THE UNIVERSITY OF NEWCASTLE UPON TYNE

DEPARTMENT OF COMPUTING SCIENCE

Performance and Reliability in Distributed Systems

by

Nigel Anthony Thomas

PhD Thesis

March 1997

Abstract

This thesis is devoted to the construction and analysis of models which can be used to evaluate the performance and reliability of distributed systems. The general object of the research therefore is to extend the types of queueing models with breakdowns which have been solved, with a particular interest in networking structures.

The systems that are studied involve various collections of servers and their associated queues. These range from isolated nodes, though parallel nodes coupled by the effect of breakdowns on arrivals, to pipelines of such parallel stages and more general networks. The issues that are explored include the influence of breakdowns and repairs on delays, job losses and optimal routeing. Obtaining performance measures for interacting queues is difficult, however a degree of abstraction has been used here which allows long run averages to be calculated (exactly in many cases) for quite complex systems. A variety of different techniques are used in order to obtain solutions to these models, including exact equations, exact numerical and approximate numerical techniques.

Acknowledgements

I would like to thank my supervisor Prof. Isi Mitrani, without his ideas this thesis would never have been started and without his encouragement never finished. Several other people at Newcastle University have also been of considerable help to me, of these special mention must go to Dr. Paul Ezhilchelvan for advice about relating this work to real world scenarios, and to Dr. Ram Chakka (now at Imperial College, London) for his patient explanations of the spectral expansion method.

This work was carried out under an EPSRC CASE Studentship in conjunction with BT Laboratories, Martlesham. I would like to thank Dr. Peter Key and his team at BTL for making me most welcome when I visited and for trying hard to understand what it was I was doing.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

Computer and communications systems play a vital role in everyday life, from aircraft navigation to the supermarket checkout we rely on them. The consequences of system failure can be serious, either in terms of safety, financial cost or just plain inconvenience to the user. Therefore a great amount of effort goes in to making these systems reliable, both in terms of hardware and software. Clearly some means of predicting performance and reliability would greatly assist in the design and management of these systems.

In traditional engineering systems, like construction, well known physical equations exist to predict the strength of a structure, however in computer systems the physical component level is far too complex to study, hence failures appear to the observer as random events. It is perhaps a natural step therefore to model systems of this type as collections of randomly occuring events which alter the state of the system in some way. If a processor is looked at as a 'black box' it appears to accept requests from some outside

source, perform some actions based on the request and deliver a response. In all but the simplest systems processors will receive many requests from several different sources (users, peripherals, other processors, etc) but will only be able to respond to a certain number at a time, and so the requests will need to be queued to await attention. Thus a computer system can be modelled as a system of one or more queues which are affected by randomly occuring events. The events of interest might be new requests arriving into the system, the service of requests being completed, or something that affects one or other of these, e.g. a breakdown in the system. The study of these models is known as *queueing theory*.

Queueing theory has long been a major method for predicting the performance of computer systems and queueing networks have long been a major topic of research interest. As computer systems have become increasingly powerful (and complex) so greater reliance has been put on them. As a result a greater amount of effort has been made to access the reliability of computer systems, therefore a substantial amount of attention has been given to incorporating breakdowns into queueing models. This thesis is devoted to the construction and analysis of models which can be used to evaluate the performance and reliability of distributed systems. The general object of the research therefore is to extend the types of queueing models with breakdowns which have been solved, with a particular interest in networking structures.

The systems that are studied involve various collections of servers and their associated queues. These range from isolated nodes, though parallel nodes coupled by the effect of breakdowns on arrivals, to pipelines of such parallel stages and more general networks. The issues that are explored include the influence of breakdowns and repairs on delays,

job losses and optimal routeing. Obtaining performance measures for interacting queues is difficult, however a degree of abstraction has been used here which allows long run averages to be calculated (exactly in many cases) for quite complex systems. A variety of different techniques are used in order to obtain solutions to these models, including exact equations, exact numerical and approximate numerical techniques.

Our initial motivation for studying models of this kind came from the telecommunications industry, where the servers are alternative gateways through which messages or packets may be routed. The pipeline and network structures presented in the later chapters are easily applicable to systems in manufacturing industries, where a node consisting of several servers may represent a stage in the manufacturing process of a product, represented by a job. Equally servers may represent computers in a network, and thus a set of parallel servers may represent a set of replicated World Wide Web servers, or database gateways.

1.2 Literature Review

General

The modelling literature contains many studies dealing with the performance and availability of systems subject to breakdowns and repairs. Problems of this type arise in areas as diverse as computing, communications, manufacturing and transport. However, most of the work has concentrated on models involving a single job queue served by one or more processors (e.g., see Avi-Itzhak and Naor [3], Gaver[30], Mitrani and Avi-Itzhak [72], Sengupta [92], Thiruvengedam [99] and White and Christie [108]). Very few results are available for systems with more than one queue, although several exist when processors are

statistically identical, e.g. Chakka and Mitrani [13], Mikou [68] and Neuts and Lucantoni [83]. An approximate solution for a general Jackson network of unreliable nodes was suggested by Mitrani in [70].

There are a number of texts which provide a general introduction into queueing theory in general, queueing networks and modelling queues with breakdowns in particular, among those I have consulted are Ajmone Marsan et al [1], Gelenbe and Pujolle [32], King [48], Kleinrock [50], Mitrani [69] and Sauer and Chandy [89]. Also a number of papers have been published which are useful in directing the reader to relevant work, notable amongst these are the surveys of Doshi [17, 18] and Disney and Konig [16] and the bibliography compiled by Takagi [96].

Single Server Models with Breakdowns

White and Christie [108] were the first to specifically consider server repair following breakdowns, or server vacations, in a queueing system. They presented a model single M/M/1 queue subject to random (negative exponentially distributed) breakdowns and repairs where no jobs are lost. They also considered the case where breakdowns can occur during the repair process, so that when one repair is completed, another must begin immediately before any service takes place (i.e. breakdowns join a higher priority queue).

Heathcote [39] modified White and Christie's model such that breakdowns can only occur when the job queue is non-empty and the service time is $k-$Erlang. Jaiswal [43] obtained a solution of White and Christie's model with multiple failures and generally distributed service and repair times, using a method referred to as 'inclusion of supplementary variable technique'. Thiruvengedam [99] used Jaiswal's method to derive solutions for White and Christie's and Heathcote's models where the service and repair times are

generally distributed.

Gaver [30] also considered a single server model with negative exponentially distributed time before failure. Jobs arrive in batches according to a compound Poisson process and service time and time to repair are generally distributed. Interruptions are considered to be preemptive resume, postponable or preemptive repeat, with several performance measures derived and compared for each.

Avi-Itzhak and Naor [3] considered 5 similar single server models, in each arrivals are assumed to be Poisson and service time and time to repair are generally distributed. The first two of these models were considered earlier by Jaiswal [43] and Thiruvengedam [99], that is, negative exponentially distributed time before failure with preemptive priority, either occuring at any time or only when the queue is non-empty. The next model assumes a negative exponentially distributed time before failure, preemptive and at any time, with repair withheld until the queue is non-empty. The fourth model assumes that the speed of service degrades with time, a job may request that the server repairs to an optimum level with a given fixed probability. The final model again assumes negative exponentially distributed time before failure, but only when the queue is empty.

Federgruen and Green [25] studied a more complicated version of the model considered by both Thiruvegedam [99] and Avi-Itzhak and Naor [3], where the failure and repair times are generally distributed. This is shown to be a far from trivial extension of the earlier work and only bounds and approximations were derived for the most general case, although exact forms are produced when the repair time is assumed negative exponential. Nicola [84] considered an M/G/1 queue subject to failures and repairs where many different types of interruption are possible (e.g. preemptive resume, preemptive repeat, postponable). The

Laplace Stieltjes transform for the completion time is derived and steady state results obtained for the case where only single failures are allowed.

A single server queue with 2 distinct classes of customer and threshold type service was studied by Nain [80]. Class 2 customers have priority until the amount of work required by class 1 type customers exceeds a certain threshold. Clearly this model is closely related to the failure models above, except that class 1 customers build up to cause an interruption of normal (class 2 customer) service. Arrivals in this model are assumed Poisson for both classes of customer, service times are exponentially and generally distributed for class 1 and 2 respectively, and the Laplace Stieltjes transform for the stationary joint distribution of server workload is derived. A slightly simpler version of this model (with exponential service time for class 2 jobs) was studied by Boxma et al [7], who determined the joint queue length distribution using both analytic techniques and the power series algorithm.

Sengupta [92] again considered a single server queue with 2 states, with generally distributed time spent in each state. Jobs arrive in Poisson streams of different type and rate depending on the server state, the interpretation for this being that more urgent jobs will be directed away from a broken server, thus changing the job profile. There is no priority associated with each job type, but the service times may have different means and distributions. A comparison is made with the GI/G/1 queue is made and exact closed form is derived for the case when failures and repairs are exponentially distributed.

Lucantoni et al [61] studied a single server model where a vacation is taken every time the queue becomes empty, and the queue is still empty when the vacation is over then another is taken (ad infinitum). The interpretation for this model is that the server has another associated queues of lower priority which only receive service when this queue is

empty. The duration of the vacation therefore is a period of service at the other (lower priority) queue. Service and vacation times are generally distributed and the arrivals are subject to a general Markovian arrival process, which includes the Markov modulated Poisson process. Arithmetically tractable equations are derived for several performance measures and comparisons are made with the GI/G/1 and M/G/1 queues with vacations.

Doshi [17, 18] has produced two surveys of queues with generally distributed service times and vacations based around a secondary class of jobs, which include several of the above models. Fischer and Meier-Hellstern [26] produced an excellent review of the literature of the Markov-Modulated Process (MMP).

Multi-server Models with Breakdowns

Mitrani and Avi-Itzhak [72] considered a model where a single queue is attached to several identical servers which are subject to independent failures and repairs. Arrivals are assumed Poisson and service time, time before failure and time to repair are all assumed to be negative exponentially distributed. This was the first multi-server model with breakdowns to be published. This problem was returned to in a study by Neuts and Lucantoni [83] with the added feature that only a limited number of servers could be repaired at a time. Mitrani and King [74] compared the model in [72] with an earlier model of theirs [75] where failures arrive into the queue as preemptive priority jobs. They used these models to test the hypothesis that a single server performs less well than many slower servers when failures occur, unlike the case where servers are reliable (assuming the same overall service capacities).

In [70] Mitrani studied a Jackson network where each node consists of a single queue and server subject to exponentially distributed breakdowns and repairs. Jobs arrive at each

node in independent Poisson streams such that every node is saturated, after completion of a service of exponentially distributed duration the job leaves the system. Jobs in a queue may transfer to another queue at any time, the rate of transfer being dependent on the operative state of the server. Exact performance measures are derived for this situation and an approximation technique is describe where the network is not saturated.

Mikou [68] analysed a tightly coupled two-node network with simultaneous breakdowns and repairs, by a far from trivial reduction to a boundary value problem. Jobs arrive in a Poisson stream to the first of the servers, after an exponentially distributed service time the job either departs the system, or passes on to the second server. After service at the second server, all jobs return to the first. Failures halt service at both servers, but the queues remain intact, both time before failure and time to repair are exponentially distributed. This problem is revisited (as an example) in chapter 7.

More recently, Mitrani and Wright [77] examined a system with N parallel queues where the consequences of a breakdown are (a) the loss of all jobs in the corresponding queue and (b) the re-direction or loss of all arrivals to that queue during the subsequent repair period. Those assumptions imply that the queue of a broken server is necessarily empty. The solution of this model utilises the *quasi-separable* nature of the system in the same way as in chapter 3 of this thesis. Idrissi-Kacemi et al. [40] have studied the case of two queues, only one of which is subject to breakdowns; all jobs present are transferred, and new jobs are redirected, to the other queue after a breakdown. A pipeline structure was considered by Ezhilchelvan et al [21], here the nodes were tri-modular redundant (a highly reliable architecture) and a good approximate solution was found.

Of the above citations, only Mitrani and Wright [77] obtain exact performance mea-

sures for a model with more than two queues.

In forming the network model in chapter 7 it was necessary to consult several reliable network models from the literature and consider the effect of imposing failures. Foremost in this literature are the survey of Disney and Konig [16] and the class of models of open and closed networks considered by Baskett et al [5]. The model in chapter 7 is based on one the earliest queueing network models proposed by Jackson [41], in which there are M nodes (called departments), each consisting of a single queue with n_M associated servers. Jobs arrive from outside the system in a Poisson stream at each node, and upon completion of an exponentially distributed service time may either leave the system or move on to another node with fixed probability. Clearly a job will eventually leave the system, but the number of services it receives and the route it takes through the system is random. Two examples are also considered in chapter 7, one based on a 2 node model proposed by Mikou [68] (outlined above) and another an extension of a 3 node model studied by Mitrani [71], where each node is a simple M/M/1 queue. All jobs arrive in a Poisson stream at node 1, and after completion of service are divided between nodes 2 and 3. All jobs completing service at node 2 move on to node 3 and all jobs completing service at node 3 leave the system. Essentially this is a simple network that allows overtaking, but without any reliability issues involved, the model presented in chapter 7 has the same structure but with independent random failures and repairs at each node.

Priority Queues

Closely related to models of server breakdowns and vacations is the area of priority queues. In a priority queue there are k classes of jobs, each with an independent arrival and service rate. Jobs of class i have greater priority of service than jobs of class $i-1$

$(0 < i \leq k)$ and as such are chosen for service in preference, either preemptively, or non-preemptively. In the preemptive case the service of a job of class i will be interrupted by the arrival of a job of class j if $j > i$ $(i, j \leq k)$, whereas in the non-preemptive case the service of a job cannot be interrupted. In both cases the highest priority job available in the queue will be served in FIFO order (within its class) when the server becomes available. Clearly therefore a server breakdown is a special case of a preemptive priority queueing system where there are 2 classes of jobs and the arrival of a high priority job (failure) precludes any further such events until after its service is complete.

The earliest priority queueing models to be studied were all non-preemptive, notable amongst these are Cobham [14] and Morse [78]. The first preemptive models were produced by White and Christie [108] who studied a 2 class M/M/1 queue as well as making the first models of server breakdowns (see above). Further single server models were presented by Heathcote [39], Jaiswal [43] (see *Single Server Models* above) and Miller [67] who surveyed the existing literature on priority queues and added additional performance measures to the models of Morse [78] and White and Christie [108]. More recently much effort has been made to derive performance measures for multi-server priority systems, notable amongst these are Mitrani and King [75], Buzen and Bondi [11] and Gail et al [27]. In addition Takine and Hasegawa [97] tackled the problem of re-sequencing a preemptive M/M/2 queue with 2 classes of job, Kouvatsos and Tabet-Aouel [57] developed an approximation method for a large class of general closed networks and Epema studied a general feedback model with either preemptive or non-preemptive scheduling. Some of these results and their significance is discussed by Doshi [18, 17].

Solution Methods

The matrix solution method used to solve many of the models in this thesis, known as *spectral expansion*, was developed by Chakka, Mitra and Mitrani [12, 13, 76]. The idea of spectral expansion has been known about for some time (see Neuts [82]) although few examples of its use in performance evaluation exist in the literature (see Elwalid et al [19] and Mitrani and Mitra [76]). There are other solution methods which would be applicable to these models, notably one based on the *matrix geometric* representation of the invariant vector used by Neuts [82]. More recently Haverkort [35] has proposed a further matrix geometric method based upon earlier work by Tijms et al [102]. A completely different approach based upon statistical methods from other fields has been taken by Kouvatsos ([56]) and others. The choice of which solution method to use depends on the size and structure of the model. Some evidence was given by Mitrani and Mitra [76] which suggests spectral expansion is the quickest of these. Having said that though, spectral expansion does have its limitations, notably when the number of eigenvalues to be found is large. Any comparative study of these solution methods is going to be a major task and one out of the range of this work, however it is a study worth undertaking before the number of possible methods extends further. The approximation technique used in chapters 5,6 and 7 is similar to an approach suggested by Marie [62], where the number of states in a Markov chain can be reduced by grouping together states with similar properties.

1.3 Contents of Subsequent Chapters

The initial direction of this research was to build a family of related single server queueing models with failures, based around the well known simple M/M/1 queue. These are pre-

sented in chapter 2, with a closed form solution presented for each model and appropriate performance measures derived.

In chapter 3 a general model is presented of the quasi-birth-and-death type. A model is defined whereby a server can be in one of n possible operative states, with different arrival and service characteristics in different states, and with transitions between states governed by an arbitrary Markov chain. In solving this model the spectral expansion method is introduced, this solution method becomes the mainstay of the solution of many of the models in the following chapters.

In chapter 4 a number of server/queue nodes of the type considered in chapter 2 are considered to operate in parallel, with a single arrival stream. In general the joint distribution of queue sizes is an intractable problem, but it is possible to derive certain long run averages as performance measures (notably the average number of jobs in the system) by considering the marginal queue size distribution for each queue. The solution of the marginal queue size distributions is a special case of the quasi-birth-death model presented in the previous chapter. As well as providing an interesting problem to solve, this model also raises several points regarding routeing issues in unreliable systems, and so some analysis of different routeing strategies is included here.

One of the main problems in solving queueing of this type is that the operational state space of the system becomes very large very quickly, therefore the question of approximate solutions needs to be addressed. In chapter 5 approximations are used to predict the optimal routeing strategy for the models defined in chapter 4, as well as predicting the performance measures involved. Here the ideas of Markov modulated arrival processes and lumping are introduced and the models defined in chapter one are used as simple approx-

imations. Numerical results are presented to illustrate the accuracy of the approximate methods.

The ideas introduced in chapter 5 are applied to an extended model in chapter 6. Here a pipeline model is defined where each node in the line has the structure of the model defined in chapter 4. Clearly, in the general case, this gives rise to a much larger state space than previously encountered and also the dependencies between the nodes mean that an exact solution is, in general, an intractable problem. However, the methods developed in chapter 5 can be applied to give a good approximation in most cases, which again is illustrated by numerical results compared with simulation. The question of optimal routeing is again addressed in this chapter as it is necessary to find whether the existence of previous nodes has an effect on optimal routeing policy.

Chapter 7 takes the obvious next extension beyond the pipeline model discussed in chapter 6 to take on the problem of more general networks. A network model is defined and the techniques illustrated in previous chapters are used to derive approximated solutions to this model. Two examples are also considered in chapter 7, one an extension of a 2 node model with feedback proposed by Mikou [68] and another an extension of a 3 node model with overtaking studied by Mitrani [71].

Much of the work presented in chapters 3, 4 and 5 has already been published elsewhere [100, 101] and we hope to publish more papers from this thesis in the near future.

Chapter 2

Models of Isolated Servers Suffering Breakdowns and Repairs

2.1 Summary

Before studying systems of many servers it is often necessary to consider the behaviour of these servers in isolation. These models form the foundations for larger models. In all the models presented here there is assumed to be a Poisson arrival stream, an unbounded queue and a FIFO scheduling strategy, i.e. these are all M/M/1 servers, but differ in the nature and effect of their failures and in the way in which their queues behave during repair periods. In each case the performance measures sought are the average number of jobs in the queue (including the job being served) and the related measure of average response time. In the final two models failures of more than one type can occur, these models are special cases of the more general model presented in chapter 3.

2.2 Model Definition

All the models in this section are based around the simplest of queueing models, known as M/M/1. An M/M/1 queue consists of a single server with an associated unbounded queue. Jobs arrive into the queue in a Poisson stream of rate λ and receive service (depart) sequentially in order of arrival with service (inter-departure) times exponentially distributed with mean $1/\mu$.

Figure 2.1: A simple M/M/1 queue

In the simple M/M/1 queue the server is assumed to be available at all times, therefore if there are one or more jobs in the queue then service will take place. This is unlikely to be the case in practice since most systems will have periods when the server is unavailable for some reason. In the models described here it is assumed that a server is available except when it suffers an unscheduled (random) breakdown. In the literature the word 'breakdown' is often replaced with 'vacation', implying the server is 'away' doing something else, both phrases are used synonymously to mean a period of unavailability.

It is easy to envisage scenarios for the various different causes and effects of failures. Server and queue may physically be parts of the same machine, therefore a catastrophic failure of that machine will not only suspend service, but will wipe clean the contents of

the queue, possibly irrevocably. Conversely, the server and queue may be entirely separate, and so failure of the server will not affect the operation of the queue, it may even still accept jobs. In some situations, e.g. in a printer queue, the head job is passed from the queue directly to the server, where it remains until service is completed, failure in this instance will result in only the head job being lost, the remainder of the queue may be unaffected. In this situation a failure of the queue will not immediately suspend service, as the head job will remain in service, however once that job is completed the server will be idle until the queue returns to operation and new jobs arrive. It is also possible to envisage situations where more than one type of failure occurs, for example where the server and queue are collocated a failure may be catastrophic, or it may just affect part of the system, i.e. the server.

In the first 5 models the server can suffer only one kind of failure, i.e. there is no distinction between the various possible causes of failure, the server is simply 'broken'. When a failure occurs the server goes through a period of repair, which when completed delivers the server back to full operation. The duration of the periods of operation and repair are exponentially distributed random variables with means $1/\xi$ and $1/\eta$ respectively. What happens to the jobs in the queue on failure and to any jobs arriving during the period of repair constitutes the differences between the models.

In the 6th model only the queue suffers failure, but it is assumed that the head job can continue in service as it is physically (and / or logically) located at the server, the remainder of the queue is lost.

In the final 2 models the server may suffer two different sorts of failure, each of which has a different rate of occurrence, different rate of repair and different consequences for

the jobs in the queue and those arriving. It is further assumed that a server may pass from one failed state to the other (with appropriate consequences), hence there are 6 possible transitions, each one of which has an associated exponentially distributed random variable.

2.3 Solution Method

Each of the models outlined above forms a Markov process, where the system state at time t is described by the pair $I(t), J(t) : t \geq 0$, where where $I(t) \subset \{0, ..., n-1\}$ represents the operational state of the system and $J(t)$ is the number of jobs in the queue. In the first six models described here there are just two operational states, broken and working (0 and 1 respectively) and in the final two there are three operational states; fully broken, partially broken and working (0,1 and 2 respectively).

Exact solutions are obtained in all cases. Performance measures are derived for these models by first finding the probability generating function of the number of jobs in the queue, $g(z)$, defined by,

$$g(z) = \sum_{j=0}^{\infty} z^j p_j \tag{2.1}$$

where p_j is the steady-state probability that there are exactly j jobs in the queue. Clearly,

$$g(1) = 1 \tag{2.2}$$

by definition. Also, it is then a relatively straight forward matter to derive the average number of jobs in the queue, $\overline{n}$ by differentiating (2.1),

$$\overline{n} = g'(1) = \sum_{j=0}^{\infty} j p_j \tag{2.3}$$

When the systems in the models concerned have more than one state of operation it is convenient to consider the joint generating functions of operative state i and queue size

j. The the queue size generating function can be decomposed into,

$$g(z) = \sum_{i=0}^{n-1} g_i(z) \tag{2.4}$$

where n is the total number of operational states,

$$g_i(z) = \sum_{j=0}^{\infty} z^j p_{i,j} \tag{2.5}$$

and, $p_{i,j}$ is the steady-state probability that the server is in operational state i and there are exactly j jobs in the queue. In the models presented in this chapter expressions for the joint generating functions can be derived from the balance equations in order to find a closed form solution for the queue size generating function and hence expressions for relevant performance measures.

It is clearly possible to form expressions for the probabilities $p_{i,j}$ directly from the queue size generating functions $g_i(z)$. This can be done by finding the partial fractions of $g_i(z)$ and expressing them as geometric series. The probabilities $p_{i,j}$ can then easily be found using 2.5. Alternatively expressions for the probabilities $p_{i,j}$ can often be formed with much less effort directly from the balance equations in terms of the derived constants.

2.4 Queue retained but arrivals lost during inoperative periods

A simple M/M/1 queue has a Poisson arrival stream of rate λ. Jobs are served in order of arrival with service times negative exponentially distributed with mean $1/\mu$. In addition the server suffers breakdowns and repairs with operative and inoperative periods negative exponentially distributed with means $1/\xi$ and $1/\eta$ respectively. Incoming jobs are accepted

into the queue only when the server is operative. Let $p_{0,j}$ and $p_{1,j}$ be the steady-state probabilities that the server is broken or operative respectively, with exactly j jobs in the queue.

These probabilities satisfy the following balance equations,

$$(\xi + \lambda)p_{1,0} = \eta p_{0,0} + \mu p_{1,1} \tag{2.6}$$

$$(\xi + \lambda + \mu)p_{1,j} = \eta p_{0,j} + \mu p_{1,j+1} + \lambda p_{1,j-1} \quad \ldots \quad j \geq 1 \tag{2.7}$$

$$\eta p_{0,j} = \xi p_{1,j} \quad \ldots \quad j \geq 0 \tag{2.8}$$

Writing the queue size probability generating function as the sum of the joint queue size generating functions representing the operative and inoperative states, gives,

$$g(z) = g_0(z) + g_1(z) \tag{2.9}$$

It is then a simple matter to re-write the balance equations in terms of the joint queue size probability generating functions $g_0(z)$ and $g_1(z)$ using the definitions given in (2.5) and (2.9),

$$\eta g_0(z) = \xi g_1(z) \tag{2.10}$$

$$(\mu - \lambda z)g_1(z) = \mu g_1(0) \tag{2.11}$$

hence, by (2.9) and (2.10),

$$g(z) = \frac{\eta + \xi}{\eta} g_1(z) \tag{2.12}$$

This, together with (2.2) and (2.9), yields the steady state probabilities that the server is operative, $g_1(1)$, or inoperative, $g_0(1)$,

$$g_0(1) = \frac{\xi}{(\eta + \xi)} \tag{2.13}$$

$$g_1(1) = \frac{\eta}{(\eta + \xi)} \tag{2.14}$$

so substituting $z = 1$ in (2.11) gives,

$$g_1(0) = \frac{(\mu - \lambda)\eta}{\mu(\eta + \xi)} \tag{2.15}$$

Thus (2.9), (2.12) and (2.15) give,

$$g(z) = \frac{(\mu - \lambda)}{(\mu - \lambda z)} \tag{2.16}$$

Which is exactly the same generating function as for the M/M/1 queue without breakdowns. Thus the average number of jobs is given by,

$$\overline{n} = \frac{\lambda}{\mu - \lambda} \tag{2.17}$$

Furthermore, it follows from (2.10), (2.11), (2.13), (2.14), and (2.15), that,

$$g_1(z) = \frac{\eta(\mu - \lambda)}{(\eta + \xi)(\mu - \lambda z)} = g_1(1)\frac{1 - \rho}{1 - \rho z}$$

and,

$$g_0(z) = \frac{\xi(\mu - \lambda)}{(\eta + \xi)(\mu - \lambda z)} = g_0(1)\frac{1 - \rho}{1 - \rho z}$$

where,

$$\rho = \frac{\lambda}{\mu}$$

Which means that the state of the server and the size of the queue are independent of each other. This is the only model presented here with that property.

The average response time for a successfully completed job is given by Little's Theorem to be the average number of jobs $\overline{n}$ divided by the average arrival rate (of successful jobs), thus,

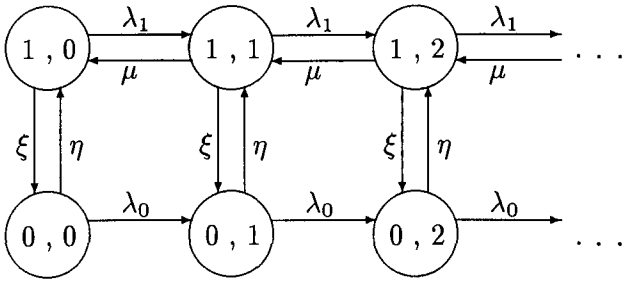
$$W = \frac{\eta + \xi}{\eta(\mu - \lambda)}$$

The ergodicity condition is the same as for the M/M/1 queue, namely $\lambda < \mu$.

2.5 Queue retained and arrivals continue during inoperative period

This model has two forms, the general case where arrivals during the repair periods have a different arrival rate than during normal operation and the special case where the arrival rate is not affected by failures. This special case is given as an example by King [48], so the result is only given as a simplification of the more general model, which is derived in full. Servers of this type were modelled in a parallel system by Neuts and Lucantoni [83], but the closed-form for a single server was not derived.

The Poisson arrival rate during operative periods is λ_1. During periods of repair the queue continues to accept jobs, but the Poisson arrival stream has a different rate, λ_0. The other assumptions are the same as the previous model. The system state diagram for this model is illustrated below.



As previously, let $p_{0,j}$ and $p_{1,j}$ be the steady-state probabilities that the server is broken or operative respectively, with exactly j jobs in the queue.

These probabilities satisfy the following balance equations,

$$(\xi + \lambda_1)p_{1,0} = \eta p_{0,0} + \mu p_{1,1} \tag{2.18}$$

$$(\xi + \lambda_1 + \mu)p_{1,j} = \eta p_{0,j} + \mu p_{1,j+1} + \lambda_1 p_{1,j-1} \quad \cdots \quad j \geq 1 \tag{2.19}$$

$$(\eta + \lambda_0)p_{0,0} = \xi p_{1,0} \tag{2.20}$$

$$(\eta + \lambda_0)p_{0,j} = \xi p_{1,0} + \lambda_1 p_{0,j-1} \quad \ldots \quad j \geq 1 \tag{2.21}$$

The balance equations can then be re-written in terms of the joint queue size probability generating functions $g_0(z)$ and $g_1(z)$ using the definitions given in (2.5) and (2.9),

$$[\lambda_0 z(1-z) + \eta z]g_0(z) = \xi z g_1(z) \tag{2.22}$$

$$[\lambda_1 z(1-z) + \xi z - \mu(1-z)]g_1(z) = \eta z g_0(z) - \mu(1-z)g_1(0) \tag{2.23}$$

Adding (2.22) and (2.23), and substituting $z = 1$, gives,

$$\mu g_1(0) = (\mu - \lambda_1)g_1(1) - \lambda_0 g_0(1) \tag{2.24}$$

As previously, the steady-state probabilities that server is broken or operative are given by (2.13) and (2.14), hence (2.24) gives,

$$\mu g_1(0) = \frac{(\mu - \lambda_1)\eta - \lambda_0 \xi}{(\eta + \xi)} \tag{2.25}$$

Substituting (2.22) in (2.9) gives,

$$g(z) = \frac{\lambda_0(1-z) + \eta + \xi}{\lambda_0(1-z) + \eta} g_1(z) \tag{2.26}$$

Equations (2.22), (2.23) and (2.24) yield,

$$g_1(z) = \frac{[\lambda_0(1-z) + \eta][(\mu - \lambda_1)\eta - \lambda_0 \xi]}{(\eta + \xi)[\eta(\mu - \lambda_1 z) + \lambda_0(\mu(1-z) - \lambda_1 z(1-z) - \xi z)]} \tag{2.27}$$

Hence,

$$g(z) = \frac{[\lambda_0(1-z) + \eta + \xi][(\mu - \lambda_1)\eta - \lambda_0 \xi]}{(\eta + \xi)[\eta(\mu - \lambda_1 z) + \lambda_0(\mu(1-z) - \lambda_1 z(1-z) - \xi z)]} \tag{2.28}$$

with the ergodicity given by $\eta\mu > \eta\lambda_1 + \xi\lambda_0$

Clearly if $\lambda_0 = 0$ then the same generating function as the previous case is obtained. Similarly if $\lambda_0 = \lambda_1 = \lambda$ then (2.28) gives the well known result (see King [48]),

$$g(z) = \frac{[\lambda(1-z)+\eta+\xi][(\mu-\lambda)\eta-\lambda\xi]}{(\eta+\xi)[(\mu-\lambda z)(\lambda(1-z)+\eta)-\lambda\xi z]} \tag{2.29}$$

The average number of jobs in the queue can now be easily obtained by differentiation, so,

$$\overline{n} = g'(1) = \frac{(\eta\lambda_1+\xi\lambda_0)(\eta+\xi)+\lambda_0\xi(\mu-\lambda_1+\lambda_0)}{(\eta+\xi)(\eta(\mu-\lambda_1)-\lambda_0\xi)} \tag{2.30}$$

This expression reduces to (2.17) in the special case $\lambda_0 = 0$. Since there are no jobs lost from the queue, the arrival rate of successful jobs is the same as the average arrival rate, namely,

$$\frac{\eta\lambda_1+\xi\lambda_0}{(\eta+\xi)}$$

So, by Little's Theorem , the average response time is given by,

$$W = \frac{(\eta\lambda_1+\xi\lambda_0)(\eta+\xi)+\lambda_0\xi(\mu-\lambda_1+\lambda_0)}{(\eta\lambda_1+\xi\lambda_0)(\eta(\mu-\lambda_1)-\lambda_0\xi)} \tag{2.31}$$

In the simpler case where $\lambda_0 = \lambda_1 = \lambda$, (2.30) coincides with the known result [3, 72],

$$\overline{n} = g'(1) = \frac{(\eta+\xi)^2\lambda+\mu\lambda\xi}{(\eta+\xi)(\eta(\mu-\lambda)-\lambda\xi)}$$

and,

$$W = \frac{(\eta+\xi)^2+\mu\xi}{(\eta+\xi)(\eta(\mu-\lambda)-\lambda\xi)} \tag{2.32}$$

2.6 Job in service and arrivals during repair lost

In this model it is assumed that when a failure occurs it is impossible to resurrect the job that was being served so that it's service can be resumed after the completed repair. This

means that the head job is lost from the queue. Apart from that the characteristics for this model are exactly the same as for the first model in this chapter, i.e. Poisson arrivals at rate λ whilst operative, no arrivals during repair, jobs served in order of arrival with service times negative exponentially distributed with mean $1/\mu$ and the server breakdowns and repairs with operative and inoperative periods negative exponentially distributed with means $1/\xi$ and $1/\eta$ respectively. Let $p_{0,j}$ and $p_{1,j}$ be the steady-state probabilities that the server is broken or operative respectively, with exactly j jobs in the queue.

These probabilities satisfy the following balance equations,

$$(\xi + \lambda)p_{1,0} = \eta p_{0,0} + \mu p_{1,1} \tag{2.33}$$

$$(\xi + \lambda + \mu)p_{1,j} = \eta p_{0,j} + \mu p_{1,j+1} + \lambda p_{1,j-1} \quad \cdots \quad j \geq 1 \tag{2.34}$$

$$\eta p_{0,0} = \xi p_{1,0} + \xi p_{1,1} \tag{2.35}$$

$$\eta p_{0,j} = \xi p_{1,j+1} \quad \cdots \quad j \geq 1 \tag{2.36}$$

The balance equations can then be re-written in terms of the joint queue size probability generating functions $g_0(z)$ and $g_1(z)$ using the definitions given in (2.5) and (2.9),

$$\eta z g_0(z) = \xi g_1(z) - \xi(1 - z)g_1(0) \tag{2.37}$$

$$(\mu + \xi - \lambda z)g_1(z) = (\mu + \xi)g_1(0) \tag{2.38}$$

Putting $z = 1$ in (2.38) gives,

$$(\mu + \xi)g_1(0) = (\mu + \xi - \lambda)g_1(1) \tag{2.39}$$

As previously, the steady-state probabilities that server is broken or operative are given by (2.13) and (2.14), hence,

$$g_1(0) = \frac{\eta(\mu+\xi-\lambda)}{(\eta+\xi)(\mu+\xi)}$$

Re-writing (2.37) and (2.38) gives,

$$(1-z)g_1(0) = g_1(z) - \frac{\eta z}{\xi}g_0(z) = \frac{(\mu+\xi-\lambda z)(1-z)}{(\mu+\xi)}g_1(z) \tag{2.40}$$

Thus substituting (2.40) and (2.39) in (2.9) gives,

$$g(z) = \frac{[(\eta+\xi)(\mu+\xi)+\xi\lambda(1-z)](\mu+\xi-\lambda)}{(\mu+\xi)(\eta+\xi)(\mu+\xi-\lambda z)} \tag{2.41}$$

So the average number of jobs in the queue is given by,

$$\overline{n} = g'(1) = \frac{\lambda\eta(\mu+\xi)+\xi\lambda^2}{(\mu+\xi)(\eta+\xi)(\mu+\xi-\lambda)}$$

This includes jobs that will be lost before completing service. The probability of a job in the queue completing service successfully is,

$$\pi = \frac{\mu}{\mu+\xi} \tag{2.42}$$

Thus the probability that there exactly j successful jobs in the queue, q_j, is given by,

$$q_j = \sum_{k=j}^{\infty} \binom{k}{j} \pi^j (1-\pi)^{k-j} p_k$$

where $p_k = p_{0,k} + p_{1,k}$.

The generating function $s(z)$ for the number of successfully completing jobs in the queue is given by,

$$s(z) = \sum_{j=0}^{\infty} z^j q_j = \sum_{j=0}^{\infty} z^j \pi^j \sum_{k=j}^{\infty} \binom{k}{j} (1-\pi)^{k-j} p_k$$

$$= \sum_{k=0}^{\infty}(1-\pi)^k p_k \sum_{j=0}^{k}\binom{k}{j}\left(\frac{\pi z}{1-\pi}\right)^j = \sum_{k=0}^{\infty} p_k(1-\pi+\pi z)^k$$

so,

$$s(z) = g(1-\pi+\pi z) \tag{2.43}$$

hence the average number of successfully completing jobs is given by,

$$s\prime(1) = \pi \sum_{k=0}^{\infty} k p_k = \frac{\mu\overline{n}}{\mu+\xi} \tag{2.44}$$

The arrival rate of successfully completing jobs is the total job arrival rate minus the rate at which jobs are lost, given by,

$$g_1(1)\lambda - \xi[g_1(1) - g_1(0)] = \frac{\eta\lambda}{(\eta+\xi)} - \frac{\xi\eta\lambda}{(\eta+\xi)(\mu+\xi)} = \frac{\eta\lambda\mu}{(\eta+\xi)(\mu+\xi)}$$

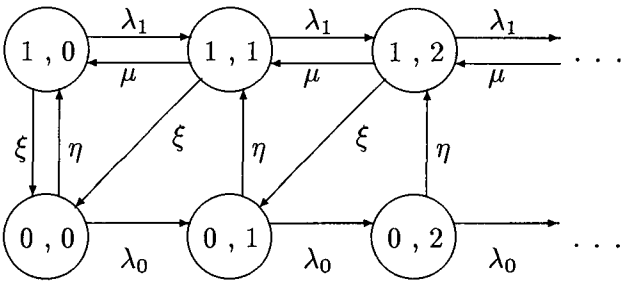
Thus, by Little's Theorem , the average response time (for a successfully completed job) is given by,

$$W = \frac{\lambda\xi + \eta(\mu+\xi)}{\eta(\mu+\xi)(\mu+\xi-\lambda)} \tag{2.45}$$

and the ergodicity condition is $\mu + \xi > \lambda$

2.7 Job in service is lost, arrivals continue during inoperative periods

In this model we make the same assumptions as the previous case, however it is further assumed that jobs arrive during the repair period. The arrival rate when the server is operative is λ_1 and λ_0 when it is broken, in both cases the arrivals are Poisson. The system state diagram for this model is illustrated below.



Taking the same approach as before, let $p_{0,j}$ and $p_{1,j}$ be the steady-state probabilities that the server is broken or operative respectively, with exactly j jobs in the system.

These probabilities satisfy the following balance equations,

$$(\xi + \lambda_1)p_{1,0} = \eta p_{0,0} + \mu p_{1,1} \tag{2.46}$$

$$(\xi + \lambda_1 + \mu)p_{1,j} = \eta p_{0,j} + \mu p_{1,j+1} + \lambda p_{1,j-1} \quad \ldots \quad j \geq 1 \tag{2.47}$$

$$(\eta + \lambda_0)p_{0,0} = \xi p_{1,0} + \xi p_{1,1} \tag{2.48}$$

$$(\eta + \lambda_0)p_{0,j} = \xi p_{1,j+1} + \lambda_0 p_{0,j-1} \quad \ldots \quad j \geq 1 \tag{2.49}$$

The balance equations can then be re-written in terms of the joint queue size probability generating functions $g_0(z)$ and $g_1(z)$ using the definitions given in (2.5) and (2.9),

$$[\lambda_1 z(1-z) + \xi z - \mu(1-z)]g_1(z) = \eta z g_0(z) - \mu(1-z)g_1(0) \tag{2.50}$$

$$(\lambda_0 z(1-z) + \eta z)g_0(z) = \xi g_1(z) - \xi(1-z)g_1(0) \tag{2.51}$$

Adding (2.50) and (2.51) gives,

$$(\mu + \xi - \lambda_1 z)g_1(z) - \lambda_0 z g_0(z) = (\mu + \xi)g_1(0) \tag{2.52}$$

As previously, the steady-state probabilities that server is broken or operative are given by (2.13) and (2.14), hence z=1 in (2.52), gives,

$$g_1(0) = \frac{(\mu+\xi-\lambda_1)\eta - \lambda_0\xi}{(\mu+\xi)(\eta+\xi)} \tag{2.53}$$

Substituting (2.51) in (2.9) gives,

$$g(z) = \frac{[\xi + \lambda_0 z(1-z) + \eta z]g_1(z) - \xi(1-z)g_1(0)}{\lambda_0 z(1-z) + \eta z} \tag{2.54}$$

And eliminating $g_0(z)$ from (2.51) and (2.50) yields,

$$g_1(z) = \frac{[\mu\lambda_0(1-z) + \eta(\mu+\xi)]g_1(0)}{\xi(\eta-\lambda_0 z) + (\mu-\lambda_1 z)[\lambda_0(1-z)+\eta]} \tag{2.55}$$

Hence, substituting (2.55) in (2.54) gives,

$$g(z) = \frac{(\mu+\xi)(\eta+\xi) + (1-z)(\xi\lambda_1 + \mu\lambda_0)}{(\eta-\lambda_0 z)(\mu+\xi-\lambda_1 z) + \lambda_0(\mu-\lambda_1 z)} g_1(0) \tag{2.56}$$

Differentiating (2.56) at $z = 1$ gives the average number of jobs in the system,

$$\overline{n} = g'(1) = \frac{(\mu+\xi)[\xi\lambda_0(\mu+\xi) + \eta(\eta\lambda_1 + \xi\lambda_0)] + \xi(\mu\lambda_0 - \eta\lambda_1)(\lambda_0 - \lambda_1)}{[\eta(\mu+\xi-\lambda_1) - \xi\lambda_0](\mu+\xi)(\eta+\xi)}$$

As previously this includes jobs that will be lost before completing service. The probability of a job in the queue completing service successfully is given by (2.42) and the relation between the average number of jobs in the queue and the average number of successful jobs in the queue is given by (2.43) and (2.44). The rate at which jobs are lost is given by the failure rate multiplied by the probability that there are one or more jobs in the queue when the server is active, i.e. $\xi[g_1(1) - g_1(0)]$. The arrival rate of successfully completing jobs is the total job arrival rate minus the rate at which jobs are lost, given by,

$$\frac{\eta\lambda_1 + \xi\lambda_0}{\eta+\xi} - \frac{\xi(\eta\lambda_1 + \xi\lambda_0)}{(\mu+\xi)(\eta+\xi)} = \frac{\mu(\eta\lambda_1 + \xi\lambda_0)}{(\mu+\xi)(\eta+\xi)}$$

So, by Little's Theorem, the average response time is,

$$W = \frac{(\mu+\xi)[\xi\lambda_0(\mu+\xi)+\eta(\eta\lambda_1+\xi\lambda_0)]+\xi(\mu\lambda_0-\eta\lambda_1)(\lambda_0-\lambda_1)}{[\eta(\mu+\xi-\lambda_1)-\xi\lambda_0](\mu+\xi)(\eta\lambda_1+\xi\lambda_0)}$$

where the ergodicity condition is given by, $\eta(\mu+\xi) > \eta\lambda_1+\xi\lambda_0$.

Clearly substituting $\lambda_0 = 0$ and $\lambda_1 = \lambda$ gives the same results as the previous model. For the special case where the arrival rate is unaffected by failures, i.e. substituting $\lambda_0 = \lambda_1 = \lambda$, the average number of jobs in the queue is given by,

$$\overline{n} = \frac{\lambda[\xi(\mu+\xi)+\eta(\eta+\xi)]}{[\eta(\mu+\xi)-\lambda(\eta+\xi)](\eta+\xi)}$$

and the average response time is given by,

$$W = \frac{\xi(\mu+\xi)+\eta(\eta+\xi)}{[\eta(\mu+\xi)-\lambda(\eta+\xi)](\eta+\xi)} \tag{2.57}$$

2.8 Entire queue lost at breakdown and arrivals lost during inoperative periods

In the previous models it was assumed that the server was in some sense independent of the queue to the extent that following a failure and subsequent repair it was possible to recover most, if not all, of the jobs in the queue. Clearly this is not always going to be the case, since in many situations the server and queue will physically be part of the same machine and so any interruption of service will also have a negative effect on the queue. In this model the case is considered where a failure of the server results in all the jobs in the queue being irrevocably lost.

A Poisson stream of rate λ enters a queue with an associated server which serve jobs in FIFO order with service times negative exponentially distributed with mean $1/\mu$. The

server suffers breakdowns and subsequent repairs with operative and inoperative periods negative exponentially distributed with means $1/\xi$ and $1/\eta$ respectively, as before. On failure all jobs in the queue, including the one currently being served, are lost, and there are no arrivals during the repair period. Intuitively this system will always be ergodic if $\xi > 0$, since failures will empty the queue, thus preventing it from becoming infinite. This type of server was considered in a parallel system by Mitrani and Wright [77], but the closed form solution for a single server was not derived.

Once again, let $p_{0,j}$ and $p_{1,j}$ be the steady-state probabilities that the server is broken or operative respectively, with exactly j jobs in the queue.

These probabilities satisfy the following balance equations,

$$\eta p_{0,0} = \xi \sum_{j=0}^{\infty} p_{1,j} \tag{2.58}$$

$$(\xi + \lambda) p_{1,0} = \eta p_{0,0} + \mu p_{1,1} \tag{2.59}$$

$$(\xi + \lambda + \mu) p_{1,j} = \mu p_{1,j+1} + \lambda p_{1,j-1} \quad \cdots \quad j \geq 1 \tag{2.60}$$

(2.2) and (2.58) give,

$$p_{0,0} = \frac{\xi}{\eta + \xi} \tag{2.61}$$

The balance equations can then be re-written in terms of the joint queue size probability generating function $g_1(z)$ using the definitions given in (2.5) and (2.9),

$$[\xi z + \lambda z(1-z) - \mu(1-z)] g_1(z) = \eta z p_{0,0} - \mu(1-z) g_1(0) \tag{2.62}$$

where $g_1(z)$ is the queue size probability generating function for the number of jobs in the system when the server is operative, such that $g_1(1) = \eta/(\eta + \xi)$.

In the previous models it has been possible to find expressions for the constants ($g_1(0)$) by setting $z = 1$ in the equation for the relevant generating function. Unfortunately that is not the case here. An equation for $g_1(0)$ can be obtained by noting that if the quadratic polynomial multiplying $g_1(z)$ in (2.62) has a root at z_0, such that $\mid z_0 \mid < 1$, then the right hand side of (2.62) must vanish at $z = z_0$. Consider the function $f(z)$ equivalent to the left hand side of (2.62),

$$f(z) = \xi z + \lambda z(1-z) - \mu(1-z)$$

Clearly, $f(1) = \xi$ and $f(0) = -\mu$, therfore, if $\xi > 0$ and $\mu > 0$, there must be some value z_0 between 0 and 1 for which $f(z_0) = 0$. Consider also the value of $f(z)$ as $z \to \infty$,

$$\lim_{z\to\infty} f(z) \to -\infty$$

Hence the other root of the square polynomial $f(z)$, z_1 must lie in the range [1,∞). The values of the roots are given by,

$$z_i = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad \ldots \quad i = 0, 1$$

where $a = -\lambda$, $b = (\xi + \mu + \lambda)$ and $c = -\mu$. Clearly,

$$b^2 - 4ac = (\xi + \mu + \lambda)^2 - 4\lambda\mu = (\mu - \lambda)^2 + \xi(2\mu + 2\lambda + \xi)$$

which is always positive, so both roots are real. Since $z_0 \leq z_1$ and $\sqrt{b^2 - 4ac} \geq 0$ then

$$z_0 = \frac{\xi + \mu + \lambda - \sqrt{(\xi + \mu + \lambda)^2 - 4\lambda\mu}}{2\lambda} \tag{2.63}$$

so substituting (2.63) in (2.62) gives,

$$g_1(0) = \frac{\xi\eta\left(\lambda + \xi + \mu - \sqrt{(\xi + \mu + \lambda)^2 - 4\lambda\mu}\right)}{\mu(\eta + \xi)\left(\lambda - \xi - \mu + \sqrt{(\xi + \mu + \lambda)^2 - 4\lambda\mu}\right)}$$

If $\xi = 0$ then both roots coincide at $z = 1$, if $\mu = 0$ then $z_0 = 0$, but then there would never be any service.

The average number of jobs can be found by differentiating (2.62),

$$\overline{n} = g'(1) = \frac{\mu(\eta + \xi)g_1(0) - (\mu - \lambda)\eta}{(\eta + \xi)\xi}$$

Once again this includes jobs which will be lost without successfully completing service. If there are n jobs in the queue then the probability that exactly j of them will successfully complete their service is given by $[\mu/(\mu+\xi)]^j[\xi/(\mu+\xi)]$ for $j = 0, 1, ..., n-1$ and $[\mu/(\mu+\xi)]^n$ when $j = n$. Thus q_j, the probability that there are j jobs in the queue that will successfully complete is given by,

$$q_j = \pi^j \left(p_{1,j} + (1-\pi) \sum_{k=j+1}^{\infty} p_{1,k} \right) \quad \dots \quad j \geq 1 \tag{2.64}$$

and,

$$q_0 = p_{0,0} + p_{1,0} + (1-\pi)[g_1(1) - p_{1,0}] \tag{2.65}$$

where π is the probability that the job in service will complete successfully, given by,

$$\pi = \frac{\mu}{\mu + \xi}$$

From this it is possible to derive an expression for $s(z)$, the probability generating function for the number of jobs in the queue that will successfully complete,

$$s(z) = \sum_{j=0}^{\infty} z^j q_j = p_{0,0} + \sum_{j=0}^{\infty} (\pi z)^j \left(p_{1,j} + (1-\pi) \sum_{k=j+1}^{\infty} p_{1,k} \right)$$

$$= p_{0,0} + g_1(\pi z) + (1-\pi) \sum_{j=0}^{\infty} (\pi z)^j \left(g_1(1) - \sum_{k=0}^{j} p_{1,k} \right)$$

$$= g(\pi z) + \frac{1-\pi}{1-\pi z} g_1(1) - \frac{1-\pi}{1-\pi z} g_1(\pi z)$$

$$s(z) = \frac{1-\pi}{1-\pi z} + \frac{\pi(1-z)}{1-\pi z} g(\pi z)$$

so the average number of successful jobs in the queue is given by,

$$s'(1) = \frac{\mu}{\xi}[1 - g(\pi)]$$

The arrival rate of successful jobs is given by the total arrival rate minus the rate of job loss, namely,

$$\lambda - \frac{\xi\lambda}{\eta+\xi} - \xi g'(1)$$

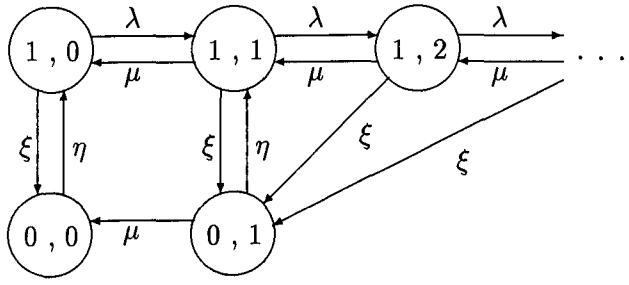
which, by Little's Theorem, gives the average response time to be,

$$W = \frac{\mu(\eta+\xi)[1-g(\pi)]}{\xi[\eta\lambda - \xi(\eta+\xi)\overline{n}]}$$

2.9 Queue lost but head job retained

In the previous model it was assumed that on failure of the server the queue was lost in entirety. Another way to view this would be that the queue had suffered catastrophic failure causing all jobs to be lost, but the server was still functional, albeit without any jobs to serve in the queue. In two of the earlier models the case was considered where the head job is passed to the server (rather than remaining in the queue), and so is lost when the server fails. In this model the case is considered where a failure occurs in the storage device where jobs are queued (e.g. a disk) resulting in all the jobs in the queue being irrevocably lost, except the head job which has been passed on to the (still functional) server. The head job will continue its service during the repair of the disk, if it completes service before repair is completed then the server will remain idle until the arrival of the first job after the queue is repaired, or alternatively if repair is completed first then the system state will return to active with one job 'in' the queue.

A Poisson stream of rate λ enters a queue with an associated server which serve jobs in FIFO order with service times negative exponentially distributed with mean $1/\mu$. The disk suffers breakdowns and subsequent repairs with operative and inoperative periods negative exponentially distributed with means $1/\xi$ and $1/\eta$ respectively, as before. On failure all jobs in the queue, excluding the one currently being served, are lost, and there are no arrivals during the repair period. This is similar to the type of server considered in the previous chapter, however a system of exactly this type does not appear in any of the literature. Intuitively this queue will always be ergodic if $\xi > 0$, since failures will empty the queue (save the head job), thus preventing it from becoming infinite. The system state diagram for this model is illustrated below.



As previously, let $p_{0,j}$ and $p_{1,j}$ be the steady-state probabilities that the server is broken or operative respectively, with exactly j jobs in the queue.

These probabilities satisfy the following balance equations.

$$\eta p_{0,0} = \xi p_{1,0} + \mu p_{0,1} \tag{2.66}$$

$$(\eta + \mu) p_{0,1} = \xi \sum_{j=1}^{\infty} p_{1,j} \tag{2.67}$$

$$(\xi + \lambda) p_{1,0} = \eta p_{0,0} + \mu p_{1,1} \tag{2.68}$$

$$(\xi + \lambda + \mu) p_{1,1} = \eta p_{0,1} + \mu p_{1,2} + \lambda p_{1,0} \tag{2.69}$$

$$(\xi + \lambda + \mu)p_{1,j} = \mu p_{1,j+1} + \lambda p_{1,j-1} \quad \ldots \quad j \geq 2 \tag{2.70}$$

The balance equations can be re-written in terms of the joint queue size probability generating function $g_1(z)$ using the definitions given in (2.5) and (2.9),

$$[\xi z + \lambda z(1-z) - \mu(1-z)]g_1(z) = \eta z p_{0,0} + \eta z^2 p_{0,1} - \mu(1-z)g_1(0) \tag{2.71}$$

where $g_1(z)$ is the queue size probability generating function for the operative periods, such that $g_1(1) = \eta/(\eta + \xi)$, $p_{0,0}$ is the probability that the disk is broken and there is no job in service and $p_{0,1}$ is the probability that the disk is broken and there is a job in service, given by (2.2), (2.66) and (2.67) to be,

$$p_{0,1} = \frac{\xi}{\eta + \mu}\left(\frac{\eta}{\eta + \xi} - g_1(0)\right) \tag{2.72}$$

From (2.66) and (2.72) it follows of course that,

$$p_{0,0} + p_{0,1} = \frac{\xi}{\eta + \xi} \tag{2.73}$$

As in the previous model it is not possible to find an expression for the constant, $g_1(0)$, which can be evaluated at $z = 1$, so once again it is necessary to find a value of z ($| z | < 1$) such that

$$f(z) = \xi z + \lambda z(1-z) - \mu(1-z) = 0 \tag{2.74}$$

Consider the function $f(z)$ at $z = 1$ and $z = 0$, $f(1) = \xi$ and $f(0) = -\mu$, therfore, if $\xi > 0$ and $\mu > 0$, there must be some value z_0 between 0 and 1 for which $f(z_0) = 0$. Consider also the value of $f(z)$ as $z \to \infty$,

$$\lim_{z \to \infty} f(z) \to -\infty$$

Hence the other root of the square polynomial $f(z)$, z_1 must lie in the range $[1,\infty)$. The values of the roots can be found using the well known quadratic roots equation

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

where $a = -\lambda$, $b = (\xi + \mu + \lambda)$ and $c = -\mu$. Clearly,

$$b^2 - 4ac = (\xi + \mu + \lambda)^2 - 4\lambda\mu = (\mu - \lambda)^2 + \xi(2\mu + 2\lambda + \xi)$$

which is always positive, so both roots are real. Since $z_0 \leq z_1$ and $\sqrt{b^2 - 4ac} \geq 0$ then

$$z_0 = \frac{\xi + \mu + \lambda - \sqrt{(\xi + \mu + \lambda)^2 - 4\lambda\mu}}{2\lambda} \tag{2.75}$$

Substituting (2.74) in (2.71) gives,

$$g_1(0) = \frac{\eta z_0 \, (p_{0,0} + z_0 p_{0,1})}{\mu(1 - z_0)} \tag{2.76}$$

And substituting (2.72) and (2.73) in (2.76) gives,

$$g_1(0) = \frac{\eta\xi z_0(\mu + \eta z_0)}{(\eta + \xi)(1 - z_0)[(\eta + \mu)\mu - \eta\xi z_0]} \tag{2.77}$$

If $\xi = 0$ then both roots coincide at $z = 1$, if $\mu = 0$ then $z_0 = 0$, but then there would never be any service.

The average number of jobs can be found by differentiating (2.71),

$$\overline{n} = \frac{(\eta + \xi)(\mu g_1(0) + \eta p_{0,0} + 2\eta p_{0,1}) - (\mu + \xi - \lambda)\eta}{(\eta + \xi)\xi} + p_{0,1}$$

substituting (2.72) and (2.73) gives,

$$\overline{n} = \frac{\eta[\xi(\eta + \xi) - (\eta + \mu)(\mu - \lambda)] + g_1(0)(\eta + \xi)[\mu(\eta + \mu) - \xi(\eta + \xi)]}{\xi(\eta + \xi)(\eta + \mu)}$$

Once again this includes jobs which will be lost without successfully completing service. If there are n jobs in the system then the probability that exactly j of them will successfully

complete their service is given by $[\mu/(\mu+\xi)]^{j-1}[\xi/(\mu+\xi)]$ for $j = 0, 1, ..., n-1$ and $[\mu/(\mu+\xi)]^{j-1}$ for $j = n$. Thus q_j, the probability that there are j jobs in the system that will successfully complete is given by,

$$q_j = \pi^{j-1}\left[p_{1,j} + (1-\pi)\sum_{k=j+1}^{\infty} p_{1,k}\right] \quad \ldots \quad j \geq 2 \tag{2.78}$$

$$q_1 = p_{0,1} + p_{1,1} + (1-\pi)\sum_{k=2}^{\infty} p_{1,k} \tag{2.79}$$

and

$$q_0 = p_{0,0} + p_{1,0} \tag{2.80}$$

where π is the probability that the job in service will complete before the next failure, given by,

$$\pi = \frac{\mu}{\mu+\xi}$$

From (2.78), (2.79) and (2.80) it is possible to derive an expression for $s(z)$, the probability generating function for the number of jobs in the queue that will successfully complete,

$$s(z) = \sum_{j=0}^{\infty} z^j q_j = p_{0,0} + zp_{0,1} + p_{1,0} + \sum_{j=1}^{\infty} \pi^{j-1} z^j \left(p_{1,J} + \sum_{k=j+1}^{\infty} p_{1,k}\right)$$

$$= g_0(z) + p_{1,0} + \frac{1}{\pi}\left(g_1(\pi z) - p_{1,0}\right) + \frac{1-\pi}{\pi}\sum_{j=1}^{\infty}(\pi z)^j \left[g_1(1) - \sum_{k=0}^{j} p_{1,k}\right]$$

$$s(z) = g_0(z) + \frac{(1-z)}{1-z\pi} g_1(z\pi) + \frac{(1-\pi)z}{1-z\pi} g_1(1)$$

so the average number of successful jobs in the queue is given by,

$$s\prime(1) = p_{0,1} + \frac{g_1(1) - g_1(\pi)}{1-\pi}$$

Jobs are only lost from the queue if a failure occurs when there are 2 or more jobs in the system (including the head job), thus the average number of jobs lost when a failure

occurs is given by,

$$\sum_{j=2}^{\infty}(i-1)p_{1,j} = g'_1(1) - [g_1(1) - g_1(0)]$$

Where $p_{1,j}$ is the probability that the queue is operative and contains exactly j jobs (including the head job). Thus the arrival rate of successful jobs is given by the total arrival rate minus the rate of job loss, namely,

$$\frac{\eta\lambda}{\eta+\xi} - \xi\left(g'_1(1) - g_1(1) + g_1(0)\right)$$

which, by Little's Theorem, gives the average response time to be,

$$W = \frac{(\eta+\xi)\left[(1-\pi)p_{0,1} + g_1(1) - g_1(\pi)\right]}{(1-\pi)\left(\eta\lambda - (\eta+\xi)\xi[g'(1) - g_1(1) + g_1(0)]\right)}$$

2.10 Two types of failure where queue remains intact after repair

In this model as well as the state where the server behaves normally (denoted by state 2), there are two states where no service occurs (states 1 and 0). In state 2, jobs arrive into the queue in a Poisson stream of rate λ_2 and are served in FIFO order with service times negative exponentially distributed with mean $1/\mu$. In state 1 jobs arrive into the queue in a Poisson stream rate λ_1 but there is no service. In state 0 there are no arrivals and no service, but the queue remains intact. Transitions between states are negative exponentially distributed random variables with mean time to transition from state i to state j equal to $1/\beta_{i,j}$.

Using the same approach as before, the probability generating function can be written as,1

$$g(z) = g_2(z) + g_1(z) + g_0(z) \tag{2.81}$$

And the balance equations are,

$$[\lambda_2 z(1-z) + \beta_{2,1}z + \beta_{2,0}z - \mu(1-z)]\, g_2(z) = \beta_{1,2}z g_1(z) + \beta_{0,2}z g_0(z) - \mu(1-z)g_2(0) \tag{2.82}$$

$$[\lambda_1(1-z) + \beta_{1,2} + \beta_{1,0}]\, g_1(z) = \beta_{2,1} g_2(z) + \beta_{0,1} g_0(z) \tag{2.83}$$

$$[\beta_{0,2} + \beta_{0,1}]\, g_0(z) = \beta_{2,0} g_2(z) + \beta_{1,0} g_1(z) \tag{2.84}$$

Two of these equations, along with $g(1) = 1$, give the steady state probabilities,

$$g_2(1) = \frac{\beta_{1,2}(\beta_{0,2} + \beta_{0,1}) + \beta_{0,2}\beta_{1,0}}{\beta_{1,2}(\beta_{2,0} + \beta_{0,2} + \beta_{0,1}) + (\beta_{0,1} + \beta_{1,0})(\beta_{2,1} + \beta_{2,0}) + \beta_{0,2}(\beta_{1,0} + \beta_{2,1})}$$

$$g_1(1) = \frac{\beta_{0,1}(\beta_{2,1} + \beta_{2,0}) + \beta_{0,2}\beta_{2,1}}{\beta_{1,2}(\beta_{2,0} + \beta_{0,2} + \beta_{0,1}) + (\beta_{0,1} + \beta_{1,0})(\beta_{2,1} + \beta_{2,0}) + \beta_{0,2}(\beta_{1,0} + \beta_{2,1})}$$

$$g_0(1) = \frac{\beta_{1,0}(\beta_{2,1} + \beta_{2,0}) + \beta_{1,2}\beta_{2,0}}{\beta_{1,2}(\beta_{2,0} + \beta_{0,2} + \beta_{0,1}) + (\beta_{0,1} + \beta_{1,0})(\beta_{2,1} + \beta_{2,0}) + \beta_{0,2}(\beta_{1,0} + \beta_{2,1})}$$

The unknown $g_2(0)$ can be found by adding 2.82 and z times 2.83 and 2.84, to give,

$$\mu g_2(0) = (\mu - \lambda_2 z) g_2(z) - \lambda_1 z g_1(z) \tag{2.85}$$

substituting z=1 gives,

$$\mu g_2(0) = \frac{(\mu - \lambda_2)[\beta_{1,2}(\beta_{0,2} + \beta_{0,1}) + \beta_{0,2}\beta_{1,0}] - \lambda_1[\beta_{0,1}(\beta_{2,1} + \beta_{2,0}) + \beta_{0,2}\beta_{2,1}]}{\beta_{1,2}(\beta_{2,0} + \beta_{0,2} + \beta_{0,1}) + (\beta_{0,1} + \beta_{1,0})(\beta_{2,1} + \beta_{2,0}) + \beta_{0,2}(\beta_{1,0} + \beta_{2,1})} \tag{2.86}$$

Combining (2.81), (2.83) and (2.84) gives,

$$g(z) = \frac{\beta_{2,1}(\beta_{0,2} + \beta_{0,1} + \beta_{1,0}) + \beta_{2,0}(b(z) + \beta_{0,1}) + b(z)(\beta_{0,2} + \beta_{0,1}) - \beta_{0,1}\beta_{1,0}}{b(z)(\beta_{0,2} + \beta_{0,1}) - \beta_{0,1}\beta_{1,0}} g_2(z) \tag{2.87}$$

and (2.85), (2.83) and (2.84) give,

$$g_2(z) = \frac{[b(z)(\beta_{0,2} + \beta_{0,1}) - \beta_{0,1}\beta_{1,0}]\mu g_2(0)}{(\mu - \lambda_2 z)[b(z)(\beta_{0,2} + \beta_{0,1}) - \beta_{0,1}\beta_{1,0}] - \lambda_1 z[\beta_{2,1}(\beta_{0,2} + \beta_{0,1}) + \beta_{2,0}\beta_{0,1}]} \tag{2.88}$$

where,

$$b(z) = \lambda_1(1-z) + \beta_{1,2} + \beta_{1,0}$$

So the average number of jobs can be calculated by differentiating (2.87) and (2.88) to give,

$$g\prime_2(1) = \frac{\mu g_2(0)\lambda_1^2[\beta_{2,1}(\beta_{0,2}+\beta_{0,1}) + \beta_{2,0}\beta_{0,1}(\beta_{0,2}+\beta_{0,1})]}{[(\mu-\lambda_2)[\beta_{1,2}(\beta_{0,2}+\beta_{0,1}) + \beta_{0,2}\beta_{1,0}] - \lambda_1[\beta_{0,1}(\beta_{2,1}+\beta_{2,0}) + \beta_{0,2}\beta_{2,1}]^2}$$

$$+ \frac{(\lambda_2+\lambda_1)[\beta_{1,2}(\beta_{0,2}+\beta_{0,1}) + \beta_{0,2}\beta_{1,0}]}{[(\mu-\lambda_2)(\beta_{1,2}(\beta_{0,2}+\beta_{0,1}) + \beta_{0,2}\beta_{1,0}] - \lambda_1[\beta_{0,1}(\beta_{2,1}+\beta_{2,0}) + \beta_{0,2}\beta_{2,1}]}$$

and,

$$\overline{n} = g\prime(1) = \frac{g\prime_2(1)}{g_2(1)} + \frac{\lambda_1(\beta_{0,2}+\beta_{0,1})\beta_{2,1}(\beta_{0,2}+\beta_{1,0}+\beta_{0,1})(\beta_{2,1}+\beta_{2,0})}{[\beta_{1,2}(\beta_{0,2}+\beta_{0,1}) + \beta_{0,2}\beta_{1,0}]^2} g_2(1)$$

The ergodicity condition is $\mu g_2(1) > \lambda_2 g_2(1) + \lambda_1 g_1(1)$

2.11 Two types of failure, one with entire queue lost, the other where the queue remains intact

As in the previous model the server can be in on of three possible states; the state where the server behaves normally (denoted by state a) or one of two states where no service occurs (states 1 and 0). In state a jobs arrive into the queue in a Poisson stream of rate λ_2 and are served in FIFO order with service times negative exponentially distributed with mean $1/\mu$. In state 1 jobs arrive into the queue in a Poisson stream rate λ_1 but there is no service. In state 0 there are no arrivals and no service and no jobs are retained in the queue. Transitions between states are negative exponentially distributed random variables with mean time to transition from state i to state j equal to $1/\beta_{ij}$.

Using the same approach as before, the probability generating function can be written as,

$$g(z) = g_2(z) + g_1(z) + p_0 \tag{2.89}$$

Where p_0 is the probability of being in state 0 and $g(z)$ is such that $g(1) = 1$. Thus the balance equations are,

$$[\lambda_2 z(1-z) + \beta_{2,1}z + \beta_{2,0}z - \mu(1-z)]\, g_2(z) = \beta_{1,2}z g_1(z) + \beta_{0,2}z p_0 - \mu(1-z)g_2(0) \tag{2.90}$$

$$[\lambda_1(1-z) + \beta_{1,2} + \beta_{1,0}]\, g_1(z) = \beta_{2,1} g_2(z) + \beta_{0,1} p_0 \tag{2.91}$$

$$(\beta_{0,2} + \beta_{0,1})\, p_0 = \beta_{2,0} g_2(1) + \beta_{1,0} g_1(1) \tag{2.92}$$

Two of these equations, along with $g(1) = 1 - p_0$, give the steady state probabilities,

$$g_2(1) = \frac{\beta_{1,2}(\beta_{0,2} + \beta_{0,1}) + \beta_{0,2}\beta_{1,0}}{\beta_{1,2}(\beta_{2,0} + \beta_{0,2} + \beta_{0,1}) + (\beta_{0,1} + \beta_{1,0})(\beta_{2,1} + \beta_{2,0}) + \beta_{0,2}(\beta_{1,0} + \beta_{2,1})}$$

$$g_1(1) = \frac{\beta_{0,1}(\beta_{2,1} + \beta_{2,0}) + \beta_{0,2}\beta_{2,1}}{\beta_{1,2}(\beta_{2,0} + \beta_{0,2} + \beta_{0,1}) + (\beta_{0,1} + \beta_{1,0})(\beta_{2,1} + \beta_{2,0}) + \beta_{0,2}(\beta_{1,0} + \beta_{2,1})}$$

$$p_0 = \frac{\beta_{1,0}(\beta_{2,1} + \beta_{2,0}) + \beta_{1,2}\beta_{2,0}}{\beta_{1,2}(\beta_{2,0} + \beta_{0,2} + \beta_{0,1}) + (\beta_{0,1} + \beta_{1,0})(\beta_{2,1} + \beta_{2,0}) + \beta_{0,2}(\beta_{1,0} + \beta_{2,1})}$$

The method employed in the last model to find the unknown $\mu g_2(0)$ cannot be used here as the elements of (2.92) will not cancel with those in (2.90) and (2.91). Substituting (2.91) in (2.90) gives,

$$(a(z)b(z) - \beta_{2,1}\beta_{1,2}z)\, g_2(z) = [b(z)\beta_{0,2} + \beta_{0,1}\beta_{1,2}]\, z p_0 - b(z)\mu(1-z)g_2(0) \tag{2.93}$$

where, $a(z) = \lambda_2 z(1-z) + \beta_{2,1}z + \beta_{2,0}z - \mu(1-z)$ and $b(z) = \lambda_1(1-z) + \beta_{1,2} + \beta_{1,0}$

Define the function $f(z)$ as,

$$f(z) = a(z)b(z) - \beta_{2,1}\beta_{1,2}z$$

If z_0 exists such that $\mid z_0 \mid < 1$ and $f(z_0) = a(z_0)b(z_0) - \beta_{2,1}\beta_{1,2}z_0 = 0$, then it is possible to equate $\mu g_2(0)$ in (2.93). If there is only one possible value of z_0 then this solution of $\mu g_2(0)$ is unique.

Like the earlier model where the queue was lost on failure, this queue will intuitively always be ergodic, since all catastrophic failures will empty the queue, thus preventing it from becoming infinite. Consider the value of the function $f(z)$ at $z = 0$ and $z = 1$, $f(0) = -\mu(\beta_{1,2} + \beta_{1,0})$ and $f(1) = \beta_{2,0}(\beta_{1,2} + \beta_{1,0}) + \beta_{2,1}\beta_{1,0}$. Clearly $f(0) < 0$ and $f(1) > 0$, by definition, so at least one root exists in the range (0,1). Now consider the value of $f(z)$ as $z \to \infty$,

$$\lim_{z\to\infty} f(z) \to \infty$$

and also,

$$f\left(\frac{\beta_{1,2} + \beta_{1,0} + \lambda_1}{\lambda_1}\right) = \frac{-[\beta_{2,1}\beta_{1,2}(\beta_{1,2} + \beta_{1,0} + \lambda_1)]}{\lambda_1}$$

Therefore one root lies in each of the following ranges, (0,1), $(1,(\beta_{1,2} + \beta_{1,0} + \lambda_1)/\lambda_1)$ and $((\beta_{1,2} + \beta_{1,0} + \lambda_1)/\lambda_1,\infty)$.

The value of z_0, and hence $\mu g_2(0)$, can easily be found by any numerical search method.

Substituting (2.91) in (2.89) gives,

$$g(z) = \frac{g_2(z)(b(z) + \beta_{2,1}) + \beta_{0,1}p_0}{b(z)}$$

Hence,

$$\overline{n} = (1 - p_0)g\prime(1) = \frac{g_2(1)\lambda_1}{(\beta_{1,2} + \beta_{1,0})^2} + \frac{\beta_{1,2} + \beta_{1,0} + \beta_{2,1}}{\beta_{1,2} + \beta_{1,0}}(1 - p_0)g\prime_2(1)$$

where,

$$g\prime_1(1) = \frac{\mu g_2(0)(\beta_{1,2} + \beta_{1,0})}{\beta_{2,0}(\beta_{1,2} + \beta_{1,0}) + \beta_{2,1}\beta_{1,0}} + \frac{p_0\,(\beta_{1,2}\lambda_1[\beta_{0,1}(\beta_{2,1} + \beta_{2,1}) + \beta_{0,2}\beta_{2,1}])}{[\beta_{2,0}(\beta_{1,2} + \beta_{1,0}) + \beta_{2,1}\beta_{1,0}]^2}$$

$$+\frac{p_0\left((\beta_{1,2}+\beta_{1,0})(\lambda_2-\mu)[\beta_{0,2}(\beta_{1,2}+\beta_{1,0})+\beta_{0,1}\beta_{1,2}]\right)}{[\beta_{2,0}(\beta_{1,2}+\beta_{1,0})+\beta_{2,1}\beta_{1,0}]^2}$$

The average arrival rate of successfully completing jobs is given by,

$$\overline{\lambda}_{succ}=\lambda_2 g_2(1)+\lambda_1 g_1(1)-\beta_{2,0}g_2(1)g\prime_2(1)-\beta_{1,0}g_1(1)g\prime_1(1)$$

where

$$g\prime_1(1)=\frac{g\prime_2(1)\beta_{2,1}(\beta_{1,2}+\beta_{1,0})+\lambda_1(\beta_{2,1}g_2(1)+\beta_{0,1}p_0)}{\beta_{1,2}+\beta_{1,0}}$$

so, Little's theorem gives,

$$W=\frac{\overline{n}}{\overline{\lambda}_{succ}}$$

2.12 Numerical Results

Although these models are included here primarily to be used in later chapters to approximate more complicated scenarios, some interesting numerical comparisons can be made.

Consider two servers with identical service, failure and repair processes, one receives jobs during repair, the other does not, and neither lose jobs on failure. If the average arrival rate into both queues is the same, i.e. $\eta\lambda=\eta\lambda_1+\xi\lambda_0$ then it is easy to show that the server not receiving jobs when broken will always out perform the one which does with respect to average response time.

$$W_1=\frac{\eta+\xi}{\eta(\mu-\lambda)}=\frac{\eta+\xi}{\eta\mu-\eta\lambda_1-\xi\lambda_0}<\frac{(\eta+\xi)(\eta\lambda_1+\xi\lambda_0)+\lambda_0\xi(\mu-\lambda_1-\lambda_0)}{(\eta\lambda_1+\xi\lambda_0)(\eta\mu-\eta\lambda_1-\xi\lambda_0)}=W_2 \tag{2.94}$$

where W_1 and W_2 are the the average response time for a server not receiving jobs when broken and a server receiving jobs when broken respectively.

Now consider the two servers with the same arrival streams ($\eta\lambda = \eta\lambda_1 + \xi\lambda_0$), but where the head job is lost on failure. Clearly in this situation the probability that there is at least one job in the queue when failure occurs will affect the average response time and there will also be the additional performance measures of job loss and throughput of successful jobs. However, (2.53) shows that the probability of the queue being empty when the server is working depends on the average arrival rate over all time, i.e.

$$p_{1,0} = \frac{\eta\lambda_1 + \xi\lambda_0}{\eta + \xi}$$

and not on the arrival rates in the working and broken states (λ_1 and λ_0). Thus if, as above, the average arrival rate at each server is the same ($\eta\lambda = \eta\lambda_1 + \xi\lambda_0$) then the job loss in each case will be identical. Clearly this is not an intuitive result as it would be reasonable to expect that varying the balance of the arrivals between the broken and operative periods, whilst keeping the overall arrival rate constant, would affect the job loss. Although the job loss at each server is identical when $\eta\lambda = \eta\lambda_1 + \xi\lambda_0$, there is still a variation in the average response time with arrival rates λ_0 and λ_1. A condition can be obtained in the same way as (2.94) to determine whether it is preferable to send jobs during repair or not. This condition can be simplified greatly if the arrival rate during repair is assumed to be the same as during normal service (i.e. $\lambda_0 = \lambda_1 = \overline{\lambda}$), then (2.45) and (2.57) give the simple condition,

$$(\mu + \xi)^2 > \overline{\lambda}(\eta + \xi)$$

which determines whether the average response time at the server receiving jobs during repair is greater than that for the server not receiving jobs during repair. In most cases this condition will hold, i.e. a server not receiving jobs whilst broken will out perform a

server with identical characteristics which does, but there are scenarios where this is not true. One such scenario is where the servers are quite heavily loaded and the repair rate exceeds the service rate, as illustrated in figure 2.2. Although this is an extreme case it is worth looking at more closely, since it is counter intuitive and not easily explained.

The intuitive explanation for this phenomenon would be that the probability of a job being lost on failure would be higher if more jobs are sent during the operative period, however it was stated earlier that this is not true since the rate of job loss is dependent only on the average arrival rate, and not on the proportion of jobs sent during active and broken periods. A more likely explanation would be that, since the load is high, sending jobs during the inoperative period will decrease the likelihood that the server will be idle when service resumes, furthermore, since the repair rate is high (compared with the service rate), few jobs will arrive during the repair period and will not experience an appreciable delay due to the repair process. This can be shown to be true by finding an expression for the probability that the queue is empty when the server is broken and jobs arrive during inoperative periods:

Eliminating $g_1(z)$ from (2.50) and (2.51) gives,

$$[\eta + \lambda_0(1-z)][\lambda_1 z(1-z) - \mu(1-z) + \xi z] - \xi\eta g_0(z) = \xi(z-1)[\lambda_1(1-z) + \mu + \xi]g_1(0) \tag{2.95}$$

Substituting $z = 0$ in (2.95) gives,

$$g_0(0) = \frac{\xi(\lambda_1 + \mu + \xi)}{\eta(\mu + \xi) + \mu\lambda_0}$$

Thus it is clear that the greater the proportion of jobs which arrive during inoperative periods the less chance there is of the server being idle when repair is complete.

Now consider the case where proportion of time spent broken or operative is constant, but the average length of the operative and inoperative periods varies, as illustrated in figure 2.3. As in figure 2.2 the arrival rate is assumed either to be unaffected by failures in the case where arrivals can occur during repair periods (i.e. $\lambda_1 = \lambda_2$) or equal on a pro rata basis where arrivals do not continue during repair (i.e. $\eta\lambda = \eta\lambda_1 + \xi\lambda_2$).

In section 2.4 it was stated that the only model where its queue size is independent of its operational state is the model where no jobs are lost on failure and there are no arrivals during repair. In the model where no jobs are lost but arrivals continue during repair it is intuitively clear that increasing the length of operative and inoperative periods, whilst keeping the proportion of time spent broken or operative constant, will increase the average size of the queue. This is because the longer inoperative periods will lead to a greater backlog of jobs to be served once repair is complete. This can easily be proved by simple manipulation of (2.31).

In the case where the head job is lost on failure and arrivals continue during repair the number of jobs in the queue will grow during long inoperative periods, in the same way as in the previous case. However, the loss of the head job on failure means that fewer jobs will be in the queue when service resumes than in the previous case. This causes the average response time to be lower than in the previous model, particularly when the failure rate is high, despite an apparent 'wastage' of service time on jobs that are destined not to complete. The same effect is more marked when there are no arrivals during repair. Here the average response time initially increases as the repair and failures rates decrease before levelling out at a value slightly less than for the case where no jobs are lost and there are no arrivals during repair. It might be expected that as the likelihood of failure

decreases that this model would converge to the same value as the case where no jobs are lost, indeed that would be the case if the failure rate decreased and the repair rate was constant. However, in this example the proportion of time that the server is inoperative is constant, hence even when failures are very rare there are still fewer jobs in the queue for the same proportion of time.

2.13 Conclusions

Expressions have been derived for various performance measures for 7 simple single server queueing systems suffering breakdown. Whilst further models involving single M/M/1 queues with breakdowns could easily be defined, the models presented here represent the extent of the realistic systems which can be expressed as simple equations. These models are principally to be used later as approximations to more complex systems, although four of the models have had their performance compared in a given set up to determine whether there are cases when it can be advantageous to send jobs to a broken server. It was found that under a no job loss situation a server not receiving jobs when broken will always perform better than a server which does, given the same overall arrival rate. However, when the models where the head job only is lost are considered, it is found that there are some cases where a server receiving jobs whilst broken can perform better than a server which does not.

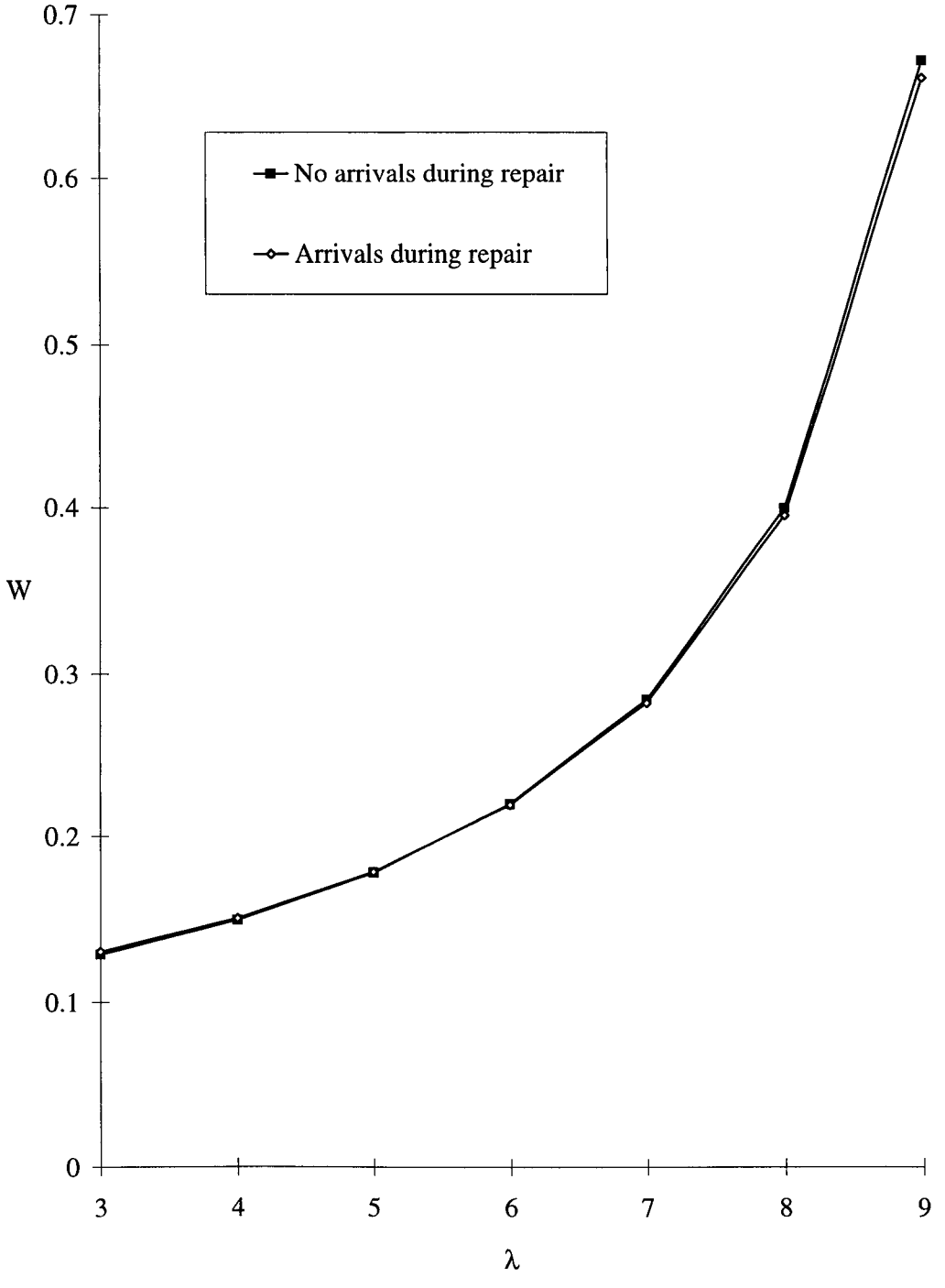


Figure 2.2: Average response time as a function of the average arrival rate for identical servers with different arrival streams

$\mu = 10, \eta = 20, \xi = 1$

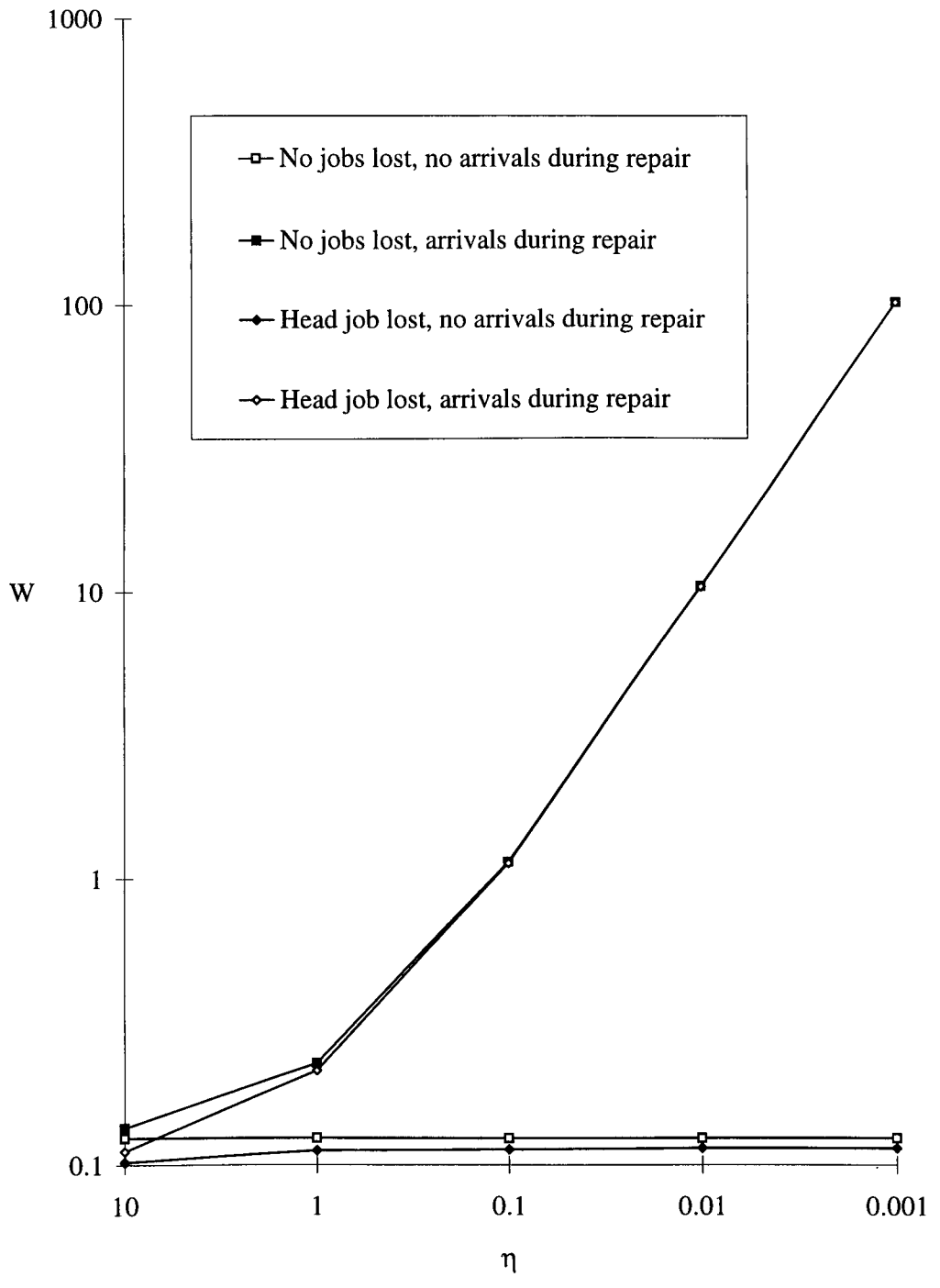


Figure 2.3: Average response time as a function of the repair rate where the proportion of time operative is constant

$\mu = 10$, $\eta = \xi/10$, $\lambda_1 = \lambda_2 = 1$, $\lambda = 1.1$

Chapter 3

A Quasi-Birth-Death Markov Process

3.1 Summary

In many practical situations a system's potential for failure varies in some known fashion that can be monitored. There are several different reasons for the operating state to change, for instance it may become more prone to failure, or be able to give a faster service, or (as in the previous chapter) the arrival rate may change, but in general two or more of these factors will be involved. Here a model is introduced where a server has n possible states of operation.

3.2 A general single server model

Consider a simple M/M/1 queue which has, in general, n distinct states of operation, numbered 1 to n. In state i ($i \leq n$) the queue accepts jobs in a Poisson stream of rate λ_i

and jobs are served in FIFO order with service time exponentially distributed at rate μ_i. Transition from state i to state i' $(i, i' \leq n)$ has no effect on jobs in the queue and takes place with time to transfer exponentially distributed at rate $\beta_{i,i'}$.

There are a number of scenarios which this model can be applied to. In general a change of state from state i to state $i-1$ $(i > 1)$ indicates that the server is experiencing a degradation of service. This may either be that the service rate decreases, the arrival rate increases, transition to a "worse" state is more likely (i.e. $\beta_{i,i'} < \beta_{i-1,i'}$ where $i \leq n$ and $i - 1 > i' \geq 1$), or any combination of these factors. In most practical situations one of these factors will be dominant and the others can, to some extent at least, be manipulated to improve performance. For instance if arrival rate is is determined by some outside factor, it might be desirable to react to an increased arrival rate by similarly increasing the service rate (temporarily freeing more power), however this may cause the system to become more unstable and so increase the possibility of complete failure (no service). Alternatively, if service rate is the dominant factor, it might be desirable to divert a proportion of the arriving jobs elsewhere. In many cases the likelihood of failure is increased by the amount of work a server is required to perform, so the transfer from a given state rates could be dependent on the arrival rate in that state.

This model is clearly one which has many applications and can be used to illustrate several different tradeoffs, it is therefore a very powerful single server model and one for which a solution would be very useful.

The system state at time t is specified by the pair $[I(t), J(t)]$, where $I(t)$ indicates the current state $(I(t) \in \{1, 2, \ldots, n\})$, and $J(t)$ is an integer whose value is the number of jobs in the queue at time t.

The condition for ergodicity of X is that the overall arrival rate is lower than the overall service capacity:

$$\sum_{i=1}^{n} \lambda_i p(i) < \sum_{i=1}^{n} \mu_i p(i) \tag{3.1}$$

where $p(i)$ is the probability that the server is in state i.

In this model the process X is a reducible Markov chain because the arrivals into, and departures from the queue during a small interval $(t, t+\Delta t)$ depend only on the server state and the size of the queue at time t.

The equilibrium distribution of X:

$$p(i,j) = \lim_{t\to\infty} P[I(t) = i\,,\, J(t) = j] \;\;,\;\; i = 1,\ldots,n \;\;,\;\; j = 0,1,\ldots \tag{3.2}$$

Given the probabilities $p(i,j)$, the average size of the queue is obtained from

$$L = \sum_{j=1}^{\infty} j \sum_{i=1}^{n} p(i,j) \tag{3.3}$$

3.3 Queue size distributions

The process X is of the *block tri-diagonal*, or *Quasi-Birth-and-Death* type. Its possible transitions are:

(a) from state (i,j) to state (i',j), if $i \neq i'$ and the transition rate from state i to state i', $\beta_{i,i'}$, is non-zero;

(b) from state (i,j) to state $(i,j+1)$, if the arrival rate in state i, λ_i, is non-zero;

(c) from state (i,j) to state $(i,j-1)$, if $j > 0$ and the service rate in state i, μ_i, is non-zero.

The balance equations for X are best written in vector and matrix form. Define the (row) vector of equilibrium probabilities of all states with j jobs in the queue:

$$\mathbf{v}(j) = [p(1,j), p(2,j), \ldots, p(n,j)] \ , \ j = 0, 1, \ldots \tag{3.4}$$

Let $A = (\beta_{i,i'})$ $(i, i' = 1, \ldots, n)$ be the matrix of instantaneous transition rates corresponding to transitions (a). It is also useful to introduce the diagonal matrix, D_A, whose i'th diagonal element is the i'th row sum of A $(i = 1, \ldots, n)$.

Let B be the diagonal matrix whose i'th diagonal element is equal to λ_i and 0 elsewhere; these elements are the instantaneous transition rates corresponding to transitions (b). Also, let C be the diagonal matrix whose i'th diagonal element is equal to μ_i and 0 elsewhere; these are the instantaneous transition rates corresponding to transitions (c).

When $j > 0$, the vectors (3.4) satisfy the balance equations

$$\mathbf{v}(j)(D_A + B + C) = \mathbf{v}(j)A + \mathbf{v}(j-1)B + \mathbf{v}(j+1)C \ , \ j = 1, 2, \ldots \tag{3.5}$$

For $j = 0$, the equation is slightly different:

$$\mathbf{v}(0)(D_A + B) = \mathbf{v}(0)A + \mathbf{v}(1)C \tag{3.6}$$

In addition, all probabilities must sum up to 1:

$$\sum_{j=0}^{\infty} \mathbf{v}(j)\mathbf{e} = 1 \tag{3.7}$$

where $\mathbf{e}$ is a column vector with n elements, all of which are equal to 1.

The above equations can be solved by several methods, see for example Neuts [82] and Haverkort [35]. Perhaps the best approach is to use *spectral expansion* (see Mitrani, Chakka and Mitra [12, 13, 73, 76]). Rewrite (3.5) in the form

$$\mathbf{v}(j)Q_0 + \mathbf{v}(j+1)Q_1 + \mathbf{v}(j+2)Q_2 = \mathbf{0} \ , \ j = 0, 1, \ldots \tag{3.8}$$

where $Q_0 = B$, $Q_1 = A - D_A - B - C$ and $Q_2 = C$. This is a homogeneous vector difference equation of order 2, with constant coefficients. Associated with it is the characteristic matrix polynomial, $Q(z)$, defined as

$$Q(z) = Q_0 + Q_1 z + Q_2 z^2 \tag{3.9}$$

Denote by z_ℓ and $\boldsymbol{\psi}_\ell$ the *generalised eigenvalues and left eigenvectors* of $Q(z)$. These quantities satisfy

$$\boldsymbol{\psi}_\ell Q(z_\ell) = \mathbf{0} \ , \ \ \ell = 1, 2, \ldots, d \tag{3.10}$$

where $d = degree\{det[Q(z)]\}$.

The eigenvalues do not have to be simple, but it is assumed that if z_ℓ has multiplicity r, then it has r linearly independent left eigenvectors. This is invariably observed to be the model in practice. Under that assumption, any solution of (3.8) is of the form

$$\mathbf{v}(j) = \sum_{\ell=1}^{d} x_\ell \boldsymbol{\psi}_\ell z_\ell^j \ , \ \ j = 0, 1, \ldots \tag{3.11}$$

where x_ℓ ($\ell = 1, 2, \ldots, d$), are arbitrary (complex) constants.

Moreover, since only solutions which can be normalised are acceptable, if $|z_\ell| \geq 1$ for some ℓ, then the corresponding coefficient x_ℓ must be set to 0. Numbering the eigenvalues of $Q(z)$ in increasing order of modulus, the spectral expansion solution of equation (3.8) can be written as

$$\mathbf{v}(j) = \sum_{\ell=1}^{c} x_\ell \boldsymbol{\psi}_\ell z_\ell^j \ , \ \ j = 0, 1, \ldots \tag{3.12}$$

where c is the number of eigenvalues strictly inside the unit disk (each counted according to its multiplicity).

In the numerical experiments carried out with this model, the eigenvalues and eigenvectors of $Q(z)$ have always been observed to be simple, real and positive.

Substituting (3.12), for $j = 0$ and $j = 1$, into (3.6), yields a set of homogeneous linear equations for the unknown coefficients x_ℓ. There are $n - 1$ independent equations in this set (rather than n) because the generator matrix of the Markov process is singular. A further, non-homogeneous equation is provided by (3.7), which now becomes

$$\sum_{\ell=1}^{n} \frac{x_\ell \boldsymbol{\psi}_\ell \mathbf{e}}{1 - z_\ell} = 1$$

These equations can be solved uniquely for the coefficients x_ℓ, if $c = n$. This turns out to be the model when (3.1) is satisfied. Indeed, the ergodicity condition is equivalent to the requirement that $Q(z)$ has exactly n eigenvalues strictly inside the unit disk.

Having determined the coefficients x_ℓ, the average number of jobs in the queue is obtained by substituting (3.12) into (3.3):

$$L = \sum_{\ell=1}^{n} \frac{x_\ell z_\ell \boldsymbol{\psi}_\ell \mathbf{e}}{(1 - z_\ell)^2} \tag{3.13}$$

3.4 Conclusions

The quasi-birth-death Markov model defined in this section is the most general model in this thesis and its solution forms the mainstay of much of the analysis that follows. The applications of the model introduced here are many and varied. Several applications are studied in the following chapters, yet more are outlined in section 8.2.

Chapter 4

Systems of Servers in Parallel where No Jobs are Lost

4.1 Summary

Jobs generated by a single Poisson source can be routed through N alternative gateways, modelled as parallel $M/M/1$ queues. The servers are subject to random breakdowns which leave their corresponding queues intact, but may affect the routeing of jobs during the subsequent repair periods.

The marginal equilibrium queue size distributions are determined by spectral expansion. This can be done, at least in principle, for any number of queues. Several routeing strategies are evaluated and compared empirically. Numerical results, including optimal routeing, are presented and possible generalisations are considered.

4.2 The model

Jobs arrive into the system in a Poisson stream with rate λ. There are N servers, each with an associated unbounded queue, to which incoming jobs may be directed. Server k goes through alternating independent operative and inoperative periods, distributed exponentially with means $1/\xi_k$ and $1/\eta_k$, respectively. While it is operative, the jobs in its queue receive exponentially distributed services with mean $1/\mu_k$, and depart upon completion. When a server becomes inoperative (breaks down), the corresponding queue, including the job in service (if any), remains in place. Services that are interrupted in this way are eventually resumed from the point of interruption. The system model is illustrated in figure 4.1.

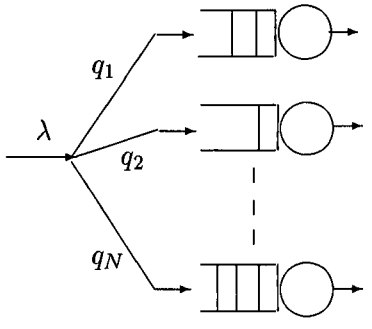


Figure 4.1: A single source split among N unreliable nodes

The *system configuration* at any moment is specified by the subset, σ, of servers that are currently operative (that subset may be empty, or it may be the set of all servers): $\sigma \subset \Omega_N$, where $\Omega_N = \{1, 2, \ldots, N\}$. There are of course 2^N possible system configurations.

The steady-state marginal probability, p_σ, of configuration σ is given by

$$p_\sigma = \prod_{k\in\sigma} \frac{\eta_k}{\xi_k + \eta_k} \prod_{k\in\overline{\sigma}} \frac{\xi_k}{\xi_k + \eta_k} \quad , \quad \sigma \subset \Omega_N \; , \tag{4.1}$$

where $\overline{\sigma}$ is the complement of σ with respect to Ω_N and an empty product is by definition equal to 1. These expressions follow from the fact that servers break down and are repaired independently of each other.

If, at the time of arrival, a new job finds the system in configuration σ, then it is directed to node k with probability $q_k(\sigma)$. These decisions are independent of each other, of past history and of the sizes of the various queues. Thus, a routeing policy is defined by specifying 2^N vectors,

$$\mathbf{q}(\sigma) = [q_1(\sigma), q_2(\sigma), \ldots, q_N(\sigma)] \quad , \quad \sigma \subset \Omega_N \quad , \tag{4.2}$$

such that for every σ,

$$\sum_{k=1}^{N} q_k(\sigma) = 1 \quad .$$

The system state at time t is specified by the pair $[I(t), \mathbf{J}(t)]$, where $I(t)$ indicates the current configuration (the configurations can be numbered, so that $I(t)$ is an integer in the range $0, 1, \ldots, 2^N - 1$), and $\mathbf{J}(t)$ is an integer vector whose k'th element, $J_k(t)$, is the number of jobs in queue k ($k = 1, 2, \ldots, N$). Under the assumptions that have been made, $X = \{[I(t), \mathbf{J}(t)], \, t \geq 0\}$ is an irreducible Markov process. The condition for ergodicity of X is that, for every queue, the overall arrival rate is lower than the overall service capacity:

$$\lambda \sum_{\sigma\subset\Omega_N} p_\sigma q_k(\sigma) \; < \; \mu_k \frac{\eta_k}{\xi_k + \eta_k} \; , \; k = 1, 2, \ldots, N \; . \tag{4.3}$$

When the routeing probabilities depend on the system configuration, the process X is not separable (i.e., it does not have a product-form solution). Consequently, the problem

of determining its equilibrium distribution is intractable for $N > 2$. In the case $N = 2$, a solution may be possible, but both the mathematical analysis and the implementation would be difficult, an outline of the solution required was given by Thomas and Mitrani [101]. On the other hand, the quantities of principal interest are expressed in terms of averages only; they are the steady-state mean queue sizes, L_k, and the the overall average response time, W, given by

$$W = \frac{1}{\lambda} \sum_{k=1}^{N} L_k \ . \tag{4.4}$$

To determine those performance measures, it is not necessary to know the joint distribution of all queue sizes; the marginal distributions of the N queues in isolation are sufficient. Unfortunately, the isolated queue processes, $\{J_k(t)\ ,\ t \geq 0\}$ $(k = 1, 2, \ldots, N)$, are not Markov. However, the performance measures can be determined by studying the stochastic processes $Y_k = \{[I(t), J_k(t)]\ ,\ t \geq 0\}$ $(k = 1, 2, \ldots, N)$, which model the joint behaviour of the system configuration and the size of an individual queue. The state space of Y_k is infinite in one dimension only, which simplifies the solution considerably and makes it tractable for reasonably large values of N. The important observation here is that Y_k is an irreducible Markov process, for every k. This is because the arrivals into, and departures from queue k during a small interval $(t\ , t+\Delta t)$ depend only on the system configuration and the size of queue k at time t, and not on the sizes of the other queues.

The next task, therefore, is to find the equilibrium distribution of Y_k :

$$p_k(i,j) = \lim_{t \to \infty} P[I(t) = i\ ,\ J_k(t) = j]\ ,$$

$$i = 0, 1, \ldots, 2^N - 1\ \ ,\ \ j = 0, 1, \ldots\ . \tag{4.5}$$

Given the probabilities $p_k(i,j)$, the average size of queue k is obtained from

$$L_k = \sum_{j=1}^{\infty} j \sum_{i=0}^{2^N-1} p_k(i,j) \,. \tag{4.6}$$

The process Y_k is of the *block tri-diagonal*, or *Quasi-Birth-and-Death* type described in the previous chapter and can easily be solved in exactly the same manner to find the probabilities $p_k(i,j)$.

4.3 Evaluation of scheduling strategies

In order to reduce the number of parameters that have to be given values when defining the routeing strategy, we shall evaluate and compare several strategies based on a single routeing vector, $\mathbf{q} = (q_1, q_2, \ldots, q_N)$. In each case, the optimisation problem is to chose the elements of that vector so as to minimise the average response time, given by (4.4).

1. *The fixed strategy.*

The most straightforward way of splitting the incoming stream is to send each job to node k with probability q_k, regardless of the system configuration. Then the N nodes are independent of each other; node k can be considered in complete isolation, as an M/M/1 queue with breakdowns and repairs. In this simple case, there is a well known explicit formula for the average queue size (see chapter 2):

$$L_k = \frac{\lambda q_k[(\xi_k + \eta_k)^2 + \xi_k \mu_k]}{(\xi_k + \eta_k)[\eta_k \mu_k - \lambda q_k(\xi_k + \eta_k)]} \,. \tag{4.7}$$

2. *The selective strategy.*

Intuitively, it seems better not to send jobs to nodes where the server is inoperative, unless that is unavoidable. This suggests the following strategy: If the subset of operative

servers in the current system configuration is σ, and that subset is non-empty, send jobs to node k only if $k \in \sigma$, with probability proportional to q_k :

$$q_k(\sigma) = \frac{q_k}{\sum_{\ell \in \sigma} q_\ell} \quad , \quad k \in \sigma \quad .$$

If σ is empty (i.e. all servers are broken), send jobs to node k with probability q_k ($k = 1, 2, \ldots, N$).

3. *The fixed(m) strategy.*

It is possible that some nodes are unable, under any circumstances, to receive jobs when broken. Suppose that the last $N - m$ nodes are of this type ($m > 0$), and that jobs are sent to the first m nodes regardless of their state. Thus, when the system configuration is σ, an incoming job can be directed to any node k for which $k \leq m$ or $k \in \sigma$, or both, with probability

$$q_k(\sigma) = \frac{q_k}{\sum_{\ell \in \{1,2,\ldots,m\} \cup \sigma} q_\ell} \quad , \quad (k \leq m) \vee (k \in \sigma) \quad .$$

4. *The selective(m) strategy.*

This strategy, like the selective one, does not send jobs to broken nodes unless that is unavoidable. In addition, the last $N - m$ nodes are completely unable to receive jobs when broken ($m > 0$). In other words, if the system configuration is σ, and $\sigma \neq \emptyset$, an incoming job is directed to node k, only if $k \in \sigma$, with probability proportional to q_k :

$$q_k(\sigma) = \frac{q_k}{\sum_{\ell \in \sigma} q_\ell} \quad , \quad k \in \sigma \quad .$$

If σ is empty, the job is sent to one of the first m nodes, with probability

$$q_k(\sigma) = \frac{q_k}{\sum_{\ell=1}^{m} q_\ell} \quad , \quad k = 1, 2, \ldots, m \quad .$$

Clearly, the fixed strategy is a special case of the fixed(m) one, when $m = N$. Similarly, the selective strategy is a special case of the selective(m) one, when $m = N$. All strategies except the fixed are evaluated by the spectral expansion method.

Intuitively it would seem that, for a given routeing vector, the selective strategy should perform better than the others, since it appears to make the best use of all servers. The fixed strategies may be expected to perform poorly, since they largely or completely disregard the current availability of servers. When the majority of the servers are quite reliable, the performance of a selective(m) strategy should not depend much on m and should resemble that of the selective strategy (since the only differences arise when all servers are broken).

This intuition is confirmed by the results in figure 4.2, where a 3-node model is solved under the three fixed and three selective scheduling strategies. In all cases, the overall average response time, W, is plotted against the job arrival rate. The nodes have different characteristics (see caption), but no advantage is taken of those differences. The routeing vector is $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, i.e. the *a-priori* splitting of the input stream is into three equal sub-streams.

There is a clear separation between the two groups of curves; every selective strategy out-performs every fixed one. The selective strategies are quite close, although the servers are not very reliable. Within the fixed strategies, it is worth noting that fixed(1) and fixed(2) start off better than fixed, but become worse when the load increases. This is because the prohibition on sending jobs to some servers when they are broken helps to balance the load at low arrival rates, but saturates the other servers when the load is high. If, instead of keeping the routeing vector constant, it is optimised for each value of λ, then

the corresponding plots do not cross: fixed(1) becomes uniformly better than fixed(2), which in turn becomes better than fixed.

Despite their plausibility, the above remarks are not universally valid. In particular, it is possible to construct examples where the fixed strategy performs better than the selective (e.g. $N = 2$, $\lambda = 10$, $\mu_1 = 30$, $\mu_2 = 10, \xi_1 = 100$, $\xi_2 = 1$, $\eta_1 = 100$, $\eta_2 = 100000$; admittedly, that example is rather contrived, with one fast and fairly unreliable server, while the other is slower and extremely reliable).

The rest of the experiments illustrate various aspects of optimal routeing, which is discussed further in the following chapter.

Figure 4.3 concerns a 2-node system where the routeing vector, $(q, 1-q)$, is varied on the range $0 \leq q \leq 1$ (remember that that vector is used in making routeing decisions only when both servers are operative or, in the case of the selective strategy, when both are broken). The average response time is plotted against q. The system parameters (see caption) are such that each server is operative approximately 90% of the time, while server 1 is 50% faster than server 2. The figure suggests the following observations, of which the first is obvious (from the definitions of the strategies), the next two are quite intuitive, and the last is somewhat counter-intuitive:

(a) When $q = 1$, the fixed and fixed(1) strategies are identical, as are the selective and selective(1) ones; when $q = 0$, the fixed(1) and selective(1) strategies are identical.

(b) The curves corresponding to the selective strategies are not only lower, but also *flatter* than those of the fixed ones; in other words, the selective strategies are less sensitive to changes in the routeing vector.

(c) For the fixed and two selective strategies, the best routeing vector sends the majority of the jobs (70% - 80%) to the faster server.

(d) For the fixed(1) strategy, it is best to send fewer jobs (40%) to the faster server than to the slower one.

To explain (d), note that under the fixed(1) strategy, node 1 is obliged to receive all jobs whenever server 2 is broken, regardless of its own state. This load should be compensated by sending it fewer jobs when there is a choice, i.e. when both servers are operative.

Figure 4.4 shows the performance of a 5-node system as a function of the job arrival rate, under an approximately optimal routeing vector. For each value of λ, a gradient search method was used to get close to the optimal vector, and the corresponding value of the average response time was plotted. The parameters are chosen so that the faster servers are also more reliable. As in figure 4.2, there is almost no difference between the selective strategies. However, the fixed strategy curves no longer cross each other. The general conclusion concerning those strategies seems to be that the more one avoids sending jobs to broken servers, the better the performance that can be achieved, provided that an appropriate routeing vector is employed.

It was shown in chapter 2 that the performance of a server can degrade significantly as the length of breakdowns increases, even when the server is fairly reliable. Figure 4.5 shows the performance of a fairly reliable 2-node system as a function of the routeing vector $(q, 1-q)$, where one node is significantly faster and more reliable than the other. The reliability of this system is of the same order as figure 4.2, but the repair and failure rates are about a factor of ten less (proportionally to the other characteristics). The effect of this is not only an across the board increase in the average response time, but also a dramatic

steepening of the curves such that the fixed and fixed m=1 curves are not finite over the given limited range of q. Clearly this may add to the difficulty of finding an optimal routeing vector and increases the penalty caused when a system is not optimised. Figure 4.6 takes this argument one step further by considering the performance of a symmetrical 2-node system as a function of the repair rate whilst keeping the proportion of time spent broken a constant. The figure shows the same effect observed in figure 1.3, namely that the continued arrival of jobs during repair means that the average queue size (and hence average response time) increases significantly as the repair rate decreases. Predictably this effect is much more marked for the fixed routeing strategy than the selective as the rate of arrivals during repair is much greater in the fixed strategy case.

A numerical search for the optimal routeing vector is expensive, and rapidly becomes more so when the number of nodes increases. It is desirable, therefore, to find a good heuristic that avoids the search and yet produces a nearly optimal performance. One candidate for such a heuristic is the following: Assign to node i a weight, w_i, given by

$$w_i = \frac{\mu_i \eta_i}{\xi_i + \eta_i} \quad , \quad i = 1, 2, \ldots, N \, .$$

This is the available service capacity of server i (the average amount of service it can provide per unit time). Let the i^{th} element of the routeing vector be

$$q_i = \frac{w_i}{\sum_{j=1}^{N} w_j} \quad , \quad i = 1, 2, \ldots, N \, .$$

Thus the suggestion is to ignore the job arrival rate and simply split the input stream in proportion to the available service capacities.

In figure 4.7, the performance of the above heuristic is compared to that of the optimal routeing vector (which does depend on λ), and also to the 'dumb' splitting based on the

vector $(\frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N})$. The experiment is carried out on a 5-node system under the selective routeing strategy. The servers have the same breakdown and repair characteristics (about 90% operative), but different speeds. The average response time is plotted against λ. It can be seen that, while the heuristic is very close to the optimal performance throughout the range of arrival rates, the equal splitting clearly fails to balance the loads at the different servers. The penalty of not using a good routeing vector can be very large.

Figure 4.8 shows the accuracy of the heuristic in predicting the optimal routeing weights for both the fixed(m) and selective(m) strategies in a 5 node system. Clearly in the selective(m) case the heuristic performs well for all values of m and as we would expect the simple fixed strategy gives a very good approximation. However, for the fixed(m) case where m¡N the heuristic prediction rapidly diverges from the optimal solution, and for lower values of m gives a very poor approximation indeed. It is also worth observing here that the selective-m minima and the fixed-m minima converge as m decreases. This is because as m decreases more servers refuse to accept jobs when they breakdown. In the selective-m case this means that in the relatively rare event of all the servers being broken, more jobs will be sent to the remaining servers which will accept jobs, thus causing a greater backlog of work at those servers, and hence a slight rise in the average response time. In the fixed-m case this means that fewer jobs will be directed to the queues of broken servers, hence reducing m improves the performance in the fixed-m strategy as more servers become *selective*.

Unfortunately, the fine performance of the heuristic under the selective routeing strategy is not replicated under the fixed ones. In particular, it performs very poorly with the fixed(m) strategy for small values of m. Another heuristic, better able to handle those

strategies, is needed, this problem is tackled in the following chapter.

Clearly most of the effects illustrated here are made more extreme by the choice of parameters and a more reliable system would not exhibit such diverse characteristics, however these effects do exist and are worthy of consideration.

4.4 Conclusions

The system considered here has a property which may loosely be described as *quasi-separability*. An individual node can be analysed in isolation of the others, provided that the full server configuration is included as a state variable. Because of that property, one can determine exactly the performance measures in models with more than two nodes. It is also possible to optimise the splitting of the input stream among the nodes, under different routeing policies. However, such an optimisation involves a search in a multidimensional space, together with the solution of many instances of the model. Computationally, this can be very expensive. A simple heuristic has been proposed, that appears to work well for selective routeing policies, but not for fixed ones. Further progress can be made either by discovering more generally applicable heuristics, or by developing fast approximate solutions whose complexity does not grow exponentially with N. Both these avenues of further research are worth pursuing.

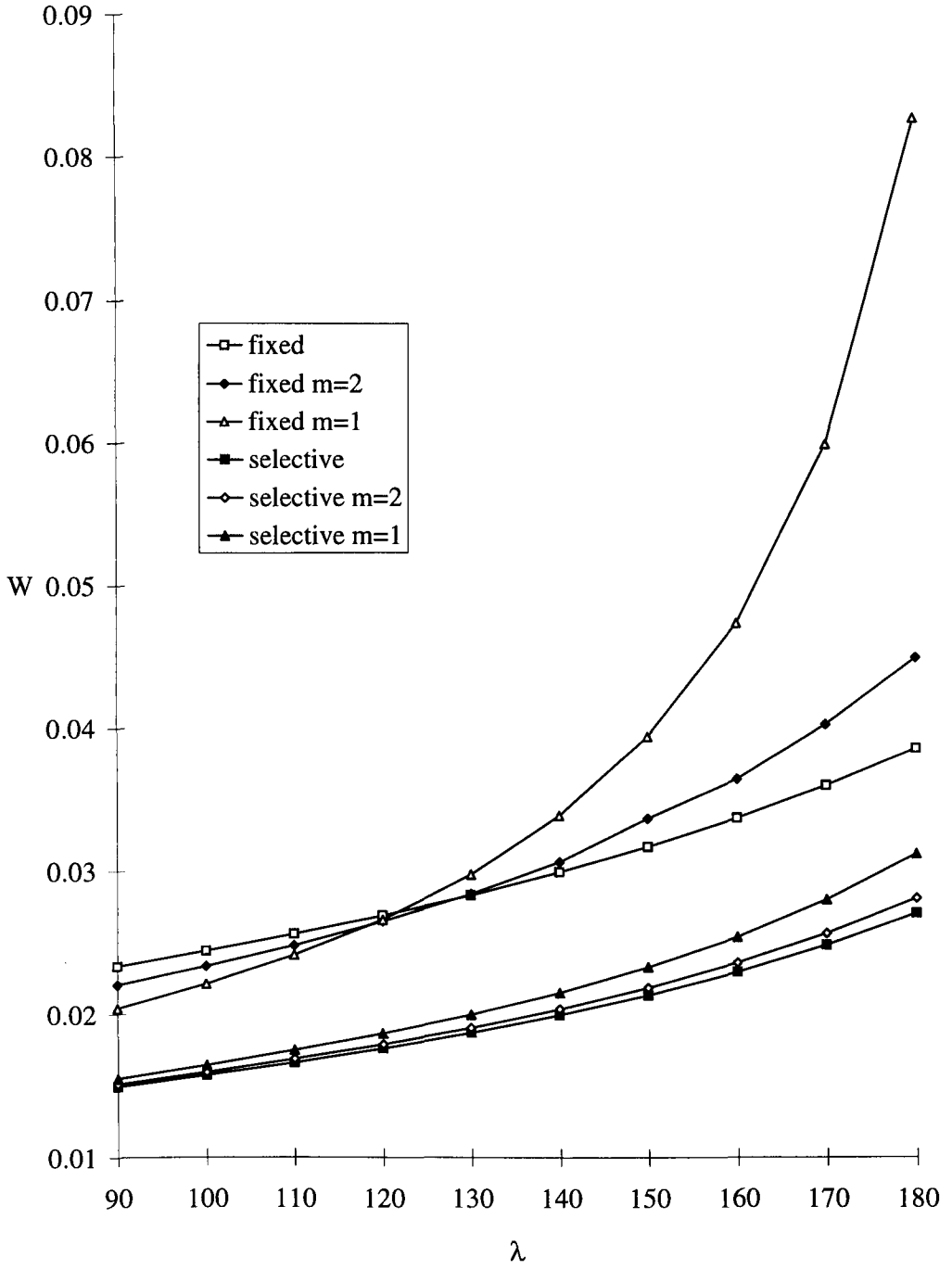


Figure 4.2: Average response time as a function of the job arrival rate.

$N = 3$, $\mu_1 = 150$, $\mu_2 = 170$, $\mu_3 = 190$, $\xi_1 = 20$, $\xi_2 = 30$, $\xi_3 = 40$, $\eta_1 = \eta_2 = \eta_3 = 50$

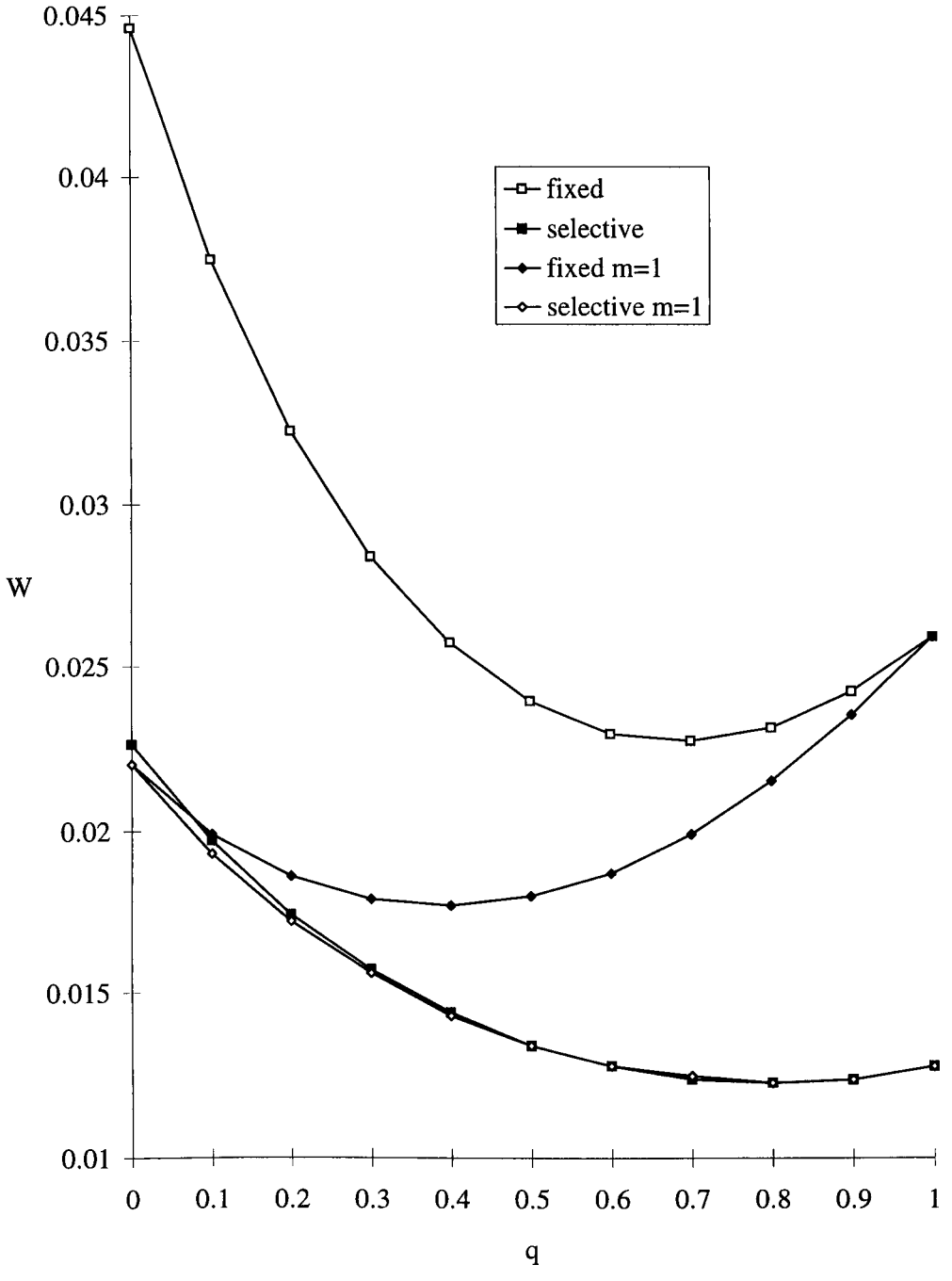


Figure 4.3: Average response time in a 2-node system

as a function of the routeing vector $(q, 1-q)$.

$\lambda = 50$, $\mu_1 = 150$, $\mu_2 = 100$, $\xi_1 = \xi_2 = 1$, $\eta_1 = \eta_2 = 10$

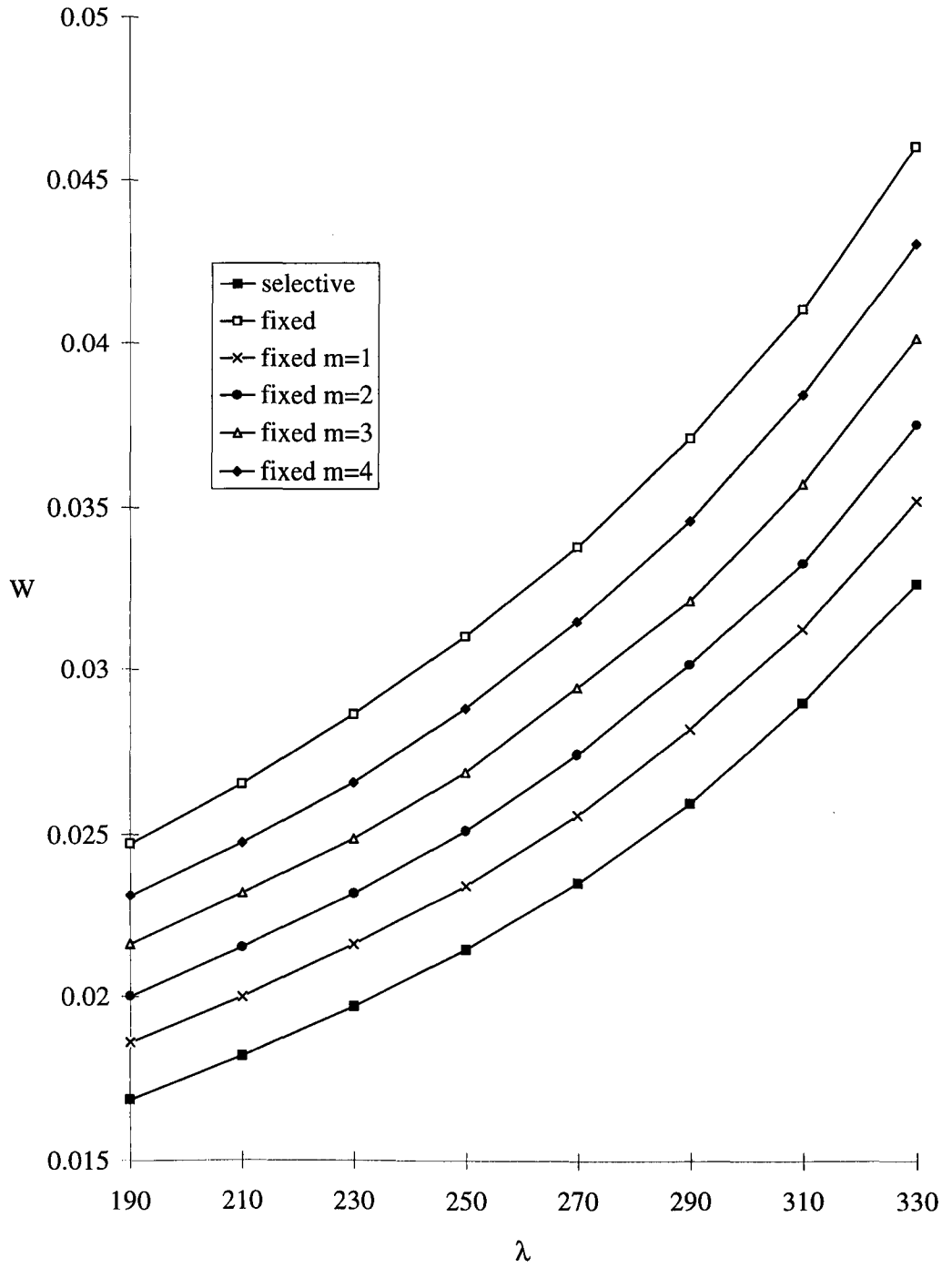


Figure 4.4: Optimised average response time as a function of the job arrival rate.

$N = 5$, $\mu_1 = 150$, $\mu_2 = 160$, $\mu_3 = 170$, $\mu_4 = 180$, $\mu_5 = 190$,

$\xi_1 = \xi_2 = \xi_3 = \xi_4 = \xi_5 = 50$, $\eta_1 = 50$, $\eta_2 = 60$, $\eta_3 = 70$, $\eta_4 = 80$, $\eta_5 = 100$

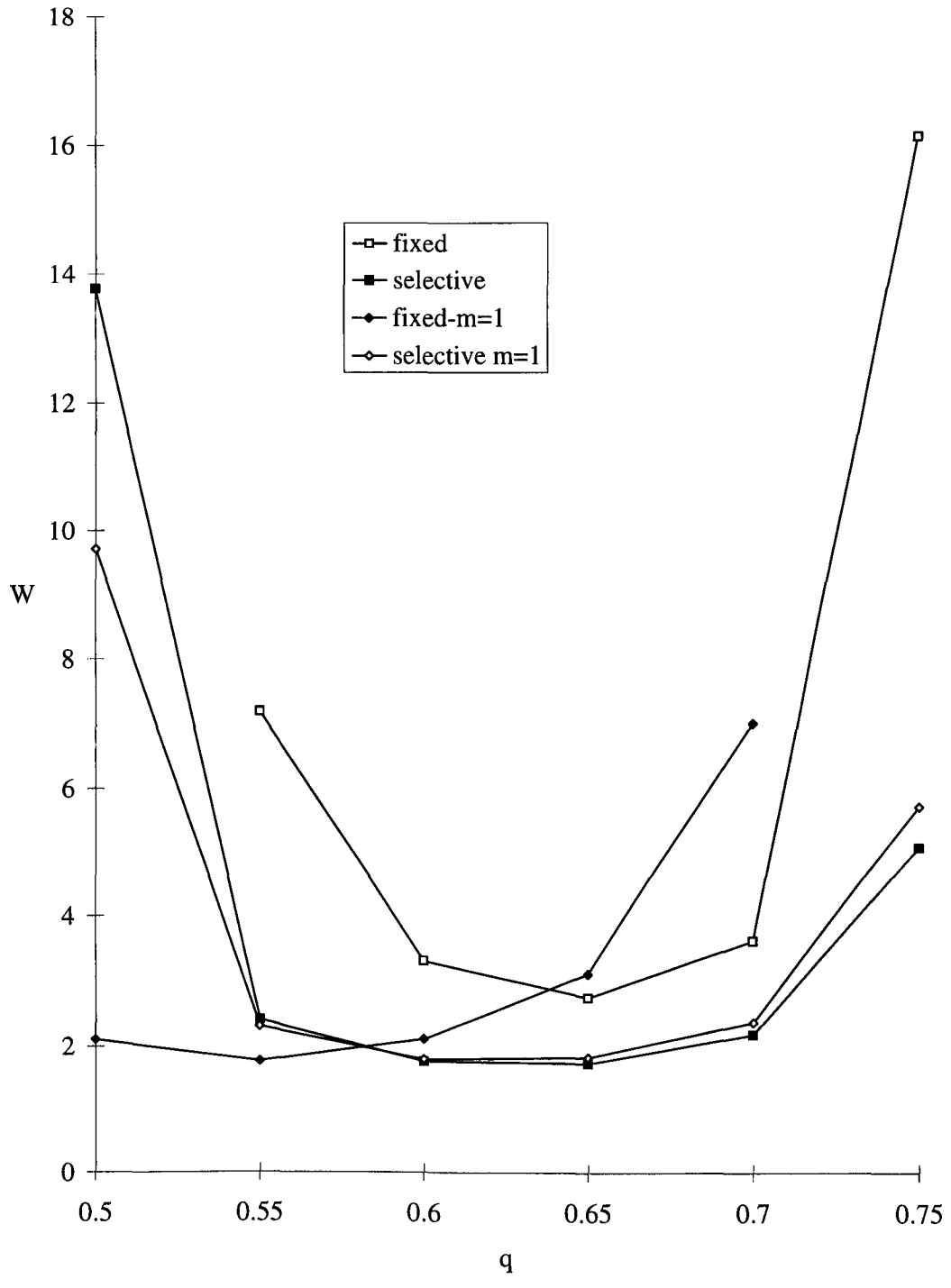


Figure 4.5: Average response time as a function of the routeing vector $(q, 1-q)$.

$N = 2$, $\lambda = 15$, $\mu_1 = 12$, $\mu_2 = 8$, $\eta_1 = 0.2$, $\eta_2 = 0.1$, $\xi_1 = \xi_2 = 0.01$

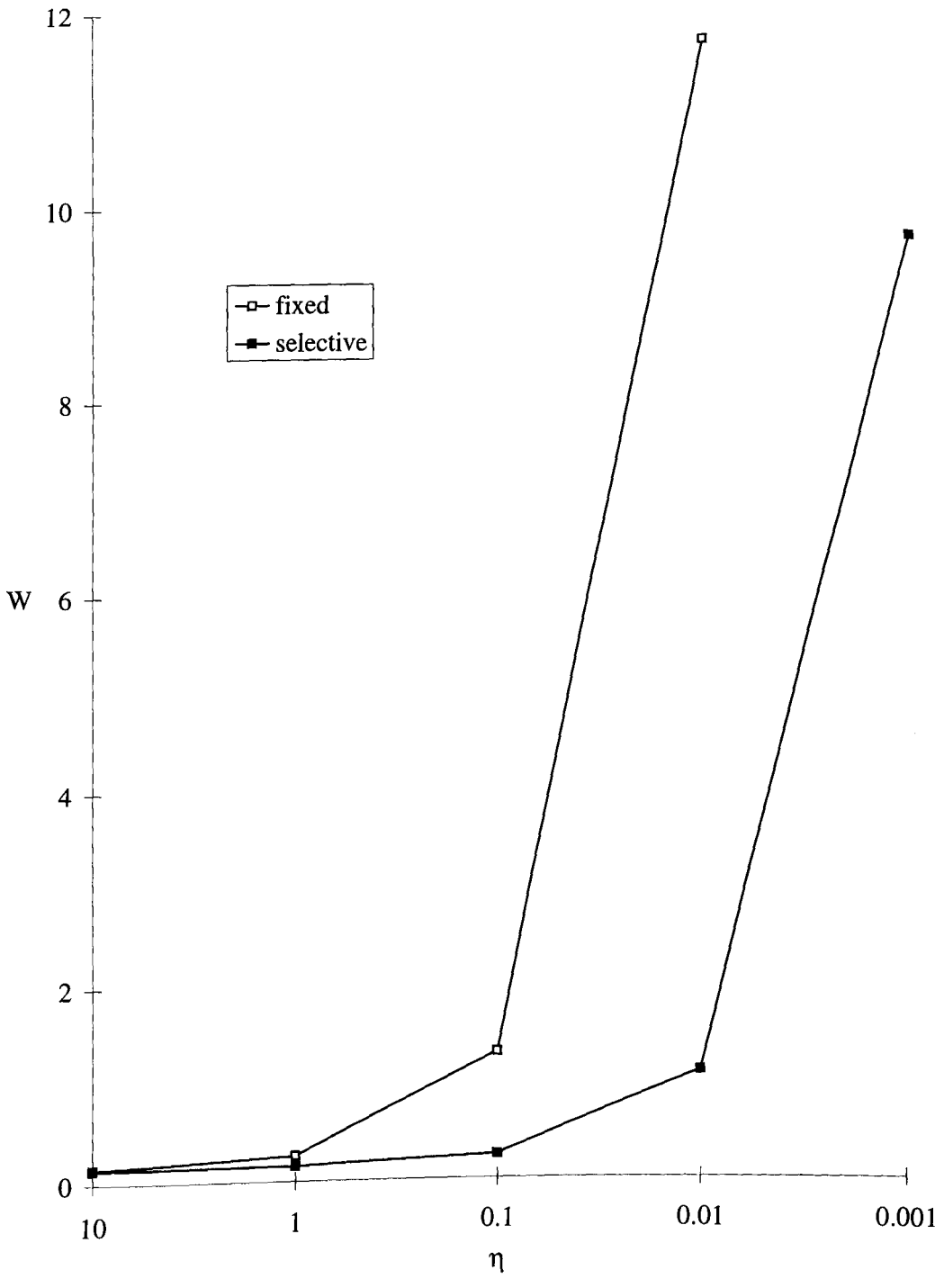


Figure 4.6: Average response time as a function of the repair rate

where the proportion of time operative is constant

$N = 2, \lambda = 4, \mu_1 = \mu_2 = 10, \eta_i = \xi_i/10, i = 1, 2$

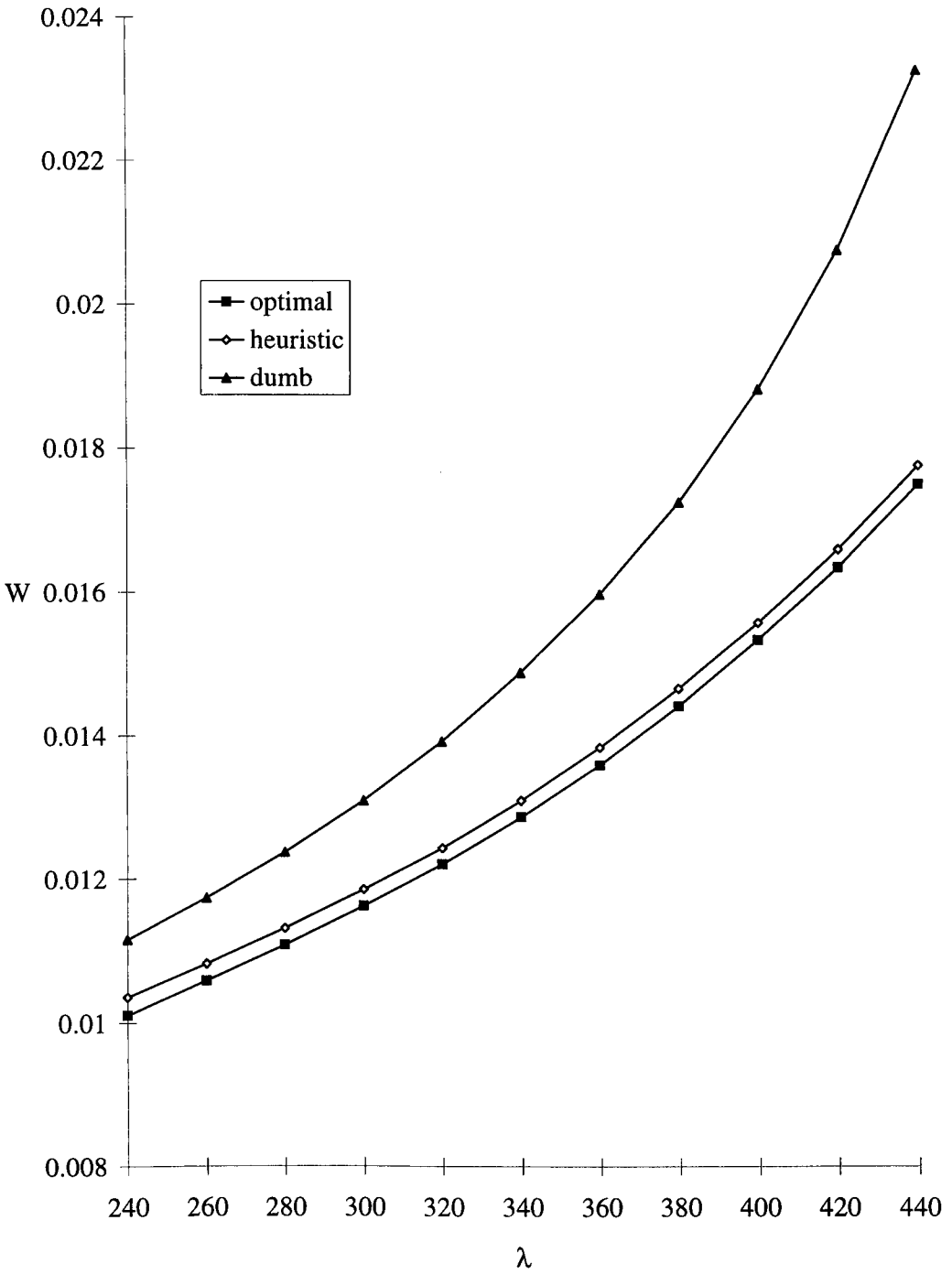


Figure 4.7: Optimised average response time as a function of the job arrival rate.

$N = 5$, $\mu_1 = 150$, $\mu_2 = 160$, $\mu_3 = 170$, $\mu_4 = 180$, $\mu_5 = 190$,

$\xi_1 = \xi_2 = \xi_3 = \xi_4 = \xi_5 = 50$, $\eta_1 = 50$, $\eta_2 = 60$, $\eta_3 = 70$, $\eta_4 = 80$, $\eta_5 = 100$

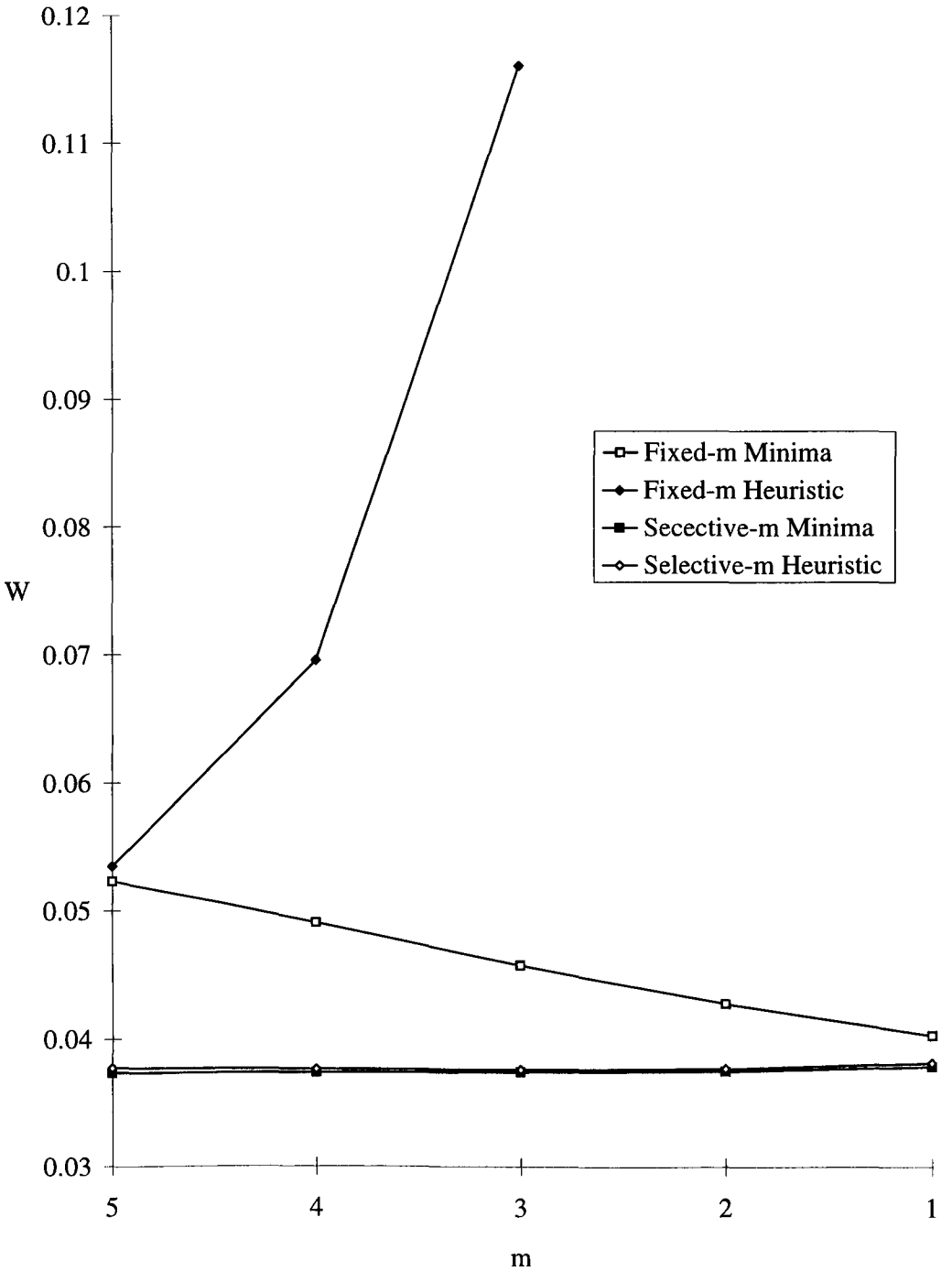


Figure 4.8: Performance of heuristic and optimal routeing, for different strategies.

$N = 5$, $\lambda = 350$, $\mu_1 = 150$, $\mu_2 = 160$, $\mu_3 = 170$, $\mu_4 = 180$, $\mu_5 = 190$,

$\xi_1 = \xi_2 = \xi_3 = \xi_4 = \xi_5 = 50$, $\eta_1 = 50$, $\eta_2 = 60$, $\eta_3 = 70$, $\eta_4 = 80$, $\eta_5 = 100$

Chapter 5

Approximate Solution of Systems of Parallel Servers

5.1 Summary

In the previous chapter a model of servers with queues in parallel was presented and a method to obtain exact numerical solution of certain performance measures was derived. It was observed that there are cases where the exact solution was impractical due to the number of system states which have to be considered, this is particularly true when an optimal solution, with respect to the job share vector $q(\sigma)$, is to be obtained. In this chapter a series of approximations are considered, both from the view of predicting the optimal job share and in approximating the performance measures themselves,. In order to compare the exact and approximate solutions a number of numerical results are presented and the limitations of the numerical solutions are discussed. A brief explanation of the optimisation routines is also included here.

5.2 Simple Approximations

Here an attempt is made to improve the heuristic by finding an approximate, but much faster solution of the model. The idea is to treat node i as an isolated single server queue modulated by a two-state Markov process. During operative periods, distributed exponentially with mean $1/\xi_i$, jobs arrive in a Poisson stream at rate λ_{i1}, and are served at rate μ_i. During inoperative periods, distributed exponentially with mean $1/\eta_i$, jobs arrive in a Poisson stream at rate λ_{i0}, and the service rate is 0. For a given strategy and routeing vector, the two arrival rates are easily determined. Let $\Omega(i)$ be the set of all server configurations in which server i is operative, and $\overline{\Omega(i)}$ be the set of all configurations in which it is inoperative. Then

$$\lambda_{i1} = \frac{\xi_i + \eta_i}{\eta_i} \lambda \sum_{\sigma \subset \Omega(i)} p_\sigma q_i(\sigma) ,$$

$$\lambda_{i0} = \frac{\xi_i + \eta_i}{\xi_i} \lambda \sum_{\sigma \subset \overline{\Omega(i)}} p_\sigma q_i(\sigma) ,$$

where the probabilities p_σ are given by,

$$p_\sigma = \prod_{k \in \sigma} \frac{\eta_k}{\xi_k + \eta_k} \prod_{k \in \overline{\sigma}} \frac{\xi_k}{\xi_k + \eta_k} \quad , \quad \sigma \subset \Omega_N , \tag{5.1}$$

where $\overline{\sigma}$ is the complement of σ with respect to Ω_N and an empty product is by definition equal to 1.

Thus the approximation consists of replacing a modulating process with 2^N states (all possible server configurations), by one with just 2 states. It should be pointed out that this approximation affects only the arrival process, not the services. Moreover, in the case of the fixed routeing strategy, the approximation coincides with the exact solution. The two arrival rates are then equal: $\lambda_{i1} = \lambda_{i0} = \lambda q_i$.

Under the simplifying assumption, it is not difficult to derive a closed-form solution for the isolated node i. The average number of jobs in it is given by

$$L_i = \frac{\eta_i\lambda_{i1} + \xi_i\lambda_{i0} + \frac{\xi_i}{\xi_i+\eta_i}\lambda_{i0}(\mu_i + \lambda_{i0} - \lambda_{i1})}{\eta_i\mu_i - \eta_i\lambda_{i1} - \xi_i\lambda_{i0}} . \tag{5.2}$$

Note that if $\lambda_{i0} = 0$, i.e. if node i does not accept jobs while broken, then (5.2) reduces to the standard result for the average queue size in an M/M/1 queue with parameters (λ_{i1}, μ_i).

A slight improvement can be made to this approximation for the selective and selective(m) strategies, without a significant increase in workload, by using the two stage failure model defined in section 2.10. Here state 2 will be equivalent to the operative state, state 1 will be the state where all servers are broken and state 0 will be the remaining states where node i is broken. There are no arrivals or service in state 0, and no service in state 1. The arrival rate in state 1 will be simply $q_i\lambda$ and the service in state 2 is μ_i. The arrival rate in state 2 is calculated in exactly the same way as above, i.e.,

$$\lambda_{ia} = \frac{\xi_i + \eta_i}{\eta_i}\lambda \sum_{\sigma\subset\Omega(i)} p_\sigma q_i(\sigma) ,$$

The transition rates from state 1 are simple, namely, $\beta_{ba} = \eta_i$, $\beta_{bc} = \sum_{j=1,j\neq i}^{N} \eta_j$

The transition rates from states 2 and 0 are based on the occupation probabilities p_σ, so,

$$\beta_{ab} = \xi_i p_{(i)} \Big/ p_a$$

$$\beta_{ac} = \xi_i (p_a - p_{(i)}) \Big/ p_a$$

$$\beta_{cb} = \left(\sum_{j=1,j\neq i}^{N} \xi_j p_{(j)} \right) \Big/ p_c$$

and,

$$\beta_{ca} = \eta_i$$

where, $p_a = \frac{\eta_i}{\eta_i+\xi_i}$ and, $p_c = \frac{\xi_i}{\eta_i+\xi_i} - \prod_{j=1}^{N} \frac{\xi_j}{\eta_j+\xi_j}$

The following optimisation procedure is now suggested: for a given strategy, find the routeing vector which minimises the *approximate* average response time, W_{approx}. The search for that vector is considerably facilitated by the ease of computing W_{approx}.

This procedure performs extremely well, not only for the selective strategy (where the crude heuristic is already quite good), but also for the various fixed(m) and selective(m) strategies. The exact value of W computed after the approximate optimisation is practically indistinguishable from that obtained by optimising exactly. The relative error is much less than 1%, and would not show up on a figure.

Another question of interest concerns the accuracy of the approximation itself, as opposed to that of its optimal routeing vector. A comparison between the exact and approximate values of W, in the context of a 5-node system under several routeing strategies, is illustrated in figures 5.1 and 5.2. In figure 5.1, the fixed(4) and the selective strategies are evaluated for different values of λ and the corresponding optimal routeing vector. In all cases, the approximation underestimates the exact response time, since the *bursty* nature of the arrivals in the exact solution is smoothed out in the approximation. The relative error is greater for the selective strategy than for the fixed one. These observations are not surprising, since the approximation reduces the variability of the arrival stream, and that reduction is greater for the selective strategy. Even the larger error does nor exceed 10%.

Figure 5.2 shows the effect of changing m in the fixed(m) and selective(m) strategies. In the former, the variability of the arrival stream increases when m decreases, and so the accuracy of the approximation decreases. The influence of m on the selective strategies is

negligible because in this system the probability that all servers are broken is very small. Again, the error is on the order of 10% or less. As in the previous chapter the parameters used in these figures are for a very unreliable system and if more realistic parameters are used then the approximations become substantially more accurate, since the arrival streams being approximated become much less bursty.

The parameters used in these two figures are the same as in figure 4.6, so a comparison of the effectiveness of heuristic prediction against approximation is worth making. As N becomes large an exact solution becomes increasingly more costly, so the approximation may be the only method which can be used, or at least substantially faster (except when N is sufficiently small). In the selective(m) case the heuristic performs very well and with this unreliable system is much more accurate than the approximation, however the heuristic fails to proved a reasonable estimate of the routeing weights for the fixed(m) strategy and in this case the approximation far out performs the heuristic. In either case the most rapid exact solution is obtained by optimising the approximation to give a near exact estimate of the optimal routeing weights. It is worth stressing here the difference between figures 5.2 and 4.8. Figure 4.8 compared the performance of the system solved exactly when the routeing vector was both estimated by a simple heuristic and calculated to give a minimum value of average response time. Figure 5.3 compares the exact solution with an approximation, in both cases using the routeing vector optimised with respect to the average response time. There is no advantage in comparing the plotted approximation in figure 5.2 with the exact solution obtained using the heuristic routeing vector in figure 4.8, as they are quite different, however it is quite clear that the routeing vectors obtained by optimising the approximations derived in this chapter is massively superior to the heuristic

used in the previous chapter.

Figure 5.3 illustrates the performance of a 2-node system as a function of the routeing vector $(q, 1-q)$. The parameters used here are such that the server is fairly reliable but the length of breakdowns is fairly long and so the curve of the average response time is relatively steep. As such it is only possible to show a small range of q before one or other of the curves extends beyond a reasonable value. Clearly the approximation to the selective strategy is very poor, being consistently less than half that of the exact calculation. As in the earlier figures the approximation for the fixed m=1 strategy performs somewhat better than for the selective. This approximation technique is most accurate for a single node when the arrival rate at that node is least affected by failures at other nodes. Therefore it is clear that the best approximation for an N-node system will be found when the routeing vector is chosen such that the sum of such effects is minimised. In this example the approximation of the selective strategy performs best when q is larger and for the fixed m=1 strategy when q is less. For both strategies the number of jobs redirected due to failures will be reduced in those cases.

In earlier chapters it has been seen that the duration of a breakdown will have a strong effect on the average response time, even when the reliability is maintained, It seems logical therefore to expect some negative effect on the accuracy of approximation if the length of breakdowns is increased. In figure 5.4 such a scenario is illustrated for a symmetrical 2-node system where the proportion of time each node is operative is kept constant, but the length of operative and inoperative periods is varied. As expected the approximations perform worst when the repair rate is least and significantly better the larger it becomes. The explanation for this has already been stated a number of times in this thesis, namely

that the longer the duration of the breakdown the more jobs will be accumulated in the queue, causing an increase in the average number of jobs and a markedly higher server usage when service resumes than the average. Hiding the breakdowns in the way described to form the approximations smooths out these effects and thus causing the approximation to greatly underestimate the average response time.

It is worth stating here that the effects observed in these final 2 examples are much more pronounced due to the presence of only 2 nodes. In general it would be unnecessary to use such approximations on 2 node systems as an exact solution can be easily calculated. If there were more nodes the burstiness of arrivals would normally be much less severe and therefore the approximations would perform better, although these inaccuracies would still exist.

5.3 More complex numerical approximations

So far only approximations giving simple equation solutions have been considered, but there is no reason why more complicated matrix solution methods should not be employed to solve more complex approximations with a far greater number of states of operation, but without the exponential growth in state space with N of the exact solution. One such model is to consider node i as either working or broken, with 0,1 or up to $N-1$ of the other servers working, giving an $2N$ operational states. The solution method involving spectral expansion has already been described, and will be revisited in subsequent chapters, therefore for the purpose of this approximation it is sufficient to express the matrices A, B and C needed to carry out this solution, corresponding to transitions between operative states, resulting from arrivals and resulting from service completions respectively.

Inoperative states are numbered 0 to $N-1$ and operative states N to $2N-1$. State 0 represents all servers broken, 1 represents node i broken and only 1 other working, etc, and state N represents node i working and all others broken, state $N+1$ represents node i working and only 1 other working, etc. So,

$$a_{j,N+j} = \eta_i \ , \ j = 0 \ldots N-1$$

$$a_{j,j-N} = \xi_i \ , \ j = N \ldots 2N-1$$

$$a_{j,j+1} = \left(\sum_{\forall \sigma s.t. S(\sigma)=j} p(\sigma) \sum_{\forall k s.t. k \in \overline{\sigma}} \eta_k \right) \Big/ p(j) \ , \ j = 0 \ldots N-2, N \ldots 2N-2$$

$$a_{j,j-1} = \left(\sum_{\forall \sigma s.t. S(\sigma)=j} p(\sigma) \sum_{\forall k s.t. k \in \sigma} \xi_k \right) \Big/ p(j) \ , \ j = 1 \ldots N-1, N+1 \ldots 2N-1$$

$$b_j = \lambda \left(\sum_{\forall \sigma s.t. S(\sigma)=j} p(\sigma) q_i(\sigma) \right) \Big/ p(j) \ , \ j = 0 \ldots 2N-1$$

$$c_j = \mu_i \ , \ j = N \ldots 2N-1$$

The number of states can be reduced further without loss of accuracy for servers which do not receive jobs when broken to just $N+1$ by lumping together all the states where node i is broken. Under the selective(m) strategy for servers receiving jobs when all servers are broken the number of states can be reduced to $2^{N-1}+2$, as the states where node i is broken can be expressed as 2 states, *all broken* and *node i broken, but not all broken.* If these reductions in state space are made then the transitions from the 'lumped' state have to be modified using the occupation probabilities, p_σ, in the same way as for the two stage failure model earlier in this chapter.

Clearly in all but the most extremely unreliable systems, the most likely states are going to be where one or fewer servers are broken. However, the more other servers which are broken, the more jobs will be directed towards node i, so these unlikely states are

significant with respect to arrivals, this is most true in the selective case where every failure causes a change in job distribution. In general it is possible to expand any one of the 'lumped' states (where j, or $j-N$, other servers are broken) into its constituent configurations to improve the approximation. Expanding states j and $j+N$ (where $j < N$) into their constituent configurations will add $(2(N-1)!/j!(N-1-j)!)-2$ states, which is clearly a large potential increase. It is possible to minimise the increase in state space by expanding only those states where 1 other server is broken or 1 other server is working, thus considering the most likely server configurations and also those configurations where service rate is highest. In general this will increase the state space by $4N-8$, however in the selective(m) strategy and the fixed(m) strategy where servers do not receive jobs when broken, no advantage is gained by considering more configurations where node i is broken and so only $4N-4$ additional states need to be considered.

5.4 Numerical Limitations

Before we leave this section, some remarks on the complexity of the exact numerical solution are in order. To compute the distribution and/or the mean of one queue in an N-node system requires the determination of 2^N eigenvalues and eigenvectors, and the solution of a set of 2^N simultaneous linear equations. The complexity of that task is on the order of 2^{3N}. Since there are N queues, the total complexity of the full solution, for one set of parameters, is $O(N2^{3N})$. This is a large computational effort even for systems of even moderate size. In addition, when the number of eigenvalues is very large, one begins to encounter numerical problems associated with ill-conditioned matrices.

The largest system we have been able to tackle had 8 nodes (256 server configurations),

then the solution for a single queue took an hour. The approximate solution is of course applicable for much larger values of N.

5.5 Conclusions

Several approximate solutions to the model presented in the previous chapter have been derived subjected to several criteria including, accuracy of predicted performance measures, accuracy of predicted routeing weights and complexity of solution. Results derived in chapter 2 were used to provide simple approximations which give a rapid solution with a good prediction of optimal routeing weights without the need to optimise the complex functions derived in chapter 4. This method of predicting optimal routeing weights gives far greater accuracy than the simple heuristics presented earlier and with only a slight increase in overheads. More complex approximations were also presented which necessitated the use of the spectral expansion solution method. Although these solutions are a lot more involved than for the simpler approximations the level of accuracy in predicting the performance measures has been shown to be greatly improved and a large saving in overheads is made over the exact solution, especially when there number of servers in parallel is large. Thus, a set of approximations has been presented which allow a rapid optimisation and accurate prediction of performance for potentially large parallel systems

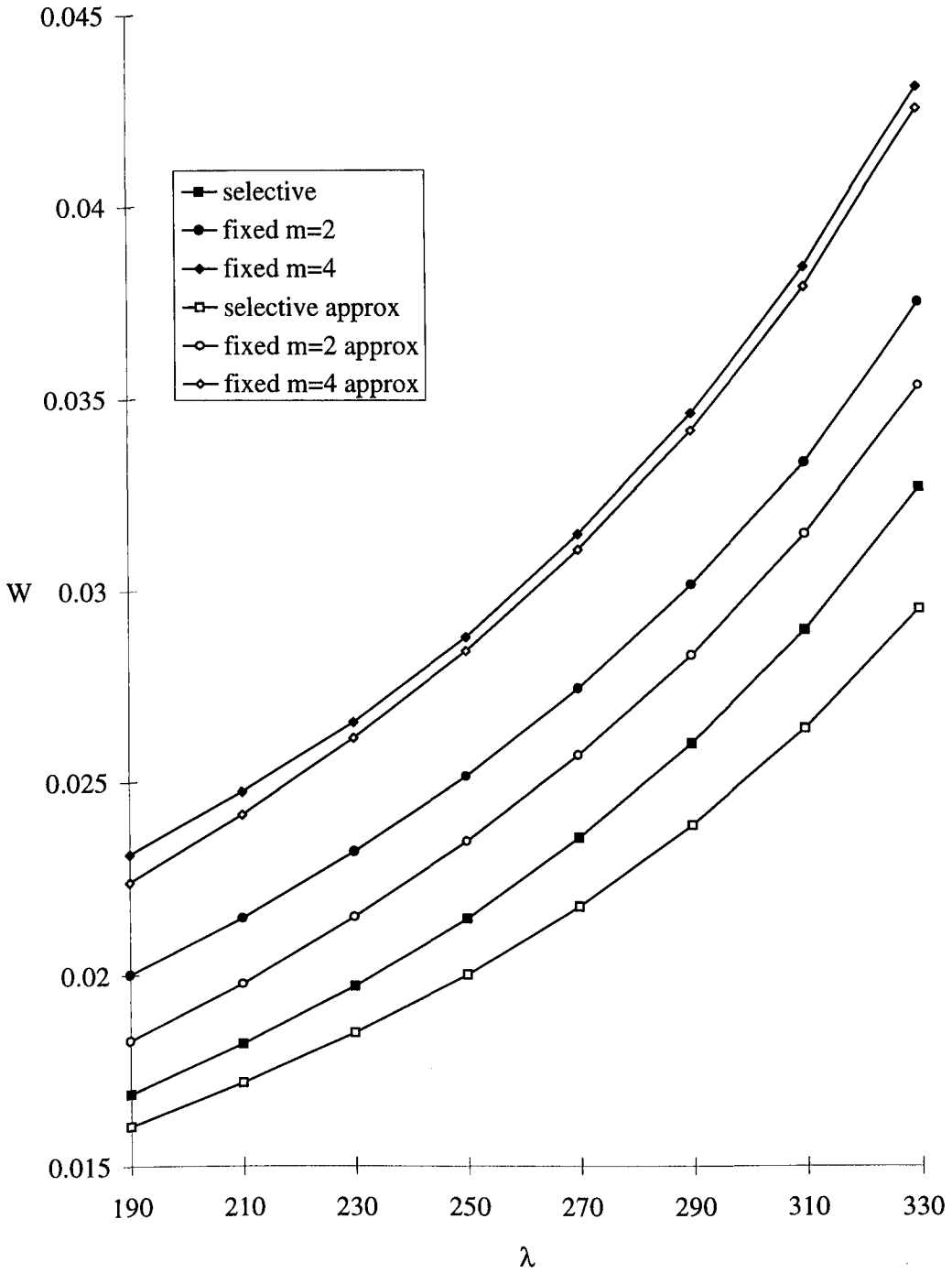


Figure 5.1: Exact and approximate solutions of average response time with optimal routeing vector for different strategies

$N = 5$, $\mu_1 = 150$, $\mu_2 = 160$, $\mu_3 = 170$, $\mu_4 = 180$, $\mu_5 = 190$,

$\xi_1 = \xi_2 = \xi_3 = \xi_4 = \xi_5 = 50$, $\eta_1 = 50$, $\eta_2 = 60$, $\eta_3 = 70$, $\eta_4 = 80$, $\eta_5 = 100$

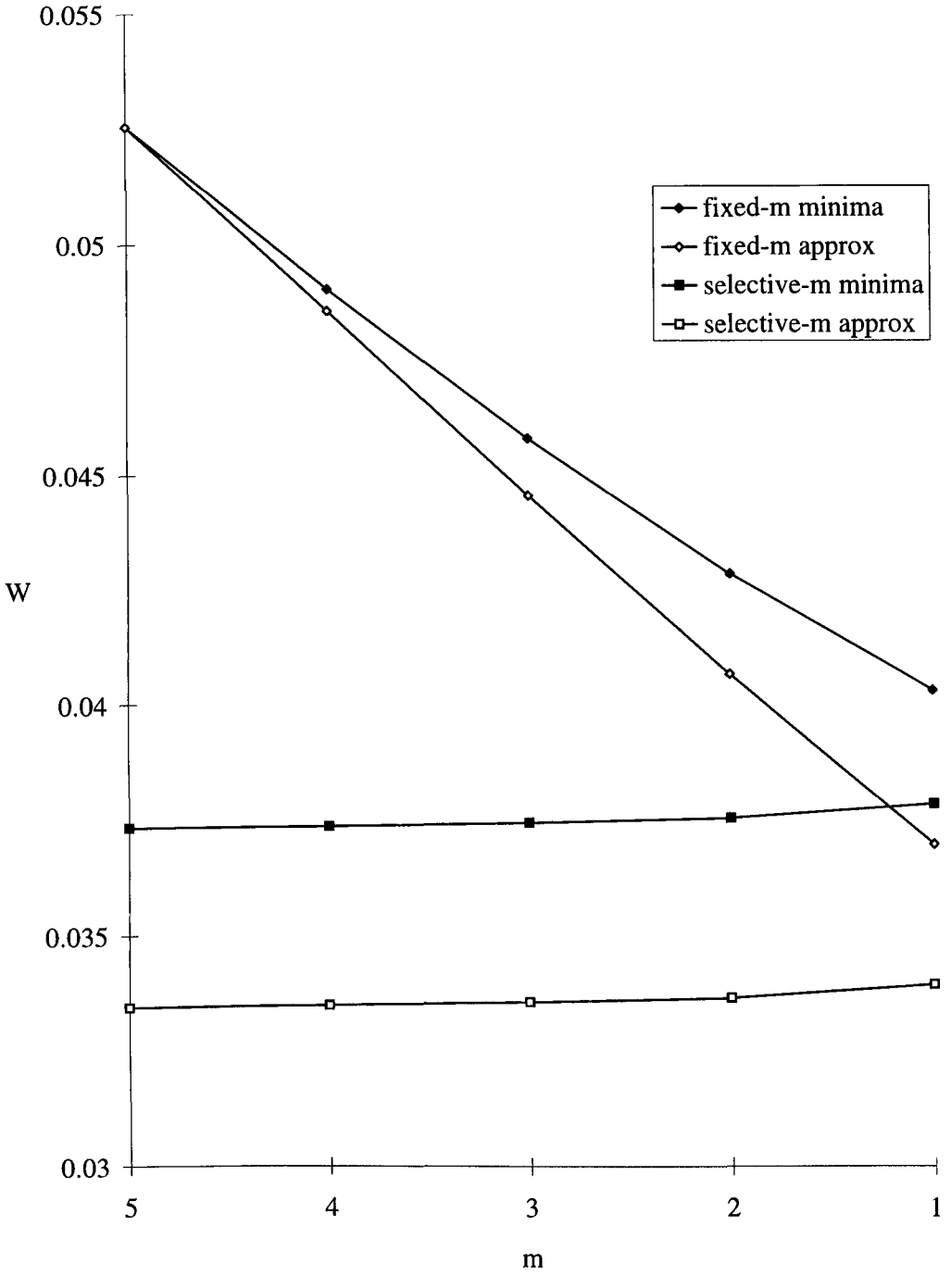


Figure 5.2: Exact and approximate solutions of average response time with optimised routeing vector, as a function of job arrival rate

$N = 5$, $\lambda = 350$, $\mu_1 = 150$, $\mu_2 = 160$, $\mu_3 = 170$, $\mu_4 = 180$, $\mu_5 = 190$,

$\xi_1 = \xi_2 = \xi_3 = \xi_4 = \xi_5 = 50$, $\eta_1 = 50$, $\eta_2 = 60$, $\eta_3 = 70$, $\eta_4 = 80$, $\eta_5 = 100$

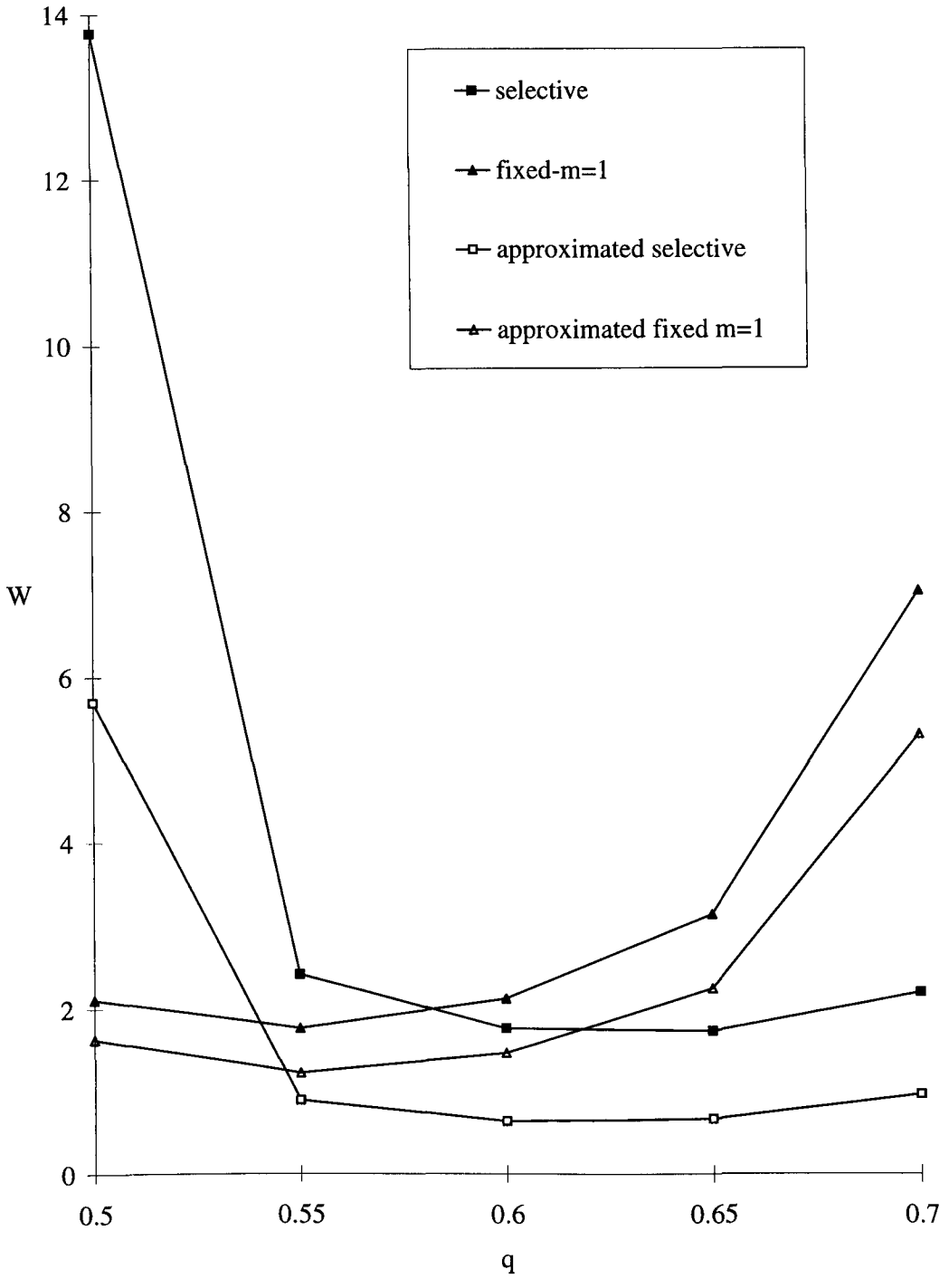


Figure 5.3: Exact and approximate solutions of average response time as a function of the routeing vector $(q, 1-q)$

$N = 2$, $\mu_1 = 12$, $\mu_2 = 8$, $\xi_1 = \xi_2 = 0.01$, $\eta_1 = 0.2$, $\eta_2 = 0.1$ $\lambda = 15$

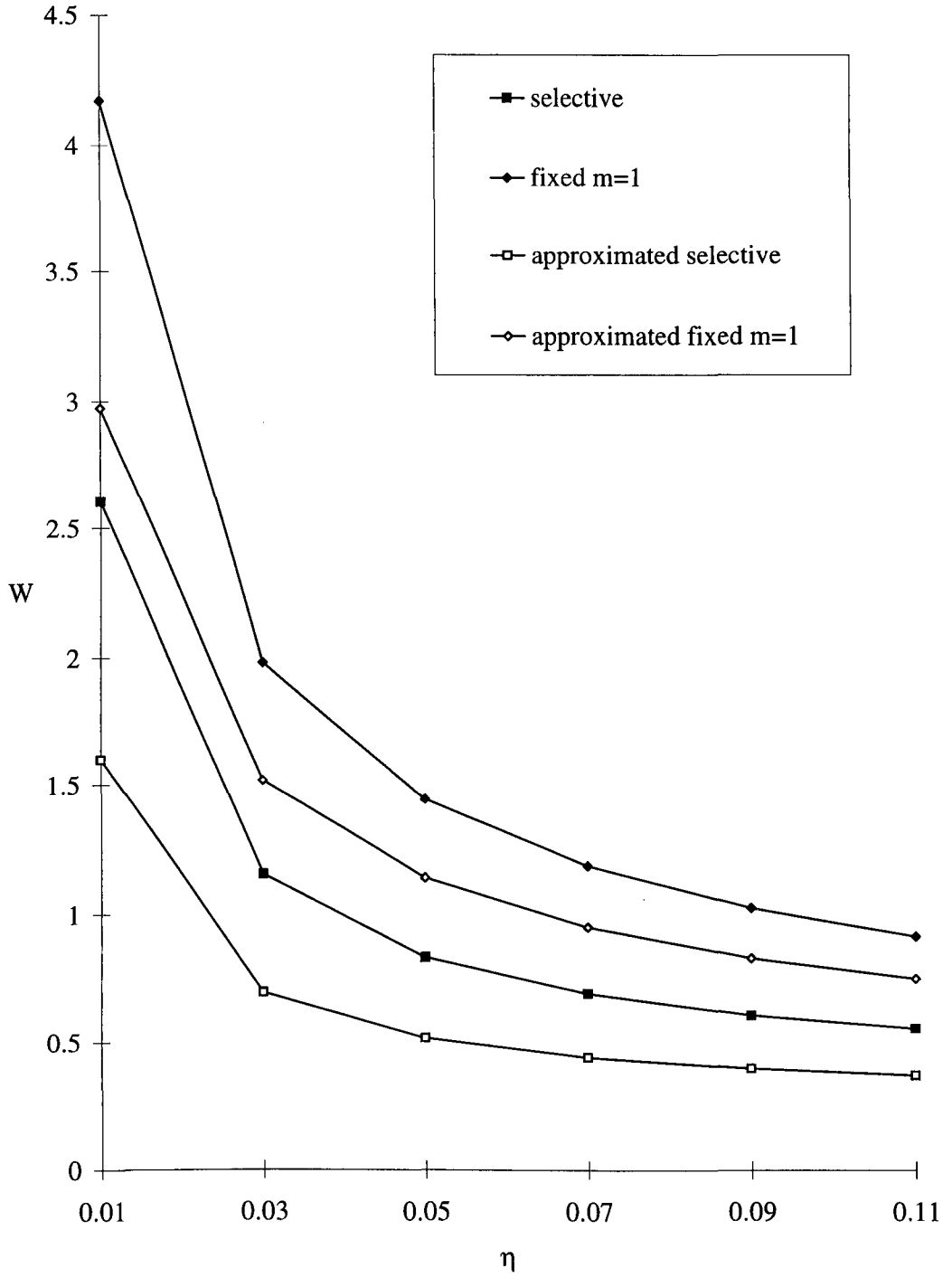


Figure 5.4: Exact and approximate solutions of average response time, with optimised routeing vector, as a function of repair rate where the proportion of time spent operative is constant

$N = 2$, $\lambda = 10$, $\mu_i = 10$, $\xi_i = \eta_i/10$, $i = 1, 2$

Chapter 6

A Pipeline with Nodes of Servers in Parallel

6.1 Summary

Jobs from a single Poisson input stream receive K independent stages of service, one at each node in the pipeline. At stage i jobs are routed through one of the N_i available nodes, modelled as $M/M/1$ queues. These nodes are subject to random failure and repairs which leave their corresponding queues intact, but may affect the routeing of jobs arriving at that stage during the subsequent repair period. Two possible approximate solutions for the marginal queue size distributions are obtained by spectral expansion and are compared with solutions obtained by simulation techniques. Two routeing strategies are considered, fixed and selective, and the relative accuracy of the approximate solutions and predicted optimal routeing vectors are discussed.

6.2 The Model

Jobs arrive into the system in a Poisson stream with rate λ. There are K stages in series and in stage i there are N_i nodes in parallel, each with an associated unbounded queue, to which incoming jobs may be directed. Server j at stage i goes through alternating independent operative and inoperative periods, distributed exponentially with means $1/\xi_{i,j}$ and $1/\eta_{i,j}$ respectively. While it is operative, the jobs in its queue receive service of an exponentially distributed duration with mean $1/\mu_{i,j}$, and leave the stage upon completion to proceed to the next (if any) stage of service. When a node becomes inoperative (breaks down), the corresponding queue, including the job in service (if any), remains in place. Services that are interrupted in this way are eventually resumed from the point of interruption. The system model is illustrated in figure 6.1.

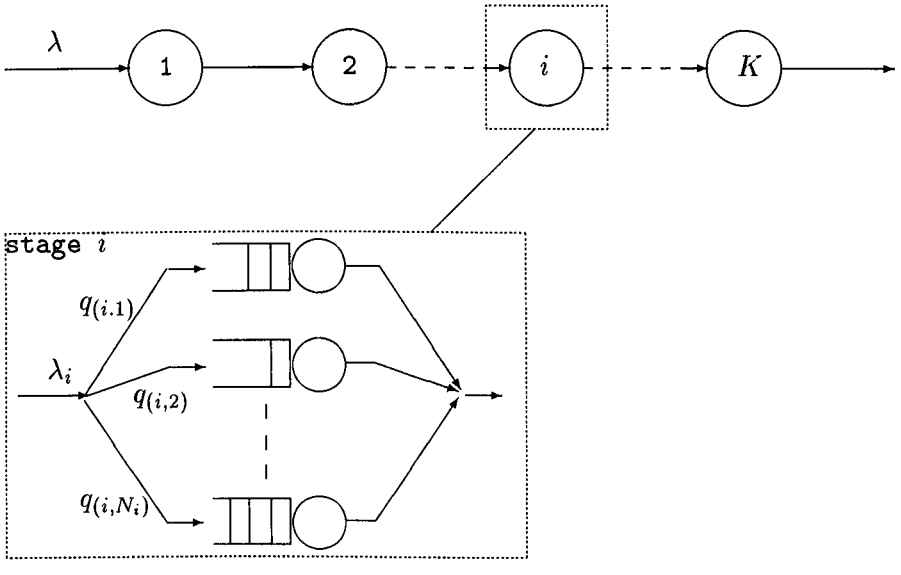


Figure 6.1: A single source to a pipeline of K stages, split between the nodes in each stage

The arrival rate at stage i is given in figure 6.1 as λ_i, but since no jobs are lost the overall arrival rate at all stages will be the same as the external Poisson arrival rate λ. However, since the arrivals at stage i depend on the departures from stage $i-1$ then the arrival stream will, in general, cease to be Poisson. The *system configuration* at any moment is specified by the subset, σ, of nodes that are currently operative (that subset may be empty, or it may be the set of all nodes): $\sigma \subset \Omega_N$, where $\Omega_N = \{(1,1),(1,2),\ldots,(1,N_1),(2,1),\ldots,(K,N_K)\}$, where the pair i,j represents node j at stage i. There are of course 2^N possible system configurations, where $N = \sum_{i=1}^{K} N_i$. In general it is more convenient to consider the subset σ_i whose elements are those nodes at stage i which are operative. The set of all nodes at stage i is denoted by Ω_{N_i}. Clearly $\sigma_i \subset \Omega_{N_i} \subset \Omega_N$ and $\sigma_i \subset \sigma$. The steady-state marginal probability, p_{σ_i}, of configuration σ_i at stage i is given by

$$p_{\sigma_i} = \prod_{j\in\sigma_i} \frac{\eta_{i,j}}{\xi_{i,j}+\eta_{i,j}} \prod_{j\in\overline{\sigma}_i} \frac{\xi_{i,j}}{\xi_{i,j}+\eta_{i,j}} \ , \ \sigma_i \subset \Omega_{N_i} \ , \tag{6.1}$$

And the steady-state marginal probability, p_σ, of configuration σ is given by

$$p_\sigma = \prod_{i,j\in\sigma} \frac{\eta_{i,j}}{\xi_{i,j}+\eta_{i,j}} \prod_{i,j\in\overline{\sigma}} \frac{\xi_{i,j}}{\xi_{i,j}+\eta_{i,j}} \ , \ \sigma \subset \Omega_N \ , \tag{6.2}$$

where $\overline{\sigma}_i$ is the complement of σ_i with respect to Ω_{N_i}, $\overline{\sigma}$ is the complement of σ with respect to Ω_N and an empty product is by definition equal to 1. These expressions follow from the fact that nodes break down and are repaired independently of each other.

If, at the time of arrival at stage i, a new job finds the stage in configuration σ_i, then it is directed to node j with probability $q_{i,j}(\sigma_i)$. These decisions are independent of each other, of past history, of the sizes of the various queues and of the state of any other stage

in the pipeline. Thus, a routeing policy at stage i is defined by specifying 2^{N_i} vectors,

$$\mathbf{q}_i(\sigma_i) = [q_{i,1}(\sigma_i), q_{i,2}(\sigma_i), \ldots, q_{i,N_i}(\sigma_i)] \ , \ \sigma_i \subset \Omega_{N_i} \ , \tag{6.3}$$

such that for every σ_i,

$$\sum_{j=1}^{N_i} q_{i,j}(\sigma_i) = 1 \ .$$

The system state at time t is specified by the pair $[I(t), \mathbf{J}(t)]$, where $I(t)$ indicates the current configuration (the configurations can be numbered, so that $I(t)$ is an integer in the range $0, 1, \ldots, 2^N - 1$), and $\mathbf{J}(t)$ is an integer vector whose k'th element, $J_k(t)$, is the number of jobs in queue k ($k = 1, 2, \ldots, N$). The integer k is used here instead of the pair i, j for simplicity, the relationship between k and i, j is a simple 1 to 1 mapping such that

$$j + \sum_{x=1}^{i-1} N_x = k$$

Under the assumptions that have been made, $X = \{[I(t), \mathbf{J}(t)], t \geq 0\}$ is an irreducible Markov process. The condition for ergodicity of X is that, for every queue i, j, the overall arrival rate is lower than the overall service capacity:

$$\sum_{\forall \sigma_i} \lambda_i p_{\sigma_i} q_{i,j}(\sigma_i) \ < \ \mu_{i,j} \frac{\eta_{i,j}}{\xi_{i,j} + \eta_{i,j}} \ , \ i = 1, 2 \ldots, K, j = 1, 2, \ldots, N_i \ . \tag{6.4}$$

When the routeing probabilities and the transfer jobs between stages of depend on the system configuration, the process X is not separable (i.e., it does not have a product-form solution). Consequently, the problem of determining its equilibrium distribution is intractable in general. On the other hand, the quantities of principal interest are expressed in terms of averages only; they are the steady-state mean queue sizes, L_k, and the the overall average response time, W, given by

$$W = \frac{1}{\lambda} \sum_{i=1}^{K} \sum_{j=1}^{N_i} L_{i,j} \ . \tag{6.5}$$

To determine those performance measures, it is not necessary to know the joint distribution of all queue sizes; the marginal distributions of the N queues in isolation are sufficient. Unfortunately, the isolated queue processes, $\{J_k(t)\ ,\ t \geq 0\}$ $(k = 1, 2, \ldots, N)$, are not Markov. As mentioned earlier the arrival stream at stage i $(i \geq 2)$ is not Poisson since it depends on the activity of all the previous stages, this makes an exact solution of the marginal queue size distributions almost as intractable a problem as solving the joint distribution of all queue sizes. However, it is possible to obtain good approximate solutions for the marginal queue size distributions by assuming the arrival stream at stage i to be Markov-Modulated Poisson. Some discussion as to how best to form the approximated arrival streams is presented afterwards.

Consider the stochastic processes $Y_{i,j} = \{[I^*(t), J_{i,j}(t)]\ ,\ t \geq 0\}$ $(i = 1, 2, \ldots, K, j = 1, 2, \ldots, N_i)$, which model the joint behaviour of the configuration and the size of an individual queue i, j, where $I^*(t)$ indicates the current approximated system configuration. In general each possible approximated system configuration, $I^*(t)$, will represent a set of one or more of the exact system configurations, σ. The number of approximated system configurations considered, from now on referred to as I_{max}, will, in general, determine the accuracy of the solution and the amount of computation required. The value of I_{max} will therefore be limited at the upper bound by the amount of computational power available and the desired rapidity of the solution and at the lower bound by the desired accuracy of the solution.

The state space of $Y_{i,j}$ is infinite in one dimension only, which simplifies the solution considerably and makes it tractable for reasonably large values of I_{max}. The important observation here is that, with the assumption of a Markov-Modulated arrival process,

$Y_{i,j}$ is an irreducible Markov process, for every i, j. This is because the arrivals into, and departures from queue i, j during a small interval $(t, t + \Delta t)$ depend only on the approximated system configuration and the size of queue i, j at time t, and not on the sizes of the other queues. As mentioned earlier, without the approximation of the arrival stream to a Markov-Modulated arrival process, this statement would not be true, since a job only arrives at stage $i + 1$ after successfully completing service at stage i, therefore making the queue size at any stage dependent on all previous stages of service.

The next task, therefore, is to find the equilibrium distribution of $Y_{i,j}$:

$$p_{i,j}(x, y) = \lim_{t \to \infty} P[I(t) = x \, , \, J_{i,j}(t) = y] \ ,$$

$$x = 0, 1, \ldots, I_{max} - 1 \ , \ y = 0, 1, \ldots \tag{6.6}$$

Given the probabilities $p_{i,j}(x, y)$, the average size of queue i, j is obtained from

$$L_{i,j} = \sum_{y=1}^{\infty} y \sum_{x=0}^{I_{max}-1} p_{i,j}(x, y) \, . \tag{6.7}$$

6.3 Approximated system configurations

In this model there are 2^N possible system configurations, which is clearly too large a number to solve for in any practical situation, hence the need for a reduced solution. In general, the arrivals at node i are dependent on all the preceding stages of service (or node configurations), however it is obvious that the nature of the arrivals at each node are most strongly linked to the configuration at the immediately preceding node. Thus one possible reduced solution method is clear, namely,

1. perform the solution described in chapter 3 on the first node - this will be an exact solution since there are no preceding nodes to affect arrivals

2. extract from that solution the appropriate performance measures and the probabilities $p_{1,j}(\sigma_1), j = 0..N_1$, where $p_{i,j}(\sigma_i)$ is the probability that queue j (at node i) is non-empty given that the configuration of node i is σ_i.

3. perform the solution described above with the approximated system configurations merely the configuration of this node and that immediately preceding it, thus $I_{Max} = 2^{N_i+N_{i-1}}$ and the arrival rate at node j (assumed Poisson) in configuration I^* is given by

$$q_j(\sigma_i) \sum_{k=0}^{N_i-1} (p_{i-1,k}(\sigma_{i-1})\mu_{i-1,k})$$

where σ_{i-1} and σ_i represent the configurations at node $i-1$ and node i respectively at given approximated system configuration I^*.

4. extract from this solution the appropriate performance measures and the probabilities $p_{i,j}(\sigma_i), j = 0..N_i$

5. repeat steps 3 and 4 for the next node until all nodes have been solved.

Clearly this solution is only possible when N_i is relatively small for all i (if $N_i+N_{i-1} \geq 8$ then the solution becomes very large) and so an alternative needs to be found. The simplest idea is to ignore all previous nodes in the solution of node i and take the arrival rate at that node to be Poisson rate λ, i.e. the same as the external arrival stream. This allows the solution of much larger parallel nodes (see chapter 3), but at the expense of all consideration of the staged nature of service. A much better alternative would be for some halfway measure, allowing reasonably large systems to be solved with some knowledge of the preceding stage taken into account. In the preceding chapter some approximate methods for the solution of a single stage parallel system were presented, the best approximate

solution coming when the most significant arrival periods were treated independently and the remainder were amalgamated into logical groups. Applying the same technique here one such solution would be to have approximated system configurations based on the current server (i, j) either working or broken, with 0,1, or up to $N_i - 1$ other servers at node i working, and 0,1, or up to N_{i-1} servers working at the previous stage, giving a total of $2N_i(N_{i-1} + 1)$ possible configurations. Another possibility is to consider all the possible configurations of node i together with those arising from having 0,1 or up to N_{i-1} servers operative at node $i-1$. These are just two examples, the best set of approximated system configurations will be determined by the server characteristics and the available computational resources and some discussion on this is included in the previous chapter. It is assumed that approximated system configurations will be chosen such that node i, j will be either operative or inoperative in any approximated configuration, but not both.

The process $Y_{i,j}$ is of the *block tri-diagonal*, or *Quasi-Birth-and-Death* type and so is a special case of the model presented in chapter 3. It can therefore be solved by spectral expansion in exactly the same way to find the probabilities $p_{i,j}(x, y)$.

6.4 Scheduling strategies

As in [100] which considered the single stage parallel system, here several strategies based on a single routeing vector are evaluated and compared, $\mathbf{q} = (q_1, q_2, \ldots, q_N)$. In each case, the optimisation problem is to chose the elements of that vector so as to minimise the average response time.

1. *The fixed strategy.*

The most straightforward way of splitting the incoming stream at stage i is to send

each job to queue j with probability q_j, regardless of the system configuration. In this simple case in the single stage model a simple equation could be used to determine the performance measures, however with the introduction of several stages this is no longer true, as the arrival process at a given stage is affected by node failures at earlier stages.

2. *The selective strategy.*

Intuitively, it seems better not to send jobs to stages where the node is inoperative, unless that is unavoidable. This suggests the following strategy: If the subset of operative nodes at stage i in the current system configuration is σ_i, and that subset is non-empty, send jobs to queue j only if $j \in \sigma_i$, with probability proportional to q_j :

$$q_j(\sigma_i) = \frac{q_j}{\sum_{\ell \in \sigma} q_\ell} \quad , \quad j \in \sigma \quad .$$

If σ is empty (i.e. all nodes are broken), send jobs to queue j with probability q_j ($j = 1, 2, \ldots, N_i$).

Note that neither of these strategies take account of the states of nodes at other stages in the system, however the existence of other stages may have an effect on the optimal routeing vector for a given strategy.

6.5 Numerical results

Numerical experiments were carried out in order to determine both the accuracy of the approximations suggested and the characteristics of the behaviour of the pipeline system. In most practical situations it is normal to find nodes with a high degree of reliability, however, as is the case with most models involving node breakdowns, systems of such nodes may behave much like nodes without breakdowns. It has been necessary, therefore,

to consider here nodes with somewhat extreme characteristics in order to highlight the strengths and weaknesses of the approximations and to show the limiting behaviour of such a system of nodes. However, it is also true to say that even nodes with a high degree of reliability may suffer rare, but prolonged, breakdowns which can have a significant effect on performance measures.

If few arrivals occur during a period of breakdown (i.e. $\eta \sim \lambda$) then the effect of a failure on the sizes of the queues at a stage will be minimal, assuming the node is reasonably reliable, just as for the single stage parallel node models considered previously. Also if the service rate is of a similar order ($\mu \approx \lambda$) then the departures will not be unduly interrupted by failures, so the arrivals at the following stage may be assumed to be nearly Poisson, hence the single stage approximation will work well for either routeing strategy.

However, if the repair rate is small compared to the arrival rate, then many arrivals will occur during a breakdown period. Under the selective routeing strategy this will cause the other nodes at that stage to be more heavily loaded, causing the queues at those nodes to grow. With the fixed routeing strategy the queue of a broken node will grow larger during a period of breakdown, leading to a large backlog of jobs if the load is sufficiently high. The solution of the model for the stage where this behaviour occurs is still exact, but the arrivals at the next stage are now distinctly 'bursty', rather than nearly Poisson and so the accuracy of the approximations is in question.

If N_i is large then the effect of an individual failure at stage i will be reduced, since one node failing out of N_i identical nodes will mean a reduction of at most $1/N_i$ in the overall service at stage i. In fact the reduction could be considerably less than $1/N_i$ if the load at stage i is not excessively high and the selective routeing strategy is used, since the

remaining nodes will be less likely to be idle. Since the arrivals at stage $i+1$ are in fact the departures from stage i then any reduction in the effect of failures at stage i will result in an improvement in the approximation of the arrivals at stage $i+1$ as a Poisson stream. Thus, although the 2-stage approximation becomes too costly to use when N_i is large, the accuracy of the simple approximation can be seen to improve in general (assuming the repair rates are sufficiently great).

Figures 6.2 and 6.3 show the average response time of a 2 stage pipeline where the are 2 nodes at each stage and all the nodes are identical. In figure 6.2 the routeing strategy is selective and in figure 6.3 it is fixed, in both cases the routeing vectors are simple and identical for each node, i.e. $(\frac{1}{2}, \frac{1}{2})$. Results are given for the simple (Poisson) approximation, the 2-stage (full Markov modulated) approximation and simulation. In both figures as the arrival rate increases the response time increases as expected and the average response time is higher under the fixed strategy. When the load is light all three methods give very similar results (for both strategies), but as the load increases the simple approximation becomes somewhat less accurate than the 2-stage approximation. In figure 6.4 the differences between the two approximate methods are highlighted further. Here the structure of the pipeline is the same, but the nodes are not reliable.

As mentioned earlier the simple approximation becomes much less accurate when the duration of the periods of inoperation is increased. This is shown in figure 6.5, where once again there are 4 identical nodes in a 2 stage pipeline, showing results for the selective strategy. The overall reliability of the nodes $(\eta/(\eta+\xi))$ remains constant, but the durations of the periods of operation and inoperation are increased exponentially. When the failure and repair rates are relatively large the effect of failures is minimal and so both

approximations work well, however as the repair and failure rates decrease the simple approximation become highly inaccurate as the arrivals become more and more 'bursty'.

The 2-stage approximation does not always give such accurate results as those shown above. With the fixed routeing strategy in particular a large backlog of jobs may build up during a period of failure, thus the probability of the queue being non-empty may be significantly less for sometime immediately following a failure than after a long period of operation. However such node characteristics would be somewhat extreme, average number of jobs in the queue would have to be small during operation, but large during inoperation, thus λ would have to be significantly less than μ ($\lambda = \mu/2$ say) and the period of inoperation would have to be very long ($\eta \ll \lambda, \eta < \lambda/10^4$ say). A simulation of such a pipeline would take an exceedingly long time to produce an accurate result.

In general the optimal routeing weights are not greatly affected by the presence of preceeding stages, but an unbalanced system will perform significantly worse as a result of increased 'burstiness'. This is illustrated in the following 4 graphs, each of which show the performance at the final stage of a pipeline only. Figure 6.6 shows the average response time at the second of 2 stages as a function of the proportion of jobs sent to node 1 when both are available (q) under a selective routeing strategy with routeing vector $(q, 1-q)$. The 2-stage approximation takes account of the behaviour at a preceeding stage which has long periods of inoperation whereas the simple approximation considers the same stage in isolation, the arrival rate is identical in both cases. Figure 6.7 shows the same system operating the fixed routeing strategy. Clearly in this (extreme) case the the optimal routeing vector is slightly altered by considering a preceeding change, but perhaps more significant is the much greater steepness exhibited by the curve of the 2-

stage approximation. Thus a routeing vector which gives a near optimal average response time when the stage is considered in isolation could give a very poor response time when the preceeding stage is taken into account.

Figures 6.8 and 6.9 compare similar results as the two previous figures with results obtained from simulations. The performance measures displayed are the average response times taken at a stage of a pipeline with either one or two previous stages. Again there is a slight difference in the optimal routeing vector between the approximations, but the 2-stage approximation is fairly accurate when compared to the optimal routeing vector found by simulation. As would be expected in this case the 2-stage approximation gives a much more accurate fit to the simulated models than does the simple approximation, although there is still an appreciable error.

In figure 6.8 a 3-stage pipeline is also illustrated. It is interesting to note that there is a significant increase in average response time calculated by simulation for the 3rd stage of a 3-stage pipeline as opposed to the 2nd stage of a 2-stage pipeline with the same parameters. Unfortunately the same cannot be said for calculations made by approximation where there is only a slight difference between the 2 and 3 stage results. Clearly therefore the earlier assertion that the performance of one stage of a pipeline is heavily dominated by its preceeding stage is not an altogether accurate one. In both figures the M-stage approximations accurately ape the curve of the simulations, albeit with some displacement. There is some deviation from this as one moves away from the optimum routeing vector, although this is much more marked in the simple approximation.

6.6 Conclusions

Under many common practical situations a good approximation to this pipeline model can be made by considering each of the stages in isolation, this is particularly true when the nodes are highly reliable, periods of inoperation are relatively short and the number of nodes at a stage is relatively large. When these conditions do not apply it is necessary to use a more involved Markov-modulated approximation such as the 2-stage approximation suggested here. In certain circumstances it may be advantageous to look for alternative approximations, more detailed than the simple approximation, but less costly than the 2-stage approximation. The exact choice of what approximation to consider will depend on many variables (the node characteristics, available computational power, desired accuracy, etc) which are out of the scope of this thesis, but are worthwhile directions of research none the less. Also it may be worth considering other heuristics to predict the optimal routeing vectors in light of the increased penalties to an unbalanced system when previous stages of service are involved.

For models with characteristics like some of those illustrated here, i.e. N_i small and fairly long periods of inoperation, simulations need too be run for a very long time before producing a steady-state result. This in itself is clear justification for attempting to find suitable approximations for these models and the inaccuracy in the predictions will, in most circumstances, be sufficiently small for the speed of the calculation to be a definite bonus.

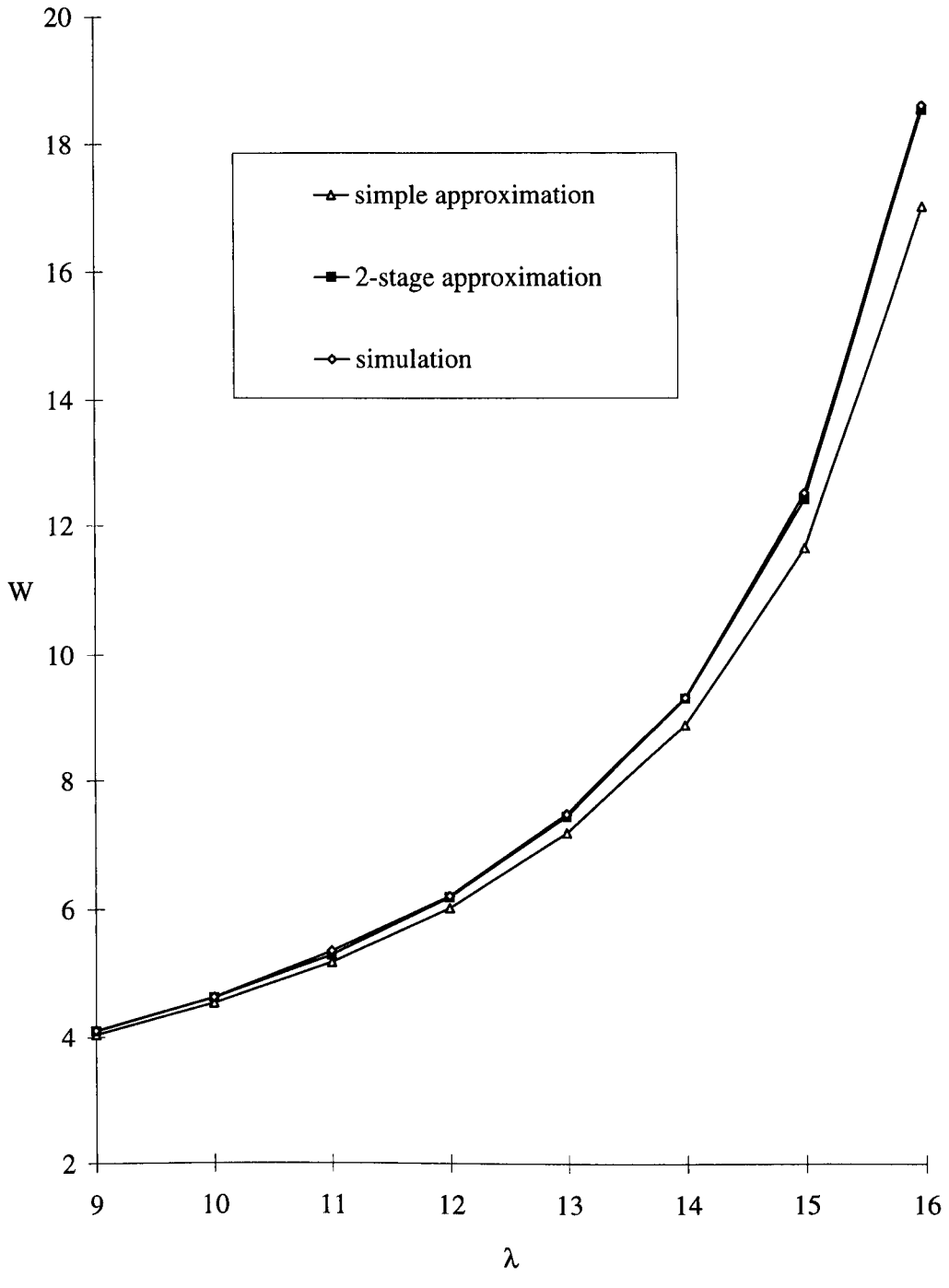


Figure 6.2: Average response time as a function of arrival rate for a 2 stage service where each stage has 2 identical servers and a fixed routeing strategy

$$M = 2,\ N_i = 2,\ \mu_{i,j} = 10,\ \xi_{i,j} = 0.01,\ \eta_{i,j} = 0.1,$$

$$i = 1, 2,\ j = 1, 2$$

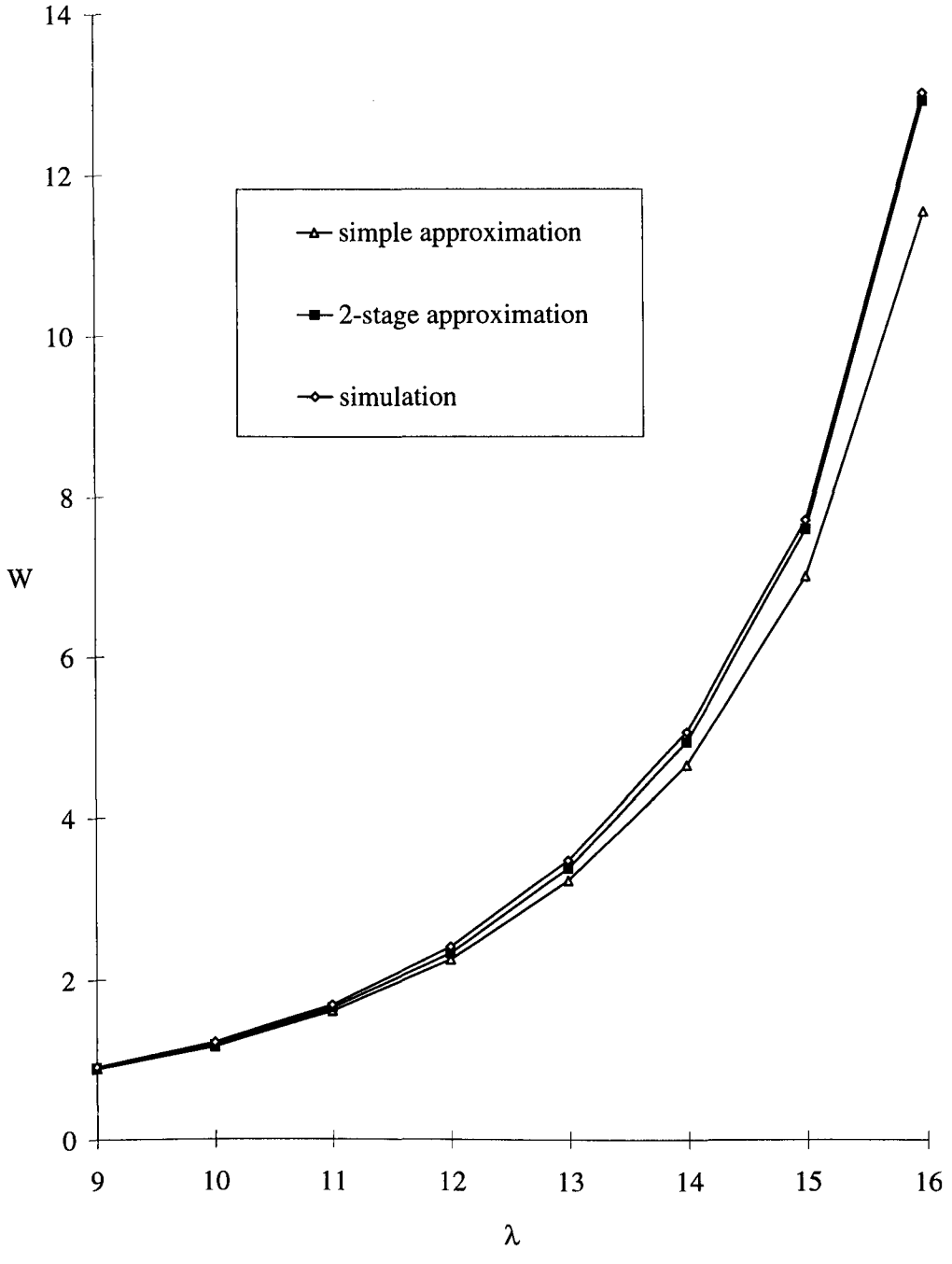


Figure 6.3: Average response time as a function of arrival rate for a 2 stage service where each stage has 2 identical servers and a selective routeing strategy

$$M = 2,\ N_i = 2,\ \mu_{i,j} = 10,\ \xi_{i,j} = 0.01,\ \eta_{i,j} = 0.1$$

$$i = 1, 2,\ j = 1, 2$$

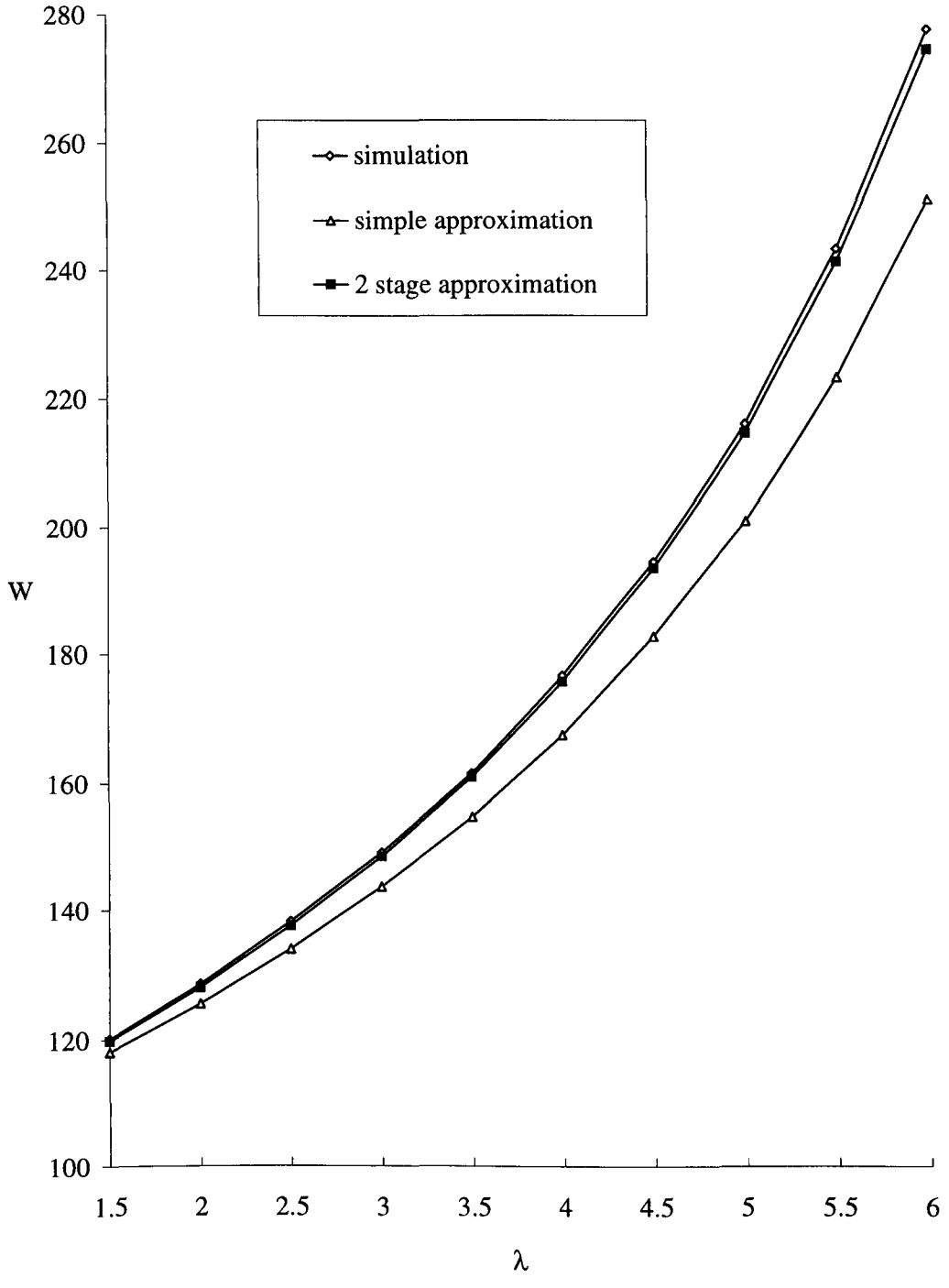


Figure 6.4: Average response time as a function of arrival rate for a 2 stage service where each stage has 2 identical servers and a fixed routeing strategy

$$M = 2,\ N_i = 2,\ \mu_{i,j} = 10,\ \xi_{i,j} = 0.01,\ \eta_{i,j} = 0.01$$

$$i = 1, 2,\ j = 1, 2$$

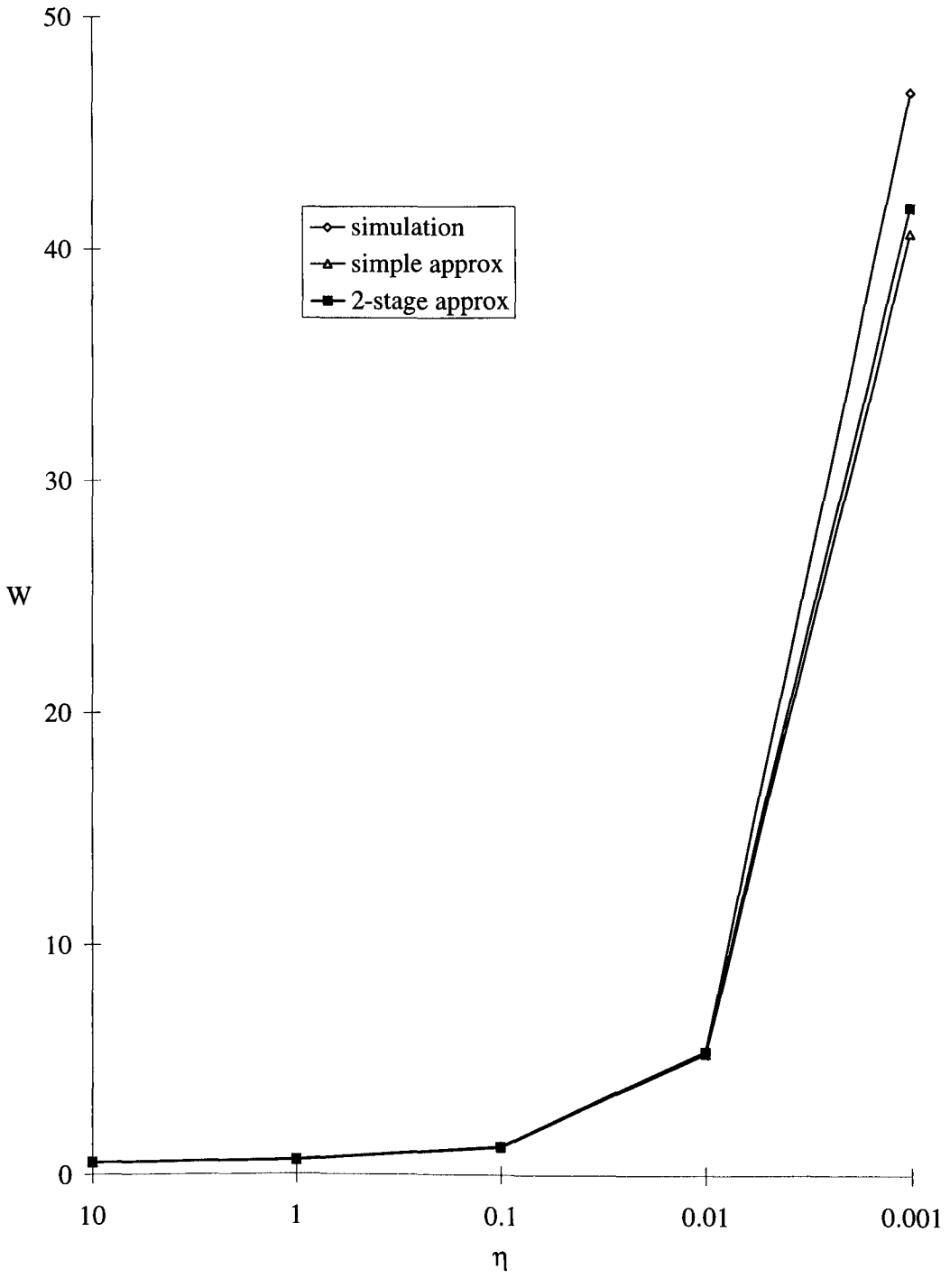


Figure 6.5: Average response time as a function of repair rate for a 2 stage service where the proportion of time operative is a constant

$$M = 2,\ N_i = 2,\ \mu_{i,j} = 10,\ \xi_{i,j} = \eta_{i,j}/10,\ \lambda = 2$$

$$i = 1, 2,\ j = 1, 2$$

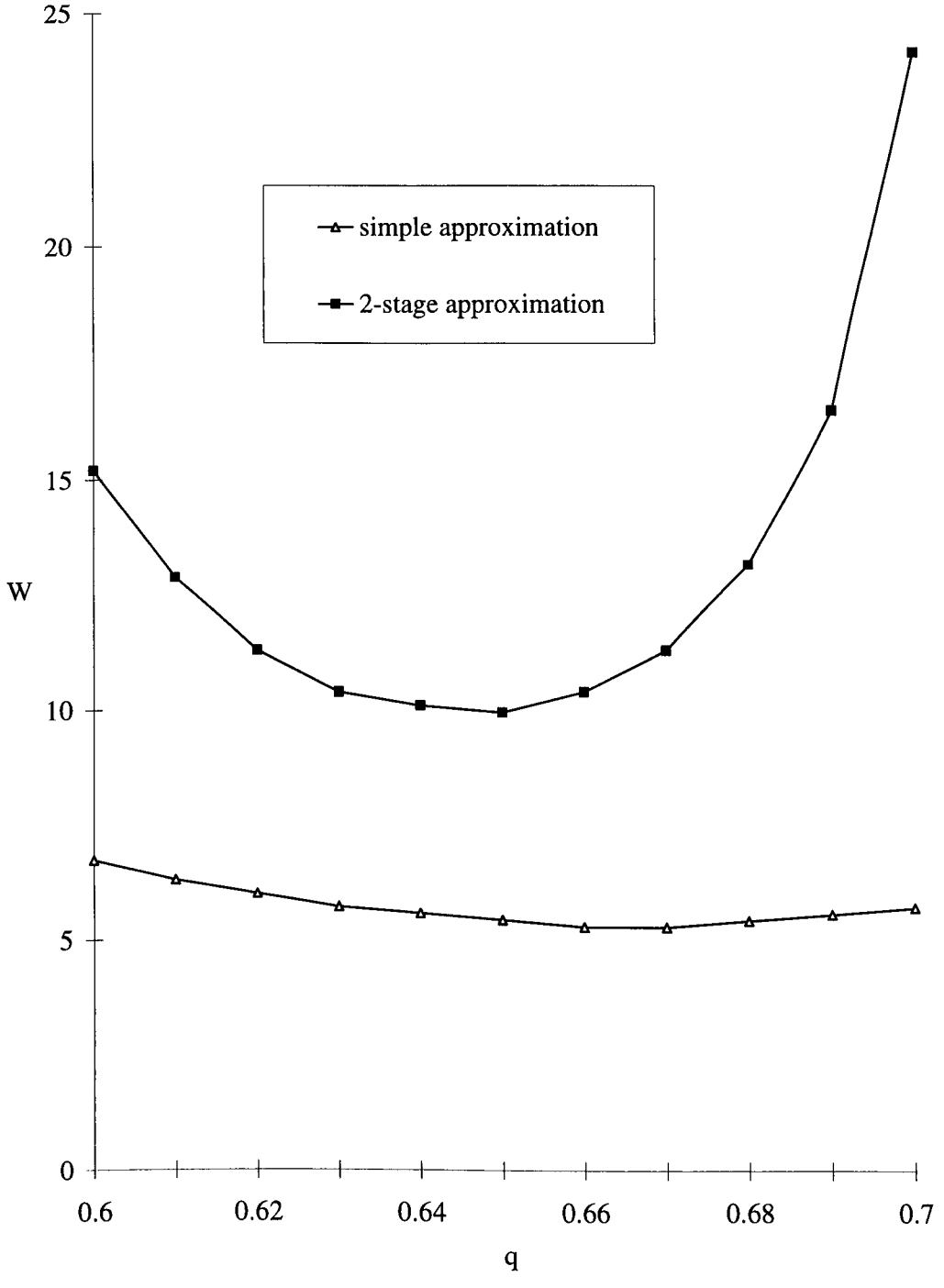


Figure 6.6: Average response time as a function of job share q

at the 2nd stage of a 2 stage pipeline with a selective routeing strategy

$M = 2$, $N_i = 2$, $\lambda = 17$, $\mu_{1,j} = 10$, $\mu_{2,1} = 14$, $\mu_{2,2} = 9$,

$\xi_{1,j} = 0.0001$, $\eta_{1,j} = 0.001$, $\xi_{2,j} = 0.01$,

$\eta_{2,1} = 0.1$, $\eta_{2,2} = 0.07$, $i = 1, 2$, $j = 1, 2$

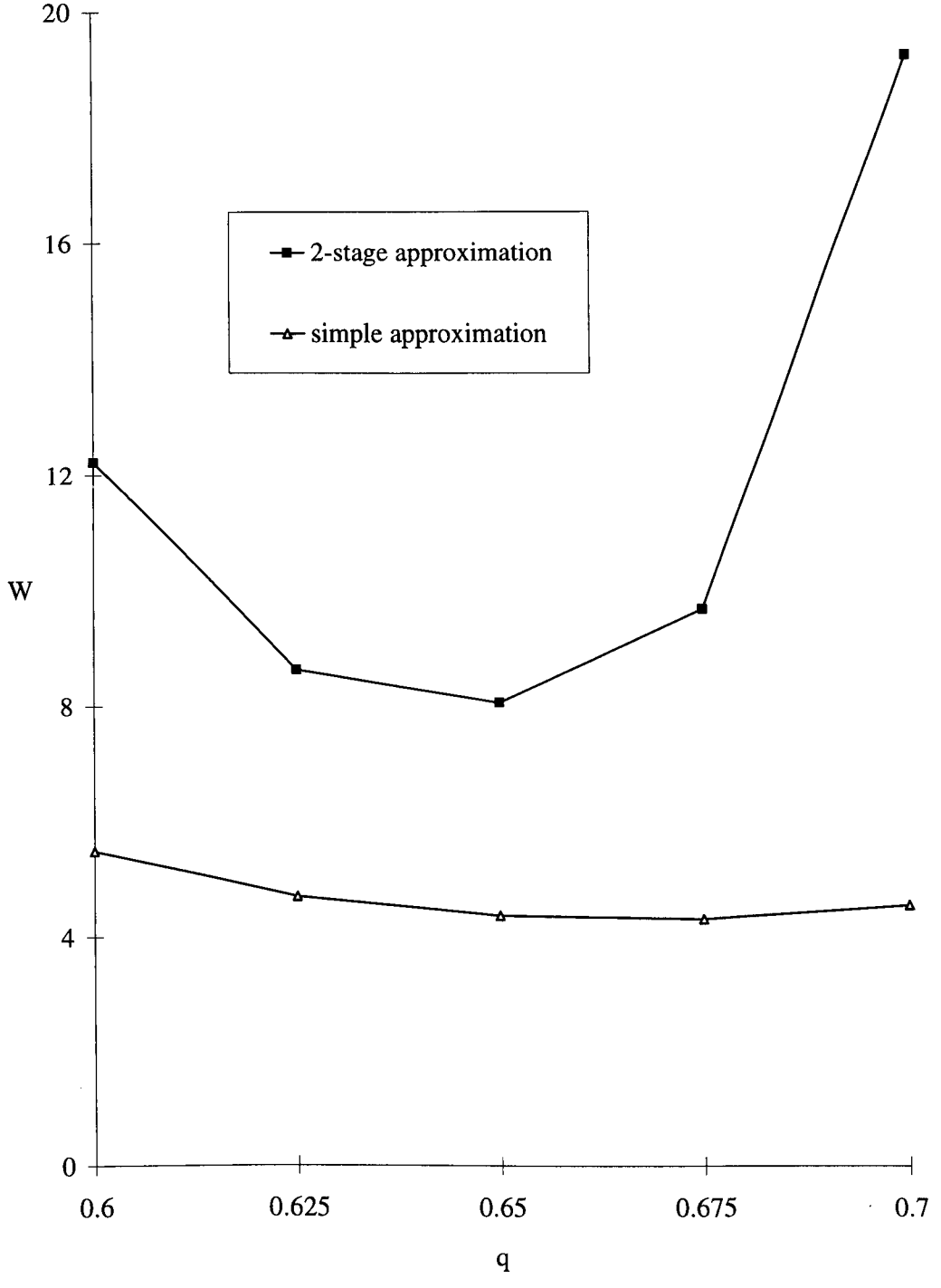


Figure 6.7: Average response time as a function of job share q

at the 2nd stage of a 2 stage pipeline with a fixed routeing strategy

$M = 2$, $N_i = 2$, $\lambda = 17$, $\mu_{1,j} = 10$, $\mu_{2,1} = 14$, $\mu_{2,2} = 9$,

$\xi_{1,j} = 0.0001$, $\eta_{1,j} = 0.001$, $\xi_{2,j} = 0.01$

$\eta_{2,1} = 0.1$, $\eta_{2,2} = 0.07$, $i = 1, 2$, $j = 1, 2$

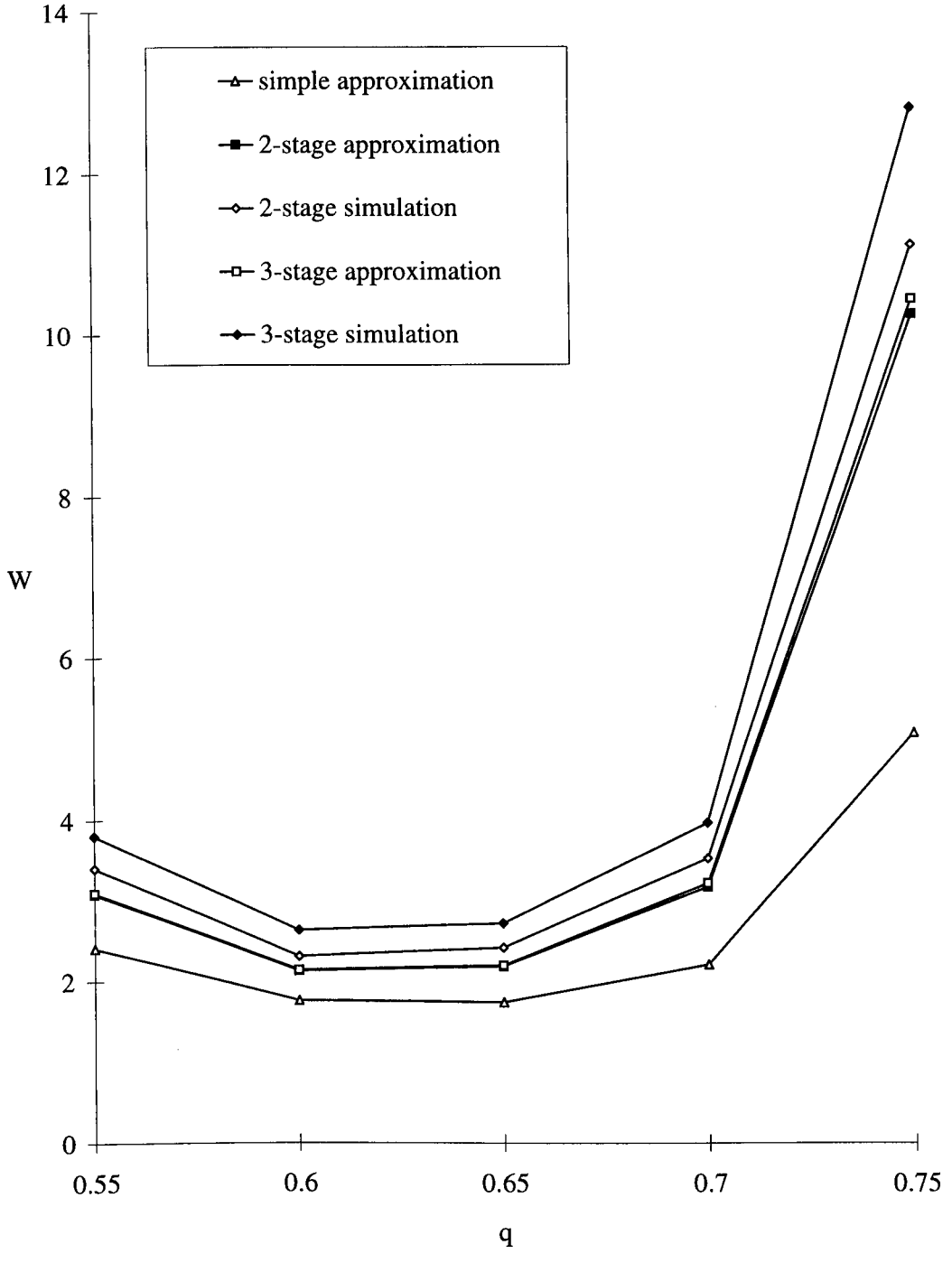


Figure 6.8: Average response time as a function of job share q

at the final stage of a pipeline with a selective routeing strategy

$N_i = 2$, $\lambda = 15$, $\xi_{i,j} = 0.01$, $\mu_{1,j} = 10$, $\eta_{1,j} = 0.2$, $i = 1..3$, $j = 1, 2$

2-stage pipeline: $\mu_{2,1} = 12$, $\mu_{2,2} = 8$, $\eta_{2,1} = 0.2$, $\eta_{2,2} = 0.1$

3-stage pipeline: $\mu_{2,j} = 10$, $\mu_{3,1} = 12$, $\mu_{3,2} = 8$, $\eta_{2,j} = 0.2$, $\eta_{3,1} = 0.2$, $\eta_{3,2} = 0.1$

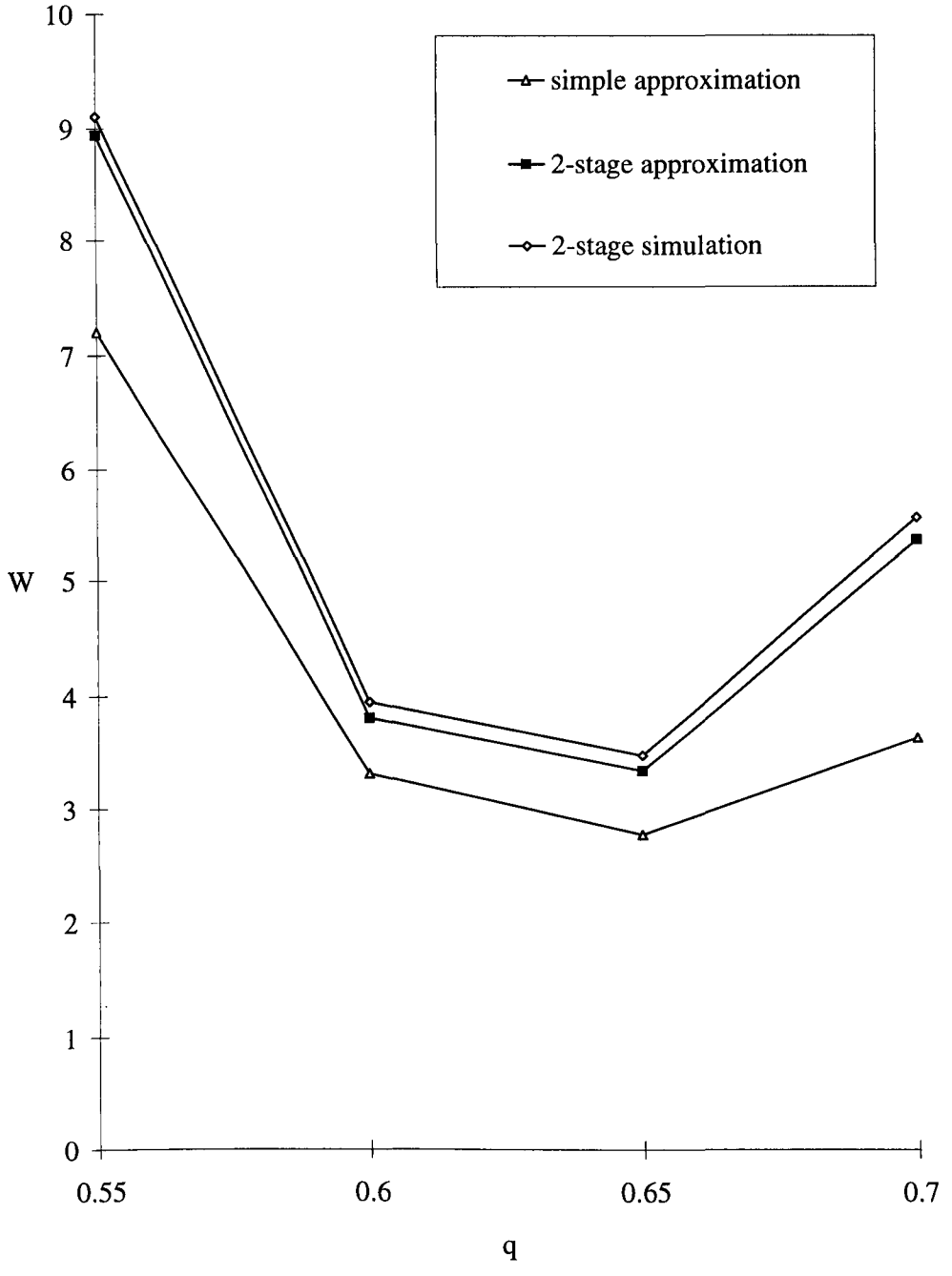


Figure 6.9: Average response time as a function of job share q at the final stage of a pipeline with a fixed routeing strategy

$N_i = 2$, $\lambda = 15$, $\xi_{i,j} = 0.01$, $\mu_{1,j} = 10$, $\eta_{1,j} = 0.2$, $i = 1..3$, $j = 1, 2$

2-stage pipeline: $\mu_{2,1} = 12$, $\mu_{2,2} = 8$, $\eta_{2,1} = 0.2$, $\eta_{2,2} = 0.1$

3-stage pipeline: $\mu_{2,j} = 10$, $\mu_{3,1} = 12$, $\mu_{3,2} = 8$, $\eta_{2,j} = 0.2$, $\eta_{3,1} = 0.2$, $\eta_{3,2} = 0.1$

Chapter 7

Networks of Servers suffering Breakdowns and Repairs

7.1 Summary

The general model presented in this section is an extension of one of the first models of a network of queues proposed, Jackson [42], by considering stages to be parallel systems of queues of the type introduced in chapter 4. Jackson studied networks from the perspective of job scheduling in an assembly plant, but the model is equally valid for a computer network. He was able to show that in a system without failures stages can be studied in isolation without loss of accuracy, i.e. in the network arrivals at each stage may be assumed to be Poisson. It was demonstrated in the previous chapter that this does not hold true when there are failures, however in chapter 6 approximations were introduced that enabled a model of multiple stage service to be solved by breaking the system down into its separate stages. This was feasible because jobs were seen to progress from one

stage to the next in a strict order, however, in a general Jackson network this is not the case as jobs may arrive from outside at any stage and upon completion of service at a stage may depart from the system or pass on to any one of the other stages. Therefore an iterative approach has been suggested using repeated approximations of the kind used in chapter 6. Two specific examples are taken from the literature to illustrate how these techniques can be adapted to suit differing requirements.

7.2 The Model

There are M stages in the system (numbered 1 to M), at stage i there are N_i servers in parallel, each with an associated unbounded queue, to which incoming jobs may be directed. Jobs from outside the system arrive at stage i in a Poisson stream with rate λ_i. Server j at stage i goes through alternating independent operative and inoperative periods, distributed exponentially with means $1/\xi_{i,j}$ and $1/\eta_{i,j}$ respectively. While it is operative, the jobs in its queue receive service of an exponentially distributed duration with mean $1/\mu_{i,j}$ Upon completion of service at a stage i, jobs either depart from the system with probability $p_{i,0}$, or move on to stage k for additional service with probability $p_{i,k}$ ($p_{i,i}$ may be non-zero). Clearly,

$$\sum_{k=0}^{M} p_{i,k} = 1$$

When a server becomes inoperative (breaks down), the corresponding queue, including the job in service (if any), remains in place. Services that are interrupted in this way are eventually resumed from the point of interruption. The system model is illustrated in figure 7.1.

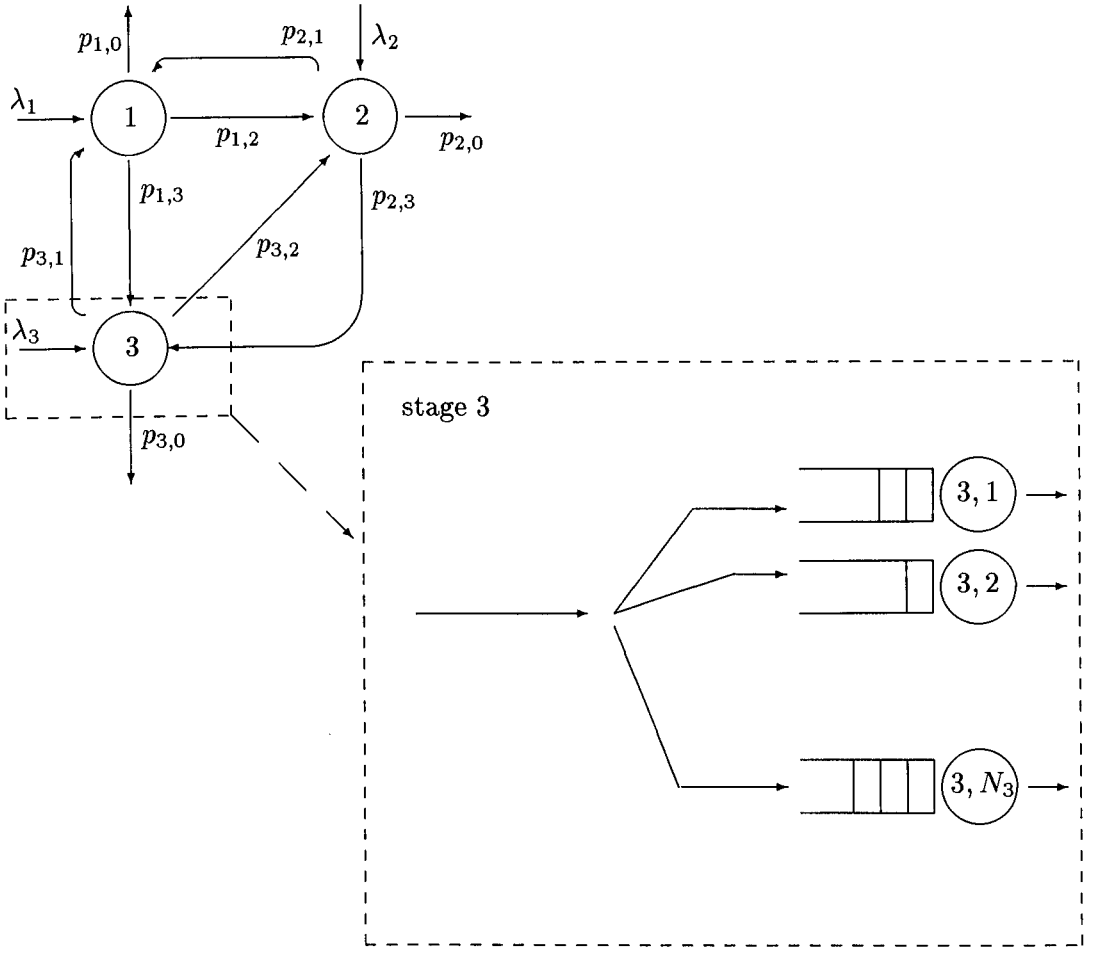


Figure 7.1: A 3 stage Jackson Network with stages of 1 or more servers in parallel

In many ways the unbounded nature of the random walk performed by jobs traversing the network is unrealistic, but may approximate the behaviour of a real system with an extremely large number of non-prioritised types of jobs. An alternative approach is the progressive staged service of the pipeline model in chapter 6, however this negates any notion of feedbacks (an important problem in network behaviour).

7.3 Approximated system configurations

In this model there are 2^N possible system configurations (where N is the total number of servers given by $N = \sum_{i=1}^{M} N_i$), which is generally too large a number to solve for in any practical situation, hence the need for a reduced solution. In general, the arrivals at stage i are dependent on all the preceding stages of service (which with feedback may mean all stages), however it is obvious that the nature of the arrivals at each stage are most strongly linked to the configuration at the immediately preceding stages. Clearly the existence of feedback loops means that a sequential, or progressive, solution is no longer possible, since it is impossible to know the arrival rate of jobs at a stage in a loop without knowing the arrival rates at all the other stages in that loop. Thus one possible reduced solution method is clear, namely,

1. select a stage i for which the rate of arrivals can be most accurately predicted

2. perform the solution described in chapter 4 on stage i

3. extract from that solution the appropriate performance measures and the probabilities $p_{i,j}(\sigma_i), j = 0..N_i$, where $p_{i,j}(\sigma_i)$ is the probability that queue j (at stage i) is non-empty given that the configuration of stage i is σ_i.

4. select a new stage k for which arrivals can be most accurately predicted using the calculations already made

5. perform the solution described in chapter 6 with the approximated system configurations based on the configuration of this stage and those immediately preceding

6. extract from this solution the appropriate performance measures and the probabilities $p_{k,j}(\sigma_k), j = 0..N_k$

7. repeat steps 4, 5 and 6 until all stages have been solved.

8. repeat the whole process using the calculated probabilities to improve the estimation of the arrival rates and hence the accuracy of the calculated performance measures and probabilities until a satisfactory level of accuracy is achieved

The best set of approximated system configurations will be determined by the server characteristics and the available computational resources and some discussion on this is included in the previous chapter. It is assumed that approximated system configurations will be chosen such that server i, j will be either operative or inoperative in any approximated configuration, but not both.

Clearly there is a large amount of scope for work in how to best perform this iterative process in order to minimise the number of iterations needed to achieve satisfaction, in particular the choice of which stage to select and the estimation of arrival rates is crucial. In both the examples presented here the choice of which stage to solve first is obvious as in both cases all arrivals from outside are directed to one stage. This is not the case in the general model and so some care may be needed to select stages in an order that will most quickly lead to an reasonably accurate solution, although in many practical situations the choice will either be obvious or make little difference to the eventual outcome.

The simplest estimation of arrival rates is merely to assume that the only arrivals at stage originate from outside the system or from stages which have already been solved. Such an estimate is likely to be reasonably accurate only for networks having a predom-

inant direction of progression of jobs or in later iterations. In the general case it may take several iterations before such a level of accuracy is achieved, therefore it would be advantageous to speed up this process somewhat, two broad approaches to doing this are suggested. Firstly the use of cruder approximations of the type used in chapter 5 will greatly reduce the number of calculations needed to obtain a reasonable approximation of the arrival streams at each stage, from then on more complex approximations of the kind used in chapter 6 can be used to derive more accurate estimate of the performance measures with fewer iterations. Even with this improvement the initial estimate of the arrivals at a stage may be so inaccurate that a reasonable level of accuracy may still take a long time to achieve, furthermore it may be that the system may be so unstable as to make such any reasonable level of accuracy impossible to achieve without a good first estimate. In such cases a heuristic is needed to predict the arrival rates at each stage, although such a heuristic is likely to be determined by the characteristics of individual networks and so is the subject of much more detailed study than is possible here.

7.4 Example: A 3 stage network with overtaking subject to failures

This example is an extension of a 3 stage network model studied by Mitrani [71]. A job arriving into the network from outside is sent to the queue at stage 1, after being served there it either proceeds to stage 2 with probability p, or proceeds to stage 3 with probability $q = 1 - p$. After service at stage 2 all jobs are sent to the queue at stage 3 and after service at stage 3 all jobs leave the system. All service times for stages 1, 2 and 3 are

independent and exponentially distributed with means $1/\mu_1$, $1/\mu_2$ and $1/\mu_3$ respectively. Failures occur randomly and independently at each stage with time before failure and time to repair are exponentially distributed with means $1/\xi_i$ and $1/\eta_i$ respectively at stage i. In all queues the service discipline is FIFO and interruptions are preemptive resume.

The system model is illustrated in figure 7.2 below.

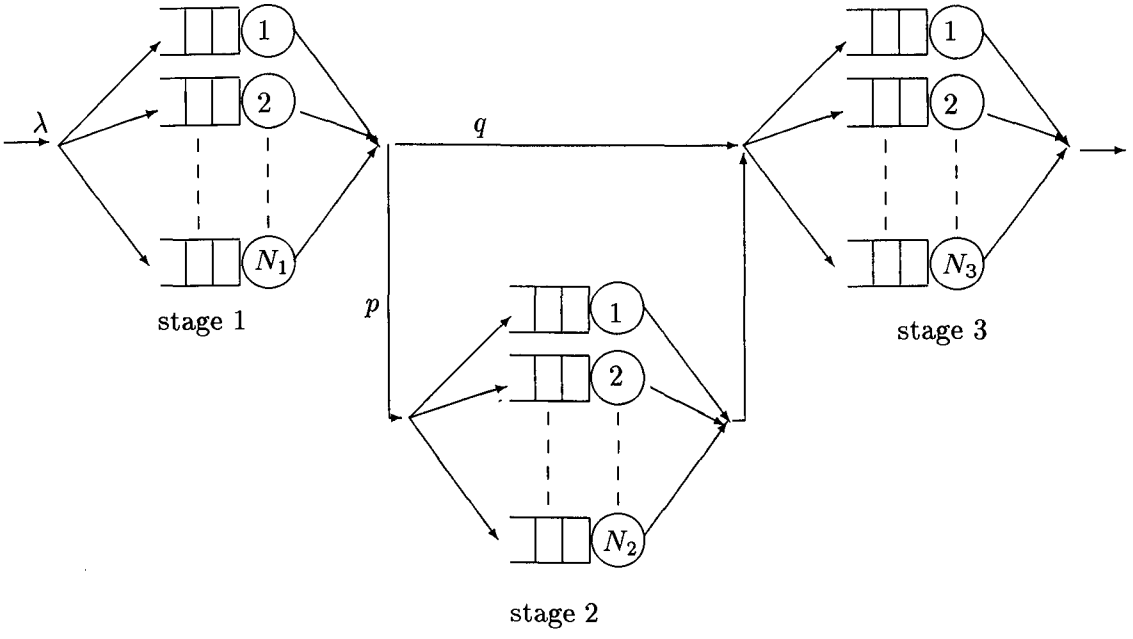


Figure 7.2: A 3 stage Network with overtaking, subject to breakdowns

Clearly the state space involved in an exact solution of even this relatively simple network model is extremely large, even obtaining exact long run average performance measures using the principal of quasi-separability is in general intractable. However, it is possible to obtain an approximate steady state performance measures using some of the models described in earlier chapters.

Applying the principal of quasi-separability gives rise to 3 separate models of queues in parallel which can be combined to give an approximate solution for the average number

of jobs in the system and hence, by Little's theorem, the average response time. Since stage 1 has no preceding stages to disrupt the Poisson arrival stream it can be modelled in isolation without approximation as a simple system of M/M/1 queues in parallel with breakdowns and no jobs lost (see chapter 4). Stages 2 and 3 do have preceding stages and so arrivals at these stages will exhibit some 'bursty' behaviour. Clearly obtaining an approximate solution for stages 2 and 3 is a slight extension of the 2-stage approximation described in chapter 6. In the case of stage 2 the only difference will be that some jobs leaving stage 1 are not directed to stage 2 and for stage 3 there are in effect 2 preceeding stages in parallel.

The accuracy of the approximation described here should be as good as for the pipeline. However, if the number of servers at stages 1 and 2 is large then deriving a good approximation at stage 3 may be numerically impractical, i.e. the number of states required will be too large. In the case of the pipeline model it was argued that if N_i is large then a simple approximation for stage $i+1$ would be reasonably accurate as the effect of a single breakdown at stage i would not significantly affect the arrivals at stage $i+1$. In this example, however, it will take half the number of servers at each stage for a good approximation to be practical, therefore in such cases the accuracy of approximation it is possible to achieve may be significantly less than for the pipeline model.

In a model of this kind where there is no feedback of jobs it is possible to derive a relatively good approximation relatively easily when only performance measures of long run averages are of interest. If transient performance measures are required then the presence of an overtaking loop may significantly increase the difficulty in obtaining a solution.

7.5 Example: A 2 stage Jackson Network subject to breakdowns

Mikou [68] has suggested a model of a tightly coupled 2 stage Jackson network. In this model jobs arrive at from outside the system in a Poisson stream of rate λ. A job arriving into the network is sent to the queue at stage 1, after being served it either leaves the network with probability p, or proceeds to stage 2 with probability $q = 1 - p$. After service at stage 2 jobs are sent to the queue at stage 1. Service times for stages 1 and 2 are exponentially distributed with means $1/\mu_1$ and $1/\mu_2$ respectively. When a breakdown occurs service is suspended at both stages and is resumed from the point of interruption when repair is complete. Time before failure and time to repair are exponentially distributed with means $1/\xi$ and $1/\eta$ respectively. In both queues the service discipline is FIFO.

The system model is illustrated in figure 7.3 below.

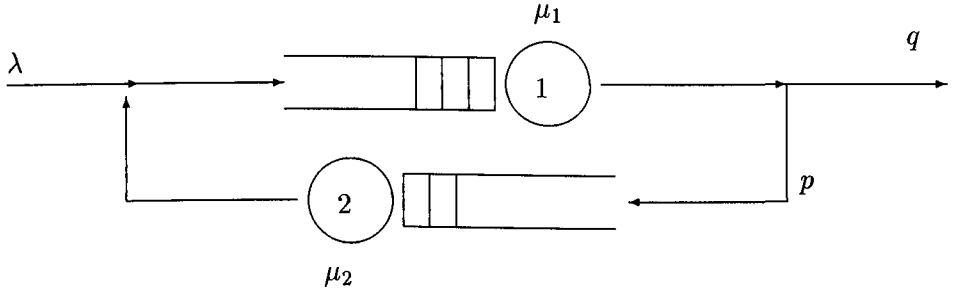


Figure 7.3: Mikou's 2 stage Jackson Network subject to breakdowns

Mikou [68] derived the generating function of the joint distribution of the sizes of the 2 queues in terms of an homogeneous Riemann-Hilbert boundary value problem. Such a solution is far from trivial, but is the only way to attain an exact solution to this model.

7.5.1 General Model

Clearly this model is not of the same form as the other models in this thesis as the same failure and repair process affects both queues, hence there are only 2 states operation; both servers operative or both broken. Whilst examples may be found where this is indeed the case, it is more common to think, as here, of non-catastrophic failures of this sort as being independent at each server in a system. To apply this kind of architecture to the general case described above each stage would consist of N_i $(i = 1, 2)$ server / queue pairs and each server would have it's own independent failure and repair processes. Such a model is illustrated in figure 7.4.

Applying the iterative process described in section 7.3 gives rise to the following process;

1. perform the solution described in chapter 4 on stage 1 using just the single external arrival stream
2. derive the probabilities $p_{1,k}(\sigma_1)$ for $k = 0..N_1 - 1$
3. perform the solution described in chapter 6 on stage 2 using the values of the probabilities $p_{1,k}(\sigma_1)$ found earlier to estimate the arrivals at stage 2
4. derive the probabilities $p_{2,k}(\sigma)$ for $k = 0..N_2 - 1$
5. perform the solution described in chapter 6 on stage 1 using the values of the probabilities $p_{2,k}(\sigma)$ found earlier to estimate the arrivals at stage 1
6. repeat steps 2 to 5 until a steady state is reached

From earlier chapters it is clear that in general the fewer servers there are at each stage,

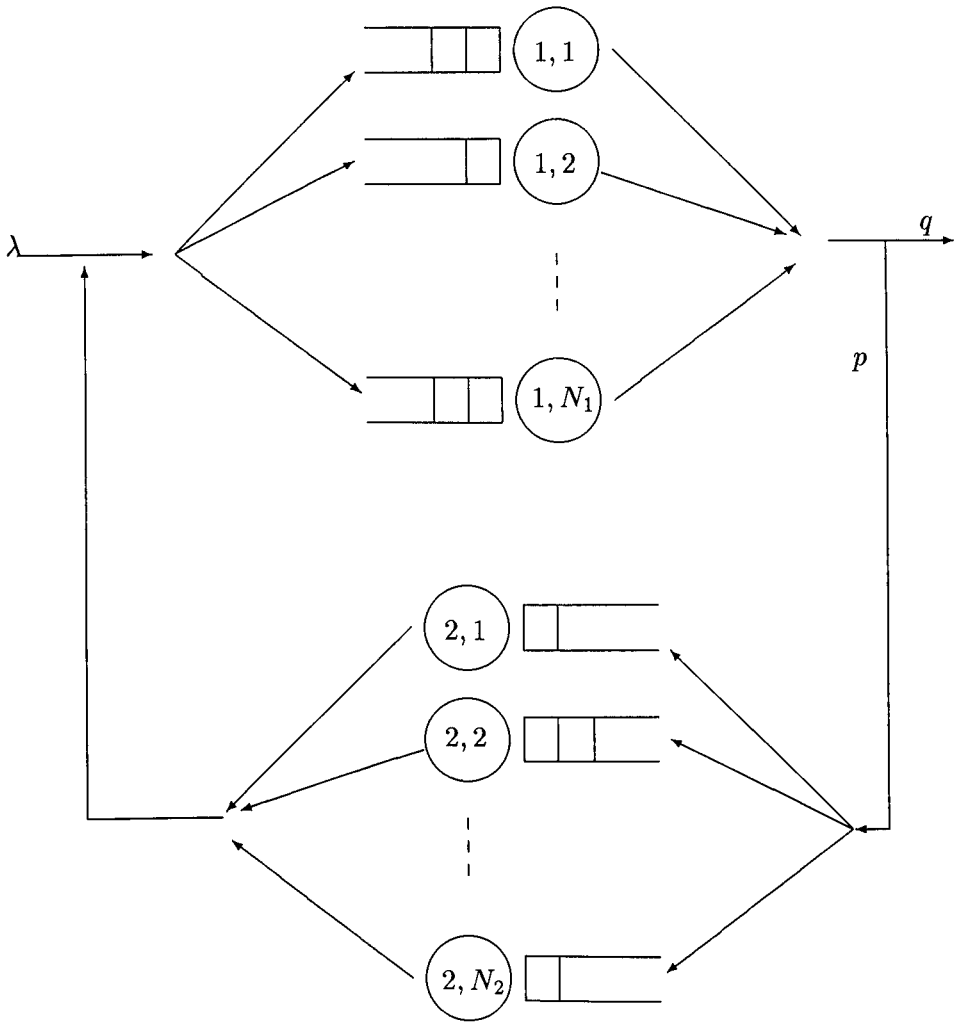


Figure 7.4: A general 2 stage Jackson Network subject to breakdowns

the longer the repair periods are and the greater the load, there worse the approximations will be. This is due principally to the same 'back-log' problem as described above for the coupled failure/repair process, although the mechanics are somewhat complicated by the presence of a feedback loop. If a failure occurs at either stage then there will be a reduction of the service capacity at that stage, causing an increase in the number of jobs in its queues. If the number of servers at the stage is small, then that reduction will be proportionally large and if the load is high the increase in queue size will be significant,

especially if the failure lasts for some time. This argument is true of all the models in previous chapters also, however in this case the reduction of service capacity at one stage will reduce the arrival tate at the other. Therefore there will be fewer jobs to be served at that queue, so the departure rate will decrease. This will mean there are fewer arrivals at the stage affected by the breakdown, so there will be a less dramatic build up of jobs. Thus the presence of the feedback loop may increase the accuracy of the approximation, especially when the external arrival rate, λ, is small and the proportion of jobs sent through the feedback, p, is large.

7.5.2 Simple approximation with correction, $N_i = 1$

Applying the methods used earlier a simple approximation to Mikou's 2 server model ($N_i = 1$), expanded to include independent failures at each server, is easily derived as follows. The first step of the iterative process entails solving stage 1 as an isolated $M/M/1$ queue with breakdowns and arrivals during repair (see section 2.5), thus the average response time at stage 1 is estimated by (2.31). Now consider stage 2 as a single server with 4 states of operation; both stages operative, stage 1 only operative, stage 2 only operative and both stages broken. The arrival rate in the first two states, λ_2 is easily found as,

$$\lambda_2 = p\mu\left(1 - \frac{g_1(0)}{g_1(1)}\right)$$

where p is the proportion of jobs directed towards stage 2, $g_1(0)$ is the probability that the queue is empty and the server is operative, given by (2.25) and $g_1(1)$ is the probability that the server is operative, given by (2.14). There are no arrivals in the remaining 2 states as the server at stage 1 is then broken. It is then a simple matter to perform a

spectral expansion solution of this model as described in chapter 3 to obtain an estimates for the average response time and the probability that the queue is empty during operative periods at stage 2. The first estimate for the average response time of the system has now been made, but unless p is particularly small, it is likely to be a gross underestimate. The next iteration now begins with stage 1 considered as a 4 state single server model in the same was as above with the arrival rate at stage 1 being estimated from the previous solution of stage 2. If p is sufficiently large this iterative process will need to be carried out several times over before a suitable level of accuracy is obtain in the estimation of the average response time of the system.

Clearly this solution is approximate since during its repair the number of jobs in the queue at stage 1 will grow, so that for some period immediately following repair the server at stage 1 will not be idle. This will mean that the arrival rate at stage 2 will be somewhat higher during this time than the average arrival rate, λ_2. This in turn will mean that the queue size at stage 2 may grow somewhat and thus the arrival rate at stage 1 may be higher. This effect may feed back through the queues many times before a steady state is reached, the longer the repair period and the greater the load then the longer the time to steady state will be.

A correction can be made to the approximation derived above by the operative state of each M/M/1 queue into 2 states, a steady state, s, and a busy state, b. The number of jobs which arrive at stage 1 during a repair period will be on average, λ_1/η_1, but in the time taken to clear this 'backlog' a further $\lambda_1^2/(\eta_1\mu_1)$ jobs will have arrived. If the feedback from stage 2 is ignored, then on average approximately $(\lambda_1\mu_1)/(\eta_1(\mu_1-\lambda_1))$ jobs will arrive before the steady state is reached, hence the average length of stay in state b

will be,

$$\frac{\lambda_1}{\eta_1(\mu_1 - \lambda_1)}$$

The proviso 'approximately' is made in the above statement as that assumes a stable Poisson arrival stream at stage 1 of rate λ_1. Clearly this will not be the case as this does not take account of the increased load now applied to stage 2 and the potential for failures there. The failure rate from both new operative states, s and b, will remain as ξ_1 and the repair rate to state b will be η_1. Thus, if the rate of transfer from state b to state s is assumed to be exponentially distributed with mean $\lambda/(\eta(\mu - \lambda))$ and the feedback from stage 2 is ignored, the approximation is modified to the solution of two 5 state $M/M/1$ queues with no job loss. A similar argument can be applied to stage 2 to improve estimation of the arrivals at stage 1 also.

7.6 Conclusions

A practical method for obtaining approximate performance measures for a general model of Jackson queueing networks with breakdowns. This model is consistent with the models presented in the previous chapters, and the approximation technique is an extension of the solution methods already used. Two specific cases of this general model, based on examples taken from the literature, are considered in somewhat greater detail. These specific models can be used to demonstrate the accuracy and stability of this approximation technique, there is scope for a large amount of work in this direction. In addition much work remains to be done in examining more specific examples to explore the questions of stability and accuracy, and also to attempt to derive any generic issues in how best to implement the approximation technique. Such a study remains outside the scope of this thesis and as

such is left as an open problem.

Chapter 8

Conclusions

8.1 Contributions

This thesis has been concerned with a family of models with breakdowns based around the simple single server $M/M/1$ model with negative exponentially distributed breakdowns and subsequent repairs. Several structures have been considered, from single server models, through models of servers in parallel and pipelines with stages consisting of servers in parallel, to general Jackson networks. Many of the models and almost all the results obtained here are new to the literature, and as such must be considered a contribution to extending the type of models which can be solved.

In the case of the single server models the objective was met of obtaining exact closed form solutions to many logical variations of the basic $M/M/1$ model. These models were principally derived in order to define the family of models of interest and to be used to approximate the more complicated scenarios in later chapters. In addition some interesting numerical comparisons were made between some of the models.

In general it is not possible to derive closed form solutions for servers in parallel, so a method was developed to consider each node in isolation and derive exact solutions to performance measures of long run averages. Four routeing strategies were defined based around a simple routeing vector, and the behaviour of systems under these strategies was compared numerically. In order to solve the models derived in this section a numerical method known as spectral expansion was employed. This method, like virtually all matrix solution methods, becomes very expensive (in terms of computational effort involved to obtain a solution) when the number of nodes in parallel is large. As a result of this it was considered necessary to derive approximations to the models of nodes in parallel based around the closed form solutions of the single server models. These approximations were seen to be successful in predicting optimal routeing vectors and also in predicting the performance measure under certain circumstances, particularly when the number of nodes is large, the nodes are reasonably reliable and the duration of breakdowns is not too long. Additionally more complex approximations were proposed that still required the use of the spectral expansion technique, although unlike the exact solutions the amount of work needed does not grow exponentially with the number of node in the system.

The techniques used to derive solutions to models consisting of servers in parallel posed an interesting question: if it was possible to consider a node in isolation in a parallel system, would it be possible to consider a node in isolation in a sequential system? In order to answer this question a model was defined of a pipeline of stages, where each stage is a system of nodes in parallel. Clearly the dependencies between stages are strong, and the presence of breakdowns at earlier stages means that the jobs arriving at a stage are no longer Poisson. An approximation method is described as an extension of the previous

methods and its accuracy is compared with simulation. In general the approximations are seen to perform well, especially when the advantageous circumstances described above hold. The circumstances in which the approximations perform worst are also those where the system is most unstable, therefore simulations need very long run times to obtain satisfactory results. This is clear justification for pursuing this approach.

The pipeline model described above can clearly be seen to be a special type of Jackson network where there is a specific direction of flow. The final model defined in this thesis is a more general Jackson network, where each stage once again consists of nodes in parallel. In the general case a job may revisit a stage many times before leaving the system. In the presence of such feedbacks an iterative approach is suggested where each iteration consists of the solution of a model of the complexity of the pipeline case above. If no feedbacks exist then this model is seen to be of similar complexity to the pipeline model as only long run performance measures are considered.

8.2 Further Work

The literature contains many models of single servers, some similar in nature to the ones described here. The limiting factor in obtaining an exact closed form solution for this class of model is the existence of only one unknown constant in the balance equations, in practice this means that there is only one operational state where service takes place. It is doubtless possible to derive closed form solutions to many more models than those described here or before, however, without a specific problem to solve it would appear to be a purely academic exercise to do so.

The solution methodology described in chapter 3 can be applied to many more general

models involving routeing and breakdowns. For example, a breakdown may be accompanied by the loss of the job in service (if any), with a given probability. The only effect of that assumption is to complicate slightly the *Death* transitions of the process Y_k: these can now be from state (i, j) to state $(i', j-1)$ ($i' = i$ if the departure is due to a service completion and $i' \neq i$ if to a breakdown). The matrix C_k is no longer diagonal but the solution procedure remains unchanged.

Similarly, a breakdown may be *caused*, with a certain probability, by the arrival of a job into a node. That complicates the *Birth* transitions of Y_k, making them from state (i, j) to state $(i', j+1)$. The matrix B_k is then no longer diagonal. Both the above effects may be present in the same model. In addition the likelihood of breakdowns may increased by the number of jobs in the queue passing some threshold level, or a series of thresholds.

It would be easy to modify the selective and selective-m strategies, used in chapters 4 and 5, by making them lose incoming jobs when all servers are broken. In all these models where losses are possible, the average number of jobs lost per unit time is an important performance measure. That quantity is obtained directly from the probabilities (4.1) and from the distributions of the processes Y_k.

Another possible generalisation concerns the introduction of more operative states. For instance, instead of being just operative or broken, a server may be *fully operative*, *partially operative* and *broken*. Perhaps when fully operative the server can both accept and serve jobs, when partially operative it can accept but not serve, and when broken it can neither accept nor serve. In general, a server could be in one of n possible operative states, with different arrival and service characteristics in different states, and with transitions between states governed by an arbitrary Markov chain. Provided that those transitions,

and the routeing decisions, do not depend on how many jobs are present at other queues, the analysis proceeds as in chapter 3.

Of course, the price paid for such an increase in generality is a corresponding increase in complexity. Changing the composition of the matrices A_k, B_k and C_k does not alter significantly the computational complexity of the solution, but changing their size does. That size is determined by the number of system configurations. If, instead of the 2 possible operative states for each server there are n states, the total number of system configurations grows from 2^N to n^N. This imposes obvious limitations on the size of problems that can be solved numerically.

Approximation techniques have been applied successfully in this thesis, and numerical analysis has shown the circumstances where these approximation perform well or poorly. As well as knowing this it would be advantageous to be able to estimate the degree of error, an so possibly apply a correction to the approximation. Since it is possible to analyse why an approximation performs badly it also seems logical that it would be possible to derive a heuristic or further approximation to estimate this correction, such as the approach suggested in chapter 7 No numerical analysis was carried out for the general network model in this thesis. Such an analysis would be a very large undertaking, possibly sufficient for an entire thesis, as such it has been left as an open problem for the future.

Bibliography

[1] M. Ajmone Marsan, G. Balbo and G. Conte, 'Performance Models of Multiprocessor Systems', MIT Press, 1986.

[2] E. Altman and G.M. Koole, 'Stochastic scheduling games with Markov decision arrival processes', *Computers and Mathematics with Applications*, 26(6), pp. 141-148, 1993.

[3] B. Avi-Itzhak and P. Naor, 'Some Queueing Problems with the Service Station Subject to Breakdowns', *Operations Research*, 11, pp. 303-320, 1963.

[4] F. Baccelli and S. Foss, 'Ergodicity of Jackson-Type Queueing Networks', *Queueing Systems*, 17(1-2), pp. 5-72, 1994.

[5] F. Baskett, K.M. Chandy, R. Muntz and J. Palacios, 'Open, Closed and Mixed Networks of Queues with different classes of customers', *Journal of the A.C.M.*, 22, pp. 248-260, 1975.

[6] O.J. Boxma and A.G. Konheim, 'Approximate Analysis of Exponential Queueing Systems with Blocking', *ACTA Informatica*, 15(1), pp. 19-66, 1981.

[7] O.J. Boxma, G.M. Koole and I. Mitrani, 'A Two-Queue Polling Model with a Threshold Service Policy', *Proceedings of MASCOTS '95*, pp. 84-88, IEEE Computer Society Press, 1995.

[8] O.J. Boxma, G.M. Koole, Zhen Liu, 'Queueing-theoretic solution methods for models of parallel and distributed systems', in *Performance Evaluation of Parallel and Distributed Systems - Solution Methods*, CWI Tract 105 and 106, CWI, Amsterdam, 1994.

[9] O.J. Boxma and P.R. de Waal, 'Multiserver Queues with Impatient Customers', Technical Report BS-R9319, CWI, Amsterdam, 1994.

[10] O.J. Boxma, 'Static optimization of queueing systems', in *Recent Trends in Optimization Theory and Applications* (ed R.P. Agarwal), World Scientific Publishing Company, 1995.

[11] J.P. Buzen and A.B. Bondi, 'The response times of priority classes under preemptive resume in M/M/m queues', *Operations Research*, 31(3), pp. 456-465, 1981.

[12] R. Chakka and I. Mitrani, 'Heterogeneous Multiprocessor Systems with Breakdowns: Performance and Optimal Repair Strategies', *Theoretical Computer Science*, 125(1), pp.91-109, 1994.

[13] R. Chakka and I. Mitrani, 'A Numerical Solution Method for Multiprocessor Systems with General Breakdowns and Repairs', *Proceedings of 6th International Conference on Modelling Techniques*, 1991.

[14] A. Cobham, 'Priority assignment in waiting line problems', *Journal of The Operations Research Society of America*, 2, pp. 70-76, 1954.

[15] J.W. Cohen and O.J. Boxma, *Boundary Value Problems in Queueing System Analysis*, North-Holland (Elsevier), 1983.

[16] R.L. Disney and D. Konig, 'Queueing networks: A survey of their random processes', *SIAM Review*, 27, 335-403, 1985.

[17] B.T. Doshi, 'Queueing Systems with Vacations', *Queueing Systems*, 1(1), pp. 29-66, 1986.

[18] B.T. Doshi, 'Single Server Queues with Vacations', *Stochastic Analysis Comp.*, pp. 217-265, 1990.

[19] A.I. Elwalid, D. Mitra and T.E. Stern, 'Statistical Multiplexing of Markov Modulated Sources: Theory and Computational Algorithms', *International Teletraffic Congress*, 1991.

[20] D.H.L. Epema, 'Mean waiting times in a general feedback queueing model with priorities', in *Performance '90*, P.B. King, I. Mitrani and R.J. Pooley (eds.), North-Holland, pp. 221-235, 1990.

[21] P.D. Ezhilchelvan, I. Mitrani and S. Shrivastava, 'A Performance Evaluation Study of Pipeline TMR Systems', *IEEE Transactions on Parallel and Distributed Systems*, 1(4), pp. 442-456, 1990.

[22] G. Fayolle and R. Iasnogorodski, 'Two Coupled Processors: The reduction to a Riemann-Hilbert Problem', *Z. Wahrsheinlichkeitstheorie*, 47, pp. 325-351, 1979.

[23] G. Fayolle, P.J.B. King and I. Mitrani, 'The Solution of Certain Two-Dimensional Markov Models', *Advances in Applied Probability*, 14(2), pp. 295-308, 1982.

[24] G. Fayolle, V.A. Malyshev, M.V. Mensikov and A.F Sidorenko, 'Probabilistic Methods for Jackson Networks', *IFIP Transactions on C-Communication Systems*, 5, pp. 209-223, 1992.

[25] A. Federgreun and L. Green, 'Queueing Systems with Service Interruptions', *Operations Research*, 34(5), pp. 752-768, 1986.

[26] W. Fischer and K.S. Meier-Hellstern , 'The Markov-Modulated Poisson Process', *Performance Evaluation*, 18, pp. 149-171, 1992.

[27] H.R. Gail, S.L. Hunter and B.A. Taylor, 'Analysis of a preemptive priority multiserver queue', in *Data Communication Systems and Their Performance*, L. de Moraes et al (eds), North Holland, 1988.

[28] D.H.L. Epema, 'Mean waiting times in a general feedback queueing model with priorities', *Performance '90*, P.B. King, I. Mitrani and R.J. Pooley (eds.), North-Holland, pp. 221-235, 1990.

[29] F.D. Gakhov *Boundary Value Problems*, Addison Wesley, 1966.

[30] D.P. Gaver, 'A Waiting Line with Interrupted Service Including Priorities', *Journal of the Royal Statistical Society Series B*, 24, pp. 73-90, 1962.

[31] R. Geist, D.E. Stevenson and R.A. Allen, 'The Perceived Effect of Breakdown and Repair on the Performance of Multiprocessor Systems', *Performance Evaluation*, 6(4), pp. 249-260, 1986.

[32] E. Gelenbe, *Multiprocessor Performance*, John Wiley and Sons, 1989.

[33] E. Gelenbe and J. Pujolle, *Introduction to Queueing Networks*, John Wiley and Sons, 1987.

[34] E. Gelenbe and I. Mitrani, 'Analysis and Synthesis of Computer Systems', *Computer Science and Applied Mathematics*, 1980.

[35] B.R. Haverkort, 'Matrix-Geometric Solution of Infinite Stochastic Petri Nets', *Proceedings of IEEE International Computer Performance and Dependability Symposium*, pp. 72-81, 1995.

[36] D.P. Heyman, 'A Priority queueing system with server interference', SIAM Jornal of Applied Mathematics, 17, 1969.

[37] A. Hordijk, G.M. Koole, J.A. Loeve, 'Analysis of a customer assignment model with no state information', *Probability in the Engineering and Informational Sciences*, 8, pp. 419-429, 1994.

[38] A. Hordijk and G.M. Koole, 'On Suboptimal Policies in Multi-Class Tandem Models, *Probability in the Engineering and Informational Sciences*, to appear.

[39] C.R. Heathcote, 'On Priority Queues', *Biometrika*, 48, pp. 57-63, 1961.

[40] O. Idrissi-Kacemi, N. Mikou and S. Saadi, 'Two Processors Only Interacting During Breakdown: The Case Where the Load is Not Lost', submitted for publication.

[41] J.R. Jackson, 'Jobshop-like queueing systems', *Management Science*, 10, pp. 131-142, 1963.

[42] J.R. Jackson, 'Networks of Waiting Lines', *Operations Research*, 5, pp. 518-521, 1957.

[43] N.K. Jaiswal, 'Preemptive Resume Priority Queue', *Operations Research*, 9, pp. 732-742, 1961.

[44] H. Kameda, 'A Finite-Source Queue with Different Customers, *Journal of the A.C.M.*, 29, pp. 478-491, 1982.

[45] H. Kameda, 'Resizeable Performance Vectors of a Finite Source Queue', *Operations Research*, 32, pp. 1358-1367, 1984.

[46] D.G. Keehn, 'Poles for Networks of Markov Queues', *IEEE Transactions on Circuits and Systems I - Fundamental Theory and Applications*, 39(7), pp. 577-582, 1992.

[47] F.P. Kelly, *Reversibility and Stochastic Networks*, Wiley, 1979.

[48] P.J.B. King, *Computer and Communication System Performance Modelling*, Prentice Hall, 1990.

[49] P.J.B. King and I. Mitrani, 'The Effect of Breakdowns on the Performance of Multi-Processors', in *Performance '81* (Ed. F.J. Kylstra), North-Holland, 1982.

[50] L. Kleinrock, *Queueing Systems, Vol. 1*, Wiley, 1975.

[51] A.G. Konheim, I. Meilijson and A. Melkman, 'Processor Sharing of Two Parallel Lines', *Journal of Applied Probability*, 18, pp. 952-956, 1981.

[52] G.M. Koole, 'Assigning multiple customer classes to parallel servers', Technical Report TW-91-07, Leiden University, 1991.

[53] G.M. Koole, 'On the optimality of FCFS for networks of multi-server queues', Technical Report BS-R9235, CWI, Amsterdam, 1992.

[54] G.M. Koole, 'Optimal repairman assignment in two maintenance models which are equivalent to routing models with early decisions, European Journal of Operations Research, 82, pp. 295-301, 1995.

[55] G.M. Koole and M. Vrijenhoek, 'Scheduling a repairman in a finite source system', Technical Report TW-95-03, Leiden University, 1995.

[56] D.D. Kouvatsos, 'Entropy maximisation and queueing network models', Annals of Operations Research, 48, pp. 63-126, 1994.

[57] D.D. Kouvatsos and N. Tabet-Aouel, 'Product-Form Approximations for an Extended Class of General Closed Queueing Networks', *Performance '90*, P. King et al (eds.), North-Holland, pp. 301-315, 1990.

[58] V.G. Kulkarni, V.F. Nicola and K.S. Trivedi, 'On Modelling the Performance and Reliability of Multimode Computer Systems', *The Journal of Systems and Software*, 6(1/2), pp. 175-182, 1986.

[59] V.G. Kulkarni and P.F. Chimento, 'Optimal Scheduling of Exponential Tasks with In-Tree Precedence Constraints on 2 Parallel Processors Subject to Failure and Repair', *Operations Research*, 40(s2), pp. s263-s271, 1992.

[60] K.K. Leung, 'Response time for an additional job served by an execution / sleep sceduling policy in priority systems', *Performance '90*, P.B. King, I. Mitrani and R.J. Pooley (eds.), North-Holland, pp. 209-219, 1990.

[61] D.M. Lucantoni, K.S. Meier-Hellstern and M.F. Neuts, 'A Single Server Queue with Server Vacations and a Class of Non-Renewal Arrival Processes', *Advances in Applied Probability*, 22(3), pp. 676-705, 1992.

[62] R.A. Marie, 'An Approximate Analytical Method for General Queueing Networks', *IEEE Transactions on Software Engineering*, SE-5(5), pp. 530-538, 1979.

[63] R.A. Marie and G. Rubino, 'An Approximation for a Multiclass ./M/1/FIFO Queue Imbedded in a Closed Queueing Network', *Journal of Systems and Software*, 1(2), pp. 31-39, 1986.

[64] B. Melamed, 'Sojourn Times in Queueing Networks', *Mathematics of Operations Research*, 7(2), pp. 223-224, 1982.

[65] J.F. Meyer, 'On Evaluating the Performability of Degradable Computing Systems', *IEEE Transactions on Computers*, 30, pp. 720-731, 1980.

[66] W.J. Meyer, *Concepts of Mathematical Modelling*, McGraw Hill, 1985.

[67] R.G. Miller, 'Priority Queues', *Annals of Mathematical Statistics*, 31, pp.86-103, 1960.

[68] N. Mikou, 'A Two-Node Jackson Network Subject to Breakdowns', *Stochastic Models*, 4, pp. 523-552, 1988.

[69] I. Mitrani, *Modelling of Computer and Communication Systems*, Cambridge University Press, 1987.

[70] I. Mitrani, 'Networks of Unreliable Computers', in *Computer Architectures and Networks* (eds. E. Gelenbe and R. Mahl), North-Holland, 1974.

[71] I. Mitrani, 'Response Time Problems in Communication Networks', *Journal of the Royal Statistical Society Series B*, 47(3), pp. 396-406, 1985.

[72] I. Mitrani and B. Avi-Itzhak, 'A Many-Server Queue with Service Interruptions', *Operations Research*, 16(3), pp. 628-638, 1968.

[73] I. Mitrani and R. Chakka, 'Spectral Expansion Solution for a Class of Markov Models: Application and Comparison with the Matrix-Geometric Method', to appear in *Performance Evaluation.*

[74] I. Mitrani and P.J.B. King, 'Multiserver Systems Subject to Breakdowns: An Empirical Study', *IEEE Transactions on Computers*, C-32, pp. 96-99, 1983.

[75] I. Mitrani and P.J.B. King, 'Multiserver systems with Preemptive Priorities', *Performance Evaluation*, 1, pp. 118-125, 1981.

[76] I. Mitrani and D. Mitra, 'A Spectral Expansion Method for Random Walks on Semi-Infinite Strips', in *Iterative Methods in Linear Algebra* (Eds R. Beauwens and P. de Groen), North-Holland, 1992.

[77] I. Mitrani and P.E. Wright, 'Routing in the Presence of Breakdowns', *Performance Evaluation*, 20, pp. 151-164, 1994.

[78] P.M. Morse, 'Queues, Inventories and Maintenance', John Wiley and Sons, New York, 1958.

[79] N.I. Muskhelishvili, *Singular Integral Equations*, P. Noordhoff, 1953.

[80] P. Nain, 'On a generalisation of the preemptive resume priority', *Advances in Applied Probability*, 18-1, pp. 255-273, 1986.

[81] M.F. Neuts, 'Models Based on the Markovian Arrival Process', *IEICE Transactions on Communications*, E75B(12), pp. 1255-1265, 1992.

[82] M.F. Neuts, *Matrix Geometric Solutions in Stochastic Models*, John Hopkins Press, 1981.

[83] M.F. Neuts and D.M. Lucantoni, 'A Markovian Queue with N Servers Subject to Breakdowns and Repairs', *Management Science*, 25, pp. 849-861, 1979.

[84] V.F. Nicola, 'A Single Server Queue with Mixed Types of Interruptions', *ACTA Informatica*, 23(4), pp465-486, 1986.

[85] V.F. Nicola, V.G. Kulkarni and K.S. Trivedi, 'Queueing Analysis of Fault Tolerant Computer Systems', *IEEE Transactions on Software Engineering*, SE-13(3), pp. 363-375, 1987.

[86] N.U. Prabhu and Y. Zhu, 'Markov-Modulated Queueing systems', *Queueing Systems Theory and Application.*, 5, pp. 215-246, 1989.

[87] T.G. Robertazzi, *Computer Networks and Systems: Queueing Theory and Performance Evaluation*, Springer-Verlag, 1990.

[88] T.G. Robertazzi and A.A. Lazar, 'On Modeling and Optimal Flow Control of the Jacksonian Network', *Performance Evaluation*, 5(1), pp.29-44. 1983.

[89] C.H. Sauer and K.M. Chandy, *Computer Systems Performance Modelling*, Prentice Hall, 1981.

[90] R. Schassberger, 'The Insensitivity of Stationary Probabilities in Networks and Queues', *Advances in Applied Probability*, 10, pp. 906-912, 1978.

[91] J.H. van Schuppen, 'Routing of freeway traffic - A discrete-time state space model and routing problems', Technical Report BS-R9232, CWI, Amsterdam, 1992.

[92] B. Sengupta, 'A Queue with Service Interruptions in an Alternating Markovian Environment', *Operations Research*, 38, pp. 308-318, 1990.

[93] E.D. Silva and M. Gerla, 'Queueing Network Models for Load Balancing in Distributed Systems', *Journal of Parallel and Distributed Computing*, 12(1), pp. 24-38, 1991.

[94] J.R. Spirn, 'Queueing Networks with Random Selection for Service', *IEEE Transactions on Software Engineering*, SE-5(3), pp. 287-289, 1979.

[95] R.P. Sundarraj, P.S. Sundararaghavan and D.R.Fox, 'Optimal Server Acquisition in Open Queueing Networks', *Journal of the Operational Research Society*, 45(5), pp. 549-558, 1994.

[96] H. Takagi, 'Bibliography on Performance Evaluation'

[97] T. Takin and T. Hasegawa, 'Resequencing Delay in preemptive priority M/M/2 queues', *Performance '90*, P.B. King, I. Mitrani and R.J. Pooley (eds.), North-Holland, pp. 109-121, 1990.

[98] Y.C. Tay, 'An Approach to Analysing the Behaviour of Some Queueing Networks', *Operations Research*, 40(s2), pp. s300-s311, 1992.

[99] K. Thiruvengadam, 'Queueing with Breakdowns', *Operations Research*, 11, pp. 62-71, 1963.

[100] N. Thomas and I. Mitrani, 'Routing Among Different Nodes Where Servers Break Down Without Losing Jobs', *Proceedings of IEEE International Computer Performance and Dependability Symposium*, pp. 246-255, 1995.

[101] N. Thomas and I. Mitrani, 'Routing among different nodes', in *Quantitative Methods In Parallel Systems* (Eds F. Baccelli, A. Jean-Marie and I. Mitrani), Springer-Verlag, 1995.

[102] H.C. Tijms and M.C.T. van der Coevering, 'A Simple Numerical Approach for Infinite-state Markov Chains', *Probability in Engineering and Information Sciences*, 5, pp. 285-95, 1991.

[103] D.F. Towsley, 'Queueing Network Models with State-Dependent Routing', *Journal of the A.C.M.*, 27(2), pp. 323-337, 1980.

[104] K. Trivedi, *Probability and Statistics with Reliability, Queueing and Computer Science Applications*, Prentice Hall, 1982.

[105] B. Vinod, 'Unreliable Queueing Systems', *Computers and Operations Research*, 12(3), pp. 323-340, 1985.

[106] P. de Waal, 'A General Model for Maintenance of Complex Systems', Technical Report BS-R9201, CWI, Amsterdam, 1993.

[107] J. Walrand, *An Introduction to Queueing Networks*, Prentice Hall, 1988.

[108] H.C. White and L.S. Christie, 'Queueing with Preemptive Priorities or with Breakdown', *Operations Research*, 6, pp. 79-95, 1958.

[109] M.E. Woodward, *Communication and Computer Networks: Modelling with Discrete-Time Queues*, Pentech Press, 1993.

[110] U. Yechiali, 'A Queueing-Type Birth-and-Death Process Defined on a Continuous Time Markov Chain', *Operations Research*, 21, pp. 604-609, 1973.

[111] Y.F. Yakushev, 'Analysis of Certain Queueing Networks', *Soviet Journal of Computer and Systems Sciences*, 23(2), pp. 42-50, 1985.

[112] Y.X. Zhu, 'Markovian Queueing Networks in a Random Environment', *Operations Research Letters*, 15(1), pp. 11-17, 1994.