k} \in S_{NM}$.

Proof

As usual, this is by induction on M. For M = 1, the result gives

$$g_{N1}(\underline{k}) = 1 + \sum_{1+N-k_1}^{N} \binom{n}{n}$$

$$= 1 + k_1 \quad \text{as required.}$$

Suppose the proposition is true for S_{NM}. If $\underline{k} \in S_{N,M+1}$ and $k_1 = 0$, then

$$g_{N,M+1}(\underline{k}) = g_{NM}(k_2,k_3,\ldots,k_{M+1}) \text{ by definition D4.1}$$

$$= 1 + \sum_{m=1}^{M} \sum_{n=1+N-\sum_{i=2}^{m} k_i}^{N-\sum_{i=2}^{m} k_i} \binom{M+n-m}{n}$$

$$= 1 + \sum_{m=1}^{M+1} \sum_{n=1+N-\sum_{i=1}^{m} k_i}^{N-\sum_{i=1}^{m-1} k_i} \binom{M+1+n-m}{n}$$

by renumbering and since $k_1 = 0$ with the term for $m=1$ being zero.

If $k_1 \neq 0$, by Definition D4.1 and Proposition P4.5,

$$g_{N,M+1}(\underline{k}) = \binom{M+N}{N} + g_{N-1,M+1}(k_1-1, k_2, \ldots, k_{M+1})$$

$$= \sum_{n=1+N-k_1}^{N} \binom{M+n}{n} + g_{N-k_1,M+1}(0, k_2, \ldots, k_{M+1})$$

by induction on k_1 or direct substitution

$$= 1 + \sum_{m=1}^{M+1} \sum_{n=1+N-\sum_{i=1}^{m} k_i}^{N-\sum_{i=1}^{m-1} k_i} \binom{M+1+n-m}{n}$$

using the above result, so completing the proof.

The function f_{NM}, the inverse of g_{NM}, may be evaluated using the following relationships which follow directly from definition D4.1.

$$g_{NM}(i, 0, 0, \ldots, 0) \leq g_{NM}(\underline{k}) < g_{NM}(j, 0, \ldots, 0)$$

for all $i,j \in Z^+$ s.t. $i \leq k_1$, $j \geq k_1 + 1$

and $g_{NM}(\underline{k}) = g_{NM}(k_1, 0, \ldots, 0) - 1 + g_{N-k_1,M-1}(k_2, k_3, \ldots, k_M)$

$(\underline{k} \in S_{NM})$

The inequality may be used to determine the number of customers at the leftmost centre in a network, and the equation enables it to be applied to a network with that centre removed, so allowing successive components of $\underline{k}$ to be determined.

The functions f_{NM} and g_{NM} are implemented in the APL package (Appendix 7) as NTS and STN respectively.

Now, let the transition matrix, T, be represented under the mapping g by T'. Then,

for $i,j \in Z^+$, $f(i), f(j) \in S_{NM}$

T'_{ij} is assigned the value $T_{f(i),f(j)}$ which is defined in section 4.3. This determines the values for every element of T' since, by definition D4.1 and proposition P4.5, $\{g(\underline{n}) \mid \underline{n} \in S_{NM}\}$ is a consecutive set of integers in the range $[\ 1, |S_{NM}|\ ]$. Henceforth, the prime on T' will be dropped, no ambiguity being present since a mapping g can only cause a permutation of rows and columns.

Clearly the size of the transition matrix T, and so of its modified form T^* and $(I-T^*)^{-1}$, will grow rapidly with the size of network considered; for the case of M servers and N customers, the number of components will be the square of $\begin{pmatrix} M+N \\ N \end{pmatrix}$. However, the matrix is very sparse, with at most M non-zero elements in any row (corresponding to transitions from each occupied centre), and sparse matrix techniques may be used to advantage with respect to both storage and computation time used.

A one dimensional representation is used in the APL package of Appendix 7 in which the non-zero matrix elements only are stored in a vector in column order for successive rows.

Two control vectors are associated with this vector; one to indicate the column positions and one to indicate the indices at which successive rows begin.

The storage requirement will be bounded above by

$$M \times \binom{M+N}{N} \quad \text{(vector of values of non-zero matrix elements)}$$

$$+ \quad M \times \binom{M+N}{N} \quad \text{(vector of their column positions)}$$

$$+ \quad \binom{M+N}{N} \quad \text{(vector of pointers to delimit row boundaries in the above)}$$

$$= \quad (2M+1)\binom{M+N}{N} \quad \text{storage elements.}$$

Of course the second and third contributions to this aggregate will require smaller storage elements than the first since they are integer valued as opposed to floating point. However, the huge saving in storage is evident. In the APL package of Appendix 7, the transition matrix is constructed in this form by the function TRM.

The actual implementation of the sparse matrix operations required for the computation of the Laplace transform, moments and approximate discrete form of the cycle time distribution is discussed in the following three sections.

4.7.3 Computation of the Laplace transform, L(s)

The formula for the Laplace transform of the cycle time distribution was derived in section 4.3 as

$$L(s) = \sum_{\alpha \in S_I} Z(\alpha) \, (I - T^*)^{-1}_{\alpha o}$$

where $Z(\alpha)$ is a product form expression for the initial state

space probabilities on arrival of the test customer at centre 1. Thus $Z(\alpha)$ is easily obtained, and in the APL package (Appendix 7) it is computed during computation of the transition matrix T for the relevant states $\alpha \in S_I \subset S_{NM}$.

The more significant problem is computation of $(I - T^*)^{-1}$. Under the mapping $f_{NM}^{-1} : S_{NM} \to Z^+$, let α map into a, S_{NM} and S_I to S'_{NM} and S'_I respectively. For brevity denote $(I - T^*)$ by X, the components of which are ordered according to the function f_{NM}.

By proposition P4.6 the matrix X is lower triangular so that the inversion requires only a back substitution process, giving greater accuracy as well as efficiency compared with a general inversion. Furthermore, for cyclic networks (but not for the more general "tree-like" networks considered in chapter 5) only the first column of the inverse is required (corresponding to state $o \in S_{NM}$).

Thus $(X^{-1})_{a1}$ is given by :

$$(X^{-1})_{11} = 1$$

$$(X^{-1})_{a1} = \sum_{j=1}^{a-1} T^*_{aj} \, (X^{-1})_{j1} \qquad (2 \le a \le W)$$

where W is the total number of states, the order of S_{NM}.

The simplicity of this calculation, because X is lower triangular, is due to the fact that the open network's states are all transient and the network will always end up in state o.

The equivalent expression derived before $(I - T^*)^{-1}$ in section 4.3 was the (finite) sum

$$\sum_{n=0}^{\infty} (T^*)^n = R \quad \text{say.}$$

Then $R = I + T^* R$

so $R_{ii} = 1 \qquad (1 \leq i \leq W)$

and $R_{ij} = \sum_{k=1}^{W} T^*_{ik} R_{kj} \qquad (1 \leq i \neq j \leq W)$

Thus usage of either the inversion or power series methods are also seen to be equivalent computationally, although compared with direct summation of the powers of a matrix, the back substitution method would clearly be superior.

The back substitution process for computing $(I-T^*)^{-1}$ is performed by the function ESB in the APL package of Appendix 7.

4.7.4 Computation of cycle time moments

Using proposition P4.2, any number of moments of the cycle time distribution may be computed. In particular, if M_p is written as

$$M_p = \sum_{\underline{n} \in S_I} Z(\underline{n})\ \Omega^{(p)}_{\underline{n}o}$$

and it is required to find M_1 and M_2, then $\Omega^{(1)}_{\underline{n}o}$ and $\Omega^{(2)}_{\underline{n}o}$ must be evaluated for $\underline{n} \in S_I$. From proposition P4.2

$$\Omega^{(1)} = (I-T)^{-1}\ T'(1)\ (I-T)^{-1}$$

and $\Omega^{(2)} = \omega^{(2,1)} + \omega^{(2,2)}$

where $\omega^{(2,1)} = (I-T)^{-1}\ T'(1)\ (I-T)^{-1}\ T'(1)\ (I-T)^{-1}$

and $\omega^{(2,2)} = (I-T)^{-1}\ T'(2)\ (I-T)^{-1}$

Now, by the state numbering definition, D4.1, the state o is allocated number 1. Let $n = f^{-1}(\underline{n})$ $(\underline{n} \varepsilon S_I)$, then in the Z^+ - space corresponding to S_{NM}, S'_{NM}, it is required to evaluate $\Omega^{(1)}_{n\,1}$ and $\Omega^{(2)}_{n\,1}$ where Ω, ω, T and T' take their representations in Z^+-space.

In the sparse matrix one dimensional representation of T and so, by simple division, of T', as described in section 4.7.2, it is a simple matter to perform the operation of *post*-multiplication by a vector or a matrix in either uncompressed form or else represented linearly with respect to columns as opposed to rows, the method used here.

Now returning to the general case, $\Omega^{(p)}_{n\,1}$ $(p \geq 1)$ is a sum of terms of the form

$$\left\{ \prod_{i=1}^{h} (I-T)^{-1}_{k_{i-1} j_i} \; T'(m_i)_{j_i k_i} \right\} (I-T)^{-1}_{k_h 1}$$ (with summation convention)

where $k_0 = n$, for some m_i, $h \varepsilon Z^+$; $1 \leq i \leq h$.

Now, it was shown in section 4.7.3 how $(I-T)^{-1}_{k\,1}$ can be evaluated by simple iteration for $k \varepsilon S'_{NM}$. Working from right to left in the expression,

$T'(m_h)_{j_h k_h} \; (I-T)^{-1}_{k_h 1}$ is also easily evaluated in the representation of T' described above, to give, say, $E_{j_h 1}$, another vector. Now,

$(I-T)^{-1}_{k_{h-1} j_h} E_{j_h 1}$, denoted by $E'_{k_{h-1},1}$, say, is the solution to the equations

$$(I-T)_{ij} \; E'_{j1} = E_{i1} \qquad (i,j \varepsilon S'_{NM})$$

which can be solved by direct back substitution as in the case of computation of $(I-T)^{-1}_{i1}$ in view of the fact that the matrix T is lower triangular. This is accomplished by the function HSB in Appendix 7 .

By continuing the process, $E_{j_\ell 1}$ and $E'_{k_{\ell-1},1}$ may then be computed for $\ell = h-1,h-2,\ldots,1$ where

$$E'_{k_{\ell-1},1} = \left\{ \prod_{i=\ell}^{h} (I-T)^{-1}_{k_{i-1}j_i} T'(m_i)_{j_i k_i} \right\} (I-T)^{-1}_{k_h 1}$$

and

$$E_{j_\ell 1} = T'(m_\ell)_{j_\ell k_\ell} E'_{k_\ell 1}$$

are partial products.

Thus, in principle, it is not a difficult programming problem to compute any number of moments, M_p. In the package implemented (Appendix 7), M_1 and M_2 are computed to provide mean cycle time and its standard deviation, by the functions THM and MOM. The post-multiplication of the matrix T in compressed form is accomplished by the function SML. A major reason for limiting the calculation to two moments only, apart from time consideration, is concerned with the future application of the Central Limit Theorem in the prediction of response time distribution, in which case higher moments are not necessary. This is discussed further in section 4.10.

4.7.5 Computation of the discrete form approximation

The discrete form approximation for cycle time distribution may be computed using a simple iterative method based on the recurrence relation, (E4.6) in section 4.6, for

Π_{st}, the (discrete form) time delay distribution corresponding to transition from state s to state t in any number of steps.

For cyclic networks (but not so for the more general tree-like networks considered in chapter 5) the state t is always state o, that representing an empty network, and so Π_{so} is a vector. Thus, no new techniques are required to compute Π_{so} :

(i) The total service rates are pre-computed during construction of the state transition matrix, so x_s is readily available for each $s \varepsilon S_{NM}$;

(ii) The initialisation of the $\Pi_{so}(j)$ for j=0 is trivial;

(iii) The post-multiplication of the transition matrix T, in sparse form, by the vector $\underline{\Pi}(j-1)$ is accomplished simply as described in the previous section, owing to the availability of the column oriented form of the vector second operand.

(iv) The unconditional, approximate discrete form of the cycle time distribution is then easily computed via the expression

$$\Pi(j) = \sum_{s \varepsilon S_I} Z(s) \; \Pi_{so}(j)$$

The functions which perform the computation of the discrete cycle time distribution approximation are DIS and DST, to be found in Appendix 7 .

4.8 Laplace transform inversion

Although an approximate, theoretically convergent method was developed to give a discrete form of cycle time distribution in the previous section, the only exact result derived is that of its Laplace transform and hence its (infinite) set of moments. Of course this is theoretically equivalent to the distribution itself, but it is impossible to interpret intuitively, although the lower moments are obviously useful. Moreover percentiles cannot be computed directly for the distribution. Thus, at first sight, it may appear that inversion of the Laplace transform is desirable.

However, as will be discussed at greater length in section 4.10, the most important distribution is that of a sum of successive cycle times; representing response time in an interactive computer system for example. In order to characterise such a distribution, usage of the Central Limit Theorem is proposed under appropriate assumptions of independence. As a result the distribution is assumed asymptotic Normal with only the first two moments therefore being required from the constituent cycle time distribution. In this way, the appearance of an individual cycle time probability distribution, whilst still undeniably useful, particularly with respect to identification of system imbalances revealed through unexpected peaks, becomes less important to the analyst.

In papers by Lazowska, [LAZ079, LAZ077a], methods are developed for fitting parameters to phase type servers in queueing network models of computer systems, by matching the Laplace transforms of their service time distributions at certain points. The chief advantage of this approach is that performance measures of the server in question may be determined via such a Laplace

transform, evaluated for certain (Laplace) parameter values, so that the calibration process is performed directly with respect to the main measurements of interest. This is not always the case when the classical method of matching the first two or possibly three [SAUE75] moments is used. This, then, provides justification for the computation of Laplace transforms of distributions of time delays in queueing networks, even when the distributions themselves may be available.

There are in fact two possible approaches to inversion of the Laplace transform of cycle time distribution:

(a) Numerical, in which from a set of values for $L(s_j)$ corresponding to values s_j $(j \in Z^+)$ of the Laplace transform parameter s, some distribution, $G_E(t_i)$ say, is estimated for times $t_i \in R^+$ $(i \in Z^+)$ to give a fit to the $L(s_j)$ which is optimal in some sense. However, this gives a discrete approximation which can be no better than the convergent result described in section 4.6. Unfortunately such numerical inversion is exceedingly difficult in view of the smoothing process implicit in the taking of a Laplace transform; in fact the definition of L(s), the transform of G(t), may be written

$$L(s) = E(e^{-sC})$$

where C is a random variable s.t.

$$Pr(C \leq t) = G(t).$$

Thus L(s) is the average w.r.t. the distribution G(t) of a smooth analytic function and any irregularities in G(t) will be smoothed out to a very great extent in the transformation. This heuristic argument indicates the difficulties that are encountered in numerical inversion of Laplace transforms. Nevertheless, programming packages which

perform such inversions have been constructed, e.g. [CAVE78].

(b) In Appendix 1, a recurrence relation is derived to give the distribution of multiple convolutions of negative exponential distributions with different means by inversion of the product of Laplace transforms. The method evaluates the residues at the poles of the Laplace transform in the complex plane to perform the Bromwich integral. The expression for L(s) derived here is a weighted average of just such convolutions, so that in principle the same method could be used. However, the method was developed for the much simpler applications described in chapter 3 and the extension required is certainly not easy. In any case the recurrence relation of Appendix 1 is exceedingly complex, even for quite simple cases, so that the computational problems alone would probably be prohibitive, in spite of the fact that identification of the poles is straightforward.

To sum up, it was not considered worthwhile to pursue the inversion of the Laplace transform, L(s), for the following reasons:

(a) An approximate discrete form for the distribution has been derived (section 4.6) and is convergent;

(b) The distribution of prime interest is that of response time, a summation of several consecutive cycle times;

(c) Inversion of Laplace transforms poses many problems.

4.9 Decomposition Methods

Cyclic, and their associated tandem, networks may be analysed in sections and the results for each such sub-network combined to provide a solution for the whole network. This approach is very similar to that adopted in [HARR78a], but here it is exact. The reason for considering such decomposition methods is one of computational efficiency, with respect to both storage and execution time, achieved by consideration of smaller state spaces.

In this section, a tandem network of M centres with a maximum of N customers, C_{NM}, having state space S_{NM} is considered. Such a network may be successively decomposed into sub-networks C_{Nm}, $1 \leq m \leq M$, with state spaces S_{Nm} where C_{Nm} consists of centres $(M - m + 1), (M - m + 2), \ldots, M$.

Let $S_{Nm}^{(I)} \subset S_{Nm}$ denote the sub-space of valid start states which may exist on entry of the test customer to the first centre of the sub-network C_{Nm}, so that

$$S_{NM}^{(I)} = S_I \quad \text{in the notation of section 4.3.}$$

Now, the order of S_{NM}, $|S_{NM}| = \binom{N+M}{N}$

$$\text{so that} \quad \frac{|S_{Nm}|}{|S_{NM}|} = \frac{M\ (M-1) \ldots\ldots (m+1)}{(N+M)\ (N+M-1) \ldots (N+m+1)} \leq \left(\frac{M}{N+M}\right)^{M-m}$$

Thus the sub-networks considered can possess considerably smaller state spaces than that of the whole network. Furthermore, as will be seen below, it is not necessary even to represent the whole of the state space S_{NM}.

Let $G^{(m)}_{\alpha\beta}(t)$ be the probability distribution function for the time taken for the network C_{Nm} to undergo transitions from state $\alpha \in S_{Nm}$ to $\beta \in S_{Nm}$ and let $T^{(m)}$ be the state transition matrix for the network.

Then, in the notation of section 4.3,

$$G(t) = \sum_{\alpha \in S_I \subset S_{NM}} Z(\alpha)\, G^{(M)}_{\alpha o}(t)$$

$$\text{or} \qquad G(t|\alpha) = G^{(M)}_{\alpha o}(t) \qquad (\alpha \in S_I \subset S_{NM})$$

Now for $\gamma \in S_{Nm'}$ $(1 \leq m' \leq M-1)$, define the state space vector $\gamma^{(m)} \in S_{Nm}$ for $m' < m \leq M$ by

$$\gamma^{(m)}_i = \begin{cases} \gamma_i & (m-m'+1 \leq i \leq m) \\ 0 & (1 \leq i \leq m-m'). \end{cases}$$

Heuristically, $\gamma^{(m)}$ represents the inclusion of state γ into S_{Nm} $(m > m')$ by allocating zero customers to all the additional centres attached on the left of $C_{Nm'}$ to form C_{Nm}.

With this notation, for all m', $m > m' \geq 1$ and for all $\beta \in S_{Nm}$ s.t. $\exists\, \beta' \in S_{Nm'}$ with $\beta = \beta'^{(m)}$,

$$G^{(m)}_{\alpha\beta}(t) = \sum_{\gamma \in S^{(I)}_{Nm'}} P^{(mm')}_{\alpha\gamma^{(m)}}\, G^{(m)}_{\alpha\gamma^{(m)}}(t) * G^{(m')}_{\gamma\beta'}(t)$$

where

$$P^{(mm')}_{\psi\eta} = X^{(m)}_{\psi\eta} - \sum_{\theta \in S^{(I)}_{Nm'}} X^{(m)}_{\psi\theta}\, T_{\theta\eta}$$

and $X^{(m)} = (I-T^{(m)})^{-1}$.

$P^{(mm')}_{\alpha\gamma^{(m)}}$ is the "first entry probability", derived in

Appendix 5, which ensures that once a state $\gamma^{(m)}$ ($\gamma \in S_{Nm'}$) has been entered, the subsequent state transitions are convoluted w.r.t. the network $C_{Nm'}$. Thus the domain of summation implicitly involved in the weighting of the convolutions is disjoint w.r.t. routes $R_{\alpha\beta}$, defined in the state space S_{Nm}.

This result follows because

(a) The time of entry into any state $\gamma \in S_{Nm'}^{(I)}$ is a Markov time; and

(b) For all $\underline{r} \in R_{\alpha\beta}$, $\exists i$ s.t. $r_i \in S_{Nm'}^{(I)}$

and so the domain of summation spans the whole of $R_{\alpha\beta}$ and is disjoint by the above argument.

In particular,

$$G(t|\alpha) = G_{\alpha o}^{(M)}(t) = \sum_{\gamma \in S_{Nm'}^{(I)}} P_{\alpha\gamma(M)}^{(Mm')} G_{\alpha\gamma(M)}^{(M)}(t) * G_{\gamma o'}^{(m')}(t)$$

for $1 \leq m' \leq M-1$

The corresponding (general) result for Laplace transforms is

$$L_{\alpha\beta}^{(m)}(s) = \sum_{\gamma \in S_{Nm'}^{(I)}} P_{\alpha\gamma(m)}^{(mm')} L_{\alpha\gamma(m)}^{(m)}(s) \; L_{\gamma\beta'}^{(m')}(s)$$

where $L_{\alpha\beta}^{(m)}(s)$ is the Laplace transform of $G_{\alpha\beta}^{(m)}(t)$.

By making this decomposition, two advantages are gained with respect to efficiency:

(a) Storage is saved by reduced state space requirement;

(b) Execution time may be saved as networks are extended to include additional centres.

First, consider claim (a). When considering the sub-network $C_{Nm'}$ as above $(1 \leq m' < m)$, the whole state space for this network must be considered since the term $G^{(m')}_{\gamma o'}(t)$ appears in the summation $(\gamma \in S_{Nm'}^{(I)})$ and

for all $\theta \in S_{Nm'}$, $\exists\, \gamma \in S_{Nm'}^{(I)}$ s.t. $\theta \in R^{(m')}_{\gamma o'}$

where $R^{(m')}_{\phi\psi}$ is the set of routes defined on $S_{Nm'}$ with start state ϕ and end state ψ and where

$$\theta \in R^{(m')}_{\phi\psi} \iff \exists\, i \in Z^{+},\ \underline{r} \in R^{(m')}_{\phi\psi} \text{ s.t. } r_i = \theta.$$

However, this state space is much smaller than S_{Nm} , particularly if N is large compared with m; recall

$$\frac{|S_{Nm'}|}{|S_{Nm}|} \leq \left(\frac{m}{N+m}\right)^{m-m'}$$

The other state space to be considered in the (convolution) summation is a sub-set, $U_{Nmm'}$ say, of S_{Nm} defined by $\theta \in U_{Nmm'} \iff \nexists\, \phi \in S_{Nm'}$ with $\phi^{(m)} = \theta$

$$\begin{aligned}\text{Thus } |U_{Nmm'}| &= |S_{Nm}| - |S_{Nm'}| \\ &= |S_{Nm}| \left(1 - \frac{m(m-1)\ldots(m'+1)}{(N+m)(N+m-1)\ldots(N+m'+1)}\right) \\ &\leq |S_{Nm}| \left(1 - \left\{\frac{m'+1}{N+m'+1}\right\}^{m-m'}\right)\end{aligned}$$

Of course, since $|U_{Nmm'}| + |S_{Nm'}| = |S_{Nm}|$ there is no overall saving in the size of the domain of summation if the computation is performed directly. However, if

$G_{\gamma o'}^{(m')}(t)$ or $L_{\gamma o'}^{(m')}(s)$ is first computed and the results saved for each $\gamma \in S_{Nm'}^{(I)}$, all the other data structures associated with $S_{Nm'}$ may be discarded.

The summation may now be performed over the domain $U_{Nmm'}$ so that the effective state space storage requirement is only $\max (|U_{Nmm'}| , |S_{Nm'}|)$ which, for significantly large problems, will be $|U_{Nmm'}|$.

Claim (b) follows naturally. Exactly the same method is employed as that described in the previous paragraph except that the results for $G_{\gamma o'}^{(m')}(t)$, or its Laplace transform, will already be known from some previous computation, so reducing execution time. Such a situation will arise if a network has been solved and is to be extended by adding (m-m') new centres to precede centre 1 in $C_{Nm'}$, but no new customers.

4.10 Response time distribution

Cycle time distribution in queueing networks is undoubtedly of value to the computer system analyst, for example see [LAZO78, HARR78a]. Given a representative queueing network model, accurate stochastic predictions may be made concerning the times taken for tasks in a computer system to complete cycles of service from a set of resources. For example percentiles may be computed so that the percentage of occurrences of some event (completion of a cycle) in each of a set of categories (time intervals) can be predicted. Applications of this type of analysis have been discussed in chapter 2 and further model details are suggested in chapter 8.

However, usually one is more interested in the distribution of response time, the time taken to accomplish a complete task which requires a number of cycles in a computer system. Thus the time delay constituted by the summation of several consecutive cycle times is of paramount importance since it represents response time distribution *conditional* on the number of cycles involved.

Obviously the method considered here cannot find the distribution of the time taken for multiple cycles in a queueing network, since in the open network there are no customers left after one cycle. In theory it should be possible to include a cycle number and to tag explicitly the test customer in each state, so adding two new dimensions to the state space, and then allow customers to return to the first centre after departure from the last in an analysis of the closed network. In this way, multiple cycle time distribution could be derived, but the enormous computational problems involved make the approach impracticable; they are severe enough for a single cycle!

If, on the other hand, rather than considering consecutive cycles one considers a random sample of cycles and uses the sum of their times to represent the response time random variable, the Central Limit Theorem may be applied to the aggregate distribution. This is a valid step since for a random sample of cycles, the resulting cycle time distributions must be independent in view of the equilibrium state space probabilities assumed at the start of each cycle of the test customer. In the case of consecutive cycles, such an equilibrium state will only exist at the start of the *first* cycle, the states existing at the starts of subsequent cycles being correlated, i.e. dependent on the initial (equilibrium) state.

When the assumptions of the Central Limit Theorem are satisfied, the response time distribution is asymptotically Normal and so only the first two moments of cycle time distribution need be computed, as in section 4.3.4 or 4.4.5. In other words, as the number of cycles sampled tends to infinity, the normalised sum of their cycle times will approach a Standard Normal distribution. Thus it is not necessary to compute cycle time moments higher than the second for this application.

The crucial requirement is that the cycles considered be uncorrelated which is implied if the network is in steady state equilibrium at the start of each cycle observed for the test customer, usually consecutive. Now, it has been stated that this in general will not be the case, but intuitively one would expect two cycles to be effectively independent (uncorrelated) if the number of transitions occurring between their start states is large compared with some (unknown) transient value, c.f. a time constant. Thus for networks with many customers, one might expect consecutive cycle start states to be independent and the Central Limit Theorem to be applicable. This postulate may be tested by means of simulation experiments and statistical tests such as the autocorrelation function. The results of such tests are reported in Appendix 8.

It would certainly be desirable to investigate the validity of this assumption for consecutive cycles via a theoretical approach. But for the moment this must remain an open question. It is tacitly assumed that the correlation between cycles is small and the Central Limit Theorem can be applied whenever some form of response time distribution is required. A different approach to response time distribution is taken by Sevcik and Lazowska [LAZ078], but there also many independence assumptions

are required, including that for successive cycles, as discussed in more detail in chapter 2.

4.11 Concluding remarks

Chapter 4 has been concerned with the exact derivation of cycle time distribution for cyclic networks. Clearly the method has limitations with respect to both efficiency and the restrictions on network structure, but its chief merit is the fact that it is exact. It is therefore possible to assess the accuracy of approximate methods such as the PSA method presented in chapter 3.

A considerable amount of detailed theory has been presented in this chapter which relates only to a fairly simple class of networks (cyclic with exponential servers). However, many of the results will be required for use in the much more general class of "tree-like" networks discussed in the next chapter, and the theory provides a sound foundation for the analysis of this superset.

§5. Exact Cycle Time Distribution for "Tree-like" Queueing Networks

5.1 The extension of the cyclic result

The method presented in chapter 4 for deriving the cycle time distribution in cyclic networks relied primarily on the order invariance property of customers in the network, so allowing the position of the test customer to be known in any state and an equivalent open network with no arrivals to be analysed. Clearly such an approach can be applied to a much greater class of networks than merely cyclic ones, although not to networks of arbitrarily interconnected centres. In this section, cycle time distribution and its related quantities derived for cyclic networks in chapter 4, Laplace transform and moments, are derived for what turns out to be the most general class of network able to be handled by the method, so called closed "tree-like" networks.

Informally, a tree-like network consists of a root segment of tandem service centres, the last of which is connected to zero or more tree-like networks or sub-trees in the sense that on departure from the last root segment centre, a customer proceeds directly to the first root segment centre of one of the connected sub-trees, according to the network routing probabilities. The "leaves" of the tree (short for tree-like network) are the last centres of sub-trees with no further sub-trees connected to them. A closed tree-like network is one in which the leaves are all connected back to the top of the tree, i.e. on departure from a leaf centre, a customer proceeds directly to the first centre of the root segment. Thus, cyclic networks are a special case of tree-like networks (no sub-trees). An

example of a closed tree-like network is shown below:

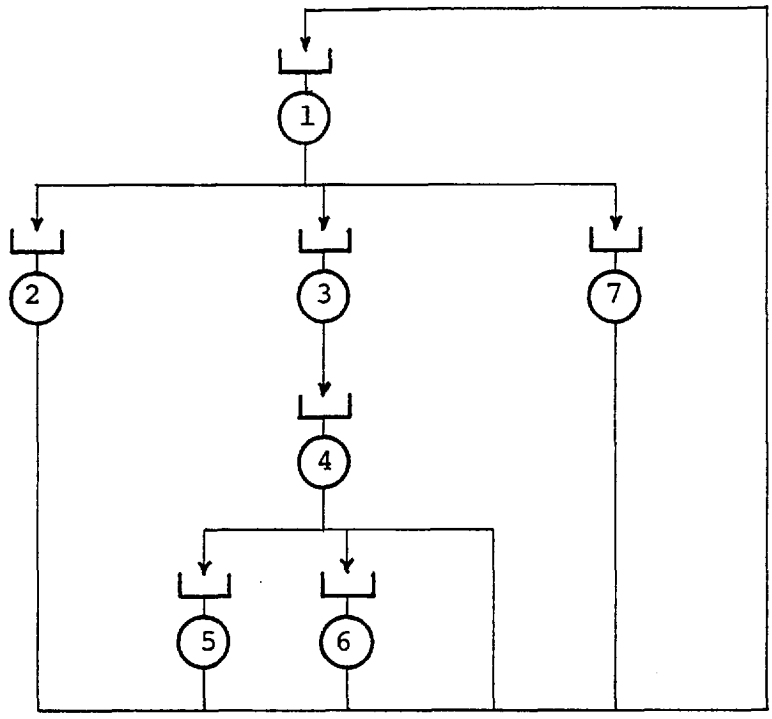


fig. 5.1 A closed tree-like network

The cycle time in a closed tree-like network is the time elapsed between successive arrivals of a customer at the first service centre in the root segment. This is equivalent, assuming instantaneous passage between centres, to the time elapsed between arrival at the first root segment centre and departure from a leaf centre.

In section 5.2 it is shown that tree-like networks are the most general class of network for which cycle time distribution can be computed using the method presented in this chapter, and in section 5.3 a 1-1 mapping between the state space and positive integers is defined, c.f. section 4.7.2, which is shown to result in a lower triangular state transition matrix. In sections 5.4, 5.5 and 5.6, recursive techniques are used to derive expressions for the Laplace transform, the moments and (approximate but convergent) discrete form of the cycle time distribution of tree-like networks. In section 5.7 significant problems encountered during the development of the programming package of Appendix 7, for numerical evaluation of the results

of sections 5.4 to 5.6 are discussed. The section closes with a summary of and some general remarks on this research.

Many of the important results of this chapter rely heavily on propositions from the previous one; part of the justification for the detailed and sometimes lengthy analysis given there. Being far more general in nature, however, the primary results of this chapter are presented as theorems rather than propositions. The inherent structure of the tree-like network suggests the use of recursive techniques which are duly applied where appropriate.

5.2 Most general extension of the method

In this section it is shown that the most general opened class of networks to which the methods of this and the previous chapter may be applied to derive cycle time distribution, called C-networks, are precisely the class of tree-like networks. As a by-product, a formal definition of tree-like networks is obtained.

Consider an open network, A, with no arrivals; M centres numbered 1,2,...,M; N customers initially; state space S_A; and routing probabilities $\{p_{ij} | 1 \leq i, j \leq M\}$.

Definition D5.1

A *segment*, $B \subset A$, is a non-empty sequence of centres $\{B_1, B_2, \ldots, B_{N_B} | B_i \in Z^+;\ 1 \leq B_i \leq M,\ 1 \leq i \leq N_B \leq M\}$

$$\left.\begin{array}{lll} \text{s.t.} & p_{B_i B_{i+1}} = 1 & \\ \text{and} & p_{a_i B_{i+1}} = 0 & \end{array}\right\} \quad 1 \leq i \leq N_B - 1,\ a_i \in A,\ a_i \neq B_i$$

Informally, then, B is a tandem sub-network of A.

Definition D5.2

A segment $B \subset A$ is *maximal* if $\nexists$ a segment $B' \subset A$, $B' \neq B$, and $n \geq 0 \in Z$ s.t.

$$B'_{i+n} = B_i \qquad (1 \leq i \leq N_B)$$

Thus if B is maximal, on departure from its last centre a customer must be able to transit to any of more than one centre or else leave the network; and customers must be able to arrive at the first centre from more than one centre unless the first centre is a starting point of the network with no arrivals from anywhere.

Definition D5.3

A *path*, C, in an open C-network, A, with no arrivals is defined by

$$C = \{C_1, C_2, \ldots, C_{N_C} \mid C_i \in Z^+;\ 1 \leq C_i \leq M,\ 1 \leq i \leq N_C \leq M\}$$

s.t.
$$\begin{cases} p_{C_i C_{i+1}} \neq 0 & (1 \leq i \leq N_C - 1) \\ p_{C_{N_C} j} = 0 & (1 \leq j \leq M) \\ p_{j C_1} = 0 & (1 \leq j \leq M) \end{cases}$$

i.e. C_1 is a starting centre for the test customer and customers can leave the network on departure from the centre numbered C_{N_C}.

In the corresponding closed C-network the definition is the same except that

$$p_{C_{N_C} C'_1} \neq 0 \quad \text{for all paths } C'$$

i.e. the last centres are connected back to the start centres, and a path is the same as a cycle.

Definition D5.4

A segment B is a *starting segment* if $\exists$ path C s.t. $C_1 = B_1$.

Definition D5.5

The relation $\sim$ is defined on centres $a,b \in A$ by $a \sim b$ iff $\exists$ segment $B \subset A$ s.t. $a \in B$ and $b \in B$.

Proposition P5.1

$\sim$ is an equivalence relation.

Proof

(i) $a \sim a$ $(a \in A)$ since $a \in$ segment $\{a\}$.

(ii) $a \sim b \Rightarrow b \sim a$ $(a,b \in A)$ trivially.

(iii) Suppose $a \sim b$ and $b \sim c$ $(a,b,c \in A)$

Then $\exists\ i_1, i_2, \ldots, i_n \in A$ $(n \in Z^+,\ 1 \leq n \leq M)$

s.t.

$$\left.\begin{aligned} p_{i_k i_{k+1}} &= 1 \\ p_{i'_k i_{k+1}} &= 0 \end{aligned}\right\} \quad \begin{aligned} &1 \leq k \leq n-1 \\ &i'_k \in A,\ i'_k \neq i_k \end{aligned}$$

and without loss of generality

$$\begin{aligned} i_1 &= a \\ i_n &= b \end{aligned}$$

the case $i_1 = b$, $i_n = a$ having a similar proof.

Then either

(a) $\exists\ j_1, j_2, \ldots, j_m \in A$ $(m \in Z^+,\ 1 \leq m \leq M)$

s.t.
$$\left.\begin{aligned} p_{j_k j_{k+1}} &= 1 \\ p_{j'_k j_{k+1}} &= 0 \\ j_1 &= b \\ j_m &= c \end{aligned}\right\} \quad \begin{aligned} &1 \le k \le m-1 \\ &j'_k \in A,\ j'_k \neq j_k \end{aligned}$$

$\therefore$ $\exists\ \ell_1, \ell_2, \ldots, \ell_{m+n}$ defined by

$$\ell_k = \begin{cases} i_k & (1 \le k \le n) \\ j_{k-n+1} & (n+1 \le k \le m+n-1) \end{cases}$$

such that

$$\left.\begin{aligned} p_{\ell_k \ell_{k+1}} &= 1 \\ p_{\ell'_k \ell_{k+1}} &= 0 \\ \ell_1 &= a \\ \ell_{m+n-1} &= c \end{aligned}\right\} \quad \begin{aligned} &1 \le k \le m+n-1 \\ &\ell'_k \in A,\ \ell'_k \neq \ell_k \end{aligned}$$

$\therefore$ $a \sim c$

or

(b) $\exists\ \ j_1, j_2, \ldots, j_m \in A \quad (m \in Z^+,\ 1 \le m \le M)$

s.t.
$$\left.\begin{aligned} p_{j_k j_{k+1}} &= 1 \\ p_{j'_k j_{k+1}} &= 0 \\ j_1 &= c \\ j_m &= b \end{aligned}\right\} \quad \begin{aligned} &1 \le k \le m-1 \\ &j'_k \in A,\ j'_k \neq j_k \end{aligned}$$

Now, if $e, f \in$ segment $B \subset A$ s.t. $p_{ef} = 1$,

for all $g \in A$, $g \neq e$, $p_{gf} = 0$ by definition D5.1

$\therefore$ either $a = b \quad (n = 1)$ so that $a \sim c$

or $b = c \quad (m = 1)$ so that $a \sim c$

or $j_{m-1} = i_{n-1}$

Applying the same argument inductively to the last case, and assuming without loss of generality that $n \leq m$

$$j_{m-n+1} = i_1 = a$$

but $j_1 = c$ and using the range $1 \leq k \leq m-n$ in the definition of $c \sim b$ above,

$$c \sim a$$

and $a \sim c$ by (ii).

Proposition P5.2

The equivalence class of $b \in B$, $[b]$, is a segment.

Proof

It is sufficient to prove that if B, B' are segments with $a, b \in B$ and $b, c \in B'$ then $B \cup B'$, appropriately ordered, is a segment, since $a, b, c \in B \cup B'$. Since the result must be true for all such a, b, c, it is required to prove that given $B \cap B' \neq \phi$, $B \cup B'$ is a segment.

Now, $\exists\, j,k \in Z^+$, $b \in B \cap B'$ s.t.

$$b = B_j = B'_k$$

Without loss of generality it is assumed that $j \geq k$.

Then

$$B'_n = B_{j-k+n} \qquad (1 \leq n \leq k)$$

by the argument of the previous proof. Similarly, by definition

D5.1 and the requirement that the routing probabilities from any centre sum to one,

$$B'_n = B_{j-k+n} \qquad (k+1 \leq n \leq N_B + k-j)$$

where it is assumed $N_B + k - j \leq N_{B'}$

for otherwise $B' \subset B$ and the result would be trivial.

Thus,

$$B \cup B' = \{B_1, B_2, \ldots, B_{j-k}, B'_1, B'_2, \ldots, B'_{N_{B'}}\}$$

and $p_{B_{j-k}B'_1} = 1$

$p_{aB'_1} = 0$ for all $a \in A$, $a \neq B_{j-k}$

since $B'_1 = B_{j-k+1}$ and by definition D5.1.

Hence, by the additional application of D5.1 to B and B' separately, $B \cup B'$ is a segment.

Corollary CP5.2

For $a \in A$, $[a]$ is a maximal segment.

Proof

Suppose $[a] \subset B$ where B is a maximal segment and let $b \in B$

Then $a, b \in B$

$\therefore$ $a \sim b$

$\therefore$ $b \in [a]$

$\therefore$ $[a] = B$

Corollary CCP5.2

Maximal segments are disjoint.

Proof

This follows since equivalence classes are disjoint in general.

Lemma L5.1

Let C,C' be distinct paths in a network A. Then A is a C-network if and only if

$\exists\ i \in Z^+ ,\ 0 \le i \le \min (N_C , N_{C'})$, s.t.

(a) for all $j \in Z^+ ,\ 1 \le j \le i$,

$$C_j = C'_j ;$$

(b) for all $j, k \in Z^+ ,\ 1 \le j \le N_{C'} ,\ i < k \le N_C ,$

$$C'_j \neq C_k ;$$

*(c) for all $j, k \in Z^+ ,\ i < j \le N_{C'} ,\ 1 \le k \le N_C ,$

$$C'_j \neq C_k .$$

Proof

Choose the maximum $i \in Z$ s.t. for all $1 \le j \le i$

$C_j = C'_j$ so that $0 \le i \le \min (N_C , N_{C'})$.

Without loss of generality, $N_C \ge N_{C'}$ and so $i < N_C$ for $C \neq C'$.

Suppose $\exists k > i$ s.t. $C_k = C'_j$ for some j, $1 \le j \le N_{C'}$.
Suppose further that the test customer has just departed from centre C_i and consider some other customer, Γ.

Case (i) $k > i + 1$. Suppose the test customer proceeds to centre C_{i+1} and Γ follows path C'. Then Γ can arrive at centre C_k before the test customer.

Case (ii) $k = i + 1$ and without loss of generality $j < i + 1$, ($C_{i+1} \neq C'_{i+1}$ by definition of i). Suppose Γ is at centre i immediately behind the test customer which proceeds to centre C'_j. Then Γ can overtake the test customer in path C' by subsequently entering centre C'_{i+1} .

---

* Condition (c) is in fact superfluous in that it may be derived from condition (b) by interchanging C,C' and j,k.

Hence, returning customers cannot be disregarded and A cannot then be a C-network.

Conversely, by virtue of the FCFS queueing discipline assumed for all networks, paths such as C and C′ above must exist in order to allow a customer returning to a start centre subsequently to be situated in front of the test customer in the latter's path.

Corollary CL5.1

Paths with different start centres in a C-network are disjoint.

Proof

By definition such paths have disjoint start centres and so by Lemma L5.1 are disjoint.

Thus it may be assumed that C-networks have only one start centre since otherwise the disjoint sub-networks may be considered independently.

Corollary CCL5.1

Paths may not include any centre more than once in a C-network, A.

Proof

Suppose path $C \subset A$ has

$$C_i = C_j \quad \text{for some} \quad i, j \in Z^+, \quad 1 \leq i < j \leq N_C.$$

Choose the maximum such i.

Define C' by

$$C'_k = \begin{cases} C_k & (1 \leq k \leq i) \\ C_{k+j-i} & (i+1 \leq k \leq N_{C'} = N_C + i-j) \end{cases}$$

Then C' is a path since

$$p_{C'_i C'_{i+1}} = p_{C_j C_{j+1}} \neq 0$$

so that $$p_{C'_k C'_{k+1}} \neq 0, \quad 1 \leq k \leq N_{C'} - 1$$

by virtue of the fact that C is a path

and $$C'_1 = C_1$$

$$C'_{N_{C'}} = C_{N_C} .$$

Now, $C'_k = C_k$ $(1 \leq k \leq i)$ and either

(i) $j < N_C$ and $C'_{i+1} = C_{j+1} \neq C_{i+1}$ since i was chosen to be maximum, or

(ii) $j = N_C$ so that $i = N_{C'}$ and C'_{i+1} does not exist.

Therefore, in either case, A is not a C-network by Lemma L5.1, a contradiction.

Informally this means that there can be no looping back in a C-network; C-networks possess the "feed-forward" property.

Theorem T5.1

A C-network, A, is defined to be either

(a) A single segment; or

(b) A (maximal start) segment from the last centre of which a customer may enter one of at least two (sub) C-networks; or

(c) A (maximal start) segment from the last centre of which a customer may either depart from the network or enter one of at least one C-networks.

In other words, a C-network is tree-like.

Proof

Case (a) is trivial since it is the tandem network solved in chapter 4.

Otherwise, we may assume by Corollary CL5.1 that there is only one start centre and so only one maximal starting segment (by Corollary CP5.2), B say.

On departure from the last centre of B, a customer must choose one of at least two paths, for otherwise B would not be maximal.

Let the set of possible successor centres be $X \neq \phi$ ($X = \phi$ gives case (a)) and let the set of paths possible in A corresponding to a transition to centre $x \in X$ be denoted by Y_x.

Then the set of all possible paths in A is

$$\left\{ \bigcup_{x \in X} Y_x \right\} \cup \{B\}$$

where {B} corresponds to a network from which it is possible to depart from the last centre of the maximal start segment, case (c).

Now denote $Y_x' = Y_x \setminus B$.

Then by Lemma L5.1

$$\bigcap_{x \in X} Y_x' = \phi$$

Therefore there exist disjoint sub-networks A_x for each $x \in X$ with centres given by $A_x = \bigcup_{C \in Y_x'} C$

Furthermore, for each $x \in X$, by application of Lemma L5.1 to the paths Y_x in A, the same Lemma applies to the paths Y_x' in A_x since all such paths have B for their first N_B centres.

$\therefore$ A_x is a C-network for all $x \in X$

and the theorem is proved.

Definition D5.6

The maximal start segment of a C-network is the *root segment.*

Definition D5.7

The C-networks connected to the last centre of a maximal segment are called *sub-trees.*

A C-network is also defined to be a sub-tree of itself.

The C-networks connected to the last centre of the root segment are called *primary sub-trees.*

These are the networks labelled A_x in the proof of Theorem T5.1.

5.3 The mapping between the state space and the positive integers

The state numbering mapping used here is identical with that of section 4.7.2 given an enumeration of the service centres in the tree-like network. In this section, such an enumeration is defined on tree-like networks and it is shown that the resulting state transition matrices are lower triangular. These matrices will again be sparse; from any state a maximum of only M, the number of centres in the network, transitions are possible; and the same representation, in one dimensional form, as was described in section 4.7.2 may be used here (see section 5.7).

Definition D5.8

The centres in a tree-like network, A, with s primary sub-trees labelled (arbitrarily) $A_1, A_2, \ldots, A_s$, are numbered as follows:

Let centre $c \in A$ be numbered $m_A(c)$. Then

$$m_A(c) = \begin{cases} i & \text{if } c = B_i \text{ where B is the root segment of A, } 1 \le i \le N_B. \\ N_B + \left(\sum_{j=1}^{r-1} N_{A_j} \right) + m_{A_r}(c) & \text{if } c \in A_r \ (1 \le r \le s) \\ & \text{where } N_{A_j} \text{ is the no. of centres in sub-tree } A_j, \ 1 \le j \le s. \end{cases}$$

Definition D5.9

The *depth* of a tree-like network, A, denoted by d(A), is the maximum number of maximal segments in any path of A.

Since by Theorem T5.1, every maximal segment is the root segment of a tree-like network, d(A) is one greater than the maximum number of (non trivial) branches in any path of A.

Lemma L5.2

(a) For all $c, c' \in A$, $1 \le m_A(c) \le N_A$ and $m_A(c) \ne m_A(c')$ for $c \ne c'$

i.e. $\exists$ an ordering of $\{m_A(c) \mid c \in A\}$ which is consecutive.

(b) The centres in any sub-tree, $D \subset A$, are numbered consecutively in the order that they are enumerated by the mapping m_D on D.

Proof

(a) The result is true for d(A) = 1 by definition D5.8. Suppose true for all tree-like networks A′ s.t. $1 \le d(A') \le n \in Z^+$ and let A be as defined in definition D5.8 with d(A) = n + 1.

For $c \in A$, by Definition D5.8,

$$\left.\begin{array}{l} 1 \le m_A(c) \le N_B \le N_A \\ \text{and} \quad m_A(c) \neq m_A(c') \quad \text{for} \quad c \neq c' \end{array}\right\} \quad c, c' \in B$$

$$\left.\begin{array}{l} N_B < m_A(c) \le N_B + N_{A_1} \\ \text{and so} \quad m_A(c) \neq m_A(c') \text{ for } c \neq c' \end{array}\right\} \quad c \in A_1,\ c' \in B \cup A_1$$

by D5.8 and the inductive hypothesis.

Thus, by induction on r, for r = 1, 2,....,s

$$\left.\begin{array}{l} N_B + \sum_{j=1}^{r-1} N_{A_j} < m_A(c) \le N_B + \sum_{j=1}^{r} N_{A_j} \\ \text{and} \quad m_A(c) \neq m_A(c') \quad \text{for} \quad c \neq c' \end{array}\right\} \quad \begin{array}{l} c \in A_r \\ c' \in B \cup \bigcup_{j=1}^{r} A_j \end{array}$$

$$\left.\begin{array}{l} \text{Thus} \quad 1 \le m_A(c) \le N_A \\ \text{and} \quad m_A(c) \neq m_A(c') \end{array}\right\} \quad c, c' \in A,\ c \neq c'$$

and (a) is proved by induction on d(A).

(b) For d(A) = 1, the only sub-tree $D \subset A$ is $D = A$ and the result is trivial.

Suppose true for all tree-like networks A′ s.t. $1 \le d(A') \le n \in Z^+$ and let A be as defined in D5.8 with $d(A) = n + 1$.

Then either

(i) $D = A$ and the result is trivial; or

(ii) D is a sub-tree of A_r for some r, $1 \le r \le s$.

∴ by hypothesis $\{m_{A_r}(c) \mid c \in D\}$ is consecutive.

∴ by definition D5.8 $\{m_A(c) \mid c \in D\}$ is consecutive.

Thus (b) is proved by induction on d(A).

Corollary CL5.2

The mapping $m : A \to Z^+$ is $1 - 1$.

Proof

The corollary is a restatement of (a).

Proposition P5.3

For a tree-like network A, with state space S_A defined according to the centre enumeration, m_A, given by definition D5.8, and the mapping $f_A : Z^+ \to S_A$ defined by D4.1, the state transition matrix, T, is lower triangular.

Proof

Consider a state transition in the state space S_A,

$$\underline{n} \to \underline{n}' \qquad (\underline{n}, \underline{n}' \in S_A)$$

Case (i)

The transition is due to a customer's exit from a centre, c say, which is the last in a maximal segment, X say ($X \subset A$). The customer therefore either enters centre c', say, in a primary sub-tree of the sub-tree with root segment X, A_X say, or else leaves the network. Thus, either

$$m_{A_X}(c) < m_{A_X}(c') \quad \text{by Definition D5.8}$$

so that $m_A(c) < m_A(c')$ by Lemma L5.2 (b),

or a customer leaves the network.

Case (ii)

The transition is due to a customer's exit from a centre, c say, which is not the last in X, using the notation of case (i).

$$\therefore \quad m_{A_X}(c) < m_{A_X}(c') \quad \text{by Definition D5.8}$$

and so $m_A(c) < m_A(c')$ by Lemma L5.2 (b).

Thus it is sufficient to show that:

Given an open network of centres numbered 1,2,...,M (where here M = N_A), with no arrivals and initially N customers, having state space S_{NM},

if a transition $\underline{n} \rightarrow \underline{n}'$ $(\underline{n}, \underline{n}' \epsilon S_{NM})$ implies that a customer has either progressed between centres numbered c and c' where $M \geq c' > c \geq 1$ or has left the network from some centre, then the state transition matrix for the network is lower triangular.

The proof of this is simple in that since customers cannot move to lower numbered centres, the proof by induction on M, the number of centres, given for Proposition P4.6 may be used.

5.4 Laplace Transform of cycle time distribution in tree-like networks

An expression for the Laplace transform of the cycle time distribution in a tree-like network is derived in this section by a recursive extension of the method described in section 4.3. The cycle time distribution, both conditional on starting state and unconditional, is again a weighted sum of convolutions of state sojourn time distributions (because of the Markov property), but more complex than for cyclic networks in view of the existence of more than one path through the network admissible as a cycle.

The notation is based on that of chapter 4 and is now given:

Let a tree-like network, A, have N customers initially, M_A centres and primary sub-trees $A_1, A_2, \ldots, A_r$ $(r \geq 0)$.

If $r = 0$ there are no primary sub-trees and the network is tandem.

Let the centres in A be numbered according to definition D5.8 and denote any centre by its number, m say, $1 \le m \le M_A$. Let the state space of A under this numbering be denoted by S_{NA} and given by

$$S_{NA} = \left\{ \underline{n} \mid \sum_{i=1}^{M_A} n_i \le N \; ; \; n_i \ge 0, \; 1 \le i \le M_A \right\}$$

where if $\underline{n} \in S_{NA}$, n_i is the number of customers at the i'th numbered centre, $1 \le i \le M_A$.

Let the set of valid initial states be denoted by $S_{NA}^{(I)}$ and defined by

$$S_{NA}^{(I)} = \left\{ \underline{n} \mid \underline{n} \in S_{NA} \; ; \; \sum_{i=1}^{M_A} n_i = N \; ; \; n_1 > 0 \right\}$$

which represents a state with N customers and the test customer at (the back of the queue of) the first centre in the root segment of A.

Let the set of routes between states $\alpha, \beta \in S_{NA}$ be denoted by $R_{\alpha\beta}^{(NA)}$ and defined as in section 4.3 by

$$R_{\alpha\beta}^{(NA)} = \left\{ (i_1, i_2, \ldots, i_n) \mid n \in Z^+; \; i_j \in S_{NA}, \; 1 \le j \le n; \right.$$
$$\left. i_1 = \alpha; \; i_n = \beta; \; T_{i_k i_{k+1}} \neq 0, \; 1 \le k < n \right\}$$

where T is the state transition matrix for S_{NA} defined below.

Let the service rate of centre i be μ_i $(1 \le i \le M_A)$, a constant for the reasons explained in section 4.2, and define θ_A, ϕ_A, λ_A by

$\theta_A(u,v)$ = no. of centre from which a departure causes a transition $u \to v$ $(u, v \in S_{NA})$

$\phi_A(u,v)$ = no. of centre at which a customer arrives on a transition $u \to v$ $(u, v \in S_{NA})$

where $\theta_A(u,v)$ and $\phi_A(u,v)$ are undefined if a one step transition $u \to v$ is invalid.

$\lambda_A(\underline{u}) = \sum_{\substack{1 \le j \le M_A \\ u_j > 0}} \mu_j$, the total service rate in state $\underline{u} \in S_{NA}$.

The state transition matrix for the embedded Markov chain, T, may be derived from the instantaneous transition rate matrix or the balance equations for A as

$$T_{uv} = \begin{cases} \dfrac{\mu_{\theta_A(u,v)} p_{\theta_A(u,v)\phi_A(u,v)}}{\lambda_A(u)} & \text{if a one-step transition } u \to v \text{ is valid} \\ 0 & \text{otherwise} \end{cases}$$

where p is the routing probability matrix of A so that for a transition, $u \to v$, caused by a customer moving within a segment, the factor would be absent in the expression for T_{uv}.

Let the cumulative probability distribution of the time spent in state u be $d_u(t)$, which is negative exponential by Appendix 4, having Laplace transform

$$D_u(s) = \frac{\lambda_A(u)}{s + \lambda_A(u)}$$

Define the modified transition matrix T^*, as in section 4.3, by

$$T^{*}_{uv} = D_u(s)\, T_{uv} \qquad (u,v \in S_{NA}).$$

Let the probability distribution function for the time to pass through A on some stochastically chosen path, conditional on initial state $\alpha \in S_{NA}^{(I)}$ be $G_{NA}(t|\alpha)$ with the unconditional distribution function for an initial equilibrium state distribution being $G_{NA}(t)$.

Let these distributions have Laplace transforms $L_{NA}(s|\alpha)$ and $L_{NA}(s)$ respectively.

In order to derive a recurrence relation for $L_{NA}(s)$ it is necessary to define one more set of states, viz. those which can introduce the test customer into a primary sub-tree after a state transition.

Let A have root segment B and define $E_{NA} \subset S_{NA}$ by

$$E_{NA} = \left\{ \underline{n} \mid \underline{n} \in S_{NA} \; ; \; n_i = 0,\ 1 \le i < N_B \; ; \; n_{N_B} = 1 \right\}.$$

Hence by definition D5.8, E_{NA} consists of states with only one customer left in the root segment of A, at its last centre. Because of the FCFS queueing discipline this must be the test customer.

Under the mapping $f_{NA}^{-1} : S_{NA} \to Z^{+}$ of Proposition P5.3, by definition D4.1, $\underline{n} \in E_{NA}$ if and only if

$$F^{(1)} = f_{NA}^{-1}(\underline{n}^{(1)}) \le f_{NA}^{-1}(\underline{n}) < f_{NA}^{-1}(\underline{n}^{(2)}) = F^{(2)}$$

where $\underline{n}^{(i)}$ is defined by

$$n_j^{(i)} = \begin{cases} 0 & (j \neq N_B) \\ i & (j = N_B) \end{cases}$$

Thus the states in E_{NA} are numbered consecutively between $F^{(1)}$ and $F^{(2)}-1$ inclusive, a fact of great use in implementation on a computer (see section 5.7 and Appendix 7).

Let the random variable for the time taken for the network A to reach state β from state α ($\alpha,\beta \in S_{NA}$) be denoted by $\tau_{\alpha\beta}$.

Theorem T5.2

The cycle time distribution, $G_{NA}(t|\alpha)$, in a tree-like queueing network, A, with root segment B, r primary sub-trees, N customers initially and start state $\alpha \in S_{NA}^{(I)}$, in which the test customer is at the first numbered centre of A, is given by

$$G_{NA}(|\alpha) = \begin{cases} H_{\alpha o}^{(NA)} & (r = 0) \\ \sum_{\beta \in E_{NA}} \sum_{j=1}^{r} \frac{P_j \mu_{N_B}}{\lambda_A(\beta)} G_{NA_j}(|\beta^{-(j)}) * d_\beta * H_{\alpha\beta}^{(NA)} & \text{otherwise} \end{cases}$$

where $H_{\alpha\beta}^{(NA)}(t) = Pr(\tau_{\alpha\beta} \leq t) \qquad (\alpha \neq \beta)$

$H_{\alpha\alpha}^{(NA)}(t) = 1$;

state o is that numbered 1, representing a network containing zero customers;

$P_j = p_{N_B K_j}$, the routing probability, in which

$$K_j = 1 + N_B + \sum_{i=1}^{j-1} N_{A_i} \qquad (1 \leq j \leq r \neq 0);$$

$\beta_i^{-(j)} = \beta_{i+K_j-1}^{(j)} \qquad (1 \leq i \leq N_{A_j})$ in which

$\beta^{(j)} \in S_{NA}$ is the state succeeding $\beta \in S_{NA}$ entered by transit of a customer from centre N_B to the first centre in sub-tree A_j $(1 \leq j \leq r)$.

Proof

Let the random variable for the time taken for the test customer to leave the network from state $\beta \in S_{NA}$ be denoted by T_β. Also, let δ_β denote the random variable for the time spent in state β.

Then, for $\alpha \in S_{NA}^{(I)}$ and $\alpha \notin E_{NA}$, $r \neq 0$,

$$G_{NA}(t|\alpha) = \sum_{\beta \in E_{NA}} \int_0^t \Pr(T_\beta \leq t-u \wedge \text{transition from state } \beta \text{ caused by test customer service completion}) \, d\Pr(\tau_{\alpha\beta} \leq u|\beta)$$

since for any path in A taken by the test customer, some state $\beta \in E_{NA}$ must be entered just before passage of the test customer to a primary sub-tree, and using the fact that $\tau_{\alpha\beta}$ is a Markov time, c.f. section 4.3.

Thus,

$$G_{NA}(t|\alpha) = \sum_{\beta \in E_{NA}} \int_0^t \left\{ \sum_{j=1}^{r} \Pr(T_\beta \leq t-u \wedge \text{test customer enters primary sub-tree } j \text{ after state } \beta) \right\} d\Pr(\tau_{\alpha\beta} \leq u|\beta)$$

The expression in { } may be written as

$$\underbrace{\sum_{j=1}^{r} P_j \frac{\mu_{N_B}}{\lambda_A(\beta)}}_{\text{state transition probability } \beta \to \beta^{(j)}} \underbrace{\int_0^{t-u} G_{NA}\,(t-u-v \mid \beta^{(j)})}_{\text{distribution of time from state } \beta^{(j)} \to \text{end of path}} \underbrace{dPr\,(\delta_\beta \le v)}_{\text{distribution of time spent in state } \beta}$$

where $\beta^{(j)}$ is, as defined in the theorem, the state succeeding β caused by transit of a customer from centre N_B (last in the root segment) to the first centre in the j'th primary sub-tree root segment, numbered K_j say, and

$P_j = p_{N_B K_j}$, the associated routing probability. By definition D5.8,

$$K_j = 1 + N_B + \sum_{i=1}^{j-1} N_{A_i} \qquad (1 \le j \le r)$$

Thus the expression { } is

$$\sum_{j=1}^{r} P_j \frac{\mu_{N_B}}{\lambda_A(\beta)} \left(G_{NA_j}\,(\,\mid \beta'^{(j)}) * d_\beta \right)(t-u)$$

omitting the arguments in the convoluted distributions, where, as given,

$$\beta'^{(j)}_i = \beta^{(j)}_{i+K_j-1} \qquad (1 \le i \le N_{A_j})$$

are the components of the state space vector for A_j taken in isolation corresponding to state $\beta^{(j)}$ in A. The sub-trees A_j may be considered separately in this way since all transitions in other sub-trees A_k $(1 \le k \ne j \le r)$ are independent by the argument of Appendix 4 and the disjointness property of C-networks (Lemma L5.1).

Thus,

$$G_{NA}(t|\alpha) = \sum_{\beta \epsilon E_{NA}} \sum_{j=1}^{r} \frac{P_j \mu_{N_B}}{\lambda_A(\beta)} G_{NA_j}(|\beta^{-(j)}) * d_\beta * H^{(NA)}_{\alpha\beta}$$

for $r \neq 0$ and $\alpha \notin E_{NA}$

where $H^{(NA)}_{\alpha\beta}(t) = Pr(\tau_{\alpha\beta} \leq t)$ as given.

The reason for partitioning routes through the network A in this way, via states $\beta \epsilon E_{NA}$, is so that the position of the test customer can be identified at all stages in the (recursive) computation.

For $\alpha \epsilon E_{NA}$, the same reasoning and resulting equation may be applied using the result that

$$H^{(NA)}_{\alpha\alpha}(t) = 1 \qquad (t \geq 0, \alpha \epsilon S_{NA})$$

For r = 0, A is a tandem network, for which the problem was solved in section 4.3, and

$$G_{NA}(t|\alpha) = H^{(NA)}_{\alpha o}(t).$$

This completes the proof.

Corollary CT5.2

$$L_{NA}(s|\alpha) = \begin{cases} (I - T^*)^{-1}_{\alpha o} & (r = 0) \\ \sum_{\beta \epsilon E_{NA}} \sum_{j=1}^{r} \frac{P_j \mu_{N_B}}{s + \lambda_A(\beta)} (I - T^*)^{-1}_{\alpha\beta} L_{NA_j}(s|\beta^{-(j)}) & \text{otherwise} \end{cases}$$

Proof

Recall, by the argument of section 4.3, that the Laplace transform of $H_{\alpha\beta}^{(NA)}(t)$ is $(I - T^*)^{-1}_{\alpha\beta}$, $(\alpha,\beta \in S_{NA})$, where T^* is as defined in this section. This is so because Corollary CCL4.1 applies to $R_{\alpha\beta}^{(NA)}$ in a tree-like network, A, since T^* is lower triangular by Proposition P5.3. This property also ensures that the condition $H_{\alpha\alpha}^{(NA)}(t) = 1$ is satisfied.

$\therefore$ since $D_\beta(s) = \dfrac{\lambda_A(\beta)}{s + \lambda_A(\beta)}$,

the corollary is proved.

Corollary CCT5.2

Finally, an expression for the Laplace transform, $L_{NA}(s)$, of the unconditional cycle time distribution, assuming an initial equilibrium state space probability distribution $Z(\alpha)$, $\alpha \in S_{NA}^{(I)}$, is:

$$L_{NA}(s) = \sum_{\alpha \in S_{NA}^{(I)}} Z(\alpha)\, L(s|\alpha)$$

where $$Z(\underline{\alpha}) = \frac{1}{g(N,A)} \prod_{i=1}^{M_A} \left(\frac{e_i}{\mu_i} \right)^{\alpha_i} \qquad (\underline{\alpha} \in S_{NA}^{(I)})$$

in which $\{e_i \mid 1 \le i \le M_A\}$ is such that

$$e_i = \sum_{j=1}^{M_A} e_j\, p_{ji} \qquad (1 \le i \le M_A)$$

and $g(N,A)$ is the normalising constant for the state sub-space

$S_{NA}^{(I)}$. This expression for $Z(\underline{\alpha})$ is the well known one for separable queueing networks, [GORD67].

This theorem, together with its corollaries, is the fundamental result of this chapter, and in particular the numerical method of section 4.4 for deriving the moments of $G_{NA}(t)$ may be applied without modification.

In the next two sections, extensions of the methods of sections 4.5 and 4.6 are applied to derive analytic expressions for the moments of cycle time distribution and an approximate discrete version of this distribution which is convergent in the sense of section 4.6.

5.5 Recursive solution for the moments

A recursive expression for the moments of cycle time distribution in tree-like networks is easily obtained via differentiation of the result of Corollary CT5.2, using Proposition P4.1 and an extension of the Leibnitz theorem for repeated differentiation of a product, which is given in Appendix 6.

Let the p'th moment of cycle time distribution for the tree-like network, A, of N customers and r primary sub-trees, conditional on start state $\alpha \in S'^{*}_{NA}$ be denoted by $M_p(NA|\alpha)p!$ In the notation of the previous section, we have:

Theorem T5.3

$$\text{Let } X^{(p)} = \sum_{\substack{\sum_{i=1}^{|\underline{m}|} m_i = p \\ m_i > 0}} \left\{ \prod_{i=1}^{|\underline{m}|} (I-T)^{-1} T'(m_i) \right\} (I-T)^{-1}$$

---

* $S'_{NA} = \bigcup_{n=1}^{N} S_{nA}^{(I)}$

where

$$T'_{\alpha\beta}(m_i) = \frac{T_{\alpha\beta}}{(\lambda_A(\alpha))^{m_i}} \qquad (\alpha,\beta \in S_{NA}).$$

Then, for $\alpha \in S'_{NA}$,

$$M_p(NA|\alpha) = \begin{cases} X^{(p)}_{\alpha o} & (r = 0) \\ \sum_{\beta \in E_{NA}} \sum_{j=1}^{r} P_j \mu_{N_B} \sum_{\substack{u+v+w=p \\ u,v,w \geq 0}} X^{(u)}_{\alpha\beta} (\lambda_A(\beta))^{-v-1} M_w\left[NA_j | \beta'(j)\right] & (r > 0) \end{cases}$$

Proof

As stated in section 4.4,

$$M_p(NA|\alpha) = \frac{(-)^p}{p!} \left[\frac{d^p}{ds^p} L_{NA}(s|\alpha) \right]_{s=0} \qquad (\alpha \in S'_{NA})$$

If r = 0, then the result is that of a tandem network and given by Propositions P4.1, P4.2.

For r > 0, by Corollary CT5.2,

$$M_p(NA|\alpha) = \frac{(-)^p}{p!} \sum_{\beta \in E_{NA}} \sum_{j=1}^{r} P_j \mu_{N_B} \left[\frac{d^p}{ds^p} \left\{ (I-T^*)^{-1}_{\alpha\beta} \frac{1}{s+\lambda_A(\beta)} L_{NA_j}(s|\beta'(j)) \right\} \right]_{s=0}$$

$$= \frac{(-)^p}{p!} \sum_{\beta \in E_{NA}} \sum_{j=1}^{r} P_j \mu_{N_B} p! \sum_{\substack{u+v+w=p \\ u,v,w \geq 0}} \frac{F^{(u)}_{\alpha\beta}(0)}{u!} \frac{(-)^v}{(\lambda_A(\beta))^{v+1}} \frac{L^{(w)}_{NA_j}\left[0|\beta'(j)\right]}{w!}$$

where the n'th derivative of a function of s, H(s) say, is denoted by $H^{(n)}(s)$ and the extension to Leibnitz's theorem has been used.

Here, $F = (I-T^*)^{-1}$ so that, by Proposition P4.1,

$$F^{(u)}(0) = (-)^u \, u \, ! \, X^{(u)}$$

since $T^*(0) = T$

and $$\left(T^{*(m)}(0)\right)_{\alpha\beta} = T_{\alpha\beta} \, D_\alpha^{(m)}(0) \qquad (m \geq 0),$$

recalling the definition from section 5.4 where

$$D_\alpha^{(m)}(0) = (-)^m \, m \, ! \, \left(\lambda_A(\alpha)\right)^{-m}$$

so that $T^{*(m)}(0) = (-)^m \, m \, ! \, T'(m)$

Furthermore,

$$M_w(NA_j \mid \beta'(j)) = \frac{(-)^w}{w!} \, L_{NA_j}^{(w)}(0 \mid \beta'(j))$$

which completes the proof.

Corollary CT5.3

The unconditional p'th moment of cycle time distribution, assuming the initial equilibrium state space probability distribution $Z(\alpha)$, $\alpha \in S_{NA}^{(I)}$, given explicitly in Corollary CCT5.2 is $p \, ! \, M_p(NA)$ where

$$M_p(NA) = \sum_{\alpha \in S_{NA}^{(I)}} Z(\alpha) \, M_p(NA \mid \alpha)$$

In theory, then, any number of moments of the cycle time distribution may be computed, but in practice the recursion involved may be excessively inefficient since the moments for the sub-trees of the network A may be recomputed many times. Clearly the magnitude of this problem will grow with the depth of the tree being analysed and with the number of the moment, p, required. These problems are considered in more detail in

section 5.7; and in section 5.8, as in 4.10, it is pointed out that it will normally only be necessary to perform the calculation for p = 1 and 2, relatively simple cases.

5.6 Recursive solution for discrete cycle time distribution

The primary result of this section, a recursive scheme for the computation of a discrete approximation to cycle time distribution in tree-like networks which is convergent, is presented below as Theorem T5.4. It follows quite simply from Theorem T5.2, the basic result, using the same approach as that adopted in section 4.6.

First, it is necessary to define some more notation:

(i) Let the discrete approximation for the cumulative distribution of the time spent by the network A in state ν, $d_\nu(t)$, be denoted by

$$Q_\nu(\ell) = d_\nu(t_\ell) \qquad (\ell = 1,2,\ldots.)$$

where $t_\ell = \ell\Delta$ for some $\Delta \in R^+$

and let $q_\nu(\ell) = Q_\nu(\ell) - Q_\nu(\ell-1) \qquad (\ell \geq 2)$

$$q_\nu(1) = Q_\nu(1)$$

(ii) Let $x_\nu = e^{-\lambda_A(\nu)\Delta}$

so that $q_\nu(1) = 1 - x_\nu$

$$q_\nu(\ell) = x_\nu\, q_\nu(\ell - 1) \qquad (\ell \geq 2)$$

(iii) Let $G_{NA}(t|\alpha)$ have discrete form

$$\Psi_\alpha^{(NA)}(\ell) = G_{NA}(t_\ell|\alpha) \qquad (\ell \geq 1,\ \alpha \in S_{NA}^{(I)})$$

(iv) Let $H_{\alpha\beta}^{(NA)}(t)$, the distribution function of the time delay between states α and $\beta \in S_{NA}$ have discrete form

$$\Psi_{\alpha\beta}^{(NA)}(\ell) = H_{\alpha\beta}^{(NA)}(t_\ell) \qquad (\ell \geq 0;\ \alpha,\beta \in S_{NA})$$

recalling that $H_{\alpha\alpha}^{(NA)}(t) = 1$ for all $t \in R$, $t \geq 0$

(v) Let the approximations for $\Psi_\alpha^{(NA)}$ and $\Psi_{\alpha\beta}^{(NA)}$, obtained by convolving the discrete approximations, Q_ν, rather than the exact d_ν, be denoted by $\Pi_\alpha^{(NA)}$ and $\Pi_{\alpha\beta}^{(NA)}$ respectively.

(vi) Define $S_{NA}^{(E+)} = \left\{\nu \mid \nu \in S_{NA};\ \exists \beta \in E_{NA} \text{ s.t. } R_{\nu\beta}^{(NA)} \neq \phi\right\}$

$\supset E_{NA}$ clearly.

Theorem T5.4

Given a mesh $\{\ell\Delta \mid \ell = 1,2,\ldots\}$ of size $\Delta \in R^+$ on the time axis, the approximate, discrete form of cycle time distribution, $\Pi_\alpha^{(NA)}$, $\alpha \in S_{NA}^{(I)}$, is given by

(i) For $r = 0$, $\alpha \in S_{NA}$,

$$\Pi_\alpha^{(NA)}(\ell) = \begin{cases} x_\alpha\, \Pi_\alpha^{(NA)}(\ell-1) + (1-x_\alpha) \sum\limits_{\gamma \in S_{NA}} T_{\alpha\gamma}\, \Pi_\gamma^{(NA)}(\ell-1) & (\ell \geq 1) \\ \delta_{\alpha o} & (\ell = 0) \end{cases}$$

(ii) For $r \geq 1$, $\alpha \in S_{NA}^{(I)}$,

$$\Pi_\alpha^{(NA)}(\ell) = \begin{cases} \sum_{\beta \epsilon E_{NA}} \Pi_\alpha^{(NA)}(\ell|\beta) & (\ell \geq 1) \\ 0 & (\ell = 0) \end{cases}$$

where for $\beta \epsilon E_{NA}$ and $\nu \epsilon S_{NA}^{(E+)}$

$$\Pi_\nu^{(NA)}(\ell|\beta) = \begin{cases} x_\nu \Pi_\nu^{(NA)}(\ell-1|\beta) + (1-x_\nu) \sum_{\gamma \epsilon S_{NA}^{(E+)}} T_{\nu\gamma} \Pi_\gamma^{(NA)}(\ell-1|\beta) & \\ \qquad (\nu \neq \beta, \ell \geq 1) & \\ 0 \qquad (\nu \neq \beta,\ \ell = 0) & \\ \sum_{j=1}^{r} \frac{P_j \mu_{N_B}}{\lambda_A(\beta)} \Omega_{\beta j}^{(NA)}(\ell) \qquad (\nu = \beta) & \end{cases}$$

and for $\beta \epsilon E_{NA}$, $1 \leq j \leq r$

$$\Omega_{\beta j}^{(NA)}(\ell) = \begin{cases} x_\beta \Omega_{\beta j}^{(NA)}(\ell-1) + (1-x_\beta) \Pi_{\beta'(j)}^{(NA_j)}(\ell-1) & (\ell \geq 1) \\ 0 \qquad (\ell = 0) & \end{cases}$$

where $\beta'^{(j)}$ is as defined in Theorem T5.2 $(1 \leq j \leq r)$.

Proof

Proceeding as in section 4.6, the approximations, $\Pi_\alpha^{(NA)}$ and $\Pi_{\alpha\beta}^{(NA)}$ are derived by convolving the discrete representations of the corresponding continuous distributions for the times spent in successive states.

For the case of r = 0 (no primary sub-trees), the problem reduces to that of section 4.6 exactly, with the same result.

For $r > 0$, from Theorem T5.2,

$$\Pi_{\alpha}^{(NA)} = \sum_{\beta \epsilon E_{NA}} \sum_{j=1}^{r} \frac{P_j \mu_{N_B}}{\lambda_A(\beta)} \Pi_{\alpha\beta}^{(NA)} * Q_\beta * \Pi_{\beta^-(j)}^{(NA_j)}$$

$$(\alpha \in S_{NA}^{(I)})$$

Let

$$\Omega_{\beta j}^{(NA)} = Q_\beta * \Pi_{\beta^-(j)}^{(NA_j)} \qquad (\beta \in E_{NA})$$

so that, by Appendix 2, since Q_β is geometric,

$$\Omega_{\beta j}^{(NA)}(\ell) = \begin{cases} x_\beta \; \Omega_{\beta j}^{(NA)}(\ell-1) + (1-x_\beta) \Pi_{\beta^-(j)}^{(NA_j)}(\ell-1) & (\ell \geq 1) \\ 0 & (\ell = 0) \end{cases}$$

which is as used in the theorem.

Thus,

$$\Pi_{\alpha}^{(NA)} = \sum_{\beta \epsilon E_{NA}} \Pi_{\alpha\beta}^{(NA)} * \sum_{j=1}^{r} \frac{P_j \mu_{N_B}}{\lambda_A(\beta)} \Omega_{\beta j}^{(NA)}$$

Suppose

$$\Pi_{\nu}^{(NA)}(|\beta) = \Pi_{\nu\beta}^{(NA)} * \sum_{j=1}^{r} \frac{P_j \mu_{N_B}}{\lambda_A(\beta)} \Omega_{\beta j}^{(NA)}$$

$$(\nu \in S_{NA}^{(E+)}, \quad \beta \in E_{NA})$$

As in section 4.6, $\Pi_{\nu\beta}^{(NA)}(0)$ is defined to be $\delta_{\nu\beta}$ and so $\Pi_{\nu\nu}^{(NA)}(\ell)=1$; for all $\ell \geq 0$; $\nu, \beta \in S_{NA}$. This agrees with the requirement $H_{\nu\nu}^{(NA)}(t)= 1$; $t \geq 0$, $\nu \in S_{NA}$.

Thus, for $\nu = \beta$,

$$\Pi_\nu^{(NA)}(\ell|\beta) = \sum_{j=1}^{r} \frac{P_j \mu_{N_B}}{\lambda_A(\beta)} \Omega_{\beta j}^{(NA)}(\ell) \qquad (\ell \geq 0)$$

For $\nu \neq \beta$, $\nu \in S_{NA}^{(E+)}$ and $\beta \in E_{NA}$, $R_{\nu\beta}^{(NA)} \neq \phi$,

$$\Pi_{\nu\beta}^{(NA)} = \sum_{r \in R_{\nu\beta}} \prod_{k=1}^{|r|-1} T_{r_k r_{k+1}} Q_{r_k} \quad \dagger$$

$$= \sum_{r_2 \in D} \sum_{r' \in R_{r_2\beta}} T_{\nu r_2} Q_\nu * \prod_{i=1}^{|r'|-1} T_{r'_k r'_{k+1}} Q_{r'_k}$$

where $D = \{\gamma | \gamma \in S_{NA},\ \exists\, r \in R_{\nu\beta} \text{ s.t. } r_2 = \gamma\}$

$$= \{\gamma | \gamma \in S_{NA},\ T_{\nu\gamma} \neq 0,\ R_{\gamma\beta} \neq \phi\}$$

$$= \{\gamma | \gamma \in S_{NA}^{(E+)},\ T_{\nu\gamma} \neq 0,\ R_{\gamma\beta} \neq \phi\}$$

But for $T_{\nu r_2} = 0$ or $R_{r_2\beta} = \phi$, there is zero contribution to the sum,

$$\therefore \quad \Pi_\nu^{(NA)}(|\beta) = \sum_{\gamma \in S_{NA}^{(E+)}} T_{\nu\gamma} Q_\nu * \Pi_\gamma^{(NA)}(|\beta)$$

and so, by Appendix 2,

$$\Pi_\nu^{(NA)}(\ell|\beta) = \begin{cases} x_\nu \Pi_\nu^{(NA)}(\ell-1|\beta) + (1-x_\nu) \sum_{\gamma \in S_{NA}^{(E+)}} T_{\nu\gamma} \Pi_\gamma^{(NA)}(\ell-1|\beta) & \\ & (\ell \geq 1) \\ 0 \qquad (\ell = 0) & \end{cases}$$

The proof is thus complete.

---

† omitting the superscript (NA) from $R_{\nu\beta}$ for brevity.

Corollary CT5.4

As $\Delta \to 0$,

$$\Pi_{\alpha\beta}^{(NA)} \to \Psi_{\alpha\beta}^{(NA)} \qquad (\alpha,\beta \in S_{NA})$$

and $$\Pi_{\alpha}^{(NA)} \to \Psi_{\alpha}^{(NA)} \qquad (\alpha \in S_{NA}^{(I)})$$

Proof

For r = 0, the result is precisely that of Proposition P4.4.

For r > 0, the result for $\Pi_{\alpha\beta}^{(NA)}$ is again that of Proposition P4.4 and thus it is sufficient to show that

$$\left\{\Pi_{\alpha}^{(NA)}(\ell|\beta) \,|\, \alpha \in S_{NA}^{(I)} \;;\; \beta \in E_{NA}\right\}$$

are convergent as $\Delta \to 0$, since S_{NA} and so E_{NA} is finite and

$$\Pi_{\alpha}^{(NA)} = \sum_{\beta \in E_{NA}} \Pi_{\alpha}^{(NA)}(|\beta)$$

The proof is by induction on the depth of the tree-like network. For

$$\alpha \in S_{NA}^{(E+)} \;;\; \beta \in E_{NA} \;;\; \alpha \neq \beta$$

$$\Pi_{\alpha}(|\beta) = \sum_{\gamma \in S_{NA}^{(E+)}} T_{\alpha\gamma}\, Q_{\alpha} * \Pi_{\gamma}^{(NA)}(|\beta)$$

so that the proof again follows via Propositions P4.3 and P4.4, the summation being finite.

Finally, for $\alpha = \beta$

$$\Pi_{\alpha}(|\beta) = \sum_{j=1}^{r} \frac{P_j \mu_{N_B}}{\lambda_A(\beta)}\, \Omega_{\beta j}^{(NA)}$$

But $\Omega_{\beta j}^{(NA)} = Q_\beta * \Pi_{\beta'(j)}^{(NA_j)}$

and by hypothesis, $\Pi_{\beta'(j)}^{(NA_j)}$ is convergent as $\Delta \to 0$

and so, via Corollary CL4.3 and Proposition P4.4

$\Omega_{\beta j}^{(NA)}$ is convergent.

The complexity of the computation of the result of Theorem T5.4 is rather awesome at first sight, with repeated recursive calls to functions representing $\Pi_\alpha^{(NA)}(|\beta)$ and $\Omega_{\beta j}^{(NA)}$ (in the notation of the Theorem).

However, the situation is not as bad as it looks, particularly for trees of small depth, and a simplification in the computation of $\Omega_{\beta j}^{(NA)}$ is possible as shown in the next section.

5.7 Significant computational problems

5.7.1 How can they be reduced?

As can be seen from the recurrence relations derived for the various results of the previous sections, the numerical computations involved are based on the linear composition, via recursive techniques, of the parallel quantities derived for tandem networks in chapter 4. However, such quantities must be computed for start and end states which are in certain sub-sets of the state space, the state immediately following departure of the test customer from a segment no longer being restricted to that of the empty network. This is discussed in more detail in section 5.7.2, but it will be noticed that no new fundamental techniques have been introduced, only the composition of a

tree-like network from its sub-trees.

As with most recursive schemes, a major problem in numerical evaluation of results is that of efficiency, with respect to both storage and computation time requirements. In order to allow any solution at all to be generated for networks of fairly modest complexity, a purely recursive program in any existing programming language is precluded and some additional techniques have been supplied to augment such recursion in the APL package of Appendix 7. Such aids to efficiency are discussed in general in section 5.7.3, and in the next section a computationally simpler expression for $\Omega_{\beta j}^{(NA)}$, defined in Theorem T5.4 of section 5.6, is provided, so improving the efficiency of the calculation of the approximate discrete form of the cycle time distribution.

5.7.2 Transitions between predefined start and end states

In sections 5.4, 5.5 and 5.6, the results derived for the Laplace transform of cycle time distribution, its moments and its approximate discrete form involved terms $(I-T^*)^{-1}_{\alpha\beta}$, $X_{\alpha\beta}^{(p)}$ and $\Pi_{\alpha}^{(NA)}(|\beta)$ respectively, where α is a "start state", $\alpha \in S_{NA}^{(I)}$, and β is a "target state" or "end state", $\beta \in E_{NA}$. X is determined solely in terms of the transition matrix, T and Π is given by an iterative relationship also involving T. Thus, the computation of such quantities requires the following:

(i) Computation of the transition matrix, T and its modified form, T^*. This is accomplished as described in section 5.4 and in order to save storage space, the representation is as a vector with 2 control vectors to delimit the rows (by means of pointers) and show the column numbers of the non zero elements. In other words, the representation is precisely that described

in section 4.7.2 for the case of cyclic networks.

(ii) Computation of the quantities for multiple target states, β. The methods used for this are parallel with those of sections 4.7.3, 4.7.4 and 4.7.5 for each of the three above listed expressions respectively, with modifications to account for the fact that β will not, in general, be represented by a state numbered 1.

The computation of $(I-T^*)^{-1}_{\alpha\beta}$ requires only very simple modification to the method. As in section 4.7.3, let α and β map into states numbered a and b respectively under the mapping f_A^{-1} of Proposition P5.3.

Then, given $b \in f_A^{-1}(S_{NA})$,

$\left\{(I-T^*)^{-1}_{ab} \mid f_A(a) \in S_{NA}\right\}$ is the solution, $\left\{x_a^{(b)}\right\}$ say, to the equations

$$\sum_{c \in f_A^{-1}(S_{NA})} (I-T^*)_{ac} \; x_c^{(b)} = \delta_{ab}$$

which is given by

$$x_a^{(b)} = 0 \qquad (1 \leq a < b)$$

$$x_b^{(b)} = 1$$

$$x_a^{(b)} = \sum_{j=b}^{a-1} T^*_{aj} \; x_j^{(b)} \qquad (a > b)$$

since T^* is lower triangular, and then

$$(I-T^*)^{-1}_{ab} = x_a^{(b)} \qquad (a,b \in f_A^{-1}(S_{NA}))$$

Thus, for any such b, the iterative procedure of section 4.7.3 may be used here, with different starting conditions. The function ESB in the APL package of Appendix 7 is just this generalisation of the function of the same name referenced for cyclic networks. Of course, results will be needed for several values of b, corresponding to $\beta \in E_{NA}$, and these are produced by repeated calls to ESB, admittedly not the most efficient method in terms of execution time, although saving on storage if the results for the complete range of b are not all saved.

In the second case, $X^{(p)}_{\alpha\beta}$, a similar problem arises. Referring to Theorem T5.3, $X^{(p)}_{\alpha\beta}$ is represented under the mapping f_A used above by a sum of terms of the form

$$\left\{ \prod_{i=1}^{h} (I-T)^{-1}_{k_{i-1}j_i} \; T'(m_i)_{j_i k_i} \right\} \; (I-T)^{-1}_{k_h b}$$

(with summation convention)

for some m_i, $h \in Z^+$; $1 \leq i \leq h$ and where $k_0 = a$.

Proceeding as in section 4.7.4, relabelling E,E′ defined therein by F,F′ respectively to avoid a clash of notation, define

$$F'_{k_{\ell-1}b} = \left\{ \prod_{i=\ell}^{h} (I-T)^{-1}_{k_{i-1}j_i} \; T'(m_i)_{j_i k_i} \right\} \; (I-T)^{-1}_{k_h b} \qquad (1 \leq \ell \leq h)$$

$$F'_{k_h b} = (I-T)^{-1}_{k_h b}$$

and $$F_{j_\ell b} = T'(m_\ell)_{j_\ell k_\ell} \; F'_{k_\ell b} \qquad (1 \leq \ell \leq h)$$

so that $X^{(p)}_{\alpha\beta}$ is given by F'_{ab} which can be computed iteratively.

Now, values of F_{ab} are required for all a,b with $f_A(a) \in S_{NA}^{(I)}$ and $f_A(b) \in E_{NA}$ so that $\{F_{ab}\}$ is not a vector as was the case for cyclic networks in section 4.7.4.

Now, $F'_{k_h b}$ may be computed (by the function ESB) as above, with the result represented in column oriented form, one column vector for each $b \in f_A^{-1}(E_{NA})$. Next, $F_{j_h b}$ may be computed, also in column form, by successively taking the inner product of $T'(m_h)$ with the columns of $F'_{k_h b}$, a simple process since $T'(m_h)$ is stored by rows, recall (i). This inner product is performed by the function MSM in Appendix 7. Then, $F'_{k_{h-1} b}$ is the solution to the equations

$$(I-T)_{ij} F'_{jb} = F_{ib} \qquad (i,j \in S_{NA})$$

which can be solved by direct back substitution, since T is lower triangular, giving F'_{jb} in column form for each b. This is performed by the function HSB in Appendix 7. Proceeding in this way to calculate $F'_{k_\ell b}$ for $\ell = h, h-1, \ldots .0$, one can see by a straightforward inductive argument that a value for $F'_{k_0 b}$ representing $X^{(p)}_{\alpha\beta}$ results.

For the case of $\Pi_\alpha^{(NA)}(|\beta)$, $(\alpha \in S_{NA}^{(E+)}, \beta \in E_{NA})$, the only significant problem is that of the computation of

$$\sum_{\gamma \in S_{NA}^{(E+)}} T_{\alpha\gamma} \; \Pi_\gamma^{(NA)}(|\beta)$$

for each $\beta \in E_{NA}$. This problem is again resolved, as for the case of $X^{(p)}_{\alpha\beta}$ above, by representing $\left\{ \Pi_\alpha^{(NA)}(|\beta) \right\}$ in column oriented form, the pre-multiplication by T then being simple,

giving a column form result. The function used in Appendix 7 to accomplish this, MSE, is a version of MSM modified to sum over the domain $S_{NA}^{(E+)}$ rather than the whole of S_{NA}.

It will be appreciated that the property of the integers-states mapping f_A that the states in the sub-spaces $S_{NA}^{(E+)}$ and E_{NA} are numbered consecutively (see section 5.4) is of great value in the implementation of functions to compute the results presented in this chapter; the relevance being, of course, to the resulting simple indexing of the part-rows or part-columns required corresponding to the various domains of summation.

5.7.3 Space and time constraints

From the results presented in sections 5.4, 5.5, 5.6 for the Laplace transform, moments and discrete form of the cycle time distribution in tree-like queueing networks respectively, it can be seen that in a direct implementation, many recursive function calls corresponding to each of the network's sub-trees would be necessary. In fact one could envisage, in each of the three cases, one call for each of the states of E_{NA} in the network for which a sub-tree is primary. Furthermore, in the cases of the Laplace transform and the discrete form calculations, a call could be necessary for each parameter value and time interval sequence number respectively. Such a large number of recursive function calls would clearly result in considerable inefficiency with respect to both storage and computation time.

The principle dilemma concerns the computation of the transition matrices for sub-trees; once the transition matrix for the whole network's state space is available, that for any

sub-space may be obtained by selecting appropriate rows and columns therefrom and re-normalising. The problem is in selection of such rows and columns. A further, if minor, complication is that under any state-integer mapping, in general the sub-space of states involved in the centre service completions in any sub-tree cannot be numbered consecutively. This presents no problem when storage space is unlimited and matrices are uncompressed, but in the linear representation described for the sparse transition matrices considered here the state-integer mapping functions must be invoked frequently.

A row in a sub-tree state transition matrix may be computed as follows:

(i) find the state vector corresponding to the row number by applying the integer-state mapping for the sub-tree;

(ii) find the row number in the state transition matrix for the whole state space corresponding to this state vector, expanded to represent zero customers at all centres not in the sub-tree in question. This involves application of the state-integer mapping function for the whole network;

(iii) map each column number associated with this row for the whole network into a column number for the sub-tree by applying (ii) and (i) in reverse.

Of course, corresponding to any sub-tree state there will be, in general, several states in the whole network with the numbers of customers at each centre of the sub-tree determined by the sub-tree state; the expansion of (ii) guarantees that any transition in the whole network is due to a service completion in the sub-tree, and so that every associated column in the whole network matrix is required for the sub-tree matrix.

Thus, the choices available concerning the transition matrices for sub-trees are as follows:

(a) Only store the transition matrix for the whole network, and whenever a row is required for a sub-tree, compute it as described in (i) - (iii) above;

(b) Since very many such (repeated) row computations will be required, (a) is very inefficient in terms of execution time and can be improved, at the expense of storage, by pre-computing as in (a) the whole of the sub-tree state transition matrix, in sparse form, and storing it;

(c) The state transition matrix may be computed from first principles for each sub-tree and stored.

It was found that option (c) turned out to be the most convenient to program and the fastest to execute in the cases tried. However, for any of the sub-tree recursive functions' evaluations a large proportion of the total execution time required is used in the computation of the state transition matrix for that sub-tree; in addition, the storage requirement of this matrix is also considerable, even when represented in the linear form described in section 5.7.2.

Thus it is desirable only to enter the function corresponding to each sub-tree once in any calculation, to compute all the results corresponding to every initial state (associated with the target states in E_{NA} for the tree of which the sub-tree is primary), and every Laplace transform parameter or time interval sequence number as appropriate. In this way it is possible not only to reduce the number of function calls

required, but also to save space by using a global variable for the state transition matrices of all sub-trees. Such use of impure recursive techniques is, of course, undesirable where resources are unlimited, but necessary here. However, the introduction of global variables demands a great deal of care in the writing of the recursive programs, viz. with respect to the order of evaluation within each function which must not call a similar function for a sub-tree if reference to its transition matrix is required subsequently. Such use of global variables may be found explicitly in Appendix 7 in the functions LT, MOM and DST for the calculations of the Laplace transform, moments and discrete form of cycle time distribution respectively.

The reduction of the number of function calls by means of storing intermediate results in this way is in effect an explicit implementation of a "Memo Rule" type of system for efficient evaluation of recursive functions, see [MICH67,HARR74]. Thus it has the same limitation; viz. that if very many such intermediate results are required, insufficient storage may be available. Thus another trade-off situation arises. Nevertheless, this storage problem is only linear here since the number of sets of intermediate results existing on the run time stack at any one time cannot exceed the total number of sub-trees in the network.

Clearly, for increasingly large networks, as the size of the sub-space E_{NA} becomes excessive, a storage overflow would ultimately occur, but for quite significant cases, such as those presented in Appendix 8, this problem did not arise.

5.7.4 Enhancement of the discrete distribution recursion

In this section, an enhancement to the efficiency of the computation of the approximate discrete cycle time distribution is described. As explained in the previous section, for this computation it is preferable to compute in parallel the results for every time interval number, ℓ, (in the notation of Theorem T5.4, section 5.6) in a single function call associated with a sub-tree.

This presents no problem, certainly in APL, the programming language of the software package of Appendix 7, for all except computation of the term $\Omega_{\beta j}^{(NA)}(\ell)$ where

ℓ is the time interval number $(1 \leq \ell \leq L$ for some $L \varepsilon Z^{+})$,

A is the tree-like network under analysis,

N is the population of the network (initially),

$\beta \varepsilon E_{NA}$,

j is the number of a primary sub-tree of A.

Evaluation of $\Omega_{\beta j}^{(NA)}$ by direct recursive methods would include a function call for each of the L values of ℓ specified. These would each call functions corresponding to the primary sub-trees of A, so overwriting the state transition matrix of A which is required for subsequent values of ℓ.

However, the expression for $\Omega_{\beta j}^{(NA)}(\ell)$ in the form of the explicit summation for the convolution involved, permits its values corresponding to multiple values of ℓ to be produced in parallel. This expression is given by the following Proposition.

Proposition P5.4

In the notation of Theorem T5.4,

$$\Omega_{\beta j}^{(NA)}(\ell) = \sum_{i=0}^{\ell-1} (x_\beta)^i (1-x_\beta) \Pi_{\beta\text{-}(j)}^{(NA_j)}(\ell-1-i)$$

Proof

By induction on ℓ or direct substitution using the definition of Theorem T5.4,

$$\Omega_{\beta j}^{(NA)}(\ell) = \begin{cases} x_\beta \Omega_{\beta j}^{(NA)}(\ell-1) + (1-x_\beta) \Pi_{\beta\text{-}(j)}^{(NA_j)}(\ell-1) & (\ell > 0) \\ 0 & (\ell = 0) \end{cases}$$

For $\ell = 0$ the sum is, of course, empty.

By substitution of this expression directly into the definition of $\Pi_\nu^{(NA)}(\ell|\beta)$ $(\nu \in S_{NA}^{(E+)})$ given in Theorem T5.4, the reference to $\Omega_{\beta j}^{(NA)}$ is eliminated and it is clear that the results for any sub-tree may now be computed in parallel for all values of ℓ, $1 \leq \ell \leq L$, as required and as accomplished in the APL function DST of Appendix 7.

5.8 Concluding remarks

5.8.1 The topics discussed

The comments relating to the analysis of tree-like networks presented here are, in the main, the same as those made for the analysis of cyclic networks in chapter 4.

Thus, this discussion will be limited to a summary of that given for cyclic networks, but related to tree-like networks, the details being available in the relevant sections of chapter 4. The following topics are considered: Laplace transform inversion, decomposition methods, response time distribution and finally some general remarks are made leading in to the following chapters.

5.8.2 Laplace transform inversion

Although at first sight a method of inverting the Laplace transform of the cycle time distribution, derived exactly for tree-like networks here, might seem attractive, it is not considered worthwhile, for the reasons given in section 4.8, in brief:

(a) An approximate discrete form for this distribution has been derived (section 5.6), and is convergent;

(b) The more important distribution is that of response time, a sum of several consecutive cycle times;

(c) Inversion of Laplace transforms poses many problems.

In fact, the Laplace transform is itself of great use for predicting performance measures directly, see section 4.8.

5.8.3 Decomposition methods

Such methods were discussed at some length for cyclic networks in section 4.9, but for tree-like networks their ease of application is at once apparent in view of the recursive nature of such networks. Indeed, the "memo rule" method of storing intermediate results, referred to in section 5.7.3, uses precisely decomposition methods, whether implemented by the system or explicitly by limiting the number of function calls. In order to achieve full decomposition techniques in the sense of section 4.9, it would only be necessary that all the results for every possible start state in the relevant sub-trees be saved permanently and so be available from run to run of the implementation. This could result in a large storage requirement in that the (initial) population of any sub-tree may take any value between one and the initial population of the whole network, so that there are many possible start states. Furthermore, in the case of calculation of the Laplace transform or the discrete cycle time distribution approximation, the number of results involved is proportional to the number of parameters specified.

Such techniques are not implemented in the package of Appendix 7.

5.8.4 Response time distribution

Of great importance to the analyst is the distribution of response time in an interactive or real time computer system. Whilst clearly closely related to cycle time distribution, it is obvious that the latter distribution is not adequate in itself, rather multiple convolutions for successive cycles being the chief objective. However, as indicated in section 4.10, to

derive such a convolution for successive cycles poses excessive, indeed prohibitive, computational problems. Thus the approach taken with regard to response time is to assume that the cycles considered in *any* constitution of a time delay,(in particular response time), are statistically independent so that the Central Limit Theorem may be applied. In this case it is only necessary to calculate the first two moments of cycle time distribution, which are given exactly by Theorem T5.3 in section 5.5. The accuracy of this independence assumption has not been tested analytically, but empirical tests on the independence of cycle times are reported in Appendix 8.

5.8.5 Summary

The exact derivation of cycle time distribution for tree-like queueing networks presented here is a considerable generalisation of the method developed for cyclic networks in chapter 4. As such it has the same limitation with respect to efficiency (in execution time and storage requirement) and similar types of application. Its own use as a practical tool for the performance analyst must be limited to simple cases only, and a major application is in the validation of approximate techniques, notably the PSA method of chapter 3, applied to tree-like networks. As emphasised in earlier chapters, it is the approximate PSA method which is intended as the major practical tool in view of its far superior efficiency (see section 6.5). Ideally a formal analysis to provide bounds on the error of the PSA method should be undertaken, at least for the now solved tree-like networks. This is discussed further in chapter 8 as a future research direction. In the following chapter an empirical validation of the PSA method is described

for the case of tree-like networks by comparing results with those of the corresponding exact methods and also with corresponding simulation runs.

Although the tree-like network is not fully general, computation of the exact cycle time distribution in non tree-like networks involves explicit tagging of a customer and an approach along the lines of [YU77]. The solution obtained in this way, as discussed in section 2.3, is a special case of results in the general theory of stochastic processes. This is impractical for non trivial cases in that it is necessary to solve a system of linear equations, the number of which is several times that of the order of the original state space of the network.

Validation of the assumptions and implementation of the exact method described in this chapter is discussed in the following chapter, and a comparison with simulated results may be found in Appendix 8.

§6. Validation of the Theoretical Models

6.1 Introduction

The purpose of validation is to support ones conviction about the accuracy of a model. Now, inaccuracies may be introduced into a model at two levels:

(a) In the process of abstraction from the physical system represented. For example, approximations must be made in order to allow development of a mathematically solvable analytic model or programmable simulation model.

(b) In approximations which may be necessary in the computation of model predictions. For example in an analytic model an infinite series may be truncated or a continuous function approximated by a discrete representation as here, see sections 3.3.3, 4.6 and 5.6.

Ideally one would like to perform a formal error analysis to assess inaccuracies of type (b)- and also of type (a) if a system with formal definition is modelled (e.g. a model itself). However, such an analysis is frequently not possible, for example in the case of our PSA results at the present time. Thus validation is often performed in either case by making statistical assessments of the error existing for a (representative) selection of environments, as represented by a choice of model parameter values.

The suite of APL functions, developed by the author and listed in Appendix 7, constitute analytic models providing numerical predictions for properties of cycle time distribution in tree-like queueing networks. These functions are based on the results derived in previous chapters and may produce exact or approximate (according to PSA) predictions. The parameter

specification permitted for the tree-like networks, in terms of network topology and customer population, is fully general, limited only by the computing resources available. Clearly for sufficiently complex networks or large populations, the size of the state spaces involved would result in excessive storage and/or CPU requirements, especially in the case of the exact analysis. This is discussed on a quantitative basis in Appendix 8.

Now, the objective of the theoretical analysis is the provision of models capable of representing a variety of computer system configurations in various environments. Thus validation of type (a) must be based on a comparison with data measured on actual systems. However, as stated in previous chapters, this type of data is of a fine level of detail, for example requiring measurements to be recorded whenever a CPU is switched or an I-O transfer is initiated. As a result, such data is costly, and therefore in practice administratively difficult, to obtain.

Because no measured data is available at present, ultimate validation of both the PSA and the (assumptions of the) exact models is not possible. However, some validation is possible:

(i) Of the PSA model with respect to the exact one by comparison of the results of the two models for networks conforming to the assumptions of the latter.

(ii) By constructing a network simulation model of a real system to generate test data on which to base validation of both analytic models. The independence of such validation is limited in that the simulation model

(a) itself requires validation,

(b) necessarily has built into it assumptions

similar to those made for the theoretical analysis, and (c) by design will be statistically well-behaved.

The next section outlines how a systematic *mutual validation* methodology, based on the above, can increase confidence in the analytic models.

6.2 A mutual-validation methodology

This is based on the following observations:

(i) Given its assumptions, we may assume that the mathematical analysis of the previous chapters has yielded correct results. However, no attempt has been made to prove their programmed *implementation* correct. Thus an independent check should be made to substantiate any numerical results.

(ii) The same applies to the approximate results of the PSA analysis for its less restrictive set of assumptions. In addition, we would also wish to validate the assumptions on which the approximation is based.

(iii) The assumptions of the simulation model can be adjusted to match either exact or PSA models. Since the computational procedure is quite different, its numerical results can provide *independent* confirmation of the accuracy of the implementation of each theoretical model. However, the simulation model itself may contain errors, but agreement between two models supports ones conviction in the accuracy of each.

This leads to the following systematic scheme for validation of the three models - exact analytic, PSA analytic and simulation:

(a) Compare the numerical predictions of each model for a "base set" of networks conforming to the assumptions of each model type; i.e. to those of the exact model. Their agreement will mutually validate the computational procedures of each in this restricted domain.

(b) At this point it is *assumed*, any proof or validation being impossible for the present, that the simulation model may be *extrapolated*, by relaxing the assumptions of (a), to represent adequately non base set network specifications.

(c) We may now assess models with respect to a simulation model, assumed validated itself. Such a process will indicate the adequacy of the PSA model for networks which conform to its own assumptions and also of both analytic models for networks in which their respective assumptions do not hold; a test on the "robustness property" (section 2.2) applied here to time delay prediction in queueing networks. The extrapolation of the simulation model may *support* but cannot validate ultimately the PSA model (in particular) in its more general domain of applicability.

6.3 Concluding remarks

The implementation of the methodology described in the previous section is a necessary and important step in the development of representative models for prediction of time delays in computer systems. This is particularly so when no measured data is available. The detailed procedure constituted

a significant part of the research reported in this thesis, but its philosophical value is rather less than that of other chapters, no fundamental advance in validation methodology being proposed. Thus the details are presented in Appendix 8.

The conclusions drawn there may be summarised as:

(i) As required, the exact and simulation models were in agreement for base set networks conforming to the assumptions of the exact model.

(ii) The PSA approximation appeared very good for networks conforming to its much less restrictive assumptions, according to comparisons with the simulated results.

(iii) The robustness property of queueing networks was emphatically not demonstrated in this study of time delays, neither for the exact nor the PSA models.

(iv) The simulated successive cycle times of a particular customer in the tree-like queueing networks considered were independently distributed. Thus it is valid, at *least* in these cases, to apply the Central Limit Theorem in order to obtain an approximation to response time distribution, see sections 4.10, 5.8.4.

The most important practical conclusion was the accuracy of the PSA model. Since its assumptions hold for a wide range of applications, it has emerged as a tool of great potential for the computer performance analyst.

Finally, the procedure detailed in Appendix 8 not only achieves mutual validation as described in the previous section but also demonstrates the methodology for the ultimate validation with respect to measured data.

§7. Analysis of Transients in Queueing Networks

7.1 The need for transient analysis

It is almost universally assumed in queueing network modelling that the network under analysis is in a state of stochastic equilibrium, i.e. that the state space probabilities are time independent. It is not disputed that this assumption will be valid for the overwhelmingly larger proportion of time periods considered. However, no analysis has been undertaken, so far as is known to the author, to determine precisely when this assumption can and can not be made; i.e. to estimate the time constant of the transient component of the time dependent state space probability distribution. Moreover, it should be recognised that there are also time intervals of great significance to the modeller during which the assumption is *not* valid. For example, the immediate effects of any disturbance to the modelled system, such as the adjustment of a scheduling parameter, may be predicted. This type of application is considered further in chapter 8.

In the following section, an iterative approximate solution is derived of the time dependent Kolmogorov differential-difference equations for the state space probabilities of Jackson or Gordon-Newell [JACK63,GORD67] type queueing networks. The result demonstrates clearly, to first order, the way in which the transient component decays and is shown to converge to the exact time dependent solution. Originally the approach was developed by the author for cyclic networks of just two centres as an improvement to the PSA approximation for cycle time distribution, by way of an analysis in continuous time, [HARR78a]. The results of this research are summarised in section 7.3 and

the intuitive assessment of the accuracy of the PSA method for various classes of network (section 3.1) is given some formal support.

7.2 Solution of the Kolmogorov equations

7.2.1 The convergent iterative solution

Let a Markovian queueing network of M exponential servers with FCFS queueing discipline have state space S and let

$P(\underline{k},t)$ = Prob(queue length at server $i=k_i$ at time t, $1 \le i \le M$) ($\underline{k} \in S$).

For notational conciseness and clarity a closed network is considered, the modifications necessary for open networks being simple.

The time dependent balance equations for the underlying Markov process are, [KLEI75],

$$P(\underline{k},t+\Delta t) = \left\{1 - \Delta t \sum_{i=1}^{M} \mu_i(k_i)\right\} P(\underline{k},t) + \Delta t \sum_{i=1}^{M} \sum_{j=1}^{M} \varepsilon(k_i) p_{ji} \mu_j(k_j+1) P(\underline{k}^{ij},t) + o(\Delta t) \qquad \text{.....(E7.1)}$$

where, for $1 \le i,j \le M$,

$\mu_i(n)$ is the service rate of server i when its queue length is $n \in Z^+$

$\mu_i(0) = 0$

$$\varepsilon(k_i) = \begin{cases} 0 & (k_i = 0) \\ 1 & \text{otherwise} \end{cases}$$

$$\underline{k}^{ij} = \begin{cases} (k_1,\ldots,k_i-1,\ldots,k_j+1,\ldots,k_M) & (i\neq j) \\ \underline{k} & (i=j) \end{cases}$$

p_{ji} is the routing probability from centre j to centre i, $j \neq i$.

The factor $\varepsilon(k_i)$ is included to suppress invalid transitions from states with negative valued queue lengths.

It is assumed, without loss of generality, that $p_{ii} = 0$, the modification of the method being simple if this is not so, [GORD67].

Rearranging the equations (E7.1) and dividing by Δt, the resulting Kolmogorov forward equations for the network are, in the limit $\Delta t \to 0$, for $\underline{k} \in S$,

$$\frac{\partial P(\underline{k},t)}{\partial t} = -\lambda(\underline{k})P(\underline{k},t) + \sum_{i=1}^{M}\sum_{j=1}^{M} \varepsilon(k_i)p_{ji}\mu_j(k_j+1)P(\underline{k}^{ij},t) \quad \ldots\ldots(E7.2)$$

where $\lambda(\underline{k}) = \sum_{i=1}^{M} \mu_i(k_i)$ is the total service rate in state $\underline{k} \in S$.

The iteration applied for solution of equations (E7.2) is defined as follows:

Given the set of first order differential equations

$$\frac{\partial P_i}{\partial t} = F_i(P_1,P_2,\ldots,P_m) \qquad (1\leq i\leq m)$$

for some (suitably smooth) functions F_i, let $P_i^{(n)}$ be the n'th order approximation for the solution, P_i. Then $P_i^{(n+1)}$ is defined to be the solution of

$$\frac{\partial P_i}{\partial t} = F_i(P_1^{(n)},P_2^{(n)},\ldots,P_{i-1}^{(n)},P_i,P_{i+1}^{(n)},\ldots,P_m^{(n)})$$

$$(1\leq i\leq m)$$

In the case of equations (E7.2), m = M, $P_i = P(\underline{k},t)$ and $P_i^{(n)}$ will be denoted by $P_n(\underline{k},t)$. This mapping is valid since the state space S is countable (M is finite); in fact finite for the closed network considered here.

Thus the (n+1)'th order approximation for the non-normalised time dependent state space probabilities is the solution to the equations

$$\frac{\partial P(\underline{k},t)}{\partial t} = -\lambda(\underline{k})P(\underline{k},t) + \sum_{1\le i\ne j\le M} \varepsilon(k_i)p_{ji}\mu_j(k_j+1)P_n(\underline{k}^{ij},t)$$

$$\therefore \frac{\partial}{\partial t}\left\{e^{\lambda(\underline{k})t}P(\underline{k},t)\right\} = e^{\lambda(\underline{k})t} \sum_{1\le i\ne j\le M} \varepsilon(k_i)p_{ji}\mu_j(k_j+1)P_n(\underline{k}^{ij},t)$$

so that

$$P_{n+1}(\underline{k},t) = e^{-\lambda(\underline{k})t}P(\underline{k},0) + e^{-\lambda(\underline{k})t}\int_0^t e^{\lambda(\underline{k})u}\left\{\sum_{i\ne j}\varepsilon(k_i)p_{ji}\mu_j(k_j+1)P_n(\underline{k}^{ij},u)\right\}du \quad .....(E7.3)$$

where $\{P(\underline{k},0)\,|\,\underline{k}\varepsilon S\}$ is the initial state space probability distribution. The normalised form is obtained by dividing by $\sum_{\underline{k}\varepsilon S} P_{n+1}(\underline{k},t)$.

The zero order approximation is chosen to be the equilibrium distribution so that

$$P_0(\underline{k},t) = \Theta(\underline{k}) \quad \text{where}$$

$$\Theta(\underline{k})\,\lambda(\underline{k}) = \sum_{1\le i\ne j\le M} \varepsilon(k_i)p_{ji}\mu_j(k_j+1)\Theta(\underline{k}^{ij}) \quad .....(E7.4)$$

$\Theta(\underline{k})$ is the well known product form solution for the equilibrium state space probabilities, [GORD67,JACK63].

This iterative scheme gives, in particular, a first order approximation

$$P_1(\underline{k},t) = e^{-\lambda(\underline{k})t} P(\underline{k},0) + \{1 - e^{-\lambda(\underline{k})t}\}\theta(\underline{k}) \quad \ldots (E7.5)$$

by substituting (E7.4) in (E7.3) and performing the simple integral.

This result is intuitively pleasing in that it is an exponentially weighted average of the initial and equilibrium state space probability distributions. The initial distribution contribution dies away exponentially with *time constant* $\lambda(\underline{k})^{-1}$, $\underline{k} \in S$, the mean time to the next service completion (at any server) in state $\underline{k}$. The equilibrium distribution is accordingly approached exponentially also. Also pleasing is that every iteration (except the zero'th) gives exact results at time $t=0$ and as $t \to \infty$, as shown in the following proposition.

Proposition P7.1

In the notation above, for all $n \in Z^+$, $\underline{k} \in S$,

$$P_n(\underline{k},0) = P(\underline{k},0)$$

and $$P_n(\underline{k},t) \to \theta(\underline{k}) \quad \text{as} \quad t \to \infty .$$

Proof

The first part of the proposition is trivial.

For the second part, suppose

$$P_m(\underline{k},t) \to \theta(\underline{k}) \quad \text{for all } \underline{k} \in S \qquad m < n \in Z^+$$

as $t \to \infty$

Then for all $\varepsilon > 0$, $\underline{k} \in S$ $\exists T_{\underline{k}} \in R^+$ s.t.

$$| P_{n-1}(\underline{k},t) - \theta(\underline{k}) | < \varepsilon/3\sigma_{\underline{k}}M^2$$

$$\text{for all} \quad t > T_{\underline{k}} \ ,$$

where $$\sigma_{\underline{k}} = \max_{1\le i,j\le M} \{\mu_j(k_j)/\lambda(\underline{k}^{ji})\}.$$

Let $$T'_{\underline{k}} = \max_{\substack{\underline{n} = \underline{k}^{ij} \\ 1\le i,j\le M}} T_{\underline{n}} \ .$$

Then, by the balance equations for $\theta(\underline{k})$ and the triangle inequality,

$$| \{\lambda(\underline{k})\}^{-1} \sum_{1\le i\ne j\le M} \varepsilon(k_i)p_{ji}\mu_j(k_j+1)P_{n-1}(\underline{k}^{ij},t) - \theta(\underline{k}) | < \varepsilon/3$$

$$\text{for all} \quad t > T'_{\underline{k}} \ .$$

Also, $\exists\, U_{\underline{k}} \in \mathbb{R}^+$ s.t. $|e^{-\lambda(\underline{k})t}| < \varepsilon/3$ for all $t > U_{\underline{k}}$.

Hence, by definition of the iteration and since

$$|P(\underline{k},0)|, |\theta(\underline{k})| \le 1 \ ,$$

$$| P_n(\underline{k},t) - \theta(\underline{k}) | < \varepsilon/3 + (1 - e^{-\lambda(\underline{k})t})\ \varepsilon/3 + \varepsilon/3$$

$$< \varepsilon \quad \text{for all} \quad t > \max(U_{\underline{k}}, T'_{\underline{k}})$$

by basic inequalities of mathematical analysis.

$\therefore$ $$P_n(\underline{k},t) \to \theta(\underline{k}) \quad \text{as} \quad t \to \infty$$

and the proposition is proved by induction.

It is now shown that the iteration defined in this section converges to the exact time dependent solution.

The proof of the convergence property requires the following Lemma.

Lemma L7.1

For $\alpha,\beta \in S$, let

$\theta(\alpha,\beta)$ be the number of the centre from which a departure causes a state transition $\alpha \to \beta$ which is undefined if a one-step transition $\alpha \to \beta$ is invalid. Similarly let $\phi(\alpha,\beta)$ be the number of the centre at which a customer arrives immediately after the one-step transition $\alpha \to \beta$, c.f. section 5.4.

Let

$$\Xi_{\beta\alpha} = \begin{cases} \dfrac{p_{\theta(\alpha,\beta)\phi(\alpha,\beta)}\mu_{\theta(\alpha,\beta)}(\alpha_{\theta(\alpha,\beta)})}{\lambda(\beta)} & \text{if a one-step transition } \alpha \to \beta \text{ is valid} \\ 0 & \text{otherwise} \end{cases}$$

Then for all $n \in Z^+$,

$$(\Xi)^n_{\beta\alpha} \leq \frac{\lambda(\alpha)}{\lambda(\beta)}$$

Proof

For $n = 1$, the result is true by definition.

Suppose inductively that it is true for all $n < m \in Z^+$.

Then,

$$(\Xi)^m_{\beta\alpha} = \sum_{\gamma \in S_\alpha} (\Xi)^{m-1}_{\beta\gamma} \frac{p_{\theta(\alpha,\gamma)\phi(\alpha,\gamma)}\mu_{\theta(\alpha,\gamma)}}{\lambda(\gamma)}$$

where the argument of μ has been omitted and

$$S_\alpha = \{\gamma \mid \text{one-step transition } \alpha \to \gamma \text{ is valid}\}$$

$$\therefore \quad (\Xi)^m_{\beta\alpha} \leq \sum_{\gamma \in S_\alpha} \frac{\lambda(\gamma)}{\lambda(\beta)} \frac{\mu_{\theta(\alpha,\gamma)}}{\lambda(\gamma)}$$

using the inductive hypothesis and that $p_{\theta(\alpha,\gamma)\phi(\alpha,\gamma)} \leq 1$ for all $\alpha,\gamma \in S$.

$$\therefore \qquad (\Xi)^m_{\beta\alpha} \leq \frac{1}{\lambda(\beta)} \sum_{\gamma \epsilon S_\alpha} \mu_{\theta(\alpha,\gamma)}$$

$$= \frac{\lambda(\alpha)}{\lambda(\beta)}$$

which proves the Lemma.

In the notation of the Lemma, for $\alpha \in S$, the iteration may be given by

$$P_0(\alpha,t) = \theta(\alpha)$$

$$P_{n+1}(\alpha,t) = e^{-\lambda(\alpha)t} P(\alpha,0)$$

$$+ \lambda(\alpha)e^{-\lambda(\alpha)t} \sum_{\beta \epsilon S} \Xi_{\alpha\beta} \int_0^t e^{\lambda(\alpha)u} P_n(\beta,u)du$$

.....(E7.6)

Theorem T7.1

The iterative scheme defined above for $P_n(\alpha,t)$, $n \in Z^+$, $\alpha \in S$, $0 \leq t \in R$, converges to the exact transient solution as $n \to \infty$.

Proof

For $t = 0$, the result is true by Proposition P7.1.

For $t > 0$, let

$$D_n(\alpha,t) = P_n(\alpha,t) - P_{n-1}(\alpha,t) \qquad (n \geq 1)$$

so that

$$D_n(\alpha,t) = \lambda(\alpha)e^{-\lambda(\alpha)t} \sum_{\beta \epsilon S} \Xi_{\alpha\beta} \int_0^t e^{\lambda(\alpha)u} D_{n-1}(\beta,u)du$$

Now let $S_n(\alpha,t) = \sup_{0<u\leq t} |D_n(\alpha,u)|$ so that

$$|D_n(\alpha,t)| \leq (1 - e^{-\lambda(\alpha)t}) \sum_{\beta \epsilon S} \Xi_{\alpha\beta} S_{n-1}(\beta,t)$$

But for all τ s.t. $0 < \tau \leq t$

$$| D_n(\alpha,\tau) | \leq (1 - e^{-\lambda(\alpha)t}) \sum_{\beta\epsilon S} \Xi_{\alpha\beta} S_{n-1}(\beta,t)$$

since $1 - e^{-\lambda(\alpha)t} \geq 1 - e^{-\lambda(\alpha)\tau}$ and $\sup_{0<u\leq\tau} |D_{n-1}(\alpha,u)| \leq S_{n-1}(\alpha,t)$

$$\therefore S_n(\alpha,t) \leq (1 - e^{-\lambda(\alpha)t}) \sum_{\beta\epsilon S} \Xi_{\alpha\beta} S_{n-1}(\beta,t)$$

Now let $x = \sup_{\alpha\epsilon S} \{1 - e^{-\lambda(\alpha)t}\}$

so that $0 < x < 1$ for $t > 0$, all service rates being finite.

Then, by Lemma L7.1

$$S_n(\alpha,t) \leq x^{n-1} \sum_{\beta\epsilon S} \left\{\frac{\lambda(\beta)}{\lambda(\alpha)}\right\} S_1(\beta,t)$$

$$= \Lambda x^{n-1}$$

where $\Lambda = \sum_{\beta\epsilon S} \left\{\frac{\lambda(\beta)}{\lambda(\alpha)}\right\} S_1(\beta,t)$ is finite since for all $\beta\epsilon S$

$\lambda(\beta) < \infty$ and $S_1(\beta,t) \leq P(\beta,0) + \theta(\beta)$ by equation (E7.5).

$\therefore$ For $m > n \epsilon Z^+$, $\alpha \epsilon S$, by the triangle inequality,

$$|P_m(\alpha,t) - P_n(\alpha,t)| \leq \sum_{j=n+1}^{m} | D_j(\alpha,t) |$$

$$\leq \sum_{j=n+1}^{m} S_j(\alpha,t)$$

$$\leq \Lambda \sum_{j=n+1}^{m} x^{j-1}$$

$$< \frac{\Lambda x^n}{1 - x}$$

Now, for all $\varepsilon > 0$, $\exists\, N \in Z^+$ s.t. for all $n > N$

$$x^n < \varepsilon(1-x)/\Lambda \quad \text{and so}$$

$|P_m(\alpha,t) - P_n(\alpha,t)| < \varepsilon$ for all $m>n>N \in Z^+$.

Thus, for all $t \in R^+$, $\alpha \in S$

$\{P_n(\alpha,t)\}$ is convergent as $n \to \infty$

by Cauchy's theorem, with limit $P_\infty(\alpha,t)$ say.

It is clear that on substitution of P_∞ for P_n and P_{n+1} in equations (E7.6) P_∞ is indeed the solution to the Kolmogorov equations (E7.2), satisfying the initial condition $P_\infty(\alpha,0) = P(\alpha,0)$ by proposition P7.1.

An alternative formulation of the iterative scheme, avoiding the need for explicit integration, is by means of power series expansions. This is discussed in the following section.

7.2.2 Expansion in power series

Proposition P7.2

For all $0 \leq n \in Z$, $\underline{k} \in S$, $P_n(\underline{k},t)$ has a power series expansion with infinite radius of convergence.

Proof

The result is trivial for $n = 0$.

Suppose true for $P_{n-1}(\underline{k},t)$, $n \in Z^+$.

Then the right hand side of equation (E7.3) has infinite radius of convergence since for any functions f_1 and f_2 of t with infinite radii of convergence,

(i) Their weighted sum and product also have power series with infinite radii of convergence.

(ii) The indefinite integral of f_1 with respect to t also has a power series with infinite radius of convergence.

Thus the proposition is proved by induction on n.

In the iteration defined in the previous section, let

$$P_n(\underline{k},t) = \sum_{m=0}^{\infty} a_{nm}(\underline{k})t^m \qquad (\underline{k} \in S)$$

A recurrence relation for the power series coefficients $a_{nm}(\underline{k})$ may be derived by substitution into equation (E7.3) as follows.

$$\sum_{m=0}^{\infty} a_{n+1,m}(\underline{k})t^m = e^{-\lambda(\underline{k})t}P(\underline{k},0) + \sum_{m=0}^{\infty} A_{nm}(\underline{k})B_m$$

where $$A_{nm}(\underline{k}) = \sum_{1\le i\ne j\le M} \varepsilon(k_i)p_{ji}\mu_j(k_j+1)a_{nm}(\underline{k}^{ij})$$

and $$B_m = e^{-\lambda(\underline{k})t}\int_0^t u^m e^{\lambda(\underline{k})u}du$$

$$= \frac{(-)^{m-1}m!}{\{\lambda(\underline{k})\}^{m+1}} \sum_{\ell=m+1}^{\infty} \frac{\{-\lambda(\underline{k})t\}^{\ell}}{\ell!}$$

(after some reduction).

Thus, by comparing coefficients of t^m, $m \ge 0$, for $\underline{k} \in S$

$$a_{n+1,m}(\underline{k}) = \frac{\{-\lambda(\underline{k})\}^m}{m!}\left\{P(\underline{k},0) - \sum_{j=0}^{m-1}\left(\frac{A_{nj}(\underline{k})}{\lambda(\underline{k})}\right)j!\{-\lambda(\underline{k})\}^{-j}\right\} \qquad (m \ge 1)$$

$$a_{n+1,0}(\underline{k}) = P(\underline{k},0)$$

and

$$a_{0,m}(\underline{k}) = \theta(\underline{k})\,\delta_{m0} \text{ is the initial condition.}$$

Therefore, using these fairly simple recurrence relations, the power series for the time dependent state space probability distribution may be computed as an alternative to the direct method of performing the integration in equation (E7.3) numerically.

7.3 Relevance to the PSA method

As remarked upon earlier, the transient analysis presented here was originally pursued as an enhancement to the PSA method for approximate computation of cycle time distribution in two-centre cyclic queueing networks, [HARR78a]. The approach taken was to derive a better approximation for the joint probability distribution of the pair of queue lengths faced by the test customer (c.f. section 8.3.3) as follows:

(i) Assume an equilibrium state space probability distribution at the time of arrival of the test customer at the first centre;

(ii) Set a time origin, t = 0, at this arrival time;

(iii) Evaluate the probability distribution of the queue length existing at the second centre on arrival of the test customer *conditional* on the queue length at the first centre at time zero. This distribution was derived approximately using time dependent state space probabilities evaluated to first order by the iterative method described in section 7.2.1.

The details may be found in Appendix 11 and yield the result, in the heavy traffic case,

$$P(q_2|q_1) \propto \theta(q_2) + \{P(q_2,0) - \theta(q_2)\} \left\{ \frac{1}{1+\lambda(q_2)\mu_1^{-1}} \right\}^{q_1}$$

in the following notation

$\underline{q} = (q_1,q_2)$ is the pair of queue lengths faced by the test customer in his cycle;

μ_1,μ_2 are the service rates of the servers (assumed constant);

$\theta(k)$ is the equilibrium state space probability for state $(N-k,k)$;

$\lambda(k)$ is the total service rate in state $(N-k,k)$.

Now, $\lambda(q_2) = \begin{cases} \mu_2 & (q_2 = N) \\ \mu_1 + \mu_2 & (1 \le q_2 \le N-1) \end{cases}$

and the factor $\left\{ \frac{1}{1 + \lambda(q_2)\mu_1^{-1}} \right\}^{q_1}$ represents the degree of the difference between the equilibrium queue length probability $\theta(q_2)$ (as assumed under PSA) and this improved approximation.

Thus the difference decreases as q_1 increases - i.e. as the queue length faced on arrival at the first centre increases. For large q_1, therefore, the interpretation of section 3.1 is justified; on arrival of the test customer at the second centre, the system will have had sufficient time to have come close to its steady state.

In fact this intuition is supported in general by the first order transient approximation. Suppose the test customer arrives at some centre, c say, at time zero, facing queue length n. To first order, the time constant

$$T = \frac{1}{\{\text{total service rate of network}\}}$$

$$\leq \frac{1}{\text{service rate of centre c}}$$

$$= \frac{1}{n} E \quad \text{(sojourn time at centre c)}$$

Thus the expected time of departure from centre c becomes very much greater (linearly) than T as the queue length n increases. Consequently, for large n the system may be assumed to be in equilibrium to a good (first order) approximation on arrival of the test customer at the next centre in his path.

7.4 Summary

In this section a convergent iterative method has been developed for the solution of the Kolmogorov forward equations for queueing networks of the Gordon-Newell type. As a result, quantitative assessment of the equilibrium assumption used in queueing network analysis may be made, in particular by consideration of time constants, and analysis of transient situations undertaken. Extension of the method to the general BCMP class of networks appears straightforward.

The iterative schemes given in section 7.2 are eminently suitable for implementation by computer, whether by the direct method of numerical integration or using power series. The most efficient method is probably the former since the power series involved will not converge rapidly, being based on the exponential series, and considerable effort has been

expended in the past on techniques for efficient numerical integration.

The method normally used for (exact) solution of a linear system of differential equations,

$$\frac{dy_i}{dt} = \sum_{j=1}^{n} M_{ij}\, y_j \qquad (1 \le i \le n)$$

or $\dot{y} = My$ in matrix form

involves diagonalisation of the matrix M. Specifically, if M has eigenvalues $\{\lambda_i \mid 1 \le i \le n\}$ with eigenvectors $\{v_i \mid 1 \le i \le n\}$

Then $\dot{v} = Dv$

where $D_{ij} = \lambda_i \delta_{ij} \qquad (1 \le i,j \le n)$

so that

$$v_i \propto e^{\lambda_i t}$$

and y_i may be obtained by inverse transformation. However, this method is totally impractical for numerical computation in view of the size of the matrix M; n is the order of the state space of the network. In fact the method is analogous to that of attempting to solve the balance equations explicitly to obtain the equilibrium state space probability distribution for a network with any non local state dependence (section 2.2).

Applications of the transient analysis presented in this chapter have been suggested (section 7.1) and are discussed in more detail in section 8.2.

§8. Applications of the Research and Areas for Future Investigation

8.1 Applications of the time delay analysis

8.1.1 Model types

Practical situations in which the ability to predict time delays is desirable were identified in chapter 2. The types of time delay may be classified into two categories:

(i) Those incurred by progressing from one centre to another along any one of a set of possible paths.

(ii) Those incurred by multiple passages of type (i).

Typically type (i) time delays arise in polling systems and type (ii) represent the response time in a computer system of a task requiring several cycles through the system's network of resources. Models for each of these types of situation are described briefly in the following two sections. Of course, such models by no means form an exhaustive set for the two categories above. For example, the time delays for messages sent in communication networks - the "end-to-end" delay, [WONG78a]- is another example of type (i).

The type of analytical model used in each case will usually be the PSA approximate model in view of its generality of application (see chapter 3), computational efficiency and expected accuracy (see Appendix 8). However, in cases represented by very simple tree-like networks satisfying the assumptions required by the exact analysis of chapter 5, the exact model may well be preferred in view of its superior accuracy and despite its inefficiency.

8.1.2 Polling Systems

In a polling type of system, the time delay of greatest interest is that incurred by sampling a set of status indicators and performing tasks associated with the status noted in each case. The set will normally form a loop and be polled continuously in a cyclic manner. Of course, any indicator's successor will not, in general, be unique so that the polling system's queueing network representation must allow branches.

This type of sampling situation is not quite that of the conventional queueing network. The tasks associated with any status indicator do not transit to another on their completion by the processor: rather the processor completes every such task before sampling the next status indicator. Nevertheless, the PSA model can easily be applied in that it requires as input only the probability distribution (or empirical relative frequencies) of the number of tasks associated with each status indicator and the distribution of their service times. The tasks may even have different service time distributions provided the probability distribution for the numbers of each type at each centre is available in some form.

The status indicator could be a "data ready" line in a multiplexor system or a sensor in a process control system and it is immediately apparent that application of the PSA method will allow prediction of the probability of a system fault through failure to complete a sampling cycle within some predefined time limit.

8.1.3 Computer systems

The prediction of time delays in computer systems is probably the most important application of the analysis presented in this thesis. The distribution of cycle time is itself of great use - for example in detecting imbalances in a computer system configuration, revealed through unexpected peaks at times greater than the mean. However, the crucial measures are those of response time (interactive systems) and turnaround time, sums of successive cycle times, as discussed in section 2.3.

In fact response time can be represented quite simply in a type (i) model. An interactive system may be represented by the configuration in fig. 8.1.

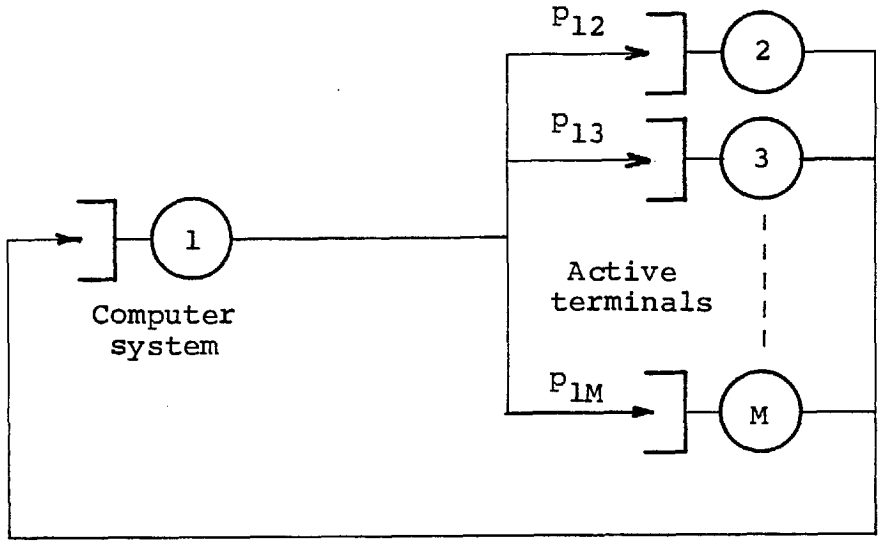


fig. 8.1 Network representing an interactive computer system

In this network, the cycle time consists of the sum of

(i) User think time, U say,

(ii) The response time of the system, R say.

A very simple model may be constructed for this

configuration in which the active terminals, or rather their users, are considered to have identical characteristics, in particular independent, identically distributed service times, U, and equal routing probabilities, p_{1i}, $2 \leq i \leq M$. Then the complete set of terminals in use (centres 2-M in fig. 8.1) may be represented as a single IS server, the number of tasks in the network being equal to the number of active terminals, M-1. The resulting network is therefore cyclic with just two centres. The tasks may have different processing time requirements, resulting in a multi-class model of the BCMP type, [BASK75], given suitable queueing discipline for the computer system server.

The probability distribution for the queue length at the computer system may be computed either using the BCMP result or empirically. The service time distribution for the computer system for each class of task may also be obtained empirically by means of controlled experiments in each one of which only one user is logged on to the actual system. By application of the PSA method to the one-centre path consisting of the computer system only, the probability distribution of response time may be predicted. An important practical advantage of this simple model is that measurements for the random variables U and R are usually available from real computer installations, so that validation problems are reduced.

This high level description of a computer system by a single server may be refined by use of a model of the system at a greater level of detail in which the individual computing resources are represented explicitly. Many such models have been constructed which find the service centre queue length probabilities, e.g. [KRZE77b,BUZE78b], and usually consist of a

server representing the processor(s) together with servers to represent the various I-O sub-systems at various levels of detail. A task created by a terminal server will require a certain number of cycles in the computer system, with some probability distribution, and the response time is the sum of these cycle times. This is the situation described in sections 4.10 and 5.8.4, and assuming independence of successive cycles, results in an asymptotically Normal distribution for the response time as the number of cycles increases, for any given task. In a model such as this, a task typically will have a fixed probability of departure from the network at one or more service centres, c.f. the probability of leaving the loop from centre γ in fig. 3.2, section 3.4. In this case the probability distribution of the number of cycles is geometric and as represented in [LAZ078], but it could equally well be obtained empirically. Either way, the asymptotic Normal distributions must be weighted according to the probabilities of their associated numbersof cycles, to give an analytic expression or (numerical) histogram respectively for the overall response time distribution.

8.2 The use of transient analysis

The principal application of the time dependent analysis of queueing networks presented in chapter 7 lies in the determination of the decay characteristics of the transient component of the state space probability distribution. In this way the length of time required before a queueing network can be considered to have attained stochastic equilibrium may be computed.

For Jackson type networks it was shown that, to first order in an iterative process, the decay was exponential with

time constant for each state's probability equal to the reciprocal of the total service rate in that state. This result will also clearly generalise to the BCMP case.

However, apart from this very general result, relevant in all modelling situations where the equilibrium assumption is made, there are several more applications of the transient analysis. The time dependent state space probabilities, computable numerically to any degree of precision specified by the modeller, may be used to describe the characteristics of the network (and so predict those of the modelled system) immediately following the setting of a time origin, representing some type of initialisation. This initialisation may take many forms in an actual computer system, typically:

(i) The literal initialisation or "starting up" of the system with some configuration of resources and tasks specified.

(ii) More generally any disturbance to a system, whether or not assumed in equilibrium, constitutes initialisation. This is because the system's characteristics at all future times depend on the nature and time of occurrence of the disturbance - clearly the system cannot be in equilibrium immediately following such an event. Disturbances may be many and varied. For example, in a dual processor system, the failure of one of the processors is quite clearly a disturbance, and the ability to predict the effect on system behaviour immediately after any such event is of great value.

A more subtle disturbance is the entry/departure of a task into/from the dispatchable set in a multiprogramming

computer system. This is, of course, represented by the arrival/departure of a customer at/from some server in an open queueing network model of the system, and the *initial* state space probability distribution is the equilibrium one, [MITR79]. However, at times immediately following the *known* time of this consequent disturbance, the state space probabilities are time dependent since a time origin has been set. In this way an analysis of the edge effects associated with such events becomes possible. This is, of course, the basis for the analysis in continuous time of time delay distributions in queueing networks discussed in chapters 2 and 7.

8.3 Future research areas

8.3.1 Outline

In the main text of this thesis, most of the remaining open questions have been identified so that a detailed discussion is not required here. Instead these research areas are summarised and some new ones identified, with elaboration where necessary in a few cases.

8.3.2 Acquisition of measured data

One of the immediate priorities, as mentioned quite frequently, is to obtain suitable data from at least one real computer system so that the validation by means of a three way comparison between analytic, simulation and empirical results may be achieved. The actual methods used to perform this validation will be precisely those described in Appendix 8;

there being more applications of each, of course, in view of the increased number of sets of data.

The actual collection of the data could be by an event driven software monitor or possibly by a hardware monitor if the system under study provided some means of identification at the hardware level of the job in use of the CPU. The latter possibility, if available, is the most attractive - to the analyst in that system performance would not be distorted by the considerable resource demands of the event driven monitor and to the installation management in that the running costs would be less for the same reason. In practice we shall be glad to accept either alternative if offered!

8.3.3 The PSA method

Perhaps the most important practical contribution of the research presented here is the PSA method of chapter 3 for the approximate prediction of time delay distributions and their moments. Consequently a considerable amount of research is planned in this area. In the immediate future, the most pressing need is the extension of the present implementation of the PSA model (Appendix 7) to cope with non exponential service time distributions, IS queueing discipline (a trivial task, see sections 3.2, 3.3.1) and state dependent service rates. In addition, it is a simple matter to incorporate into the model networks with service centre (class) queue length probability distributions based on the BCMP result, [BASK75], or operational measurement (the empirical case). In the former case, LCFS queueing discipline could either be excluded or the approximation given in section 3.2 could be used.

These enhancements would result in a PSA model able

to be applied in its full generality, but only to cycle times in tree-like queueing networks. Thus the next stage in the development will be to extend the model for application to time delays in general, in networks of general structure. This stage will be based on the methods discussed in chapter 3 and will involve a considerable design and programming effort - recall, for example, the problem of loops, section 3.4. Validation of this fully generalised model will be by precisely the methods described in this thesis (involving generalisation of the simulator, therefore), hopefully with the availability of measured data also, as discussed above.

The more fundamental research required in the area of the PSA method concerns the acceptability of its approximations. In terms of the method's predictions, empirical tests have indicated that such approximations are indeed acceptable. However, a direct empirical test may be made on the fundamental assumption of the method: independence of the queue length distributions for each centre in every valid path through the network. The test in question is the ACF test, [CHAT75], applied to cycle times in Appendix 8. In this case it would be applied to a sample from the sequence of successive queue lengths faced by a test customer in a simulated network or by a particular task in an actual computer system. An alternative test for independence of a data sample uses spectral analysis, [JENK68], by computing the sample's cumulative periodogram which should approximate to a straight line. This test could be applied both to the cycle times and queue lengths samples.

Useful and convenient though these statistical tests are, they may only be applied in *specific instances* and what is really required to assess the degree of approximation in the PSA method is a *formal error analysis*. Such an error analysis

would not be expected to provide exact results for the error in every network specification. Otherwise the exact solution would be known so rendering the research worthless since either:

(i) The form of exact solution could be sufficiently efficient in execution for practical purposes that the approximate method would become superseded,

or (ii) The computation of the error would be too inefficient for use as a practicable tool.

As an example, an error analysis in the second category has been accomplished in this thesis for tree-like networks, viz. the difference between the exact and PSA solutions.

Thus it is proposed to attempt to derive upper and lower *bounds* on the error as relatively simple expressions in terms of the network specifications. As a first step it would appear simplest to consider the case of tree-like networks for which an exact method of solution is known, bearing some resemblance to the PSA analysis. The approach taken could be based on that taken by the author in an analysis of another approximate method for cyclic networks, [HARR79a]. This method is an enhancement of the PSA method in that the joint probability distribution of the queue lengths faced by the test customer in any path is computed by an *exact* algorithm, not based on the assumption of independence of the servers. As in the PSA method, however, the customer's sojourn time distribution for each centre is assumed independent of the queue lengths faced at *other* centres in the path. This is valid for centres already departed from (by the Markov property) but not for those still to be entered. Hence the name of the method: "future path

independence" or FPI. It was shown in [HARR79a] that the FPI method gives an upper bound for the Laplace transform of the cycle time distribution in cyclic queueing networks. In view of this result and the closeness of the PSA and FPI methods it may be worthwhile, in an attempt to obtain an upper bound on the PSA method, to make an analytic comparison between them - at least to pursue an analogous development for the PSA method. However, this approach would only provide one bound, and that on the Laplace transform of the cycle time distribution.

Thus, the problems involved in a formal analysis providing efficiently computable bounds on the error of the PSA method appear considerable even for tree-like networks. They will presumably be even greater for more general networks for which no simple exact analytic solution is known.

A compromise between the empirical and theoretical approaches could be as follows:

(i) Network specifications for which the approximation of the PSA method is expected to be poor should be identified. In particular identification of the *worst* cases in any class of networks is most important. Heuristically such networks would be cyclic by the argument of section 3.1, a view which is supported by the validation process described in Appendix 8. However, analytic definition of such worst case networks is required.

(ii) Empirical tests could be made to estimate the error in these cases and in this way the maximum error for any class of network under analysis could be predicted.

8.3.4 Other research areas

In the previous section the most significant research areas have been discussed, but others have been identified in the main text and are listed here together with three new ones:

(i) The importance of response time has been emphasised in various places, and validation of the underlying independence assumption via the autocorrelation function or cumulative periodogram suggested. Another simple check would be to examine samples of measured response times and to test the hypothesis that they are drawn from a linear combination of Normal distributions, weighted according to the probability distribution for the number of cycles required, see section 8.1.3.

(ii) Inversion of the Laplace transforms of time delay distributions could be investigated. Two possible methods for this were identified in chapter 4: numerical and analytical. Both methods have their problems and the latter has been given for the PSA method with exponential server networks in Appendix 1. It has also been pointed out in section 4.8 that there is little practical value from such an exercise which is primarily of academic interest therefore.

(iii) In order to provide a truly practical tool for the computer performance analyst, the implementation of the results of the PSA and exact methods for time delay analysis must be made as efficient as possible. This will involve established techniques such as the efficient handling of sparse matrix operations, recursion and storage management. It may also require the use of specialised techniques such as decomposition methods, sections 4.9 and 5.8.3.

(iv) The exact analytical method may be extended to apply to networks in which each server may have any service time distribution possessing rational Laplace transform. The queueing discipline will still be restricted to FCFS. The extension is based on the method of stages, [BASK75, COX55] and because of the FCFS queueing discipline will, in general, introduce blocking (section 2.1 and [BASK75]) at the first stage of each service centre. However, the state space transition matrix for the generalised tree-like network is easily constructed for any state dependencies in the service rates. The non overtaking property of the generalised tree-like network is preserved by the FCFS queueing discipline of the stages and the blocking property which therefore permits parallel stages to be used - the most general case. Blocking presents no additional problem in this analysis in that the state transition matrix must be analysed explicitly in the exact analysis. There are two additional difficulties resulting from this generalisation. The first is an increase in computational inefficiency arising from expansion of the state space by the addition of the stages. The second is that no equilibrium state space probability distribution is known in closed form for FCFS queueing discipline and non exponential servers. Thus the cycle time distribution and its related results may only be derived analytically conditional on the initial state of the network. Of course in practice it may be that empirical equilibrium state space probabilities are available.

(v) As regards the transient analysis, it was observed in chapter 7 that the method derived for Jackson type networks should be easy to generalise to the BCMP class. This extension is proposed. In addition actual implementation of the analysis

as a software package has yet to be undertaken. The predictions of such an implementation should be validated with respect to data obtained for transient situations; by simulation experiments and, ultimately, by monitoring actual computer systems.

(vi) A controller can be designed to optimise system performance measures continuously by automatic tuning of the system's parameters, resulting for example in dynamic scheduling algorithms. The (operational) parameter adjustments may be based on the performance predictions of a model corresponding to optimal setting of the (model) parameter values, [KRIT78], the input to the model being based on current workload characteristics. However, a computer *installation* model has been proposed, [LEHM79a], by which workload characteristics themselves may be predicted. Given a validated model such as this, the control system described above could make (operational) parameter adjustments based on *anticipated* rather than current workload characteristics.

Note that this type of parameter adjustment constitutes a form of initialisation in the sense of section 8.2. Thus the immediate effects of such a disturbance to the system, for example some form of instability, may be predicted by application of the transient analysis as discussed in section 8.2.

§9. Conclusion

The fundamental philosophy of this thesis is concerned with the adequacy of the representation of models of computer systems. Thus it was first argued that to achieve a good representative model a phenomenological approach to modelling is required. This will result in the identification of the performance measures genuinely requiring prediction (as opposed to those easily produced by some established class of model) as well as the determination of the type and structure of the model ideally suited for such prediction. Of great interest to the modeller at present are the techniques of queueing network analysis, in view of the closeness of structure between (abstract) queueing networks and (real) computer systems, and such techniques are strongly supported here as a means for representative computer system modelling. The specific type of analysis undertaken would depend on the phenomenological study. The measures of most interest, to both users and installation management, are the time delays incurred by individual tasks (e.g. programs) in computer systems - not only their mean values but also their relative frequency histograms, or at least estimates for some higher moments.

The principal results of the research discussed here are, then, the theoretical solutions for the distribution of time delays in queueing networks. The solutions of greatest importance are the exact ones which provide, in addition to significant academic interest and achievement, standards against which to assess the adequacy of approximate methods, e.g. theoretical or simulation. Such exact methods were developed first for cyclic and then for tree-like queueing networks (chapters 4

and 5) with a view to their actual implementation so that computational efficiency was always considered an important requirement. This resulted, in particular, in the choice of the class of networks analysed, viz. those possessing the non-overtaking property (tree-like networks, chapter 5) for which expansion of the state space is not required. Nevertheless, the exact method does have severe practical limitations, both with its domain of application and its computational inefficiency. As a practical tool for the performance analyst the theoretical method assuming permanent stationarity (chapter 3) is of far more use since it is applicable in a very wide range of situations, is efficient in execution and appears to provide accurate approximations (Appendix 8).

Indeed, the emergence of the PSA method as such a potentially valuable tool is the most important *practical* achievement of the research reported. Though we are convinced of the justification for the approximations and the correctness of the derivation, it must nevertheless be accepted that the method has not been validated against measured data. Ultimately only such validation can give total confidence in the validity of the results.

Perhaps of less practical, but of significant academic interest is the transient analysis of chapter 7 which provides a very simple, convergent, iterative solution to the time dependent Kolmogorov equations for queueing networks, showing the nature of the decay of the transient component. This novel approach to the problem (to the author's best knowledge) is very much more efficient in execution than the conventional method involving eigenvalue analysis. In fact such transient solutions have several applications in practice, as discussed in chapter 8, for example to quantify time periods in which the

equilibrium assumption is valid in a queueing network model or to predict the immediate effects of disturbances to a system.

Finally, it is anticipated that this research into modelling methodology in general and analysis of time delays in particular will provide a sound foundation for a longer term research objective. This is to integrate an efficient software package, able to predict accurately the probability distribution of time delays in a variety of computer system configurations (e.g. cycle time, response time, turnaround time), into a larger scale model of computer installations, [LEHM79a]. The installation model would represent dynamic workload characteristics making use of feedback from the time delay prediction component which would itself be part of the sub-model of the computer system, the processor of the applied workload. An application of such a model was given in the context of scheduling in section 8.3.4.

Such a *dynamic* model would result in a better understanding of the interrelationships existing between the various components of an installation (applications of the user community, user workload, computer system), their short term variation and long term evolution. Thus more effective management and control of the installation would be made possible.

APPENDIX 1

The Laplace transform, $F(\underline{r},p)$, of cycle time distribution as given in equation (E3.10) in section 3.2.1 may be inverted by evaluation of the Bromwich contour integral defined in general by

$$f(t) = \frac{1}{2\pi i} \int_{\gamma-i\infty}^{\gamma+i\infty} e^{pt} F(p)\,dp$$

in which $F(p) = L\{f(t)\}$ and

$\gamma > \gamma_o = \mathrm{Re}(p_1)$ where p_1 is the singularity of $F(p)$ with the largest real part, [SPAI70].

Assuming all singularities are isolated poles (as in our case) the contour could be as shown in fig. A1 and

$$f(t) = \frac{1}{2\pi i} \cdot 2\pi i \sum \{\text{residues of } \{e^{pt}F(p)\} \text{ at poles}\}$$

where the residue of $G(p)$ at p_o is the coefficient of $\frac{1}{p - p_o}$ in the Laurent expansion of $G(p)$ about p_o.

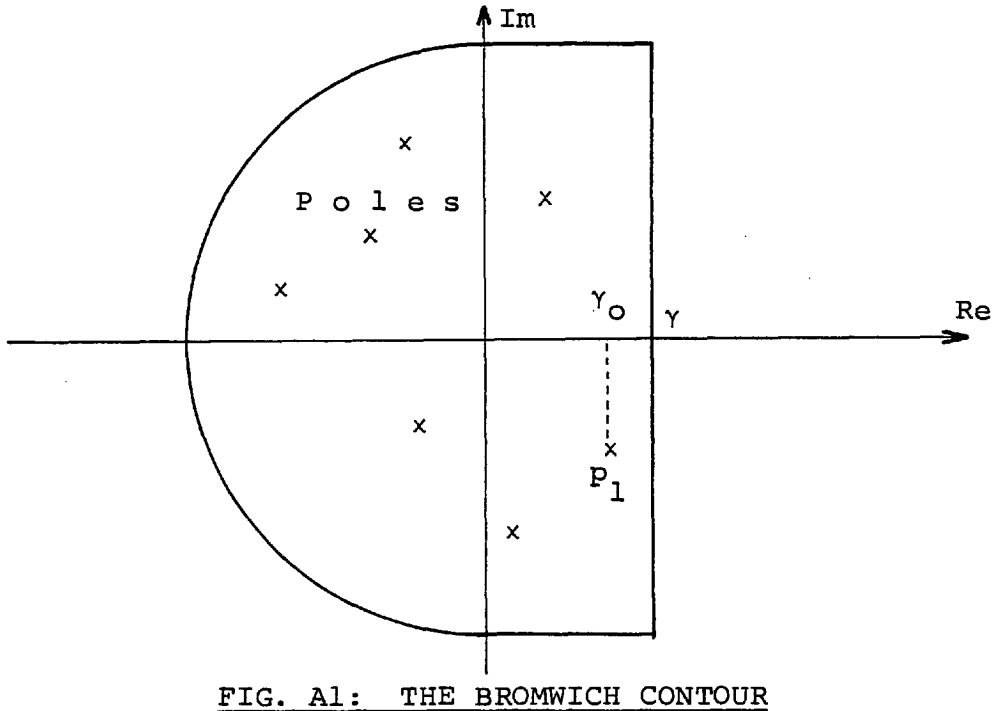


FIG. A1: THE BROMWICH CONTOUR

$$\therefore \quad \psi(\underline{k},t) = \sum_{j=1}^{L} \left\{ \text{residue of } \prod_{i=1}^{L} \left(\frac{\mu_i}{\mu_i+p} \right)^{k_i} e^{pt} \text{ at } p = -\mu_j \right\}$$

$$= \left(\prod_{i=1}^{L} \mu_i^{k_i} \right) \sum_{j=1}^{L} \frac{1}{(k_j-1)!} \lim_{p \to -\mu_j} \left\{ \frac{d^{k_j-1}}{dp^{k_j-1}} \left\{ \frac{e^{pt}}{\prod_{i \neq j} (p+\mu_i)^{k_i}} \right\} \right\}$$

Let $$\psi_j(\underline{k},t) = \frac{1}{(k_j-1)!} \lim_{p \to -\mu_j} \left\{ \frac{d^{k_j-1}}{dp^{k_j-1}} \left\{ \frac{e^{pt}}{\prod_{i \neq j} (p+\mu_i)^{k_i}} \right\} \right\}$$

.....(EA.1)

$$\text{then } \psi(\underline{k},t) = \left(\prod_{i=1}^{L} \mu_i^{k_i} \right) \sum_{j=1}^{L} \psi_j(\underline{k},t)$$

$$\text{Now, } \psi_j(\underline{k},t) = \frac{1}{(k_j-1)!} \sum_{\ell_j=0}^{k_j-1} \left\{ \frac{(k_j-1)!}{(k_j-1-\ell_j)!\,\ell_j!} t^{\ell_j} e^{-\mu_j t} \lim_{p \to -\mu_j} \left\{ \frac{d^{k_j-1-\ell_j}}{dp^{k_j-1-\ell_j}} \left\{ \frac{1}{\prod_{i \neq j} (p+\mu_i)^{k_i}} \right\} \right\} \right\}$$

using Leibnitz's rule and where the arbitrary summation index has been chosen to be ℓ_j.

$$\therefore \psi_j(\underline{k},t) = \frac{e^{-\mu_j t}}{(k_j-1)!} \sum_{\ell_j=0}^{k_j-1} \left\{ \frac{(k_j-1)!\, t^{\ell_j}}{(k_j-1-\ell_j)!\ell_j!} \sum_{\substack{\sum_{i\neq j}\ell_i = k_j-1-\ell_j \\ \ell_i \geq 0}} \frac{(k_j-1-\ell_j)!}{\left(\prod_{i\neq j}\ell_i!\right)} \right.$$

$$\left. \prod_{i\neq j} \frac{(-1)^{\ell_i}(k_i+\ell_i-1)!}{(k_i-1)!(\mu_i-\mu_j)^{k_i+\ell_i}} \right\}$$

using the generalisation of Leibnitz's rule given in Appendix 6 and indexing suitably.

$$\therefore \psi_j(\underline{k},t) = \frac{e^{-\mu_j t}}{\prod_{i=1}^{L}(k_i-1)!} \sum_{\substack{\sum_{i=1}^{L}\ell_i = k_j-1 \\ \ell_i \geq 0}} \left\{ \frac{(k_j-1)!\, t^{\ell_j} (-)^{k_j-1-\ell_j}}{\prod_{i=1}^{L}\ell_i!} \right.$$

$$\left. \prod_{1\leq i\neq j\leq L} \frac{(k_i+\ell_i-1)!}{(\mu_i-\mu_j)^{k_i+\ell_i}} \right\}$$

.....(EA.2)

A recurrence relation for $\psi_j(\underline{k},t)$ is easily derived from (EA.1) as follows:

$$\psi_j(\underline{k},t) = \frac{t}{k_j-1} \lim_{p\to-\mu_j} \left\{ \frac{1}{(k_j-2)!} \frac{d^{k_j-2}}{dp^{k_j-2}} \left\{ \frac{e^{pt}}{\prod_{i\neq j}(p+\mu_i)^{k_i}} \right\} \right\}$$

$$- \sum_{\ell\neq j} \frac{k_\ell}{k_j-1} \cdot \lim_{p\to-\mu_j} \left\{ \frac{1}{(k_j-2)!} \frac{d^{k_j-2}}{dp^{k_j-2}} \left\{ \frac{e^{pt}}{(p+\mu_\ell)\prod_{i\neq j}(p+\mu_i)^{k_i}} \right\} \right\}$$

$$= \frac{t}{k_j-1} \psi_j(\underline{k}^{j-},t) - \frac{1}{k_j-1} \sum_{\substack{\ell=1 \\ \ell\neq j}}^{L} k_\ell \psi_j(\underline{k}^{j-,\ell+},t) \qquad \text{.....(EA.3)}$$

where $\underline{k}^{j-} = (k_1,\ldots,k_j-1,\ldots,k_L)$

$\underline{k}^{j-,\ell+} = (k_1,\ldots,k_j-1,\ldots,k_\ell+1,\ldots,k_L)$

and for $k_j \geq 2$ and $k_i \geq 1$, $1 \leq i \neq j \leq L$

For $k_j = 1$

$$\psi_j(\underline{k},t) = \lim_{p\to-\mu_j} \left\{ \frac{e^{pt}}{\prod_{i\neq j}(p+\mu_i)^{k_i}} \right\}$$

$$= \frac{e^{-\mu_j t}}{\prod_{1\leq i\neq j\leq L}(\mu_i-\mu_j)^{k_i}} \qquad \text{.....(EA.4)}$$

Now let $Q_j(\underline{k},t) = \psi_j(\underline{k},t) \prod_{i=1}^{L} (k_i-1)!$. Then, by substitution in (EA.3) and (EA.4) or by direct derivation from (EA.1)

$$Q_j(\underline{k},t) = tQ_j(\underline{k}^{j-},t) - \sum_{\ell \neq j} Q_j(\underline{k}^{j-,\ell+},t) \quad \text{.....(EA.5)}$$

$(k_j \geq 2 \quad \text{and} \quad k_i \geq 1 \ , \ 1 \leq i \neq j \leq L)$

with boundary condition

$$Q_j(\underline{k},t) = \left\{ \prod_{1 \leq i \neq j \leq L} \frac{(k_i-1)!}{(\mu_i-\mu_j)^{k_i}} \right\} e^{-\mu_j t} \quad \text{.....(EA.6)}$$

$(k_j = 1 \quad \text{and} \quad k_i \geq 1, \quad 1 \leq i \neq j \leq L)$

A product form solution does exist for equation (EA.5), viz:

$$Q_j(\underline{k},t) = N_j \prod_{i=1}^{L} x_i^{k_i}$$

where $\sum_{i=1}^{L} x_i = t$ and N_j is independent of $\underline{k}$, but this expression cannot satisfy the boundary condition, (EA.6).

By inspection of the case for L=2 and the coefficients in the expansion

$$(t - \sum_{\substack{i=1 \\ i \neq j}}^{L} x_i)^{k_j}$$

which would arise in a product form solution if this existed, it can be seen that the solution of equations (EA.5) and (EA.6) should be:

$$Q_j(\underline{k},t) = \sum_{\substack{L \\ \sum_{i=1} r_i = k_j - 1}} \frac{(k_j-1)!}{\prod_{i=1}^{L} r_i!} t^{r_j} (-)^{k_j-1-r_j} Q_j(\underline{r}',t)$$

$$(1 \le i \le L)$$

.....(EA.7)

where $r'_i = \begin{cases} k_i + r_i & 1 \le i \ne j \le L \\ 1 & i = j \end{cases}$

That this is the correct solution may be shown as follows - Using (EA.7) with $k_j \ge 2$,

$$t Q_j(\underline{k}^{j-},t) - \sum_{\ell \ne j} Q_j(\underline{k}^{j-,\ell+},t)$$

$$= \sum_{\Sigma r_i = k_j - 2} \frac{(k_j-2)!}{\prod_{i=1}^{L} r_i!} t^{r_j+1} (-)^{k_j-2-r_j} Q_j(\underline{r}',t)$$

$$- \sum_{1 \le \ell \ne j \le L} \left\{ \sum_{\Sigma r_i = k_j - 2} \frac{(k_j-2)!}{\prod_{i=1}^{L} r_i!} t^{r_j} (-)^{k_j-2-r_j} Q_j(\underline{r}'^{\ell+},t) \right\}$$

$$= \sum_{\substack{\Sigma r_i = k_j - 1 \\ r_j > 0}} \frac{r_j (k_j-2)!}{\prod_{i=1}^{L} r_i!} \; t^{r_j} \; (-)^{k_j - 1 - r_j} \; Q_j(\underline{r}', t)$$

$$- \sum_{1 \le \ell \ne j \le L} \left\{ \sum_{\substack{\Sigma r_i = k_j - 1 \\ r_\ell > 0}} \frac{r_\ell (k_j-2)!}{\prod_{i=1}^{L} r_i!} \; t^{r_j} (-)^{k_j - 2 - r_j} \; Q_j(\underline{r}', t) \right\}$$

The restrictions that $r_j > 0$ and $r_\ell > 0$ may be removed because of the multiplicative factors r_j and r_ℓ so that the R.H.S. becomes:

$$\sum_{i=1}^{L} \sum_{r_i = k_j - 1} \frac{(k_j-2)! t^{r_j} (-)^{k_j - 1 - r_j} \; Q_j(\underline{r}', t)}{\prod_{i=1}^{L} r_i!} \left\{ r_j + \sum_{\ell \ne j} r_\ell \right\}$$

$$= Q_j(\underline{k}, t) \quad \text{since} \quad \sum_{\ell=1}^{L} r_\ell = k_j - 1 \text{ in the summation.}$$

Indeed by substituting the boundary condition, (EA.6) for $Q_j(\underline{r}', t)$ in (EA.7) and using the definition of Q_j in terms of ψ_j, the result (EA.2) is arrived at.

APPENDIX 2

In this appendix it is shown that

$$R(j) = \begin{cases} x\,R(j-1) + (1-x)\,Q(j-1) & (j \geq 2) \\ (1-x)\,Q(0) & (j=1) \\ 0 & (j=0) \end{cases}$$

where R is the cumulative discrete probability distribution of r which is the convolution of probability distributions p and q, where q has cumulative distribution Q and p is geometric, defined by

$$p(i) = \begin{cases} (1-x)\,x^{i-1} & (i \geq 1) \\ 0 & (i=0) \end{cases}$$

Let X_p and X_q be random variables with probability distributions p and q respectively. Then

$$R(j) = \text{Prob}\,(X_p + X_q \leq j)$$

$$= \sum_{i=0}^{j} \text{Prob}\,(X_p = j-i)\ \text{Prob}\,(X_q \leq i)$$

$$= \sum_{i=0}^{j-1} p(j-i)\,Q(i)$$

and

$$r(j) = \sum_{i=0}^{j-1} p(j-i)\,q(i)$$

similarly, or by differencing w.r.t. j.

$$\therefore\ R(j) = \begin{cases} \sum_{i=0}^{j-2} x\,p(j-1-i)\,Q(i) + p(1)\,Q(j-1) & (j \geq 2) \\ p(1)\,Q(0) & (j=1) \\ 0 & (j=0) \end{cases}$$

with a similar result for r(j) which proves the result.

APPENDIX 3

It is shown here that the convolution operation, applied to two distributions of positive random variables, is commutative. Consider the convolution, H(t), of distributions F(t) and G(t), $t \in R^+$.

$$H(t) = [F * G](t)$$

$$= \int_0^t F(t-u)\, dG(u)$$

$$= \int_0^t d\{F(t-u)G(u)\} - \int_0^t G(u)\, d_u F(t-u)$$

$$= 0 + \int_0^t G(t-v)\, d_v F(v)$$

since $F(0) = G(0) = 0$ and by the change of variable

$$v = t-u.$$

$\therefore$ $H = F * G = G * F$

as required.

APPENDIX 4

The minimum of n independent negative exponentially distributed random variables is itself negative exponential with inverse mean equal to the sum of the inverse means of its constituent components.

Proof

Let $T = \min_{1 \le i \le n} T_i$

where T_i is an exponentially distributed random variable with mean μ_i^{-1}.

$$\Pr(T > t) = \Pr(T_i > t \; ; \; 1 \le i \le n)$$

$$= \prod_{i=1}^{n} \Pr(T_i > t) \quad \text{by independence}$$

$$= \prod_{i=1}^{n} e^{-\mu_i t}$$

$$= \exp\left\{-t \sum_{i=1}^{n} \mu_i\right\}$$

APPENDIX 5

Computation of first entry probabilities to target states

In this appendix an expression is derived for the probability, P_{st}, of a network entering state $t \in S_T \subset S$ (the state space) at some future time, conditional on the network starting in state s and so that no other state in S_T is entered before t. S_T is the set of "target states" with the property that having transited to a state not in S_T from a state in S_T, the network can never return to any state in S_T.

The result is required in the decomposition analysis presented in section 4.8 for cyclic networks.

Let X_{st} be the probability of passage from state s to t in any number of steps, including zero if s = t,

i.e. $\underline{\underline{X}} = \sum_{n=0}^{\infty} \underline{\underline{T}}^n$ where T is the state transition matrix of the network, by Corollary CCL4.1,

so $\underline{\underline{X}} = (I-T)^{-1}$.

Now

$$X_{st} = \sum_{u \in S} X_{su}\, T_{ut} \quad \text{for } s \neq t \qquad \text{.....(EA.8)}$$

$$= \sum_{u \in S_T} X_{su}\, T_{ut} + \sum_{u \notin S_T} X_{su}\, T_{ut}$$

But S_T was defined to be such that for all $t \in S_T$, if $u \notin S_T$ s.t. $T_{tu} \neq 0$,

then for all $v \in S$ s.t. $X_{uv} \neq 0$, $v \notin S_T$.

Thus

$$\sum_{v \notin S_T} X_{sv} T_{vt} = P_{st}$$

by considering equation (EA.8) with t = v, so that any route implicit in X_{sv} entering S_T gives zero contribution, and so

$$P_{st} = X_{st} - \sum_{u \in S_T} X_{su} T_{ut} .$$

APPENDIX 6

In this appendix, the rule of Leibnitz for repeated differentiation of a product of two functions is extended to products of any number of terms.

Let $F(x) = \prod_{i=1}^{n} u_i(x)$ $\qquad (n \in Z^+, n \geq 1)$

Then $F^{(p)}(x) = p! \sum_{\substack{\sum_{k=1}^{n} j_k = p \\ j_k \geq 0}} \prod_{k=1}^{n} \frac{u_k^{(j_k)}(x)}{j_k!}$

where the index (p) denotes differentiation w.r.t. x p times.

Proof

The proof is by induction on p.

(i) For $p = 1$, the result yields

$$F^{(1)}(x) = \sum_{D} \prod_{k=1}^{n} u_k^{(j_k)}(x)$$

where $D = \{(j_1, j_2, \ldots, j_n) \mid j_k = 1,\ j_i = 0,\ 1 \leq i \neq k \leq n;\ 1 \leq k \leq n\}$

which is known to be true.

(ii) Suppose the result is true for $F^{(p)}(x)$.

Then

$$F^{(p+1)}(x) = \frac{d}{dx} F^{(p)}(x)$$

$$= p! \sum_{\substack{\Sigma j_k = p \\ j_k \geq 0}} \sum_{i=1}^{n} \prod_{k=1}^{n} \frac{u_k^{(j_k + \delta_{ik})}(x)}{j_k!}$$

$$= p! \sum_{i=1}^{n} \sum_{\substack{\Sigma j'_k = p+1 \\ j'_k \geq 0 \\ j'_i > 0}} j'_i \prod_{k=1}^{n} \frac{u_k^{(j'_k)}(x)}{j'_k!}$$

where $j'_k = j_k + \delta_{ik}$

$$= p! \sum_{i=1}^{n} \sum_{\substack{\Sigma j_k = p+1 \\ j_k \geq 0}} j_i \prod_{k=1}^{n} \frac{u_k^{(j_k)}(x)}{j_k!}$$

dropping the primes, since the terms for $j_i = 0$ give zero contribution, giving the result for $F^{(p+1)}(x)$ since

$$\sum_{i=1}^{n} j_i = p+1.$$

APPENDIX 7

The following APL functions produce numerical predictions of quantities related to the cycle time distribution in tree-like queueing networks. They implement the analytic results presented in chapters 3 (under PSA) and 5 (exact). The package may be used interactively by typing "CYCLE" when the workspace has been loaded.

```
[  0] CYCLE
[  1] 'CYCLE TIME DISTRIBUTION PACKAGE FOR CYCLIC QUEUEING NETWORKS VERSION 3',2⍴⎕CR⍬AVC157⍬
[  2] 'P.G.HARRISON, NOVEMBER 1978';⎕CR;⎕CR;'INPUT NO. OF CENTRES'
[  3] SPEC⍴LAP←NET←,MEM←(2 0)⍴CEN←⍴⍴RPM←(M,M)⍴CSR←(M←⍞)⍴0
[  4] →(M=CEN←1)/L1
[  5] →0⍴⍞←'*****ABORT...TOTAL NO. OF CENTRES COMES TO ',⍕CEN←1
[  6] L1:'INPUT NO. OF CUSTOMERS'
[  7] I←M!M←N←⍞
[  8] 'TOTAL NO. OF STATES = ',⍕I×E←×/1
[  9] L2:→(M≠⍴E←E,E←×(⍴E)↑,RPMC;1+⍴E])/L2
[ 10] X←E←CSR
[ 11] L8:'DO YOU WISH TO USE DECOMPOSITION OR DIRECT METHODS?'
[ 12] →('YN'=1↑⍞)/L12,L4
[ 13] →L3×⍴⍞←'TRY AGAIN'
[ 14] L12:'INPUT SIZE OF KNOWN NETWORK'
[ 15] MK←⍞
[ 16] 'INPUT VECTOR OF LAPLACE TRANSFORMS FOR EACH STATE WITH ',(3 0 ⍕N),'CUSTOMERS'
[ 17] RST←⍞
[ 18] ⍞←SSN←0,0,(×\(N+⍳N+M)+⍳⍴3+I+3+M)
[ 19] TRM
[ 20] →END
[ 21] L4:'INPUT ROOT HEAD CENTRES THE SUBTREES FOR WHICH ARE TO HAVE THEIR TRANSITION MATRICES DISPLAYED'
[ 22] SBD←⍞
[ 23] 'INPUT ONE OF THE FOLLOWING :-'
[ 24] '''L'' FOR LAPLACE TRANSFORM OF CYCLE TIME DISTRIBUTION'
[ 25] '''D'' FOR DISCRETE APPROXIMATION OF CYCLE TIME DISTRIBUTION'
[ 26] '''M'' FOR MEAN AND STANDARD ERROR OF CYCLE TIME DISTRIBUTION '
[ 27] '''N'' FOR MEAN AND STANDARD ERROR OF CYCLE TIME DISTRIBUTION EVALUATED NUMERICALLY'
[ 28] →(¯1<A←'LDNM'⍳1↑⍞)/L4
[ 29] ⍎A↓'4 3 ⍴''LTRDISNM THM'
[ 30] END:ERROR
```

```
[  0] DIS;N
[  1] 'INPUT TIME STEP SIZE FOR CYCLE TIME PROBABILITY DISTRIBUTION'
[  2] D←⍞
[  3] 'INPUT NO. OF TIME STEPS REQUIRED'
[  4] N←⍞
[  5] L1:'DO YOU REQUIRE EXACT OR APPROX. RESULTS ASSUMING PERMANENT STATIONARITY? INPUT ''E'' OR ''A'''
[  6] →('EA'=1↑⍞)/L3,L2
[  7] →L1×⍴⍞←'TRY AGAIN'
[  8] L2:→L4×⍴⍴PD←PSD N
[  9] L3:TRM 1
[ 10] PD←SSP←.×(1 0 0 ,N) DST SSI/⍳T
[ 11] L4:⎕CR'DISCRETE CYCLE TIME DISTRIBUTION:'
[ 12] ⎕CR,'      TIME          PROB.',⎕CR,30⍴'-'
[ 13] (⍳9 3 ⍴(N,1)⍴D×⍳N), 15 3 ⍴(N,1)⍴PD
[ 14] ⎕CR
[ 15] PD PLOT 120 15 ,D,0
```

```
[  0] Z←A DST I;L;J;TRE;NSB;NRT;OM;TRG;SJ;X;HD;Q;P;PP;T1;A;TST;TL
[  1] ⍝A[1]:TREE PTR,A[2]:OLD M,A[3]:OLD HD,A[4]:MAX. TIME,I:INITIAL STATES
[  2] →(1=L←⍴A[4])↓Z←0)/0
[  3] →(1=TRE←⍴A[1])/L7
[  4] X←(A[2],'0')⍴J←0
[  5] L6:→((⍴I)≠1↓⍴X←X,A[2] NTS I[J←J+1])/L6
[  6] L7:T←N!N←OM←M←MEM[2;MEM[1;]⍳HD]+1-HD←NET[TRE]
[  7] NRT←NET[TRE+1]
[  8] →(TRE=⍴⍴TST←M↑((NRT-1)⍴J←0),1)/L0
[  9] TRM HD
[ 10] I←⍳J←0
[ 11] L8:→((⍴X)[2]≠⍴I←I,STN M↑(HD-A[3])↓X[;J←J+1])/L8
[ 12] L0:Q←0,*-(1↓TSV)×D
[ 13] →((0=NSB←NET[TRE+2]),N>1)/L1,L2
[ 14] ⍝ONLY 1 CUST
[ 15] →L3×⍴⍴TRG←,I←NRT-1
[ 16] L2:TRG←¯1↓(STN TST)↓0,⍳STN 2×TST
[ 17] L3:SJ←TRANC[X←TRANI[TRG]∘.+⍳NSB]
[ 18] ⍝⍳NSB BY LABELING DEFN.
[ 19] A←CSR[TL]×((RPM[TL;]≠0)/RPM[TL←HD+NRT-1;])∘.+TSV[TRG]
[ 20] PP←+/[1](3 1 2⍴Q(L,⍴A)⍴A)×GTJ L
[ 21] T←⍴Q
[ 22] M←OM
[ 23] TRM HD
[ 24] Z←((⍴I),0)⍴Q←Q(⍴⍴P)⍴(-(⍴P←((T-TRG[1]-1),⍴TRG)⍴0)[1])↑Q
[ 25] T1←I-TRG[1]-1
[ 26] X←Q(⍴⍴P)⍴(-(⍴P)[1])↑TSV
[ 27] L5:P←(P×Q)+(1-Q)×('TRAN' MSE P)+X
[ 28] P←(0,⍳¯1+(⍴P)[1])⍴P
[ 29] P[⍳⍴TRG;1]←PP[;1+(⍴Z)[2]]
[ 30] P←(-0,⍳¯1+(⍴P)[1])⍴P
[ 31] →(L=¯1↑⍴Z←Z,+/P[T1;])/0
[ 32] →L5
[ 33] L1:Z←(T,0)⍴⍴P←1,(T-1)⍴0
[ 34] L4:→(L≠¯1↑⍴Z←Z,P←(P×Q)+(1-Q)×('TRAN' SML P)+1,1↓TSV)/L4
[ 35] Z←Z[I;]
```

```
[ 0] Z←I ESB T;J
[ 1] Z←((I-1)⍴0),1
[ 2] L:J←TRANI[I]+⍳TRANI[I+1]-TRANI[I←I+1]
[ 3] →(T≠⍴Z←Z,(TRAN[J]+T[I])+.×Z[TRANC[J]])/L
```

```
[ 0] Z←GTJ L;N;C;B;A;QQ;TRAN;TRANI;TRANC;TSV
[ 1] →(L=⍴⍴Z←(NSB,(⍴TRG),0)↑C←(NSB,(⍴TRG),L)⍴0)/0
[ 2] A←(NSB,⍴A)⍴A←⍪1,[1]×\[1]((L-N←1),⍴TRG)⍴QQ←Q[TRG]
[ 3] B←1-Q(L,⍴TRG)⍴QQ
[ 4] L1:→(NSB≥N←N+⍴⍴C[N;;]←B×(NET[TRE+2+N],OM,HD,L) DST,SJ[;N])/L1
[ 5] N← 1 1 0 ×⍴C
[ 6] L2:→(L≠¯1↑⍴Z←Z,+/(N↑C)×⍉(N←N+ 0 0 1)↑A)/L2
[ 7] ZGTJ←Z
```

```
[ 0] Z←A HSB T;J;I;R
[ 1] Z←,(R←(⍴T)↑A)[I←1]
[ 2] L:J←TRANI[I]+⍳TRANI[I+1]-TRANI[I←I+1]
[ 3] →(T≠⍴Z←Z,R[I]+(TRAN[J]+T[I])+.×Z[TRANC[J]])/L
```

```
[ 0] Z←PTH LPA P;NSB;EOR;I
[ 1] ⍝P[1]:SUB TREE POINTER,P[2]:LENGTH OF PATH VECTOR SO FAR
[ 2] EOR←PTH,NET[P[1]]+0,⍳¯1+NET[P[1]+1]
[ 3] →((I←⍴Z←⍳0)≠NSB←NET[P[1]+2])/L1
[ 4] →0×⍴PIX←PIX,P[2]+⍴Z←EOR
[ 5] L1:→((I=NSB),⍴⍴Z←Z,EOR LPA NET[P[1]+2+I←I+1],P[2]+⍴Z)/0,L1
```

```
[  0] Z←A LT I;J;TRE;NSB;NRT;OM;TRG;RATES;SJ;T;X;T1;T2;T3;TR;HD;PAR;K;TL;TST
[  1] ⍝AC1]:TREE PTR,AC2]:OLD M,AC3]:OLD HD,AC4+] LAP TRANS PARAMS,I:INITIAL STATES
[  2] →(1=TRE←AC1])/L7
[  3] X←(AC2],0)⍴J←0
[  4] L6:→((⍴I)≠1↓⍴X←X,AC2] NTS IEJ←J+1])/L6
[  5] L7:T←N!N←OM←M←MEMC2;MEMC1;]⍳HD]+1-HD←NETETRE]
[  6] NRT←NETETRE+K←1]
[  7] →(TRE=⍴⍴TST←M↑((NRT-1)⍴T←Z←J←0),1)/L0
[  8] TRM HD
[  9] I←⍳0
[ 10] L8:→((⍴X)E2]≠⍴I←I,STN M↑(HD-AC3])↓XE;J←J+1])/L8
[ 11] L0:→((0=NSB←NETETRE+2])⍱N>1)/L1,L2
[ 12] ⍝ONLY 1 CUST
[ 13] →L3×⍴⍴TRG←,I-NRT-1
[ 14] L2:TRG←¯1↓(STN TST)↓0,⍳STN 2×TST
[ 15] L3:SJ←TRANCETRANIETRG]∘.+⍳NSB]
[ 16] ⍝⍳NSB BY LABELING DEFN.
[ 17] RATES←(RPMETL;]≠0)/RPMETL←HD+NRT-1;]
[ 18] T3←CSRETL]←TSVETRG]∘.+3↓A
[ 19] T2←((PAR←¯3↓⍴A),(⍴I),⍴TRG)⍴K←1+T←T1←J←0
[ 20] L5:→(T≠(⍴TRG)×⍴⍴T2EK;;T]←(TRGET←T+1] ESB TSV+AC3+K])EI])/L5
[ 21] →(PAR≥K←K+1+T←0)/L5
[ 22] L4:→(NSB≠J×⍴⍴T1←T1+RATESEJ]×(NETETRE+2+J],OM,HD,3↓A) LT SJE;J←J+1])/L4
[ 23] →0×⍴⍴Z← 2 1 2 ⍟T2+.×T1×T3
[ 24] L1:Z←((⍴I),0)⍴0
[ 25] L10:Z←Z,(1 ESB TSV+AC2+K←K+1])EI]
[ 26] →((⍴A)≠2+K)/L10
[ 27] ⍝NO SUBS


[  0] LTR
[  1] 'INPUT VALUES FOR LAPLACE TRANSFORM PARAMETERS, S'
[  2] DEN←,0
[  3] LL:'DO YOU REQUIRE EXACT OR APPROX. RESULTS ASSUMING PERMANENT STATIONARITY? INPUT ''E'' OR ''A'''
[  4] →('EA'=1↑0)/L1,L2
[  5] →LL×⍴⍴0←'TRY AGAIN'
[  6] L1:TRM 1
[  7] →L3×⍴⍴LAP←SSP←.×(1 0 0 ,DEN) LT SSI/⍳I
[  8] L2:LAP←PSA
[  9] L3:'LAPLACE TRANSFORM OF CYCLE TIME DISTRIBUTION'
[ 10] CR,'       S              L.T.ES]',CR,30⍴'-'
[ 11] (9 3 ⍕((⍴DEN),1)⍴DEN), 20 3 ⍕((⍴LAP),1)⍴LAP
[ 12] CR
[ 13] LAP PLOT 120 15 ,(DENC2]-DENC1]),DENC1]



    [ 0] Z←MEAN N
    [ 1] G←(1,⍴X)⍴1
    [ 2] L1:→(N≠¯1+1↑⍴G←G,E1]+\X×,(¯1,⍴X)↑G)/L1
    [ 3] G←,GE;⍴X]
    [ 4] Z←(N×GEN+1])÷XE1]×GEN]


    [  0] Z←A MOM I;J;TRE;NSB;NRT;OM;TRG;RATES;SJ;T;X;T1;T2;T3;TR;HD;PAR;K;TL;TST
    [  1] ⍝AC1]:TREE PTR,AC2]:OLD M,AC3]:OLD HD,I:INITIAL STATES
    [  2] →(1=TRE←AC1])/L7
    [  3] X←(AC2],0)⍴J←0
    [  4] L6:→((⍴I)≠1↓⍴X←X,AC2] NTS IEJ←J+1])/L6
    [  5] L7:T←N!N←OM←M←MEMC2;MEMC1;]⍳HD]+1-HD←NETETRE]
    [  6] NRT←NETETRE+K←1]
    [  7] →(TRE=⍴⍴TST←M↑((NRT-1)⍴T←Z←J←0),1)/L0
    [  8] TRM HD
    [  9] I←⍳J←0
    [ 10] L8:→((⍴X)E2]≠⍴I←I,STN M↑(HD-AC3])↓XE;J←J+1])/L8
    [ 11] L0:→((0=NSB←NETETRE+2])⍱N>1)/L1,L2
    [ 12] ⍝ONLY 1 CUST
    [ 13] →L3×⍴⍴TRG←,I-NRT-1
    [ 14] L2:TRG←¯1↓(STN TST)↓0,⍳STN 2×TST
    [ 15] L3:SJ←TRANCETRANIETRG]∘.+⍳NSB]
    [ 16] ⍝⍳NSB BY LABELING DEFN.
    [ 17] RATES←(RPMETL;]≠0)/RPMETL←HD+NRT-1;]
    [ 18] T3←⍟C1](2×X←TSVETRG])⍱E1](X←X÷TSVETRG]),E0.5](⍳0),X←CSRETL]←TSVETRG]
    [ 19] TR←(T,0)⍴0
    [ 20] L9:→(⍴TRG)≠1↓⍴TR←TR,TRGET←T+1] ESB TSV)/L9
    [ 21] T2←(3,(⍴I),⍴TRG)⍴K←1+T←T1←0
    [ 22] T2E1;;]←TRCI;]
    [ 23] X←('TRAN' MSM TR)÷K×K←⍳(⍴TR)⍴1,1↓TSV,⍳T←J←0
    [ 24] L52:→(T≠(⍴TRG)×⍴⍴T2E2;;T]←(TRE;T]÷XE;T←T+1] HSB TSV)EI])/L52
    [ 25] TR←(('TRAN',⍳T←0) MSM TR)÷K×K
    [ 26] X←X÷K
    [ 27] L53:→(T≠(⍴TRG)×⍴⍴T2E3;;T]←2×((TRE;T] HSB TSV)+XE;T←T+1] HSB TSV)EI])/L53
    [ 28] L4:→(NSB≠J×⍴⍴T1←T1+RATESEJ]×(NETETRE+2+J],OM,HD) MOM SJE;J←J+1])/L4
    [ 29] X←(T1E1;]×T3E1;]),E1]((T1E1;]×T3E2;])+T1E2;]×T3E1;]),E0.5] 1 2 1 +.×T1×⍟E1] T3
    [ 30] ⍝1 2 1 ARE BINOMIAL COEFFS IN LIEBNITZ EXPANSION
    [ 31] Z←T2+.×⍟X
    [ 32] →0×⍴⍴Z←ZE1;;1],E1](+/E1] 1 2 1 ⍟⍟ ¯1 0 ¯1 ↓Z),E0.5] 1 2 1 +.× 1 2 1 ⍟⍟Z
    [ 33] L1:Z←(X←('TRAN' SML 1 ESB TSV)÷K←1,1↓TSV×2) HSB TSV
    [ 34] Z←Z,E0.5] 2×((('TRAN' SML Z)÷K) HSB TSV)+(X←1,1↓TSV) HSB TSV
    [ 35] Z←1,E1] ZE;I]
    [ 36] ⍝NO SUBS
```

```
[ 0] Z←A MSE M;I;X;J;K;D
[ 1] Z←(0 1 ×⍴M)⍴I←-J←-(⍴TSV)-(⍴M)[1]
[ 2] L:X←⍳A,'I[I]←⍳',A,'I[I+1]-',A,'I[I+I+1]'
[ 3] ⍎'→((⍴TSV)≠I×⍴⍴Z←Z,[1]⍳',A,'[D←X]+.×M[(D←+/0.5>K)↓K←J←',A,'C[X];])/L'
```

```
[ 0] Z←A MSM M;I;X
[ 1] Z←(0 1 ×⍴M)⍴I←0
[ 2] L:⍎'→((⍴TSV)≠1↑⍴Z←Z,[1]⍳',A,'[X]←.×M[',A,'C[X←',A,'I[I]←⍳',A,'I[I+1]-',A,'I[I+I+1]];])/L'
```

```
[  0] NM;EPS;DR1;A;DR2;B;C;SS;T1;T2
[  1] 'INPUT DEGREE OF PRECISION'
[  2] EPS←⎕
[  3] T1←T←T2←M
[  4] DEN← 1 2⍴+A← 0 0
[  5] [1]'DO YOU REQUIRE EXACT OR APPROX. RESULTS ASSUMING PERMANENT STATIONARITY? INPUT ''E'' OR ''A'''
[  6] →('EA'=1↑⍞)/L3,L2
[  7] →L1×⍴⍴⎕←'TRY AGAIN'
[  8] L2:⎕←DR1←(¯1+B←(T3←PSA)[1])÷C←1↑DEN←DEN←10
[  9] ⎕←DR2←(-DR1-(T3[2]-B)÷C)÷C
[ 10] →((0≠⌈/A←(DR1,DR2)×EPS<(|DR2-A[2])⌈|DR1-A[1]),1)/L2,L4
[ 11] L3:TRM 0≠T←T1←M←T2
[ 12] SS←SSI/⍳T
[ 13] ⎕←DRI←(¯1+B←(T3←SSP←.×(1 0 0 ,DEN) LT SS)[1])÷C←1↑DEN←DEN←10
[ 14] ⎕←DR2←(-DR1-(T3[2]-B)÷C)÷C
[ 15] →(0≠⌈/A←(DR1,DR2)×EPS<(|DR2-A[2])⌈|DR1-A[1])/L3
[ 16] L4:'MEAN OF CYCLE TIME DISTRIBUTION =',(A←-((⍴A)⍴A⍳'.')-⍴A←⍳EPS)⍳M1←DR1
[ 17] 'STANDARD ERROR = ',A⍳SE←((M2←DR2)-M1×M1)*0.5
```

```
[ 0] K←MM NTS NK;I;T;N
[ 1] T←N←⍴K←⍳I←0
[ 2] L:T←⍪\⍪1,×\((MM-I)↓⍳N←MM-I←I+1)÷⍳N←T
[ 3] NK←NK-1↓T←, 2 1 ⍉(T≥NK-0.1)/(⍪0,⍳N),[0.5] 0,¯1↓T
[ 4] →(MM≠⍴K←K,N-T←T[1])/L
```

```
[ 0] Z←NWS K;K1;K2;T1;T2;J
[ 1] →(0=+/Z←((1,T2←⍴K)⍴K1←K≠0),0)/J←0
[ 2] K2←K1/⍳T2
[ 3] L:T1←T1+0=T1←+/CNO←(P←((T2,T2)↑(2⍴¯1+H)↓RPM)[K2;J←J+1;])≠0
[ 4] K1←(T1×T2)⍴0
[ 5] K1[(0,T2×⍳¯1+T1)+CNO/⍳T2]←1
[ 6] K1←(T1,T2)⍴K1
[ 7] Z←Z,[1](K1+(T1,T2)⍴K-K2[J]=⍳T2),CSR[¯1+H+K2[J]]×((⍳0=+/P),CNO/P
[ 8] →(J≠⍴K2)/L
[ 9] Z←1⌽[1] Z
```

```
[ 0] PRTTRM;ΠPW;A;B;ROW;I;J
[ 1] ΠPW←120←I←0
[ 2] A←((1+(⍴T←M)),0)⍳STS
[ 3] ((2⍴B)⍴' '),(((B←¯1↑⍴A),4)⍴' |  '),((-B-⍳×C)⍴((C-B+1)⍴0),1,(B←⌈0.5×¯1+C←1+⍴1⍳⌈/CSR)⍴0)\⍴A
[ 4] (((C×T)+B+3)⍴'- ')
[ 5] L:→(T=I←I+(⍴⍴⎕←((B+⍴⍴ROW←T⍴0)⍴' '),'|')[1])/0
[ 6] ROW[TRANC[J]]←TRANC[J←TRANI[I]+⍳TRANI[I+1]-TRANI[I]]
[ 7] →L×⍴⍴Π←A[I;],' ¯ ',(C←I)⍳ROW
```

```
[ 0] Z←PSA;PTH;PIX;PPR;G;R;A;B;J
[ 1] PTH←(⍳PIX←0)⌿LPA⌿1 0
[ 2] PPR←PPR⍪+/PPR←EC(0=+/RPM)/⍳M
[ 3] G←(1,M)⍴1+R←Z←0
[ 4] L1:→(N≠⍴1+1↑⍴G←G,[1]+\X×,(⍴1,M)↑G)/L1
[ 5] G←GC;M
[ 6] L2:J←PTH[PIX[R]+⍳B←PIX[R+1]-PIX[R←R+1]]
[ 7] A←×\ 1 3 2 ⍴((⍴DEN),N,B)⍴X[J]×CSR[J])÷DEN∘.+(N,B)⍴CSR[J]
[ 8] Z←Z+(PPR[R]×+/A×(⍴A)⍴((B,N)⍴GC⌽⍳N])-X[J]∘.×GC⌽⍳N-1],0)÷×/X[J]×GCN]
[ 9] →(R≠⍴PPR)/L2
```

```
[ 0] Z←PSD L;PTH;PIX;PPR;G;R;A;B;J;C;D;P;E;MQL;I;K;CSR
[ 1] PTH←(⍳PIX←0)⌿LPA⌿1 0
[ 2] PPR←PPR⍪+/PPR←EC(0=+/RPM)/⍳M
[ 3] G←(1,M)⍴1+R←Z←0
[ 4] L1:→(N≠⍴1+1↑⍴G←G,[1]+\X×,(⍴1,M)↑G)/L1
[ 5] G←GC;M
[ 6] L2:E←1,L⍴0
[ 7] J←PTH[PIX[R]+⍳B←PIX[R+1]-PIX[R←R+1]]
[ 8] I←(QDS[J]≠'F')/J
[ 9] MQL←(+/(((⍴I),N)⍴GCN+1-⍳N])××\1,⍪((N-1),⍴I)⍴X[I])÷GCN]
[10] CSR←CSR
[11] CSR[I]←CSR[I]÷MQL
[12] Q←⍪-D×CSR[J]
[13] I←0
[14] L3:D←1,L⍴P←A←0
[15] C←0,(1-Q[I])×1,×\(L-1)⍴Q[I←I+1]
[16] →(QDS[J[I]]='F')/L6
[17] A←C
[18] →L7
[19] L6:K←L←2
[20] L4:D[K]←(K↑D)+.×⌽(K←K-1)↑C
[21] →(K≠1)/L4
[22] A←A+D×(X[J[I]]⍪P-1)×GCN+1-P]-X[J[I]]×(0,G)[1+N-P←P+1]
[23] →(P≠N)/L6
[24] A←A÷GCN]
[25] L7:K←L←2
[26] L5:E[K]←(K↑E)+.×⌽(K←K-1)↑A
[27] →((K≠1),I≠B)/L5,L3
[28] Z←Z+PPR[R]×1↓E
[29] →(R≠⍴PPR)/L2
```

```
[ 0] Z←PSM;PTH;PIX;PPR;G;R;A;B;J;C;D;I;MQL
[ 1] PTH←(⍳PIX←0)⌿LPA⌿1 0
[ 2] PPR←PPR⍪+/PPR←EC(0=+/RPM)/⍳M
[ 3] G←(1,M)⍴1+R←Z←0
[ 4] L1:→(N≠⍴1+1↑⍴G←G,[1]+\X×,(⍴1,M)↑G)/L1
[ 5] G←GC;M
[ 6] L2:A←⍴1 3 2 ⍴(D×(A+1)∘.÷C),[0.5] D←(A←⍳N)∘.÷C←CSR[J←PTH[PIX[R]+⍳B←PIX[R+1]-PIX[R←R+1]]]
[ 7] I←(QDS[J]≠'F')/J
[ 8] MQL←(+/(((⍴I),N)⍴GCN+1-⍳N])××\1,⍪((N-1),⍴I)⍴X[I])÷GCN]
[ 9] A[2;J⍳I;]←⍪(N,⍴I)⍴MQL÷CSR[I]
[10] A[1;J⍳I;]←⍪(N,⍴I)⍴2×MQL×MQL÷(CSR[I]×CSR[I])
[11] C←1,×\⍪((N-1),B)⍴X[J]
[12] A←(+/A×(⍴A)⍴CX((B,N)⍴GC⌽⍳N])-X[J]∘.×GC⌽⍳N-1],0)÷GCN]
[13] D←+/+/A[2;]∘.×A[2;]
[14] Z←Z+PPR[R]×(⌽+/A)+0,D-A[2;]+.×A[2;]
[15] →(R≠⍴PPR)/L2
```

```
[ 0] Z←A SML V;I;X
[ 1] Z←⍳I←0
[ 2] L:⍎'→((⍴TSV)≠⍴Z←Z,',A,'[X]+.×V[',A,'C[X←',A,'I[I]+⍳',A,'I[I+1]-',A,'I[I←I+1]]])/L'
```

```
[  0] SPEC NWK;SUB;A;I;J;OFF
[  1] SUB←⍴MEM←MEM,CEN,0
[  2] →((0≠1↑NWK),1)/L1,L0
[  3] L1:'CENTRE ',(⍕CEN),' : HEAD OF SUB-NETWORK ',(⍕NWK[2]),' CONNECTED TO CENTRE ',⍕NWK[1]
[  4] 'INPUT ROUTING PROBABILITY TO THIS CENTRE FROM CENTRE ',⍕NWK[1]
[  5] NET[NWK[3]]←1+⍴NET
[  6] RPM[NWK[1];CEN]←0
[  7] L0:'INPUT NO. OF CENTRES IN ROOT SEGMENT'
[  8] NET←NET,CEN,A←⍞
[  9] RPM[CEN+⍳1+⍳A-1;]←((A-1),M)⍴(CEN⍴0),1,(M-CEN)⍴0
[ 10] I←CEN←A
[ 11] 'INPUT SERVICE RATES'
[ 12] L2:J←⍴⍞←'CENTRE NO. ',(⍕CEN),' : '
[ 13] CSR[CEN]←⍎⍞
[ 14] →(I≠CEN←CEN+1)/L2
[ 15] 'INPUT NO. OF ATTACHED SUB-NETWORKS'
[ 16] OFF←-A-⍴NET←NET,A,(A←⍞←J←0)⍴0
[ 17] L3:→(A=J)/L4
[ 18] SPEC(I-1),J,OFF←J←J+1
[ 19] →L3
[ 20] L4:MEM[SUB[1];SUB[2]]←CEN-1
```

```
[ 0] Z←STN K;M;N;SK
[ 1] SK←N←+/K,M←¯1+Z←N=N←N
[ 2] L:→Lx⍳SK≠N←N-K[M]+⍴⍴Z←Z++/(-K[M])↑x\((M-M)↓⍳M←N-M←1+M)÷⍳N
```

```
[ 0] THM;A
[ 1] L1:'DO YOU REQUIRE EXACT OR APPROX. RESULTS ASSUMING PERMANENT STATIONARITY? INPUT ''E'' OR ''A'''
[ 2] →('EA'=1↑⍞)/L3,L2
[ 3] →L1x⍴⍴⍞←'TRY AGAIN'
[ 4] L2:→L4x⍴⍴A←1,PSM
[ 5] L3:TRM 1
[ 6] A←SSP←.xPNE←⍉ 1 0 0 MOM SSI/⍳T
[ 7] L4:'MEAN CYCLE TIME = ',⍕MCT←A[2]
[ 8] 'STANDARD ERROR = ',⍕SE←(A[3]-A[2]xA[2])*0.5
```

```
[  0] TRM H;I;A;B;C;J;MAT;STS
[  1] STS←(0,M)⍴⍴TRAN←TRANC←TSV←⍳TRANI←,I←0
[  2] →(H≠1)/L1
[  3] SSP←SSI←⍳0
[  4] L1:→(T<I←1+I+⍴A←⍳J←0)/L2
[  5] →(0=B←¯1+1↑⍴MAT←NWS C←,(¯1,M)↑STS←STS,[1] M NTS I)/L3
[  6] L:→(B≠⍴A←A,STN ¯1↓MAT[J←J+1;])/L
[  7] L3:TRANC←TRANC,A
[  8] TSV←TSV,+/(B←¯1↓MAT[J+1;])/M↑(¯1+H)↓CSR
[  9] TRANI←TRANI,⍴TRAN←TRAN,,(J,¯1)↑MAT
[ 10] →(H≠I)/L1
[ 11] →L1x⍳~¯1↑SSI←SSI,B[1]∧N=+/C
[ 12] →L1x⍴⍴SSP←SSP,x/X*C
[ 13] L2:SSP←SSP÷+/SSP
[ 14] →(~H←SBD)/0
[ 15] CR,'TRANSITION MATRIX FOR SUB NETWORK WITH ROOT CENTRE ',⍕H
[ 16] PRTTRM
```

APPENDIX 8

A8 The Mutual Validation Process

A8.1 Introduction and outline

This appendix describes in detail the mutual validation process discussed in chapter 6. Three model types were compared:

(i) Exact analytic.

(ii) Approximate (PSA) analytic.

(iii) Simulation.

The networks represented by the models all had tree-like topologies.

Analytic predictions were produced by the APL package listed in Appendix 7 and simulated results by the author's network simulator, see section 6.1. The numerical results, presented as tables and graphs in Appendices 9 and 10 respectively, are also discussed in some detail in this Appendix.

The validation process consisted of the following steps:

(i) Various network specifications were defined to conform with the assumptions of the theoretical analysis of chapter 5.

In all, nine networks were considered (the *base set*), their specifications being given in the following section.

(ii) Predictions were made, both for the exact analytical method and that using PSA, for each network defined in (i) using the implementation of Appendix 7.

(iii) Simulated results were generated, first for each of the cases in the base set defined in (i), and then for various modifications of these cases where the underlying assumptions of the exact theoretical model are violated. In particular, different queueing disciplines and service time distributions (with the same mean) were used. In all, twenty-one simulation runs were made in this way.

(iv) The exact and approximate theoretical predictions were compared, by methods to be described in section A8.3, for each base set case.

(v) The simulated results for each of the cases defined in (iii) were re-organised to represent cycle times for a single customer as well as for all customers. This resulted in two sets of simulated data.

(vi) For each simulation run and for each set of results given by (v) independence tests were performed by applying the autocorrelation function to the samples of cycle times. This can be used to assess the justification for application of the Central Limit Theorem for response time prediction. Independence is also required if the formulae for the estimates of the mean and variance of the cycle time distribution as simple averages are to be applied, and if a simple histogram is to be used to

represent the distribution itself.

(vii) For each simulation run, the moments derived from each set of results were compared with the theoretical exact and approximate counterparts, under their own assumptions if those of the simulation run were incompatible. There were thus four comparisons for each network.

(viii) Finally the one-sample Kolmogorov-Smirnov statistical test, [SIEG56], was applied to find the confidence level for each set of simulated results from each simulation run being drawn from a distribution given by the corresponding theoretical discrete form(defined in section 5.6). The tests were applied for both the exact and PSA theoretical discrete form distribution approximations, giving four in all for each network specification. The theoretical results were based on their own assumptions if those of the corresponding simulation could not be accommodated.

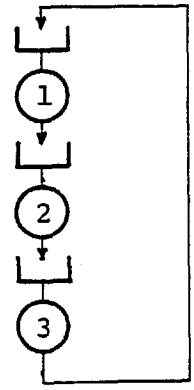
In the next section the networks used in the validation process steps (i) and (iii) are defined. This is followed by two sections in which the actual statistical analyses and interpretations are given for the comparisons of exact with approximate theoretical results and simulated with theoretical results respectively.

A8.2 Specifications of the networks used in the validation process

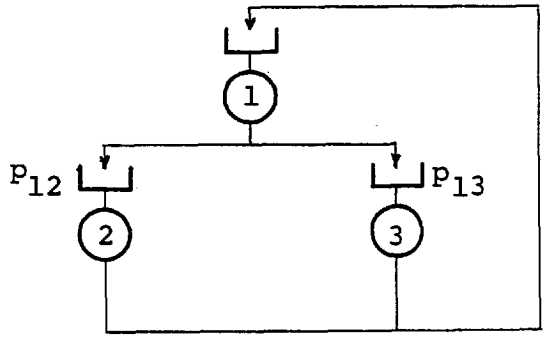
A8.2.1 Networks for the theoretical models

The nine tree-like networks (base set) specified for analysis by the exact theoretical model were based on the three topologies shown below.

(i) Cyclic



(ii) Simplest non cyclic tree-like network



(iii) More complex

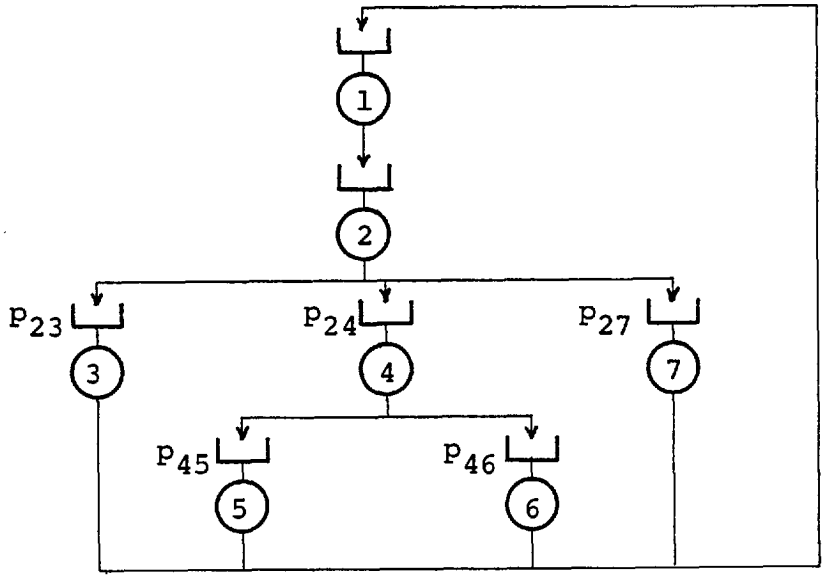


fig. A8.1

fig. A8.1 Network topologies

The particular network specifications were chosen so as to constitute a sample covering the most significant characteristics of tree-like networks; i.e. cyclic case (the most simple), branching in its simplest form with a selection of routing probabilities, and more complex configurations involving tandem segments (definition D5.1, section 5.2) of more than one centre and several (multiple) branches.

In cases (ii) and (iii), $\{p_{ij} | i,j \in Z^+\}$ are the routing probabilities of the network. The networks were parameterised in various ways as shown in table A8.1, by choices of network population (topologies one and two) and of routing probabilities or service rates (topology two). The labelling of the nine network specifications is not consecutive since the number of networks considered is increased, for the simulation runs, to 21, including the original nine interspersed throughout.

A8.2.2 Networks for the simulation models

The network specifications used for the simulation runs consisted of the base set discussed in the previous section, for which theoretical predictions were made, together with twelve other specifications derived by violating some of the underlying assumptions of the base set. The violations were made by choice of queueing disciplines other than FCFS and/or by choice of service time distributions other than negative exponential for one or more service centres in a base set specification. The details of each of the 21 specifications considered are presented in table A8.1 below.

Specification label	Network topology	Number of customers	Queueing disciplines	Routing probabilities	Service rates	Service time distributions	Associated base network	Network used for PSA predictions
S1*	(i)	2	FCFS for all servers	All one	Server 1:1 Server 2:2 Server 3:3	All negative exponential	S1	S1
S2*	(i)	4	as S1	as S1	as S1	as S1	S2	S2
S3*	(i)	6	as S1	as S1	as S1	as S1	S3	S3
S4	(i)	4	as S1	as S1	as S1	All Erlang(2)	S2	S2
S5†	(i)	4	PS for all servers	as S1	as S1	as S1	S2	S5
S6*	(ii)	2	FCFS for all servers	$p_{12} = 0.5$ $p_{13} = 0.5$	Server 1:1 Server 2:2 Server 3:2	All negative exponential	S6	S6
S7*	(ii)	4	as S6	as S6	as S6	as S6	S7	S7
S8*	(ii)	6	as S6	as S6	as S6	as S6	S8	S8
S9*	(ii)	4	as S6	$p_{12} = 0.4$ $p_{13} = 0.6$	as S6	as S6	S9	S9
S10*	(ii)	4	as S6	$p_{12} = 0.4$ $p_{13} = 0.6$	Server 1:1 Server 2:2 Server 3:0.5	as S6	S10	S10
S11	(ii)	4	as S6	as S6	as S6	All Erlang(2)	S7	S7
S12	(ii)	4	LCFS for all servers	as S6	as S6	as S6	S7	S13

Table A8.1 Network Specifications

Specification label	Network topology	Number of customers	Queueing disciplines	Routing probabilities	Service rates	Service time distributions	Associated base network	Network used for PSA predictions
S13†	(ii)	4	PS for all servers	as S6	as S6	as S6	S7	S13
S14	(ii)	4	Centre 1:LCFS Centre 2:FCFS Centre 3:FCFS	as S6	as S6	as S6	S7	S16
S15†	(ii)	2	Centre 1:PS Centre 2:FCFS Centre 3:FCFS	as S6	as S6	as S6	S6	S15
S16†	(ii)	4	as S15	as S6	as S6	as S6	S7	S16
S17†	(ii)	6	as S15	as S6	as S6	as S6	S8	S17
S18	(ii)	4	as S15	as S6	as S6	All Erlang(2)	S7	S16
S19*	(iii)	4	FCFS for all servers	$p_{12} = 1.0$ $p_{23} = 0.2$ $p_{24} = 0.5$ $p_{27} = 0.3$ $p_{45} = 0.4$ $p_{46} = 0.6$	Server 1:1 Server 2:2 Server 3:3 Server 4:4 Server 5:5 Server 6:3 Server 7:2	All negative exponential	S19	S19
S20†	(iii)	4	Centre 1:PS all others: FCFS	as S19	as S19	as S19	S19	S20
S21	(iii)	4	as S20	as S19	as S19	Centre 1: Erlang(2) all others negative exponential	S19	S20

Table A8.1(cont.)

In this table, the network specification labels are marked as follows:

(i) With an asterisk if the assumptions of both theoretical models are satisfied (base networks).

(ii) With a dagger if the assumptions of the PSA theoretical model only are satisfied.

Theoretical predictions were made for the base set specifications and corresponding simulated results produced for the complete set. The assumptions required by the PSA approximate theoretical method were satisfied for the *complete* set of specifications excluding those involving LCFS queueing discipline (see section 3.2). In fact the implementation of service time distributions other than negative exponential has not been completed so that these cases also are excluded from theoretical analysis under PSA. However, six networks in addition to the base set could be analysed by this method, the remaining six each being represented by the one of these with closest specifications, as shown in table A8.1.

Comparisons were made between the exact and approximate theoretical results and between the theoretical and simulated results in a mutual validation procedure. Of course, in the case of the base set specifications, it is necessary that the second comparison shows good agreement, certainly for the *exact* theoretical results, since the two model types are based on the same set of assumptions about the network characteristics. Thus the validation methodology is effectively to:

(a) Validate the simulation and PSA models with respect to the exact theoretical model for the base set networks;

(b) Validate the *theoretical* models in non base set cases with respect to the extrapolated simulation model. The extrapolation is *assumed* valid, no formal justification being

possible for the present.

The results of these comparisons follow in the next two sections.

A8.3 Comparison of exact and approximate theoretical results

A8.3.1 The approach to validation of the approximate method

For both the exact and approximate (PSA) theoretical methods, numerical predictions were made, for each network specification of the base set, of the following quantities:

(a) The mean and the variance of the cycle time distribution, given by the formulae for the first and second moments in Theorem T5.3, section 5.5 and equation E3.4, section 3.2 respectively;

(b) The approximate discrete form of the cycle time distribution, given by the formulae in Theorem T5.4, section 5.6 and section 3.3.3 respectively. The adequacy of the approximation may be assessed by comparing the mean and the variance estimated from the discrete form distribution with the analytic counterparts of (a). The results of this test are discussed in section A8.3.3.

In addition, the Laplace transform was computed for the specifications S2 and S7 (see table A8.1).

The validity of the approximate method was assessed by comparing its predictions and those of the exact method with respect to the following quantities:

(i) The predicted values for cycle time standard error. Recall from section 3.5 that the mean value given under PSA is exact;

(ii) The discrete forms for the distribution evaluated according to (b).

The results of these comparisons are presented in sections A8.3.2andA8.3.4 respectively. A rough guide as to the closeness of the approximation is obtained by inspection of the Laplace transforms of the two methods. Graphs for the network specifications S2 and S7 of these Laplace transforms may be found in Appendix 10 and suggest a satisfactory fit.

Of course, another way of judging the adequacy of the approximate method is by comparison of its predictions with simulated results (and real data when available). This test is described as part of the comparison between simulated and theoretical results in general in section A8.4. Here, in fact, it provides the only validation in the cases of the six specifications which can be represented under PSA but not by the exact analytical method.

A8.3.2 Comparison of standard errors

In table A9.1, Appendix 9, the standard deviations for each of the base networks, computed by the exact and approximate (PSA) theoretical methods are displayed together with their percentage difference. The mean values (the same for both methods) are also shown in the first column in order to provide an idea of scale. On inspection of the table it can be seen that the methods give results in good agreement. The least good results occur in the cases of networks S2, S3, S10 and S19. The first two of these are cyclic networks for which PSA was not expected to provide a very good representation (section 3.1). Of course the network S1 is also cyclic, but the number of customers is only 2 in this case so that the situation is closer

to that of a network with just one customer, for which PSA gives exact results (section 3.5). The network S10 was designed to be imbalanced, so giving a high standard deviation of cycle time distribution and perhaps requiring a rather more detailed analysis of the passage of the test customer than the overall, equilibrium representation provided under PSA. In the case of S19, the network is more complex, but not in the sense of (ii) in section 3.1 since there is only one arrival stream to each centre. Thus the states existing on arrival of the test customer at each centre in its path are highly conditional on the initial state space distribution by the argument of (i) in section 3.1.

Nevertheless, the results of table A9.1 are very promising and provide support for further investigation into the PSA model.

A8.3.3 Validity of the discrete form approximations

For each of the nine base networks, discrete form approximations were computed for the cycle time distribution by the exact and approximate methods. For the latter method this was also performed for the other networks in which the only change in the specifications was the use of PS queueing discipline, see section A8.2.2. In each case, the range considered on the time axis was between the origin and four times the mean value of the cycle time, as given in table A9.1. The number of time steps used was 50, resulting in a mesh size, Δ_i, of eight per cent of the mean. $(1 \leq i \leq 21)$.

Denote the resulting discrete form cumulative distributions by

$$\left\{ HE_i(j) \mid i \in S_E \right\}$$

for the exact analysis and

$$\left\{ HA_i(j) \mid i \in S_A \right\}$$

for the approximate analysis

where $S_E = \{1,2,3,6,7,8,9,10,19\}$

$S_A = \{1,2,3,5,6,7,8,9,10,13,15,16,17,19,20\}$

represent the networks analysed theoretically and

$j = 1,2,....,50$

is the time interval number corresponding to time $j\Delta_i$.

The adequacy of these discrete approximations may be assessed by comparing the means and standard deviations computed analytically, given in table A9.1, with the corresponding quantities computed from the discrete form distributions as follows. Define

$$MHE_i = \sum_{j=1}^{50} (j-\tfrac{1}{2})\,\Delta_i \left\{HE_i(j) - HE_i(j-1)\right\}$$

where $HE_i(0) = 0, \quad i \in S_E$

$$VHE_i = \sum_{j=1}^{50} \left\{(j-\tfrac{1}{2})\Delta_i - MHE_i\right\}^2 \left\{HE_i(j) - HE_i(j-1)\right\}$$

$$SHE_i = \sqrt{VHE_i}\ .$$

Then MHE_i and SHE_i are estimates of the mean and standard deviation of the cycle time distribution according to the (exact method) discrete approximation. Similarly estimates for the PSA analysis, MHA_i and SHA_i, may be made, $i \in S_A$. These values were computed for each network in the sets S_E, S_A and are displayed in tables A9.2 and A9.3 alongside their analytic counterparts. The values for MHE_i and MHA_i are *shift compensated* in the following way.

Recall from Proposition P4.3 that

$$\Pi_{\alpha\beta}(j) \leq \Psi_{\alpha\beta}(j) \leq \Pi_{\alpha\beta}(j+\ell_{\alpha\beta})$$

where $\alpha,\beta \in S$, the state space of a cyclic network,
$j \in Z^+$,

$\Psi_{\alpha\beta}$ is the discrete form probability distribution for the time delay between states α and β and $\Pi_{\alpha\beta}$ the approximation to $\Psi_{\alpha\beta}$ derived by the exact analysis,

$\ell_{\alpha\beta}$ is the number of transitions involved in the routes from α to β.

This result applies to cyclic networks, but the result is clearly true, by the argument of proposition P4.3, for tree-like networks if $\ell_{\alpha\beta}$ is chosen to be the maximum length of the routes between states α and β, $m_{\alpha\beta}$ say. A similar result obviously applies to the PSA analysis also.

Thus the approximated discrete form distributions may be shifted to the right of the true discrete forms by up to $m_{\alpha\beta}\Delta$ time units for mesh size Δ. For small Δ this is of no consequence since $m_{\alpha\beta}$ is always finite and here the effective shift is never larger than around 6Δ. The shift, which will not in general be uniform over the whole time axis, will result in an estimate for the mean which is too large. For the mesh sizes chosen as described above, $\Delta \sim 0.2 - 0.4$, the shift is significant and so is removed by shift compensation in which the first few points (corresponding to small values of j) of the discrete approximation are omitted. Thus, the matching of means in tables A9.2, A9.3 is unimportant, but note that the

shift compensation has little effect on standard deviation which becomes the main criterion for the validation. For the smaller mesh size used in section A8.4.5,shift compensation was unnecessary and not applied. Note further that for non cyclic tree-like networks, the number of convolutions involved in the PSA method is smaller than for the exact method since, under PSA, transitions in paths other than the one taken by the test customer are not considered. Thus, in general, shift compensation is less for the PSA method.

On inspection of tables A9.2 and A9.3 it is immediately apparent that the standard deviations estimated by the approximate discrete forms for the cycle time distributions are in close agreement with their analytically derived counterparts. It can also be seen that the accuracy decreases as the number of transitions involved in the test customer's passage through a network increases. This is exactly what one would expect since further approximation is introduced whenever a convolution is performed, and application of the convolution operation is in 1-1 correspondence with these transitions.

Further validation tests are performed in sections A8.4.4 and A8.4.5 by comparing with simulated results.

A8.3.4 Comparison of the discrete form distributions

Having compared the approximated discrete form distributions, derived by the exact and PSA methods, with respect to their standard deviation estimates, in this section the validity of the latter is assessed by comparison with the former. Two comparisons are made as shown in fig.A8.2 for two continuous curves:

(i) Vertical: The maximum vertical distance between the two discrete form distributions for base networks $i \in S_E$ is

$$D_i^{(V)} = \max_{1 \le j \le 50} \left\{ |HE_i(j) - HA_i(j)| \right\}$$

The physical interpretation of this measure is that it represents the maximum error in the value of a percentile, based on PSA, placed at any point on the time axis;

(ii) Horizontal: In order to compute the maximum horizontal distance between the two discrete form distributions for base networks $i \in S_E$ it is first necessary to define their inverse functions. These were defined on a mesh of 100 points on the cumulative probability distribution axis (the P-axis), viz. $0.01\,i$ $(0 \le i \le 99)$, the probability values, of course, lying in the range $[0,1)$. The method used was as follows:

Given mesh $P_i = 0.01i$ $(0 \le i \le 99)$ on the P-axis, mesh $t_j = j\Delta$ $(1 \le j \le 50)$ on the time axis and monotonic increasing function $f : \{t_j | 1 \le j \le 50\} \to \{P_i | 0 \le i \le 99\}$, then an inverse function of f, f^{-1} is given, using linear interpolation, by

$$f^{-1}(P_i) = t_j + (t_{j+1} - t_j) \frac{\{P_i - f(t_j)\}}{f(t_{j+1}) - f(t_j)}$$

where j is such that $f(t_j) \le P_i \le f(t_{j+1})$ $(1 \le j \le 50)$, the ratio is unity if $f(t_j) = f(t_{j+1})$ and $f(t_{51}) = 1$.

Since the discrete form distribution approximations are monotonic increasing, the inverse function of HE_i, HIE_i say,

is given by f^{-1} in the above where $f = HE_i$ and $\Delta = \Delta_j$. The inverse, HIA_i, of HA_i is defined similarly.

Thus the maximum horizontal distance between the two discrete form distributions for base networks $i \in S_E$ is

$$D_i^{(H)} = \max_{1 \leq j \leq 100} \left\{ |HIE_i(j) - HIA_i(j)| \right\}.$$

The physical interpretation of this measure is that it is the maximum error (in time units) in the positioning of a given percentile on the time axis.

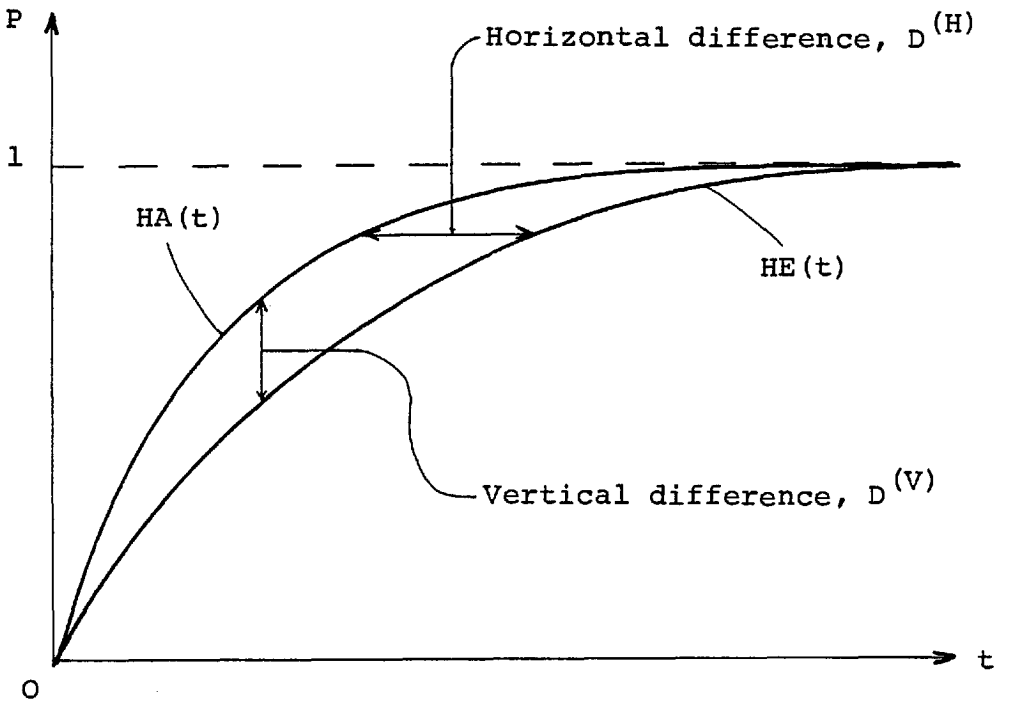


fig. A8.2 Vertical and horizontal comparisons

In tables A9.4 and A9.5 the maximum vertical and horizontal differences between the exact and PSA discrete form distribution approximations are shown. The corresponding abscissa and ordinate values are also given, together with the (exact) mean of the distribution. From these tables it can be seen that the PSA method provides results in close agreement with

the exact one. The worst cases are for networks S2,S3,S10 and S19, which is as found in section A8.3.2. The discrete form distributions may be seen in graphical form, along with the corresponding histograms derived from simulated results, for various networks in Appendix 10.

A8.3.5. General assessment of the PSA method

The results discussed in the previous sections and presented in tables A9.1-A9.5 lead to the conclusion that the PSA method of analysing cycle times in tree-like networks provides good approximations to the exact results, the poorest accuracy being obtained in the types of cases predicted in section 3.1. Furthermore, the method is computationally efficient and so provides the basis for a practical tool in computer performance evaluation.

The ultimatetest remains validation with respect to actual data, or simulated data in its absence, which is discussed in the following section. In this way the adequacy of the approximation with respect to network specifications more general than the tree-like case can be judged.

A8.4 Comparison of theoretical and simulated results

A8.4.1 The approach to validation

As discussed in section A8.1,sequences of cycle times, together with identification of the customer completing its cycle, were obtained using a simulation program for tree-like networks. The pair (cycle time, customer identification) is defined to be an *element* in what follows. A sequence of around 1000 such elements was generated for each of the 21

network specifications given in table A8.1. Now, each network was initialised with its customers at certain service centres at the start of its simulation. Thus an equilibrium situation could not be assumed initially and so only the last 600 elements were considered for statistical analysis. That is, it is assumed that steady state equilibrium would have been attained after 400 arrivals to the first (root) centre in the closed tree-like network modelled. Such an assumption is quite commonplace in contemporary queueing network modelling and its validity is not in doubt. However, this validity could be assessed quantitatively via the transient analysis presented in chapter 7.

Two sets of data were produced from each of the sequences of 600 cycle time/customer identifier elements:

(i) The 600 cycle times in their order of occurrence;

(ii) The sequence of cycle times pertaining to the identifier of one particular customer. This sequence, then, contains a much smaller number of elements, of expected value 600 ÷ n where n is the number of customers in the network specification.

In order that statistics based on data such as this be unbiased, it is necessary that the data in the sample be *independent*. Thus the first test made is to assess the independence of each of the two sets of data for each of the network specifications by use of the autocorrelation function. The results of the test are reported in the next section, and suggest very strongly that the independence condition is not violated for sequence (ii). For sequence (i) this was less conclusive.

Thus, unbiased estimates for the mean and standard deviation can be obtained by the usual formulae and then compared with similar quantities obtained for both of the corresponding theoretical models. These results are discussed in section A8.4.3. In this way a check can be made that the results are sufficiently close for the nine base network specifications; the simulation and analytic models are, in these cases, based on the same assumptions. In addition, the theoretical predictions for the base networks can be tested for cases in which their underlying assumptions are violated in various ways - a check on the "robustness" property applied to time delay distributions. Of course, the PSA theoretical method can model most of the non-base networks (all except those given in section A8.2.2) and so tests on its generality can be made empirically.

The final statistical test used in the validation process is the Kolmogorov-Smirnov (KS) test. This was used to compare, for each network specification, the theoretical discrete form distribution approximations with the cumulative relative frequency histograms derived from each of the corresponding two sets of simulated data. This is discussed in section A8.4.4 and gives rather disappointing results because the size of mesh used is too great in the theoretical analysis.

Thus there follows a section in which some of the theoretical predictions are refined by use of a smaller mesh size. The section closes with a short summary and general assessment of the analytic methods.

A8.4.2 Independence tests

Given a time series $\{ z_t \mid t \in Z \}$

the autocorrelation function (ACF) at lag k, ρ_k, is defined by, [CHAT75]

$$\rho_k = \frac{E(z_t z_{t-k})}{E(z_t^2)}$$

where E denotes expectation.

ρ_k has an unbiased estimate, r_k given by

$$r_k = \frac{\sum_{t=1}^{N-k} z_t z_{t+k}}{\sum_{t=1}^{N} z_t^2}$$

for a sample of data $\{z_t \mid t = 1,2,\ldots,N\}$.

For the series to be independently distributed,

$$\rho_k = 0 \quad \text{for} \quad k \geq 1.$$

Now, the so-called large lag standard error, σ_k, of r_k for the k'th lag, where r_ℓ is deemed to have become zero for lags $\ell \geq k$, is given by, [BART46]

$$\sigma_k \simeq \frac{1}{\sqrt{N}} .$$

Thus the independence requirement becomes

$$r_k \simeq 0 \quad \text{for} \quad k \geq 1$$

with confidence band $\pm 2/\sqrt{N}$ outside of which less than 5% of the estimates computed for r_k should lie if this condition is to be satisfied. Estimates for the first 50 lags of the ACF

were made for each network specification and each sample of simulated cycle times.

In tables A9.6 and A9.7, for each network specification the large lag standard error, the ACF estimates for the first 12 lags and the number of estimates in the first 50 lags lying outside the $\pm 1/\sqrt{N}$ and $\pm 2/\sqrt{N}$ confidence bands are displayed. In table A9.6 the results are based on the complete sets of simulated cycle times, regardless of customer identification, and in table A9.7 on the simulated cycle times for a single customer. Intuitively it is to be expected that the first table will reveal the lesser degree of independence, especially for the cases of cyclic networks with FCFS queueing discipline. This is because one would expect a customer's cycle time to be highly correlated with that of the customer completing the previous cycle; this customer always being the same under the assumptions specified. On inspection of the tables, it is at once apparent that the samples associated with the cycle times of a single customer satisfy the independence test given above for all the network specifications. This is not so for most of the samples based on the complete set of cycle times, many of which show a first lag ACF significantly different from zero.Thus independence is in doubt and the above intuition is well founded.

It is therefore concluded that the statistical analysis may proceed, certainly for the "single customer" samples, with estimation of the mean, standard deviation and cumulative histogram for each network specification. This is discussed in the following section.

There is, however, a further important consequence, perhaps even more important from the modelling viewpoint, of the independence property of the single customer samples. This

concerns response time distribution (see sections 4.10, 5.8.4). Since it is strongly suggested that it is valid to assume that successive cycle times of any particular customer are independently distributed, the Central Limit Theorem may be applied to their sum. In this way a good approximation to response time distribution should be possible by use of just the mean and standard deviation of the cycle time distribution.

A8.4.3 Comparison of moments

For each network specification and each sample of simulated cycle times the mean and the variance were estimated by the usual formulae*; certainly a valid step for the single customer sample type in view of the independence properties established in the previous section. Four comparisons were then made on the mean and standard deviation values:

(i) Between the estimates based on the complete sample of simulated cycle times and the corresponding theoretical

  (a) exact predictions

  (b) PSA predictions;

(ii) Between the estimates based on the single customer sample of simulated cycle times and the corresponding theoretical

  (a) exact predictions

  (b) PSA predictions.

---

* For the sample $\{Z_t | 1 \leq t \leq N\}$ these estimates are

$$\bar{Z} = \frac{1}{N} \sum_{t=1}^{N} Z_t \quad \text{for the mean and}$$

$$\frac{1}{N} \sum_{t=1}^{N} (Z_t - \bar{Z})^2 \quad \text{for the variance.}$$

The results of these comparisons are presented in tables A9.8 to A9.11. For cases in which the network specification violated the assumptions for tree-like networks, the *exact* theoretical results for the associated base networks were used in (i)(a) and (ii)(a). For networks S4, S11, S12, S14, S18 and S21, other specifications (as given in table A8.1) had to be used in the PSA models of (i)(b) and (ii)(b) since the PSA analysis cannot handle LCFS queueing discipline or, for the present, non exponential service times. In such cases, LCFS was represented by PS queueing discipline and the Erlang (2) service time distribution by negative exponential.

Any conclusions based on (i) (tables A9.8, A9.9) must carry less weight than those based on (ii) (tables A9.10, A9.11) since significant correlation was observed in the sampled data used, section A8.4.2. However, it turned out that the comparisons (i) were not in disagreement with (ii) to any great extent.

For the case of mean value comparisons, the results based on either of the simulated data samples were in close agreement with both theoretical counterparts for every network specification. This is as expected since the mean cycle time for any path is the sum of the mean sojourn times for each centre in the path, see section 3.5. The mean sojourn time for a centre is unaffected by choice of queueing discipline or, of course, service time distribution with same mean. Thus the means are always those of the associated base networks. Furthermore, for the base networks, good agreement was obtained with respect to the standard deviations; particularly so in the comparisons with the results of the exact theoretical method. This is as was hoped, for otherwise an error must have been

present in the simulation program or abnormal simulation runs must have occurred since the underlying assumptions of both models are identical.

However, for non-base networks, the exact theoretical model's predictions of the standard deviation for the associated base networks differed considerably from the simulated results, table A9.10. The worst cases were specifications S12, S14, S16, S17, S18 which all involve a change of queueing discipline to PS or LCFS at the root centre in network topology (ii), fig. A8.1, and also at the other centres in case S12. It can be seen that the use of these queueing disciplines causes an increase in the cycle time standard deviation, especially for LCFS. It is perhaps a little surprising that the error involved in case S13 where all centres have PS queueing discipline is about half that for case S16 which differs from S13 in that only the root centre has PS discipline, the others FCFS. It is possible that this result is due to an exceptional simulation run. One would also expect poor results for network S15, but here an operational error in the running of the simulation is suspected in that the results are identical with those of network S6; indeed this is also true of the histogram representations of the cycle time distribution. Thus it is concluded that the robustness property does not hold in the context of time delay distributions and that the exact method must be restricted to modelling networks which conform to its own assumptions.

It can be seen from table A9.11 that the PSA method gave far better standard deviation predictions than the exact method for the non-base network comparisons, especially in cases conforming to the assumptions of the PSA model. The predictions

are particularly impressive for cases involving PS discipline (except for network S15 for which the data is suspect) and less so for LCFS which had to be represented by PS anyway. As stated above, the results were also perfectly adequate for the base networks.

Thus confidence in the PSA model in its full generality is increased. The final statistical test performed on the method relates to the predicted forms of the cycle time distributions themselves and is included in the following section.

A8.4.4 Comparison of distributions: the KS test

For each network specification, each sample of simulated cycle times was used to form a cumulative relative frequency histogram with time-axis mesh defined as for the corresponding base network in section A8.3.3. Four comparisons, defined in (i) and (ii) of the previous section, are therefore possible, those involving the complete samples, (i), being of much less significance in view of their inability to satisfy the independence condition. Graphs showing various combinations of cumulative histograms, based on the approximate and exact theoretical analyses and on sampled data may be found in Appendix 10.

The comparisons were made by application of the KS statistical test to the maximum absolute (vertical) difference, D, over the time axis mesh, between the relative frequencies according to the appropriate theoretical predictions and the simulation samples. The results of these tests are presented in tables A9.12 and A9.13 in Appendix 9 for comparisons (i) and (ii) respectively. Thus the table of greatest significance is A9.13, see above.

Again, as discussed in the previous section, it is

to be hoped that, for the base networks, the KS test will indicate a good fit between the discrete form distribution predicted by the exact method and the corresponding histogram constructed from the simulated data. However, this was not found to be the case, as can be seen in table A9.13 in which the significancelevels are very low. Not surprisingly therefore, the results are very poor in non-base network comparisons for the exact theoretical method; some of the maximum differences, D, approaching 0.5. The results for the PSA method are not especially good either, particularly for the cases it could not model explicitly.

The cause of this lack of fit by the theoretical methods was assumed to be that their mesh sizes were too large; this problem had already been suggested by the need for shift compensation of the resulting discrete form distributions. A smaller mesh size was not chosen originally for the sole reason of computational efficiency. Evaluation by the exact method requires a great deal of computing power which increases considerably with the number of points in the time-axis mesh. Thus any significant decrease in mesh size for the *whole* base set of networks was ruled out, and it was preferred to define the mesh size for all networks and modelling methods in the same way. For the PSA method the computational problems involved in decreasing the mesh size are insignificant, see section A8.4.6. Consequently it was decided to decrease the mesh size in *every* case for the networks modelled by the PSA method, and in just two of the base set cases for the exact method. The results are reported in the next section.

Finally, it may have been noticed that the predictions of the PSA method are better than those of the exact method in some of the *base set* cases. This is simply explained by

the following two reasons:

(i) The effects of shift compensation; the size of the shift is restricted to an integral number of time steps.

(ii) The PSA method requires fewer convolutions in general (see section A8.3.3),so that the resulting (additional) approximation is less than for the exact method - recall each convolution operation introduces some error. Thus it may well be that this single inaccuracy incurred by the exact method outweighs the double one incurred by the PSA, approximate method, particularly in view of the accuracy of the latter method seen in previous sections.

A8.4.5 Use of a finer mesh

In order to obtain better approximations for various discrete form distributions by the theoretical methods, use of a smaller mesh size is necessary (and sufficient,by the convergence property of Corollary CT5.4). However, since the *range* considered on the time axis should not be decreased, this being defined independently of the required precision, the number of time steps required must increase in inverse proportion to the mesh size. In each of the cases considered in this section, the mesh size was reduced to one quarter of its original value given in section A8.3.3resulting in the number of points in the mesh being increased to 200. In order to compare each resulting discrete form distribution with the histogram constructed from the corresponding simulation run, by means of the KS test discussed in the previous section, the mesh must be the same for both cases. For reasons concerning

computer storage availability, the raw simulation data was not kept locally and the histograms were not easy to re-construct. Thus the theoretical distributions were re-cast into a mesh of 50 points by taking every fourth point.

The network specifications analysed in this way were S2 and S7 for the exact method, and the same subset as in the previous sections for the PSA method. Only two cases were considered for exact analysis for reasons of computer resource usage. The increased precision had the result that shift compensation was no longer necessary in any of the cases considered - a promising start to the analysis. The improvement obtained may be seen graphically in Appendix 10 for some cases.

The results were analysed in precisely the same way as were their predecessors in previous sections and tables A9.14 to A9.19 are analogous to tables A9.2 to A9.5 and A9.12, A9.13. The conclusions to be drawn from these tables are self evident and briefly discussed as follows:

(i) From tables A9.14, A9.15 it can be seen that in every case the discrete form approximation gave standard deviation estimates much closer to their true values than were obtained for the original mesh, and similar means. This is the more impressive in view of the absence of shift compensation, and demonstrates examples of convergence of the method.

(ii) From tables A9.16, A9.17 it can be seen that the approximate discrete forms for the exact and PSA methods became closer to each other for network S7. However, for network S2, the results show a marginally greater difference. This is assumed to be due to the effects of the shift compensation which was applied for the larger mesh size only, but rounding errors may also have played a part.

(iii) Finally, from tables A9.18, A9.19 (the former being of less importance because of the correlation in the sample of simulation data) it can be seen that the goodness of fit according to the KS test increased considerably. Indeed, for every test involving a network with no specification violating the assumptions of the corresponding theoretical model, the hypothesis that the sampled data was drawn from the theoretical discrete form distribution could not be rejected, even at quite low confidence levels (given by higher numerical values in the tables). As in (ii), the exact results for network S2 were slightly poorer.

The tests for networks with associated base network S2 or S7 (excluding S2 and S7 themselves) were again poor for the exact analysis, as expected in view of the discussion of section A8.4.3. For the cases not modelled directly by the PSA method, the KS test gives improved, although not really adequate results.

The conclusion (iii) is particularly pleasing from the point of view of the PSA method - in the cases of the exact method applied to specifications S2 and S7, it was only to be expected for reasons already given. Indeed, the main practical achievement of the validation discussed here has been the emergence of the PSA method as a potentially valuable tool for the computer performance analyst.

A8.5 Conclusions

A8.5.1 Assessment of the exact theoretical method

The validation process discussed in this appendix has confirmed that the predictions for properties of cycle times made by the exact theoretical model are in close agreement with those of the network simulator for specifications which conform to the assumptions of tree-like networks. However, if these assumptions are violated, for example by use of queueing disciplines other than FCFS, the accuracy of the predictions becomes very poor. Thus, it may be concluded that the robustness property of queueing network analysis discussed in section 2.2 does not apply to the modelling of cycle time distribution by this method. Consequently the domain of situations in which the exact theoretical analysis may be applied is somewhat limited, viz. to tree-like queueing networks.

Furthermore, the exact model, in the form of a computer program (Appendix 7) which produces numerical predictions based on the analytic results of chapter 5, has a tremendous appetite for computing resources, even for the solution to quite simple networks. The resources required include primarily both main storage and CPU time. In fact, in order to obtain the solutions for network specification S19 (table A8.1) several APL functions had to be modified to use temporary auxiliary storage (APL files) for intermediate results so as to free main storage for use in expression evaluation. This avoided a storage overflow but increased processing time; in fact other such trade-offs between execution time and storage requirements were also necessary to prevent such an overflow. Finally, the package took more than one hour, on a very powerful APL implementation, to compute the 50 point discrete form approximation of the cycle time distribution

- hence the reluctance to re-run with a mesh of 200 points!

Thus it is apparent that this case of a tree-like network with 7 service centres and 4 customers is close to the limit of complexity permitting practicable solution by the exact method. Certainly the method must fairly soon become impracticable as networks become more complex, in terms of more servers or more customers, because of the sheer size of the state space and the operations involved.

For a network of M servers and N customers, the size of the state space is $\binom{M+N}{N}$ and the storage requirement of the state transition matrix alone is of the order of $M\binom{M+N}{N}$, see section 4.7.2. The number of operations involved in the computation of the defining expression of any result in chapter 5 will be of at least this order since every such expression includes at least one reference to the transition matrix. In fact the order will generally be a lot higher. The expression for the second moment for example is a matrix product with five (indirect) references and the computation for the discrete form distribution involves many operations on intermediate data objects of size comparable with the transition matrix. Furthermore, in the latter case, the execution time requirement is directly proportional to the number of time-axis mesh points. Thus it can be seen that the computing requirements with respect to both storage and execution time, of the exact theoretical model increase *at least* combinatorially with the number of centres and population of the tree-like network.

As a result, the exact method is not only rather limited by its domain of applicability but also by its computing resource requirements. Nevertheless, it is an excellent practical tool for the *simple* cases in which it is applicable, as well

as a basis for validation of approximate methods.

A8.5.2 Assessment of the PSA method

For the network specifications considered in the validation process which conform to the assumptions of the PSA method†, it has been established in this appendix that the PSA model provides good predictions concerning the distribution of cycle time. These assumptions are very general in nature, only LCFS queueing discipline not being admissible in the specifications of table A8.1 (in addition to non exponential service time distributions). Thus the PSA method can be applied with confidence, in a domain of situations which is far more general than for the exact method.

Furthermore, it is also far more efficient than the exact method. The quantities relating to cycle time distribution, conditional on the choice of some given path, can be computed as some composition of the same quantities evaluated for the individual service centres in the path, see chapter 3. This is because of the assumption that the service centres in any path of the test customer through the network modelled operate independently. For example, the composition referred to may be convolution (for the distributions themselves), a product (for their Laplace transforms) or more complex (for the moments). Thus the storage requirement of this method reduces to that of storing the intermediate results in the sequence of composition operations together with that of the computation of the results for a single centre and performing the composition.

Thus, for a network of M centres and N customers and finite number of paths for the test customer, the execution time

---

† The PSA method as implemented at present cannot represent non exponential service time distributions.

requirement is of the order of

(Number of possible paths) × M × N

since M is an upper bound for path lengths and the number of operations required for a single centre is of order N.

The storage requirement is of a smaller order in that the results need not be retained for every individual server computation; only the intermediate results need be kept in the sequences of compositions (referred to above) and paths' weightings. Thus it can be seen that the PSA method is very much more efficient than the exact method, with respect to both storage and execution time. This was quite evident in practice: the computation of the discrete form distribution approximation for network S19 took less than a minute to compute for the PSA method compared with more than one hour for the exact method.

Thus the PSA method provides a far more practical tool for the computer performance analyst than the exact method in view of its superior efficiency and greater generality of application. It is tentatively assumed that the PSA method can be applied as a representative model to any network specification which conforms to its underlying assumptions. This assumption was shown to be acceptable for the selection of tree-like networks considered in this appendix. However, intuitively, one would expect the method to provide a better representation in networks *not* possessing the non-overtaking property of tree-like networks since in such cases the network can reach equilibrium more quickly (see section 3.1). Thus the assumption is intuitively valid.

For various tree-like networks violating the assumptions of the PSA method, notably involving LCFS discipline, it was seen in this appendix that the method's predictions were

barely adequate. Thus the robustness property (c.f. section 2.2) is again not valid, although it is not contradicted so drastically as in the case of the exact method, on the basis of the numerical results presented here. This is as expected, given adequate representation of the networks which can be modelled explicitly by the PSA method, in view of the greater generality of application which permits more choices for a network structure *close* to the one to be modelled. For example, in the cases considered in this appendix, the PSA method could use PS queueing discipline in place of LCFS whereas the exact method was restricted to use of FCFS, with poorer predictions.

Finally, the need for a formal error analysis of the PSA method should again be stressed. This poses many problems and is discussed in more detail in chapter 8. Intuitively the method appears to provide representative models for a wide variety of network specifications. However, validation has been performed for only 15 cases (see table A8.1), and while offering support for the method, does not *prove* its adequacy. However, and in the absence of such an error analysis, one may follow the contemporary modelling approach, "if it works, do it", as discussed in section 2.2, given, of course, stringent empirical tests as justification.

A8.5.3 Ultimate validation

As discussed in chapter 6, the process described in this appendix is only a systematic mutual validation of three models with respect to each other. Thus, although the results give support to ones conviction in the accuracy of each model type, ultimately validation *must* be performed with respect to data measured on at least one actual computer system.

APPENDIX 9

Tables showing the results of the validation process described in chapter 6 and Appendix 8.

Network Specification	Mean Cycle Time	Standard devn. (exact)	Standard devn. (PSA)	Percentage difference
S1	2.576	1.441	1.520	5.4
S2	4.245	1.962	2.187	11.5
S3	6.091	2.421	2.738	13.1
S6	2.250	1.451	1.488	2.6
S7	4.045	2.035	2.115	3.9
S8	6.006	2.491	2.574	3.4
S9	4.048	2.033	2.119	4.2
S10	5.607	3.916	4.241	8.3
S19	4.246	1.995	2.171	8.8

Table A9.1 Cycle time standard deviations predicted by exact and PSA theoretical methods

Network Specification	Mean Cycle Time	Mean estimated by disc. distn.	Percentage error	Standard devn.	Std. devn. est. by disc. distn.	Percentage error
S1	2.576	2.560	.6	1.441	1.460	1.3
S2	4.245	4.324	1.9	1.962	2.069	5.5
S3	6.091	6.224	2.2	2.421	2.619	8.2
S6	2.250	2.221	1.3	1.451	1.459	.6
S7	4.045	4.038	.2	2.035	2.102	3.3
S8	6.006	6.270	4.4	2.491	2.573	3.3
S9	4.048	4.043	.1	2.033	2.100	3.3
S10	5.607	5.624	.3	3.916	4.039	3.2
S19	4.246	4.266	.5	1.995	2.193	9.9

Table A9.2 Discrete form distribution for exact theoretical method

Network Specification	Mean Cycle Time	Mean estd. by disc. distn.	Percentage error	Standard devn.	Std. devn. estd. by disc. distn.	Percentage error
S1	2.576	2.602	1.0	1.520	1.539	1.3
S2	4.245	4.344	2.3	2.187	2.317	5.9
S3	6.091	6.341	4.1	2.738	3.010	9.9
S5	4.245	4.333	2.1	3.101	2.872	7.4
S6	2.250	2.314	2.9	1.488	1.466	1.5
S7	4.045	4.209	4.1	2.115	2.179	3.0
S8	6.006	6.320	5.2	2.574	2.673	3.8
S9	4.048	4.215	4.1	2.119	2.185	3.1
S10	5.607	5.719	2.0	4.241	4.426	4.3
S13	4.045	3.846	4.9	3.455	3.053	11.6
S15	2.250	2.195	2.5	1.762	1.617	8.2
S16	4.045	3.900	3.6	3.453	3.058	11.4
S17	6.006	5.741	4.4	5.382	4.671	13.2
S19	4.246	4.036	4.9	2.171	2.329	7.3
S20	4.246	4.442	4.6	3.025	3.039	.5

Table A9.3 Discrete form distribution for the PSA theoretical method

Network Specification	Max. Vert. difference	Time	Percentile	Mean value
S1	.01728	3.50	.77	2.574
S2	.04098	2.38	.14	4.245
S3	.04216	8.77	.82	6.091
S6	.03020	2.16	.54	2.250
S7	.03332	4.53	.62	4.045
S8	.01338	8.17	.77	6.006
S9	.03402	4.53	.62	4.048
S10	.04121	3.59	.33	5.607
S19	.06594	2.98	.26	4.246

Table A9.4 Vertical comparison of exact and PSA theoretical methods

Network Specification	Max. hor. difference	Time	Percentile	Mean value
S1	.25181	7.32	.99	2.576
S2	.65570	10.52	.99	4.245
S3	1.08396	13.54	.99	6.091
S6	.21225	7.00	.99	2.250
S7	.36968	10.31	.99	4.045
S8	.34344	12.58	.98	6.006
S9	.39341	10.31	.99	4.048
S10	1.66766	18.04	.99	5.607
S19	.41237	2.21	.13	4.246

Table A9.5 Horizontal comparison of exact and PSA theoretical methods

Network Spec.	Large lag SE	Autocorrelation function for first 12 lags												No. in first 50 outside LLSE	LLSE x 2
S1	.041	.371	-.049	.089	-.028	-.020	.005	-.025	-.017	.005	.015	-.003	-.005	12	4
S2	.041	.671	.412	.167	-.128	-.067	-.033	-.005	.033	.044	.050	.024	-.020	14	9
S3	.041	.792	.617	.437	.249	.093	-.078	-.037	-.006	.003	.007	-.015	-.049	16	12
S4	.041	.650	.370	.142	-.168	-.142	-.115	-.123	-.073	-.012	.022	.016	-.023	12	6
S5	.041	.199	.225	.163	.026	-.001	-.083	.016	-.023	.006	-.001	.026	.020	11	5
S6	.041	.370	.062	.041	-.015	-.030	.016	-.022	-.014	.011	.031	.031	-.023	10	2
S7	.041	.613	.466	.252	.066	.020	-.003	-.017	-.023	.007	.013	-.016	-.036	14	10
S8	.041	.750	.639	.481	.330	.179	.065	.011	.006	-.008	-.009	-.034	-.058	17	14
S9	.041	.650	.486	.251	.078	.002	-.018	.003	.010	.033	.011	-.017	-.045	13	9
S10	.041	.407	.275	.185	.044	-.051	-.037	-.071	-.091	.002	-.033	.015	.038	14	5
S11	.041	.658	.429	.181	-.068	-.114	-.119	-.109	-.057	-.027	.008	.033	-.017	10	7
S12	.041	-.042	-.046	.024	.019	-.041	-.048	-.011	-.013	-.035	-.015	.015	-.027	6	3
S13	.041	.050	.106	.066	.014	-.024	.003	-.027	.011	.068	.000	.000	.041	11	3
S14	.041	-.009	-.056	-.020	.030	-.041	-.028	-.040	.010	-.054	.013	-.031	.001	4	2
S15	.041	.370	.062	.041	-.015	-.030	.016	-.022	-.014	.011	.031	.031	-.023	10	2
S16	.041	.090	.126	.045	.030	-.033	-.019	-.031	-.014	.033	.044	.005	.021	10	4
S17	.041	.035	.021	.002	-.061	.002	.051	.004	.080	-.013	-.038	-.002	-.020	9	1
S18	.041	-.060	-.005	-.065	.015	-.033	-.065	-.018	-.050	-.025	-.020	-.001	.003	4	1
S19	.041	.596	.422	.204	.022	-.048	-.040	-.022	.003	.037	.061	.031	.023	14	10
S20	.041	.123	.095	.088	-.004	.078	.017	-.054	-.013	-.008	-.020	-.077	-.021	12	7
S21	.041	-.011	-.014	.023	.022	-.050	-.069	-.044	-.059	.020	.014	.043	.020	5	1

Table A9.6 Autocorrelation of the complete sets of simulated data

Network Spec.	Large lag SE	Autocorrelation function for first 12 lags												No. in first 50 outside LLSE	No. in first 50 outside LLSE x 2
S1	.058	-.063	-.070	.063	-.043	.027	.022	-.128	-.041	-.027	-.007	.048	.086	4	1
S2	.082	-.164	.041	-.005	-.135	.048	.135	-.078	.186	-.122	.043	-.004	-.066	9	2
S3	.100	-.088	.020	-.107	.140	-.015	.083	.043	-.090	-.028	.011	-.018	-.175	7	1
S4	.082	-.194	-.079	.058	-.109	.064	-.208	.013	.151	-.188	.049	-.037	.033	6	2
S5	.082	-.137	-.059	-.044	-.034	.025	-.023	.089	-.112	.128	.049	-.011	-.019	6	1
S6	.058	.050	.006	.030	-.070	-.025	-.049	-.141	-.061	-.064	-.110	.123	-.051	7	2
S7	.082	-.008	-.036	-.029	-.106	.116	.145	-.030	.000	-.057	.010	.013	.046	5	1
S8	.100	-.034	-.102	-.108	.251	.097	-.068	.004	-.042	-.035	-.012	-.040	-.109	4	2
S9	.082	.039	.024	-.018	-.036	.020	.078	.057	-.058	.042	.030	.017	-.098	3	1
S10	.081	.042	.020	-.114	.039	-.072	.047	-.106	-.024	.074	.113	.065	-.114	7	1
S11	.081	-.067	-.178	-.025	-.010	.007	-.122	.003	.044	-.071	-.084	.000	.125	9	1
S12	.083	-.038	.096	-.003	-.028	.059	-.123	-.064	-.042	-.045	-.039	-.016	.022	6	3
S13	.081	.176	.056	-.076	.011	.140	.011	.036	-.094	-.090	.060	.028	-.090	7	3
S14	.084	-.016	.065	-.045	.128	-.071	-.101	-.019	-.056	-.065	-.049	-.049	-.030	5	4
S15	.058	.050	.006	.030	-.070	-.025	-.049	-.141	-.061	-.064	-.110	.123	-.051	7	2
S16	.080	.047	-.144	-.044	.055	.042	-.039	-.125	.012	.007	.021	.080	-.071	5	1
S17	.095	-.018	.207	.131	-.016	.102	-.068	-.024	-.031	-.155	-.028	-.036	-.181	12	2
S18	.082	-.004	.052	.111	.192	.067	-.033	-.003	-.104	-.102	-.029	-.075	-.135	11	2
S19	.082	-.002	.008	.036	-.108	.097	.199	-.010	-.017	.052	-.020	-.021	-.113	5	2
S20	.082	-.093	-.015	.004	.065	.086	-.035	.053	.107	.021	-.094	.141	.008	7	1
S21	.080	-.112	.010	-.114	-.019	-.100	-.056	.015	.017	-.032	-.003	-.051	.117	9	1

Table A9.7 Autocorrelation function of the single customer sets of simulated data

Network Specification	Theoretical mean	Estimated mean	Percentage difference	Theoretical S.D.(exact)	Estimated S.D.	Percentage difference
S1	2.576	2.482	3.6	1.441	1.519	5.4
S2	4.245	4.051	4.6	1.962	2.045	4.3
S3	6.091	5.868	3.7	2.421	2.442	.9
S4	4.245	3.918	7.7	1.962	1.519	22.6
S5	4.245	4.051	4.6	1.962	2.529	28.9
S6	2.250	2.152	4.4	1.451	1.265	12.8
S7	4.045	3.912	3.3	2.035	1.826	10.3
S8	6.006	5.816	3.2	2.491	2.246	9.8
S9	4.048	3.958	2.2	2.033	1.823	10.3
S10	5.607	5.518	1.6	3.916	3.434	12.3
S11	4.045	3.896	3.7	2.035	1.393	31.5
S12	4.045	3.888	3.9	2.035	5.311	161.0
S13	4.045	3.912	3.3	2.035	2.979	46.4
S14	4.045	3.888	3.9	2.035	5.315	161.2
S15	2.250	2.152	4.4	1.451	1.265	12.8
S16	4.045	3.912	3.3	2.035	2.956	45.3
S17	6.006	5.800	3.4	2.491	5.706	129.1
S18	4.045	3.910	3.3	2.035	3.386	66.4
S19	4.246	4.023	5.2	1.995	1.844	7.6
S20	4.246	4.023	5.3	1.995	2.713	36.0
S21	4.246	3.955	6.8	1.995	2.625	31.6

Table A9.8 Comparison of exact theoretical and complete sample simulated results via mean and standard deviation.

Network Specification	Theoretical mean	Estimated mean	Percentage difference	Theoretical S.D. (PSA)	Estimated S.D.	Percentage difference
S1	2.576	2.482	3.6	1.520	1.519	.0
S2	4.245	4.051	4.6	2.187	2.045	6.5
S3	6.091	5.868	3.7	2.738	2.442	10.8
S4	4.245	3.918	7.7	2.187	1.519	30.5
S5	4.245	4.051	4.6	3.101	2.529	18.5
S6	2.250	2.152	4.4	1.488	1.265	15.0
S7	4.045	3.912	3.3	2.115	1.826	13.7
S8	6.006	5.816	3.2	2.574	2.246	12.7
S9	4.048	3.958	2.2	2.119	1.823	13.9
S10	5.607	5.518	1.6	4.241	3.434	19.0
S11	4.045	3.896	3.7	2.115	1.393	34.1
S12	4.045	3.888	3.9	2.115	5.311	151.1
S13	4.045	3.912	3.3	3.455	2.979	13.8
S14	4.045	3.888	3.9	2.115	5.315	151.3
S15	2.250	2.152	4.4	1.762	1.265	28.2
S16	4.045	3.912	3.3	3.453	2.956	14.4
S17	6.006	5.800	3.4	5.382	5.706	6.0
S18	4.045	3.910	3.3	2.115	3.386	60.1
S19	4.246	4.023	5.2	2.171	1.844	15.1
S20	4.246	4.023	5.3	3.025	2.713	10.3
S21	4.246	3.955	6.8	2.171	2.625	20.9

Table A9.9 Comparison of PSA theoretical and complete sample simulated results via mean and standard deviation.

Network Specification	Theoretical mean	Estimated mean	Percentage difference	Theoretical S.D.(Exact)	Estimated S.D.	Percentage difference
S1	2.576	2.482	3.6	1.441	1.539	6.8
S2	4.245	4.045	4.7	1.962	2.100	7.0
S3	6.091	5.872	3.6	2.421	2.307	4.7
S4	4.245	3.918	7.7	1.962	1.558	20.6
S5	4.245	4.104	3.3	1.962	2.528	28.9
S6	2.250	2.136	5.1	1.451	1.290	11.1
S7	4.045	3.923	3.0	2.035	1.864	8.4
S8	6.006	5.804	3.4	2.491	2.344	5.9
S9	4.048	3.966	2.0	2.033	1.803	11.3
S10	5.607	5.452	2.8	3.916	3.312	15.4
S11	4.045	3.878	4.1	2.035	1.377	32.3
S12	4.045	3.995	1.2	2.035	5.758	182.9
S13	4.045	3.796	6.1	2.035	2.601	27.8
S14	4.045	4.080	.9	2.035	5.581	174.2
S15	2.250	2.136	5.1	1.451	1.290	11.1
S16	4.045	3.738	7.6	2.035	3.210	57.7
S17	6.006	5.150	14.2	2.491	5.795	132.7
S18	4.045	3.900	3.6	2.035	3.168	55.7
S19	4.246	4.016	5.4	1.995	1.752	12.2
S20	4.246	4.031	5.1	1.995	2.571	28.9
S21	4.246	3.812	10.2	1.995	2.267	13.6

Table A9.10 Comparison of exact theoretical and single customer sample simulated results via mean and standard deviation.

Network Specification	Theoretical mean	Estimated mean	Percentage difference	Theoretical S.D.(PSA)	Estimated S.D.	Percentage difference
S1	2.576	2.482	3.6	1.520	1.539	1.3
S2	4.245	4.045	4.7	2.187	2.100	4.0
S3	6.091	5.872	3.6	2.738	2.307	15.7
S4	4.245	3.918	7.7	2.187	1.558	28.7
S5	4.245	4.104	3.3	3.101	2.528	18.5
S6	2.250	2.136	5.1	1.488	1.290	13.3
S7	4.045	3.923	3.0	2.115	1.864	11.9
S8	6.006	5.804	3.4	2.574	2.344	8.9
S9	4.048	3.966	2.0	2.119	1.803	14.9
S10	5.607	5.452	2.8	4.241	3.312	21.9
S11	4.045	3.878	4.1	2.115	1.377	34.9
S12	4.045	3.995	1.2	2.115	5.758	172.2
S13	4.045	3.796	6.1	3.455	2.601	24.7
S14	4.045	4.080	.9	2.115	5.581	163.8
S15	2.250	2.136	5.1	1.762	1.290	26.8
S16	4.045	3.738	7.6	3.453	3.210	7.0
S17	6.006	5.150	14.2	5.382	5.795	7.7
S18	4.045	3.900	3.6	2.115	3.168	49.8
S19	4.246	4.016	5.4	2.171	1.752	19.3
S20	4.246	4.031	5.1	3.025	2.571	15.0
S21	4.246	3.812	10.2	2.171	2.267	4.4

Table A9.11 Comparison of PSA theoretical and single customer sample simulated results via mean and standard deviation.

Network Specification	D value (exact method)	D value (PSA method)	Conf. level (exact method)	Conf. level (PSA method)
S1	.08036	.08581	.00	.00
S2	.09998	.08581	.00	.00
S3	.11845	.15212	.00	.00
S4	.11855	.13559	.00	.00
S5	.16197	.07527	.00	.00
S6	.05281	.08137	.05	.00
S7	.06709	.09987	.00	.00
S8	.12057	.13233	.00	.00
S9	.06777	.09836	.00	.00
S10	.10087	.11085	.00	.00
S11	.10658	.13598	.00	.00
S12	.29728	.15163	.00	.00
S13	.17562	.06085	.00	.01
S14	.29895	.16091	.00	.00
S15	.05281	.05842	.05	.01
S16	.17395	.06948	.00	.00
S17	.36875	.10873	.00	.00
S18	.23823	.12402	.00	.00
S19	.10447	.084 6	.00	.00
S20	.17058	.11642	.00	.00
S21	.22118	.16304	.00	.00

Table A9.12 KS test for the complete sample of simulated data with the theoretical methods.

Network Specification	D value (exact method)	D value (PSA method)	Conf. level (exact method)	Conf. level (PSA method)
S1	.08743	.09260	.01	.01
S2	.12498	.10656	.01	.05
S3	.13431	.14670	.05	.01
S4	.14187	.14225	.00	.00
S5	.18373	.07326	.00	.20
S6	.07812	.10668	.05	.00
S7	.07555	.10469	.20	.05
S8	.14531	.14049	.01	.01
S9	.07529	.10891	.20	.05
S10	.10732	.11489	.05	.01
S11	.12088	.15237	.01	.00
S12	.29962	.16406	.00	.00
S13	.18086	.07820	.00	.20
S14	.29174	.15889	.00	.00
S15	.07812	.05968	.05	.20
S16	.22276	.12655	.00	.01
S17	.46308	.20306	.00	.00
S18	.24684	.11757	.00	.01
S19	.12114	.10082	.01	.05
S20	.17558	.11122	.00	.01
S21	.22770	.17714	.00	.00

Table A9.13 KS test for the single customer sample of simulated data with the theoretical methods.

Network Specification	Mean	Estimated mean	Percentage error	Std. devn.	Estimated S.D.	Percentage difference
S2	4.245	4.331	2.0	1.962	1.998	1.9
S7	4.045	4.055	.3	2.035	2.052	.8

Table A9.14 Changes to table A9.2 resulting from finer mesh (exact method).

Network Specification	Mean	Estimated mean	Percentage error	Std. devn.	Estimated S.D.	Percentage difference
S1	2.576	2.542	1.3	1.520	1.508	.8
S2	4.245	4.292	1.1	2.187	2.216	1.3
S3	6.091	6.289	3.3	2.738	2.802	2.3
S5	4.245	4.055	4.5	3.101	2.889	6.8
S6	2.250	2.183	3.0	1.488	1.457	2.1
S7	4.045	4.034	.3	2.115	2.129	.6
S8	6.006	6.116	1.8	2.574	2.602	1.1
S9	4.048	4.037	.3	2.119	2.133	.7
S10	5.607	5.490	2.1	4.241	4.168	1.7
S13	4.045	3.718	8.1	3.455	3.064	11.3
S15	2.250	2.112	6.1	1.762	1.620	8.1
S16	4.045	3.730	7.8	3.453	3.064	11.3
S17	6.006	5.472	8.9	5.382	4.684	13.0
S19	4.246	4.320	1.7	2.171	2.211	1.8
S20	4.246	4.190	1.3	3.025	2.963	2.1

Table A9.15 Discrete form distribution for the PSA theoretical method (finer mesh)

Network Specification	Max. vert. difference	Time	Percentile	Mean value
S2	.05044	2.38	.12	4.245
S7	.01914	1.94	.11	4.045

Table A9.16 Changes to table A9,4 resulting from finer mesh (vertical)

Network Specification	Max. hor. difference	Time	Percentile	Mean value
S2	.48727	1.82	.04	4.245
S7	.22438	1.13	.02	4.045

Table A9.17 Changes to table A9.5 resulting from finer mesh (horizontal).

Network Specification	D value (exact method)	D value (PSA method)	Conf. level (exact method)	Conf. level (PSA method)
S1	.08036	.07110	.00	.00
S2	.10687	.07603	.00	.00
S3	.11845	.13273	.00	.00
S4	.11064	.11928	.00	.00
S5	.17925	.06177	.00	.01
S6	.05281	.04518	.05	.15
S7	.06442	.06813	.01	.00
S8	.12057	.09994	.00	.00
S9	.06777	.07087	.00	.00
S10	.10087	.10457	.00	.00
S11	.10394	.10969	.00	.00
S12	.30462	.13725	.00	.00
S13	.18296	.07087	.00	.00
S14	.30629	.14040	.00	.00
S15	.05281	.07616	.05	.00
S16	.18129	.07392	.00	.00
S17	.36875	.07957	.00	.00
S18	.24081	.10841	.00	.00
S19	.10447	.10903	.00	.00
S20	.17058	.08815	.00	.00
S21	.22118	.12757	.00	.00

Table A9.18 KS test for the complete sample of simulated data with the theoretical methods (finer mesh)

Network Specification	D value (exact method)	D value (PSA method)	Conf. level (exact method)	Conf. level (PSA method)
S1	.08743	.07782	.01	.05
S2	.13258	.09770	.01	.10
S3	.13431	.12702	.05	.05
S4	.13726	.12657	.00	.01
S5	.19691	.05332	.00	.20
S6	.07812	.06470	.05	.15
S7	.07627	.06868	.20	.20
S8	.14531	.11160	.01	.15
S9	.07529	.07619	.20	.20
S10	.10732	.10861	.05	.05
S11	.11807	.12253	.01	.01
S12	.30573	.14828	.00	.00
S13	.18705	.07691	.00	.20
S14	.29908	.13838	.00	.00
S15	.07812	.07521	.05	.05
S16	.22758	.10782	.00	.05
S17	.46308	.17464	.00	.00
S18	.25166	.12076	.00	.01
S19	.12114	.12569	.01	.01
S20	.17558	.08638	.00	.20
S21	.22770	.14167	.00	.00

Table A9.19 KS test for the single customer sample of simulated data with the theoretical methods (finer mesh)

APPENDIX 10

The following plots were produced to demonstrate the characteristics of predicted cycle time distribution for a selection of the network specifications given in Table A8.1. The predictions were made by various combinations of the simulation, exact and PSA theoretical models as follows:

Plots (i) - (ii)

Networks S2 and S7: Laplace transform of cycle time probability distribution computed by the exact and PSA theoretical methods.

Plots (iii) - (vii)

Networks S2, S7, S10, S17, S19: Cumulative discrete form cycle time probability distribution computed by the exact and PSA theoretical methods and the simulator.

Plot (viii)

Discrete cycle time probability distribution for networks S6, S7 and S8 computed by the exact theoretical method. This shows the change in the form of the distribution in network configuration (ii) as the number of customers increases from 2 to 4 and 6.

Plot (ix)

Cumulative discrete form cycle time probability distribution computed by the PSA method for networks S8 and S17. This shows the change in the form of the distribution for network configuration (ii) with 6 customers when the queueing discipline at the root centre is changed from FCFS to PS.

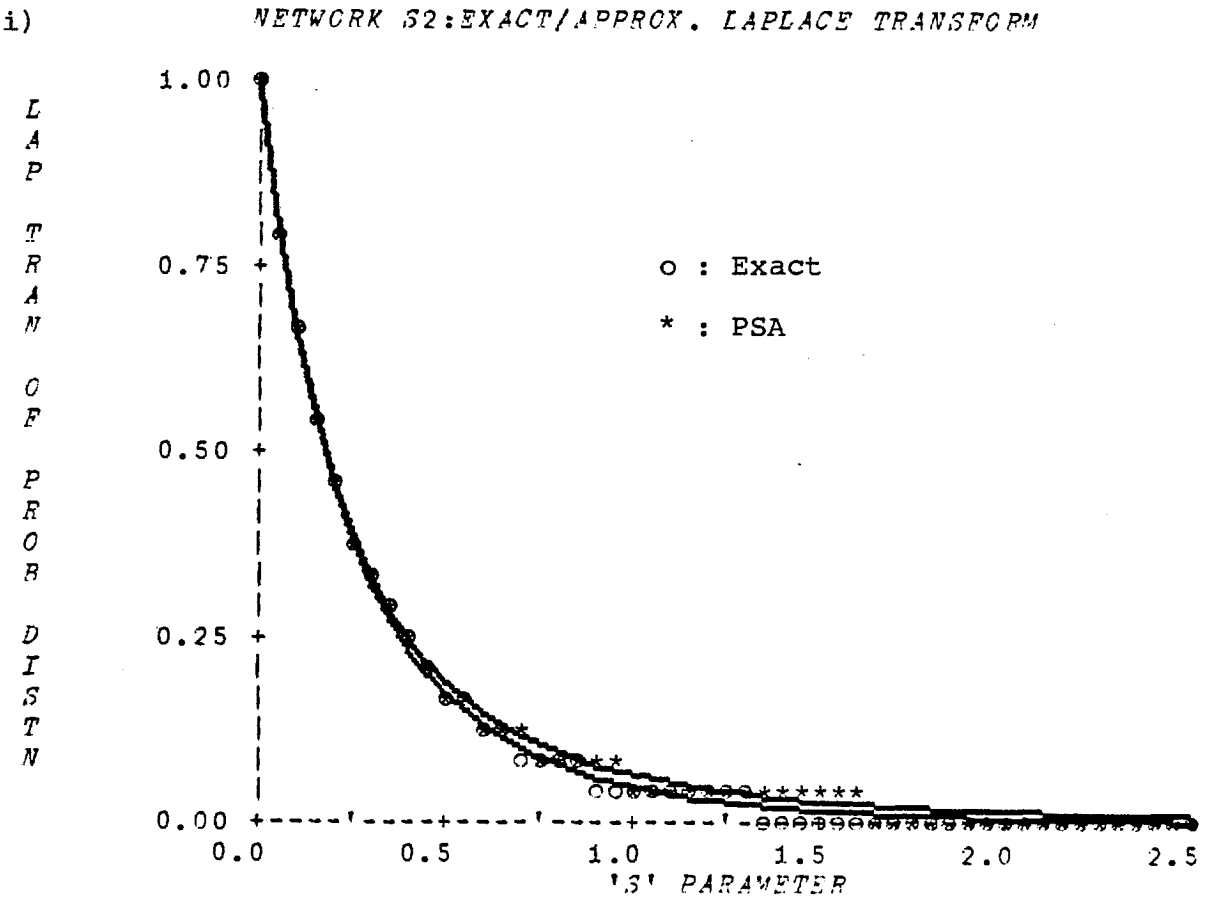
Plot (x)

Cumulative discrete form cycle time probability distribution computed by the exact method for network S7 showing the effect of decreasing the mesh size.

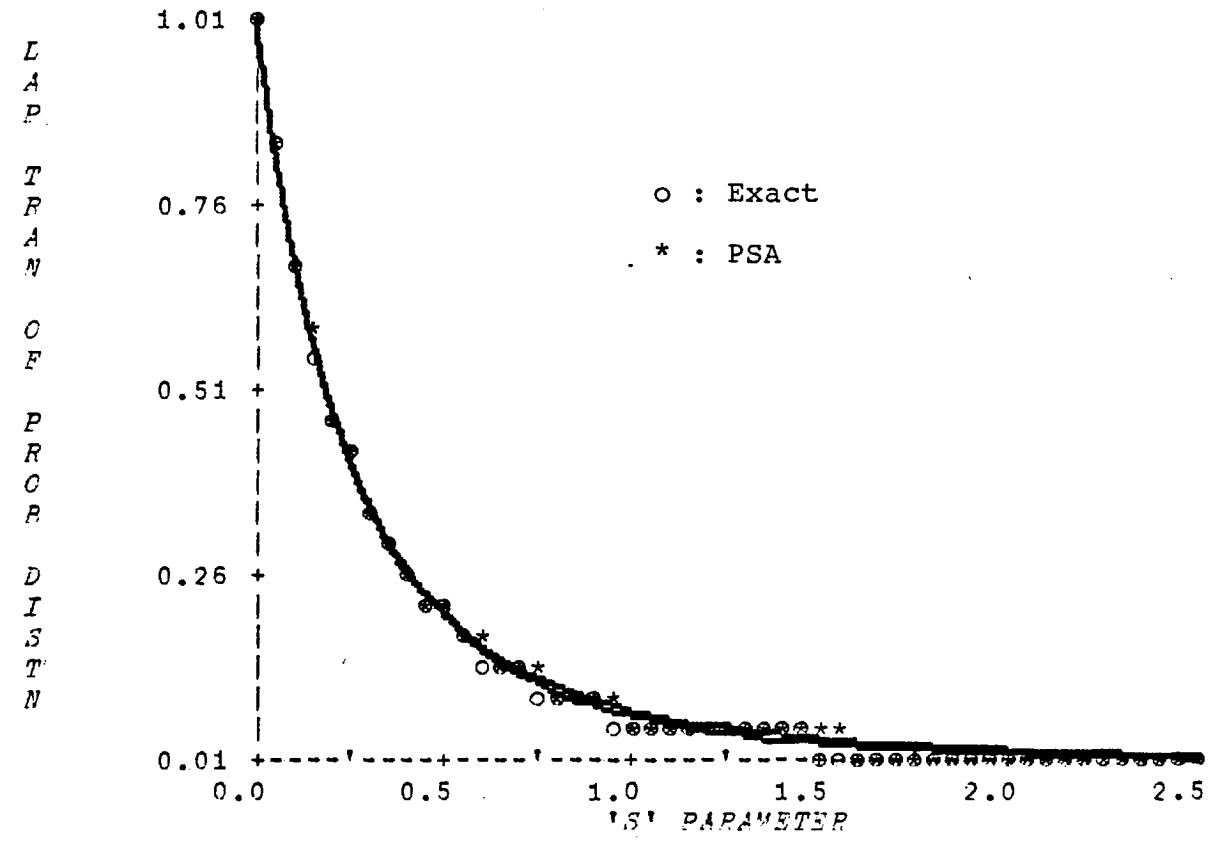
Plot (xi)

Cumulative discrete form cycle time probability distribution computed by the PSA method for network S17 showing the effect of decreasing the mesh size.

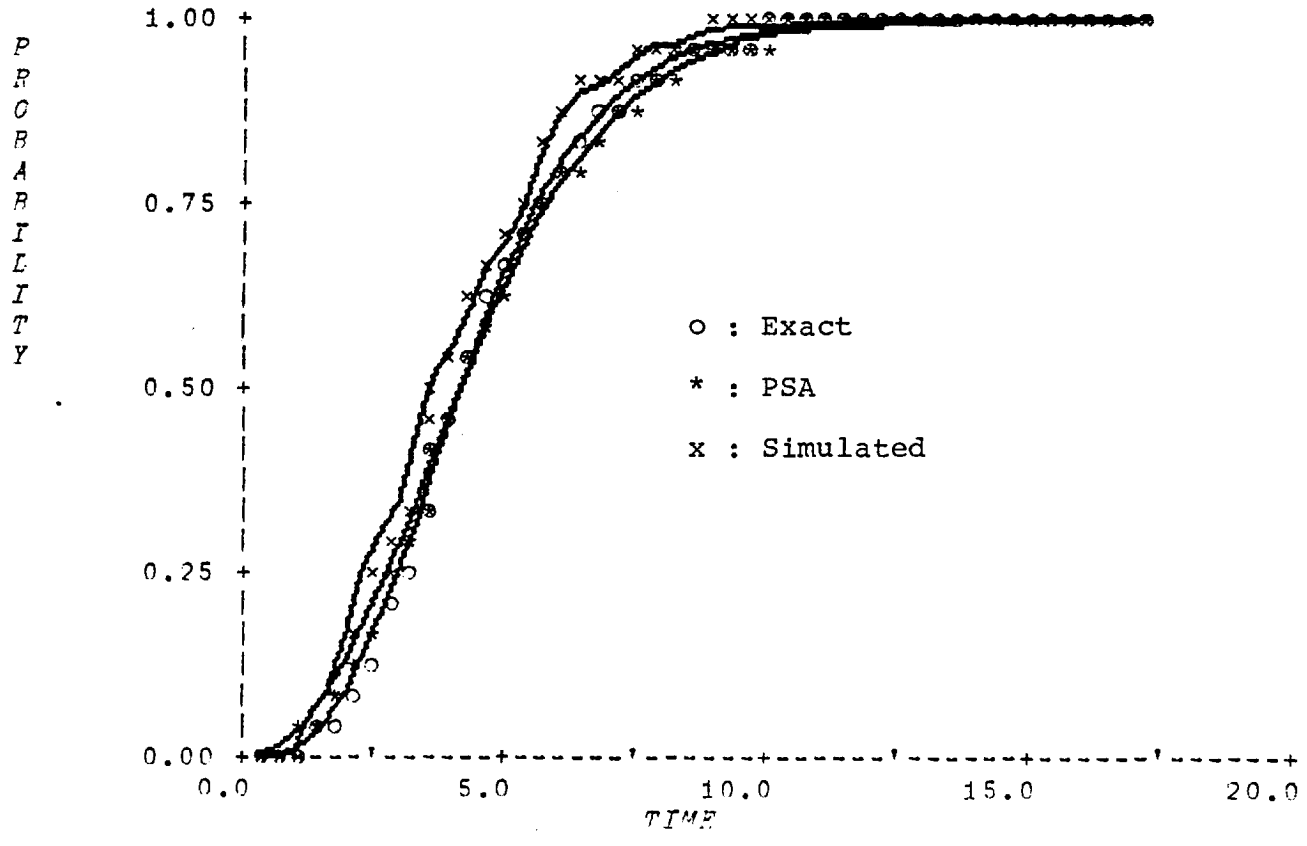
(i)



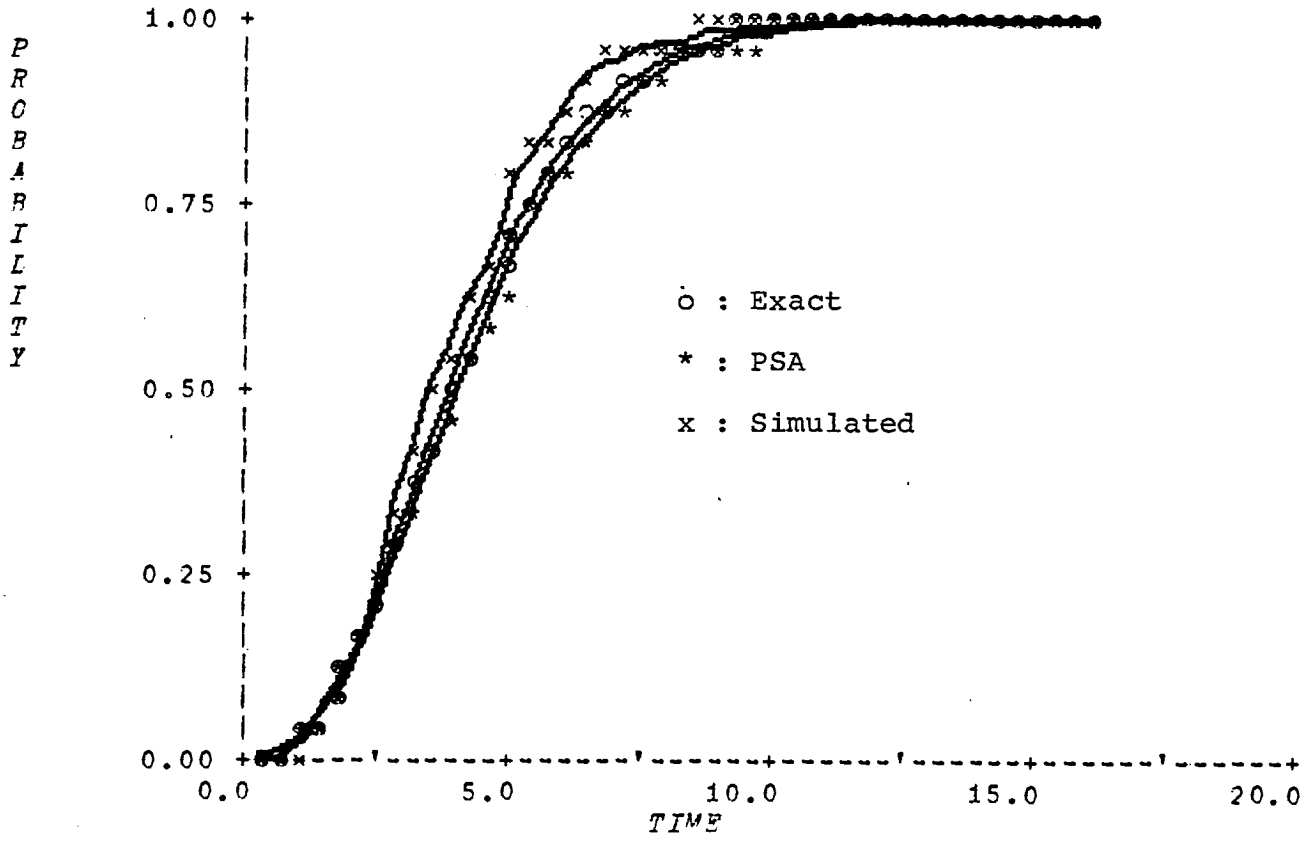
(ii)


NETWORK S7:EXACT/APPROX. LAPLACE TRANSFORM
LAP TRAN OF PROB DISTN
1.01
0.76
0.51
0.26
0.01
0.0
0.5
1.0
1.5
2.0
2.5
'S' PARAMETER
o : Exact
* : PSA

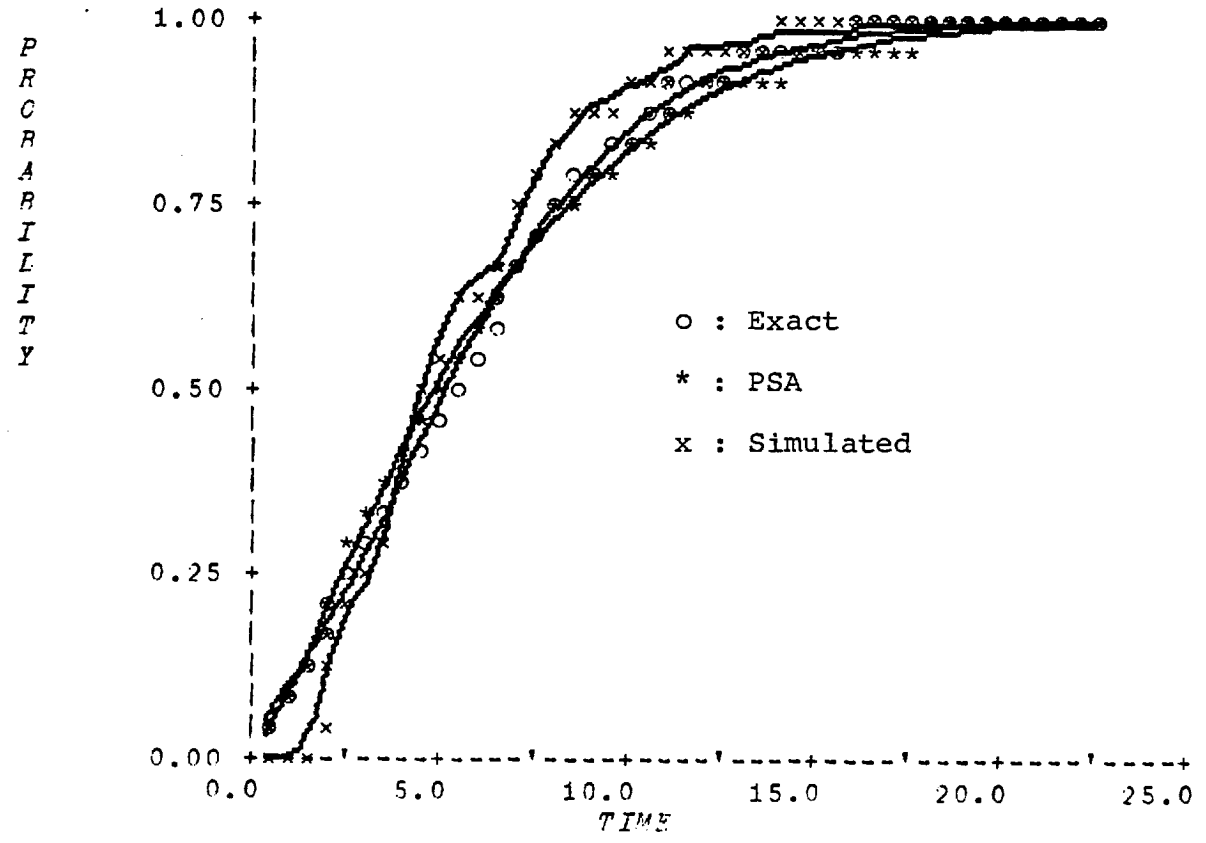
(iii)


NETWORK S2:EXACT/APPROX./SIMULATED CUM. HISTOGRAM
PROBABILITY
1.00
0.75
0.50
0.25
0.00
0.0
5.0
10.0
15.0
20.0
TIME
o : Exact
* : PSA
x : Simulated

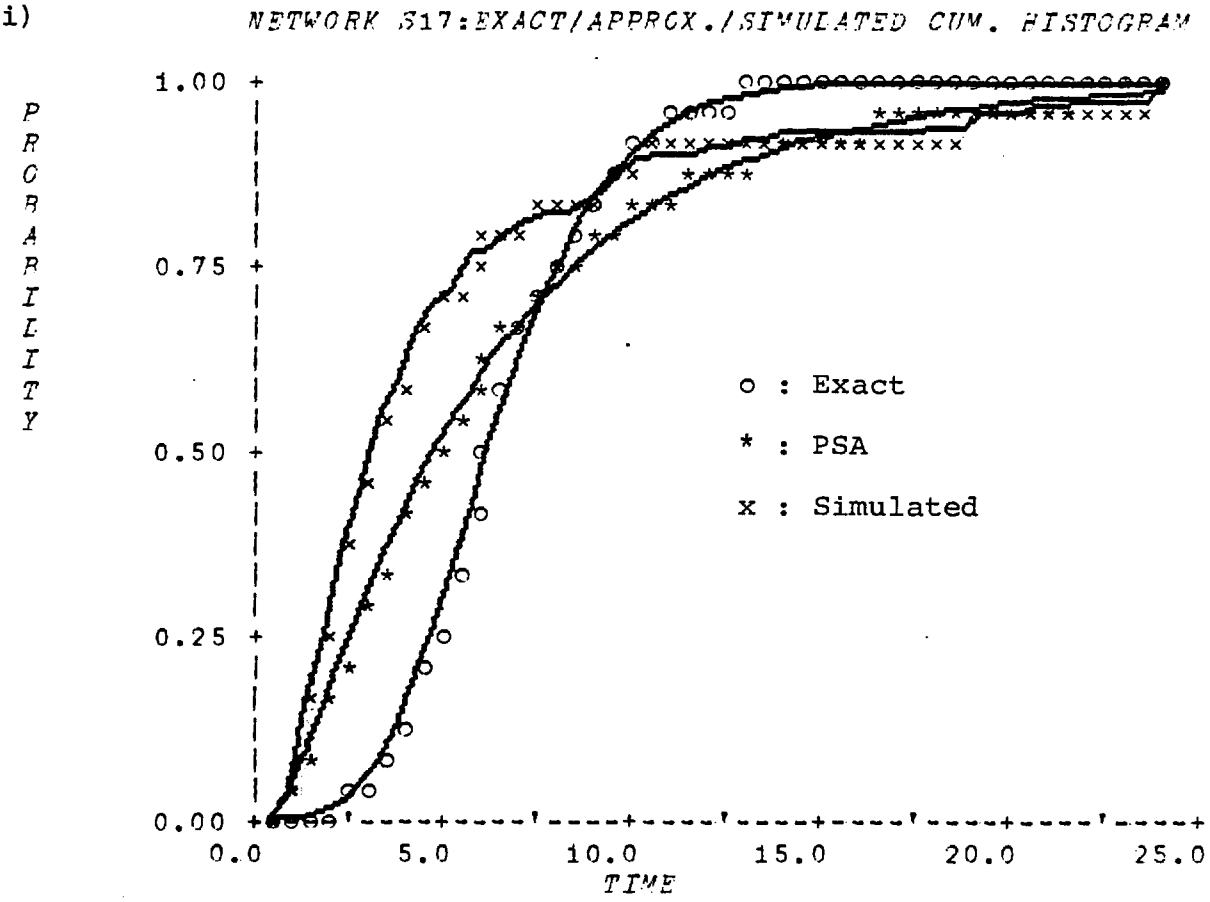
(iv)


NETWORK S7:EXACT/APPROX./SIMULATED CUM. HISTOGRAM
PROBABILITY
1.00
0.75
0.50
0.25
0.00
0.0
5.0
10.0
15.0
20.0
TIME
o : Exact
* : PSA
x : Simulated

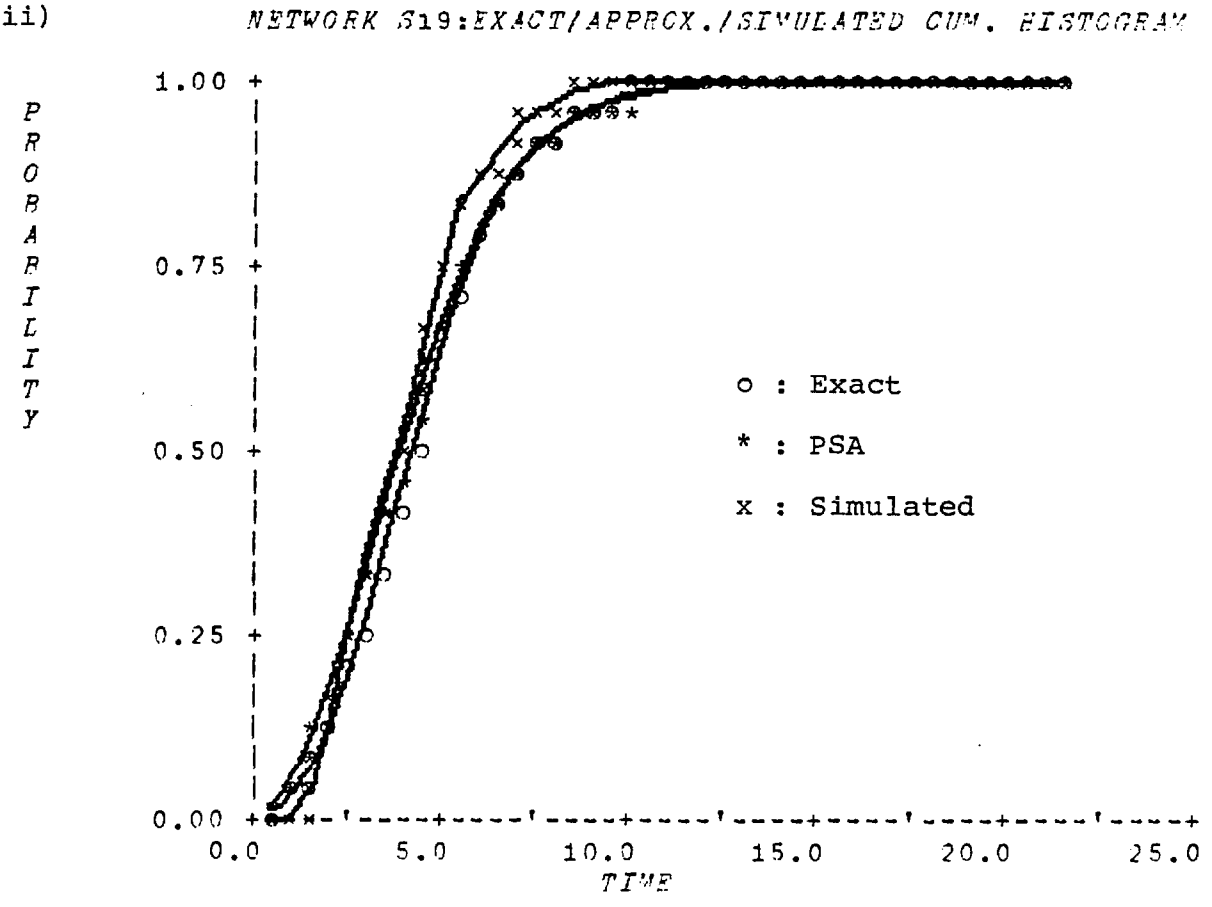
(v)


NETWORK S10:EXACT/APPROX./SIMULATED CUM. HISTOGRAM
PROBABILITY
1.00
0.75
0.50
0.25
0.00
0.0
5.0
10.0
15.0
20.0
25.0
TIME
o : Exact
* : PSA
x : Simulated

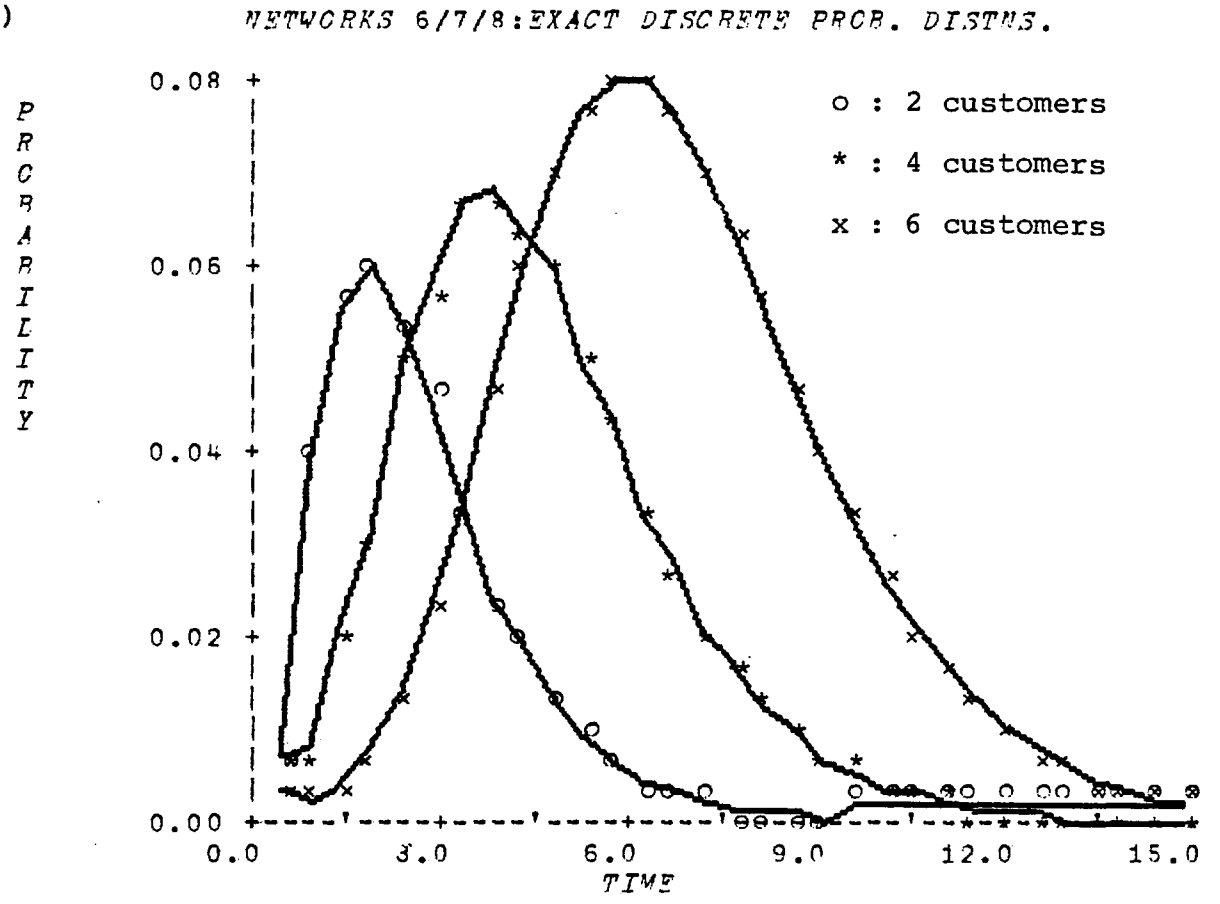
(vi)


NETWORK S17:EXACT/APPROX./SIMULATED CUM. HISTOGRAM
PROBABILITY
1.00
0.75
0.50
0.25
0.00
0.0
5.0
10.0
15.0
20.0
25.0
TIME
o : Exact
* : PSA
x : Simulated

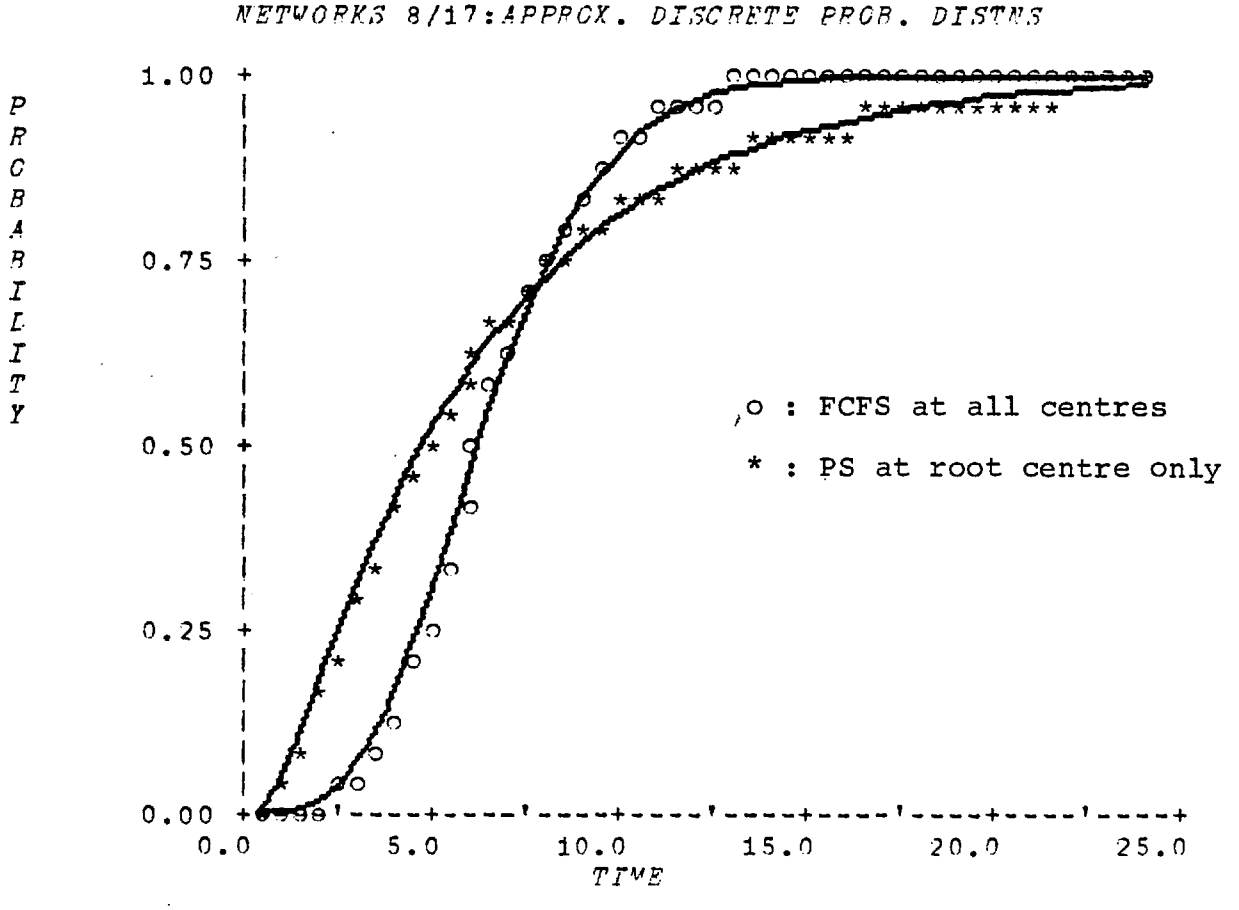
(vii)


NETWORK S19:EXACT/APPROX./SIMULATED CUM. HISTOGRAM
PROBABILITY
1.00
0.75
0.50
0.25
0.00
0.0
5.0
10.0
15.0
20.0
25.0
TIME
o : Exact
* : PSA
x : Simulated

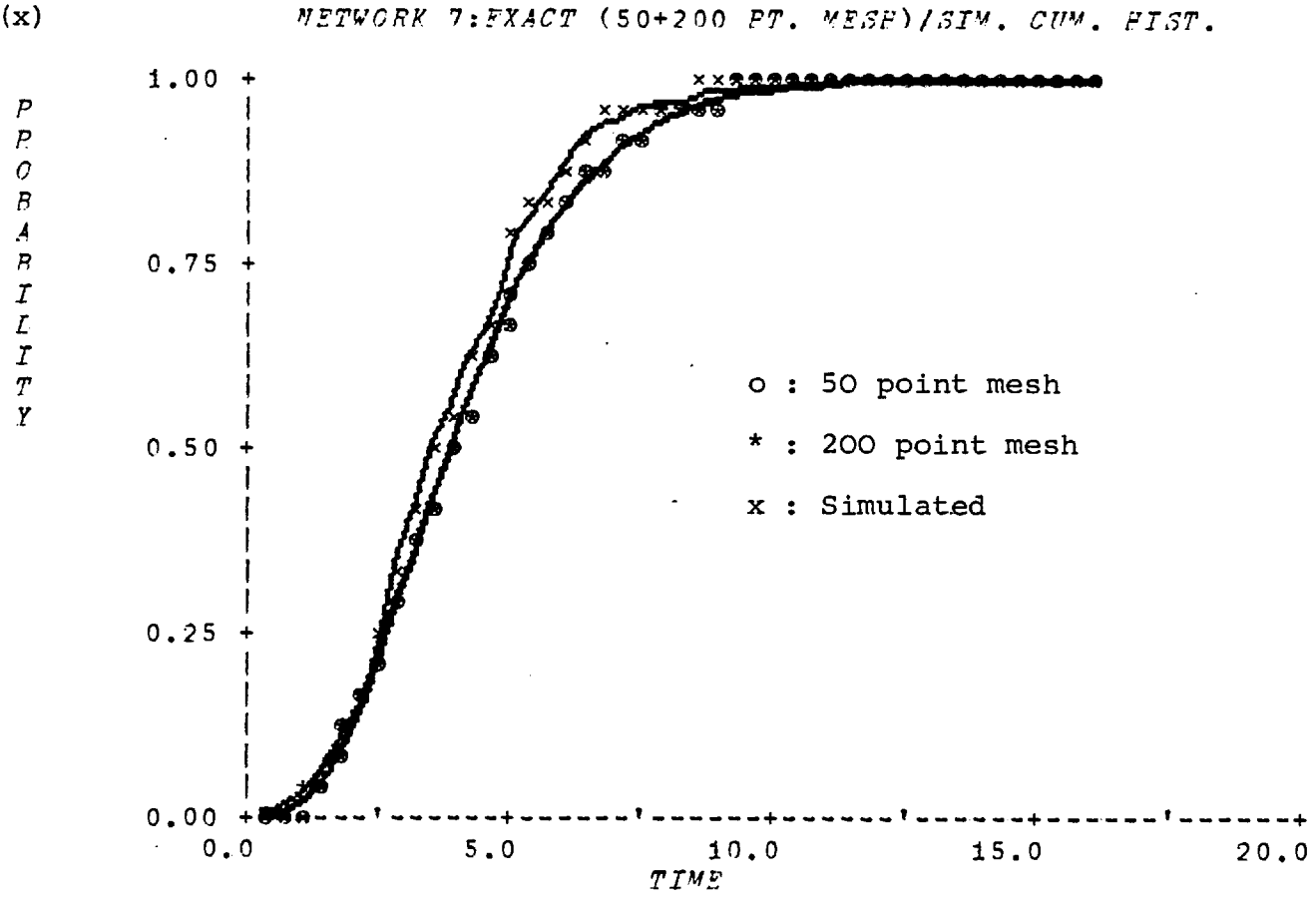
(viii)


NETWORKS 6/7/8:EXACT DISCRETE PROB. DISTNS.
PROBABILITY
0.08
0.06
0.04
0.02
0.00
0.0
3.0
6.0
9.0
12.0
15.0
TIME
o : 2 customers
* : 4 customers
x : 6 customers

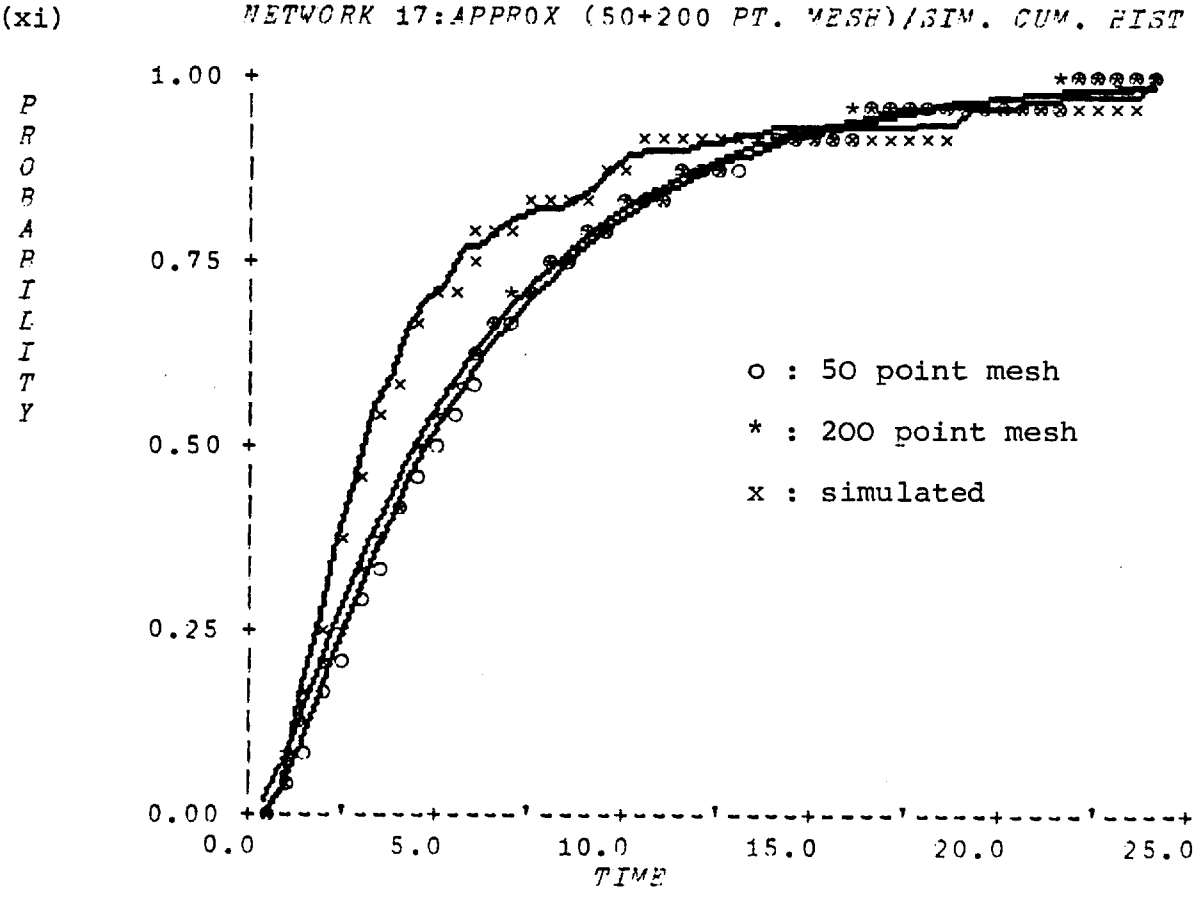
(ix)


NETWORKS 8/17:APPROX. DISCRETE PROB. DISTNS
PROBABILITY
1.00
0.75
0.50
0.25
0.00
0.0
5.0
10.0
15.0
20.0
25.0
TIME
o : FCFS at all centres
* : PS at root centre only

(x)


NETWORK 7:EXACT (50+200 PT. MESH)/SIM. CUM. DIST.
PROBABILITY
1.00
0.75
0.50
0.25
0.00
0.0
5.0
10.0
15.0
20.0
TIME
o : 50 point mesh
* : 200 point mesh
x : Simulated

(xi)


NETWORK 17:APPROX (50+200 PT. MESH)/SIM. CUM. DIST
PROBABILITY
1.00
0.75
0.50
0.25
0.00
0.0
5.0
10.0
15.0
20.0
25.0
TIME
o : 50 point mesh
* : 200 point mesh
x : simulated

APPENDIX 11

In a two-centre cyclic network with population N, the probability distribution of the queue length faced at the second centre conditional on that faced at the first by the test customer is derived, to a first order approximation, by an analysis in continuous time.

Let the pair of queue lengths faced be given by the random variables q_1, q_2 and suppose the test customer arrives at the first centre at time t = 0. Then it is required to find $P(q_2|q_1)$. Denote the state space of the network by S and let state $\underline{k} = (k_1, k_2) \in S$ have time dependent probability $P(k_2,t)$ and equilibrium (time independent) probability $\Theta(k_2) = \lim_{t\to\infty} P(k_2,t)$. It is not necessary to specify k_1 in the arguments since $k_1+k_2=N$. Now

$$P(q_2|q_1) = \int_0^\infty P\{q_2,t|q_1\text{'th centre 1 departure occurs in time interval } (t,t+dt)\}$$

$$\times \Pr\{q_1\text{'th centre 1 departure occurs in time interval } (t,t+dt)\}$$

The approximation is now made that the random variables, q_1 and q_2, for the queue lengths faced by the test customer are assumed independent. The Markov property is also assumed so that

$$P(q_2|q_1) = \int_0^\infty \underbrace{P(q_2,t|1 \le q_2 \le N)}_{\text{Prob. distn. of queue lengths at time t.}} \underbrace{\frac{e^{-\mu_1 t}(\mu_1 t)^{q_1-1}}{(q_1-1)!}}_{\text{Prob. of } q_1-1 \text{ departures from centre 1 in } (0,t) \text{ (Poisson distn.)}} \underbrace{\mu_1 dt}_{\text{Prob. of departure from centre 1 in } (t,t+dt) \text{ of test customer.}}$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{Erlang } (q_1) \text{ distn.}}$$

where μ_1, μ_2 are the centre service rates, assumed constant, and $q_1, q_2 > 0$. Using the first order, non-normalised approximation

to the solution of the Kolmogorov equations,

$$P(q,t) = e^{-\lambda(q)t}\{P(q,0) - \theta(q)\} + \theta(q)$$

where $\lambda(q)$ is the total service rate when there are q customers at the second centre (N-q at the first).

$$\therefore \quad P(q,t \mid 1 \le q \le N) = \frac{P(q,t)}{\sum_{k=1}^{N} P(k,t)}$$

so that

$$P(q_2|q_1) = \int_0^\infty \frac{\left\{\theta(q_2)e^{-\mu_1 t} + \{P(q_2,0)-\theta(q_2)\}e^{-\{\mu_1+\lambda(q_2)\}t}\right\}(\mu_1 t)^{q_1-1}\mu_1 dt}{\sum_{k=1}^{N}\{\theta(k)+[P(k,0)-\theta(k)]e^{-\lambda(k)t}\}(q_1-1)!}$$

Thus $P(q_2|q_1)$ may be computed numerically and the PSA approximation (chapter 3) for the cycle time distribution improved using $P(\underline{q}) = P(q_2|q_1)\theta'(N-q_1)$ for the joint probability distribution of the queue lengths faced by the test customer. $\theta'(k)$ is the equilibrium probability of state (N-k,k) seen by the test customer on arrival at the first centre, [MITR79], c.f. section 4.3.

The improvement arises because it is no longer necessary to assume independence of the queue lengths faced by the test customer, c.f. section 8.3.3.

For large N,

$$P(q_2|q_1) \propto \int_0^\infty \{P(q_2,0)-\theta(q_2)\}e^{-\{\mu_1+\lambda(q_2)\}t}\frac{(\mu_1 t)^{q_1-1}}{(q_1-1)!}\mu_1 dt + \theta(q_2)$$

$$= \{P(q_2,0) - \theta(q_2)\}\left\{\frac{1}{1+\frac{\lambda(q_2)}{\mu_1}}\right\}^{q_1} + \theta(q_2)$$

BIBLIOGRAPHY

[ACM78] ACM Computing Surveys 10, 3
'Queueing Network Models of Computer System Performance'

*[BARB76] Barbour, A.D.,
'Networks of queues and the method of stages', Adv. appl. prob. 8, 584-591

[BARD79] Bard, Y.,
'Some extensions to multiclass queueing network analysis',
4th Int. Symp. on Modelling and Performance Evaluation of Computer Systems, Vienna, February 1979

[BART46] Bartlett, M.S.,
'On the theoretical specification of sampling properties of autocorrelated time series',
J. Royal Stat. Soc., B8, 27, 1946

[BASK75] Baskett, F., Chandy, K.M., Muntz, R.R., Palacios, F.G.,
'Open, Closed, and Mixed Networks of Queues with Different Classes of Customers',
JACM 22, 2

*[BEIL76] Beilner, H., Kritzinger, P.S.,
'A computer installation management methodology',
Research report 76/7, Dept. of Computing & Control, Imperial College, London

[BOUH79] Bouhana, J.,
'Homogeneous approximation of general queueing networks',
4th Int. Symp. on Modelling and Performance Evaluation of Computer Systems, Vienna, February 1979

[BUZE73] Buzen, J.P.,
'Computational algorithms for closed queueing networks with exponential servers',
CACM 16, 9

[BUZE78a] Buzen, J.P.,
'Operational analysis: an alternative to stochastic modelling',
in Proc. Int. Conf. Performance Computer Installations, North Holland

[BUZE78b] Buzen, J.P.,
'A queueing network model of MVS',
Comput. Surv. 10, 3

[CAVE78] Cavers, J.K.,
'On the fast Fourier transform inversion of probability generating functions',
J. Inst. Maths. Applics 22, 275-282

* [CHAN75a] Chandy, K.M., Herzog, U., Woo, L.,
'Parametric analysis of queueing networks',
IBM J. Res. Develop. Jan. 1975

[CHAN75b] Chandy, K.M., Herzog, U., Woo, L.,
'Approximate analysis of general queueing networks',
IBM J. Res. Develop. Jan. 1975

[CHAN77] Chandy, K.M., Howard, J.H., Towsley, D.F.,
'Product form and local balance in queueing networks',
J. ACM 24, 2

* [CHAN78] Chandy, K.M., Sauer, C.H.,
'Approximate methods for analyzing queueing network models of computer systems',
Comput. Surv. 10, 3

[CHAT75] Chatfield, C.,
The Analysis of Time Series: Theory and Practice,
Chapman and Hall

[CHOW77a] Chow, W.,
'The cycle time distribution of exponential cyclic queues',
IBM Res. Rep. RC 6484

[CHOW77b] Chow, W.,
'The cycle time distribution of exponential central server queues',
IBM Res. Rep. RC 6765

[COUR75] Courtois, P.J.,
'Decomposability, Instabilities, and Saturation in Multiprogramming systems',
CACM 18, 7

[COUR77] Courtois, P.J.,
Decomposability,
Academic Press, New York

[COX55] Cox, D.R.,
'A use of complex probabilities in the theory of stochastic processes',
Proc. Cambridge Phil. Soc. 51, 313-319

[DENN77] Denning, P.J., Buzen, J.P.,
'Operational Analysis of Queueing Networks',
Proc. 3rd Int. Symp. on Measuring, Modelling and Evaluating Computer Systems,
North Holland.

[DENN78] Denning, P.J., Buzen, J.P.,
'The operational analysis of queueing network models',
Comput. Surv. 10, 3

[FAYO79] Fayolle, G.,
'Solutions of functional equations arising in the analysis of two server queueing models',
4th Int. Symp. on Modelling and Performance Evaluation of Computer Systems, Vienna, February 1979

[FELL62] Feller, W.,
An Introduction to Probability Theory and its Applications, Vol I,
John Wiley

* [FELL66] Feller, W.,
An Introduction to Probability Theory and its Applications, Vol II,
John Wiley

[GELE75] Gelenbe, E.,
'On approximate computer system models',
JACM, 22, 2

*[GELE76a] Gelenbe, E., Muntz, R.R.,
Probabilistic Models of Computer Systems - Part I (Exact Results),
Acta Informatica 7

*[GELE76b] Gelenbe, E., Pujolle, G.,
The Behaviour of a Single Queue in a General Queueing Network,
Acta Informatica 7

[GORD67] Gordon, W.J., Newell, G.F.,
'Closed queueing systems with exponential servers',
Oper. Res. 15, 254-265

[GRAS77a] Grassmann, W.K.,
'Transient solutions in Markovian queues',
European Journal or Operational Research 1

[GRAS77b] Grassmann, W.K.,
'Transient solutions in Markovian queueing systems',
Comput. & ops. res. 4, 1

[HARR74] Harrison, P.G.,
'Alternatives to the assignment statement',
MSc Thesis, Dept. of Computing & Control,
Imperial College, London.

[HARR78a] Harrison, P.G.,

'Cycle time distribution in queueing network models of multiprogramming computer systems',
Research report 78/23, Dept. of Computing & Control, Imperial College, London.

[HARR78b] Harrison, P.G.,

'Structured Modelling of a Multiprogramming Computer System Configuration using Queueing Network Analysis',
Research report 78/24, Dept. of Computing & Control, Imperial College, London.

[HARR79a] Harrison, P.G.,

'An exact derivation of cycle time distribution in cyclic queueing networks',
Research report 79/15, Dept. of Computing & Control, Imperial College, London.

*[HARR79b] Harrison, P.G.,

'Cycle times in tree-like queueing networks',
Research report 79/16, Dept. of Computing & Control, Imperial College, London.

[JACK63] Jackson, J.R.,

'Jobshop-like Queueing Systems',
Man. Sci. 10, 1

[JENK68] Jenkins, G.M., Watts, D.G.,

Spectral Analysis and its Applications,
Holden-Day

[KELL75] Kelly, F.P.,

'Networks of queues with customers of different types',
J. Appl. Prob. 12

*[KELL76] Kelly, F.P.,

' Networks of queues',
Adv. Appl. Prob. 8

[KIEN79] Kienzle, M.G., Sevcik, K.C.,

'A methodical approach to modelling of computer systems',
4th Int. Symp. on Modelling and Performance Evaluation of Computer Systems, Vienna, February 1979

[KING69] Kingman, J.F.C.,

'Markov Population Processes',
J. Appl. Prob. 6, 1-18

[KLEI75] Kleinrock, L.,

Queueing Systems I,
John Wiley

* [KLEI76] Kleinrock, L.,
Queueing Systems II,
John Wiley

[KOBA74a] Kobayashi, H.,
'Application of the diffusion approximation to queueing networks. I: Equilibrium queue distributions',
J. ACM 21, 2

[KOBA74b] Kobayashi, H.,
'Application of the diffusion approximation to queueing networks. II: Nonequilibrium distributions and applications to computer modelling',
J. ACM 21, 3

* [KRIT77] Kritzinger, P.S., Krzesinski, A.E., Teunissen, P.,
'Performance prediction of a large operational computer system using a multiclass queueing network model',
Report No. RW77-03, Dept. of Computer Science, University of Stellenbosch, South Africa.

[KRIT78] Kritzinger, P.S., Krzesinski, A.E., Teunissen, P.,
'Design of a control system for a timesharing computer system',
Proc. Int. Conf. on the Performance of Computer Installations, ICPCI 78

[KRZE77a] Krzesinski, A.E., Teunissen, P.,
'A multiclass network model of a multiprogramming timesharing computer system',
Proc. IFIP Conf., Toronto, August 1977, pp.481-486

[KRZE77b] Krzesinski, A.E., Teunissen, P.,
'An analysis of the page size problem using a network analyzer',
Proc. 3rd Int. Symp. on Measuring, Modelling, and Evaluating Computer Systems, North Holland.

* [LAM77] Lam, S.S.,
'Queueing networks with population size constraints',
IBM J. Res. Develop. July 1977

[LAZO77a] Lazowska, E.D.,
The use of percentiles in modelling CPU service time distributions ,
in Chandy, K.M., Reiser, M., eds., *Computer Performance*, North Holland, 1977

*[LAZO77b] Lazowska, E.D.,
'Characterising service time and response time distributions in queueing network models of computer systems',
PhD Thesis & Tech. Rep. 85, Computer Systems Research Group, Univ. Toronto.

[LAZO78] Lazowska, E.D., Sevcik, K.C.,

'Approximating response time distributions in queueing networks',
Computer Systems Research Group, University of Toronto.

[LAZO79] Lazowska, E.D., Addison, C.A.,

'Selecting parameter values for servers of the phase type',
4th Int. Symp. on Modelling and Performance Evaluation of Computer Systems, Vienna, February 1979

[LEHM79a] Lehman, M.M., Principal Investigator,

'Dynamics of Evolution of Computing Systems',
E.R.O. Research Proposal, Dept. of Computing & Control, Imperial College, London.

[LEHM79b] Lehman, M.M.,

'Laws, evolution and conservation in the large-program life-cycle',
Journal of Systems and Software, No.3, to appear.

*[LIPS77] Lipsky, L., Church, J.D.,

'Applications of a queueing network model for a computer system',
Comput. Surv. 9 (3)

Mecklenburg, A.W.
[MECK78] 'The evaluation of computer performance by means of state-dependent queueing network models',
Ph.D Thesis, Dept. of Computing & Control,
Imperial College, London.

[MICH67] Michie, D.,

'Memo functions : A language facility with rote learning properties',
Research memorandum MIP-R29, Dept. of Machine Intelligence and Perception, University of Edinburgh.

[MITR79] Mitrani, I., Sevcik, K.C.,

'The distribution of queueing network states at input and output instants',
4th Int. Symp. on Modelling and Performance Evaluation of Computer Systems, Vienna, February 1979

[PUJO79] Pujolle, G.,

'Analytic methods for multiprocessor system modelling',
4th Int. Symp. Mod. Perf. Eval. Comp. Sys., Vienna,1979

[REIS79] Reiser, M.,

'Mean value analysis of queueing networks, a new look at an old problem',
4th Int. Symp. Mod. Perf. Eval. Comp. Sys., Vienna, 1979

*[ROSE78] Rose, C.A.,

'A measurement procedure for queueing network models of computer systems',
ACM Comput. Surv. 10, 3

[SAUE75] Sauer, C.H., Chandy, K.M.,
'Approximate analysis of central server models',
IBM J. Res. Develop. 19, 3

*[SCHW78] Schwetman, H.D.,
'Hybrid simulation models of computer systems',
Commun. ACM 21, 9

[SHED79] Shedler, G.S.,
'Regenerative simulation of response times in networks of queues, III. Passage through subnetworks',
IBM Res. Rep. RJ 2466

[SHUM77] Shum, A.W., Buzen, J.P.,
'A method for obtaining approximate solutions to closed queueing networks with general service times',
Proc. 3rd Int. Symp. on Measuring, Modelling and Evaluating Computer Systems, North Holland.

[SIEG56] Siegel, S.,
Nonparametric Statistics for the Behavioural Sciences,
McGraw-Hill

[SPAI70] Spain, B., Smith, M.G.,
Functions of Mathematical Physics,
Van Nostrand

[WONG78a] Wong, J.W.,
'Distribution of end-to-end delay in message-switched networks',
Comput. Networks, 2, 1

[WONG78b] Wong, J.W.,
'Queueing network modelling of computer communication networks',
Comput. Surv. 10, 3

[YU77] Yu, P.S.,
'Passage time distributions for a class of queueing networks: Closed, open or mixed, with different classes of customers with applications to computer system modelling',
Tech. report no. 135, Dept. of Electrical Engineering and Computer Science, Stanford University.

---

* Not referenced specifically in the main text.