Routing Among Servers with Breakdowns and Retained Queues

N. Thomas and I. Mitrani

TECHNICAL REPORT SERIES

No. 527  September, 1995
Routing Among Servers with Breakdowns and Retained Queues

N. Thomas and I. Mitrani

Abstract

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 routing 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 routing strategies are evaluated and compared empirically. Numerical results, including optimal routing, are presented and possible generalizations are considered.
Bibliographical details

THOMAS, Nigel Anthony
Routing Among Servers with Breakdowns and Retained Queues
[By] N. Thomas and I. Mitran
(University of Newcastle upon Tyne, Computing Science, Technical Report Series, no. 527)

Added entries
UNIVERSITY OF NEWCASTLE UPON TYNE.
MITRANI, Israel

Abstract

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 routing 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 routing strategies are evaluated and compared empirically. Numerical results, including optimal routing, are presented and possible generalizations are considered.

About the author

Nigel Thomas is currently completing his PhD at the University of Newcastle upon Tyne entitled "Performance and Reliability in Distributed Systems" for which he is supported under a CASE studentship by British Telecommunications Ltd. His current research is principally concerned with evaluating performance models of queuing systems suffering random breakdowns and with routing issues in an unreliable environment.

Isi Mitran received a PhD degree from the University of Newcastle in 1973. He is now a Professor in the Department of Computing Science. His interests are in the areas of Markov processes, queues, simulation, probabilistic modelling, parallel and distributed systems, and communication networks.

Suggested keywords

BREAKDOWNS AND REPAIRS OPTIMIZATION QUEUEING MODELS

Suggested classmarks (primary classmark underlined)
Dewey (18th): 519.82 001.64404
U.D.C. 519.872 519.687
Routing Among Servers with Breakdowns and Retained Queues

Nigel Thomas  Isi Mitrani

Computing Science Dept, University of Newcastle,
Newcastle upon Tyne NE1 7RU
Email: n.a.thomas@newcastle.ac.uk, isi.mitrani@newcastle.ac.uk
June 7, 1995

Abstract

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 routing 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 routing strategies are evaluated and compared empirically. Numerical results, including optimal routing, are presented and possible generalizations are considered.

1 Introduction

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 [1, 8, 13, 14]). Very few results are available for systems with more than one queue. An approximate solution for a general Jackson network of unreliable nodes was suggested in
Mikou [5] analysed a tightly coupled two-node network with simultaneous breakdowns and repairs, by a far from trivial reduction to a boundary value problem. More recently, Mitran and Wright [11] 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. Idrissi-Kacemi et al. [6] 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.

Of the above citations, only [11] obtains exact performance measures for a model with more than two queues.

Here we consider a system where jobs from a common incoming stream may be directed to one of \( N \) alternative nodes, each of which consists of a single server and an unbounded queue. The service, breakdown and repair processes at the different nodes are independent of each other and have different parameters, in general. The consequences of a breakdown at a server are not too catastrophic: service stops and the existing jobs remain in place; new arrivals during the subsequent repair period may or may not be re-directed to other nodes, depending on the routing policy. There are no job losses.

The routing policies that are examined are almost static. That is, the choice of where to send an incoming job is Bernoulli, independent of past history and of the current queue sizes. However, the probability of selecting a given node may depend on the current server configuration, i.e. on which servers are operative and which are not.

Our motivation for studying this system comes from the field of networking: the jobs are messages generated by some source, and the servers are alternative gateways through which those messages may be routed. Gateways are subject to failures that interrupt service for random periods of time. The source finds out about such failures and may redirect traffic. This naturally raises the question of how to set the routing probabilities. The main purpose of the analysis, therefore, is to determine performance measures so that different routing policies can be evaluated and compared.

The model and its parameters are specified in section 2. Ideally, one would like to find the joint stationary distribution of the set of operative servers and the numbers of jobs in the corresponding queues. To determine that distribution, it is necessary to solve a non-separable
multidimensional Markov process, which is an intractable problem in the general case. The only case for which the joint distribution may be attainable is $N = 2$.

On the other hand, the performance measures of practical interest are mainly concerned with local or global averages, e.g. the average number of jobs present at a given node or the overall average response time. To calculate such performance measures, it is enough to determine the marginal queue size distributions. This last problem can be solved, at least in principle, for arbitrary $N$ (section 3). Problems of comparison and optimization of routing policies can thus be tackled numerically. Several such numerical evaluations are reported in section 4. Various generalisations of the model, where the same solution methodology applies, are mentioned in section 5.

2 The model

Jobs arrive into the system in a Poisson stream with rate $\lambda$. 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 1.

![Figure 1: A single source split among $N$ unreliable nodes](image-url)
The system configuration at any moment is specified by the subset, $\sigma$, 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, $p_\sigma$, of configuration $\sigma$ is given by

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

where $\bar{\sigma}$ is the complement of $\sigma$ with respect to $\Omega_N$ 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 $\sigma$, 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 routing policy is defined by specifying $2^N$ vectors,

$$q(\sigma) = [q_1(\sigma), q_2(\sigma), \ldots, q_N(\sigma)], \quad \sigma \subset \Omega_N,$$

such that for every $\sigma$,

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

The system state at time $t$ is specified by the pair $[I(t), J(t)]$, 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 $J(t)$ 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, $X = \{[I(t), J(t)] : t \geq 0\}$ 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:

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

When the routing 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. 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

$$W = \frac{1}{\lambda} \sum_{k=1}^{N} L_k.$$  \hspace{1cm} (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 $Y_k = \{[I(t), J_k(t)] \mid t \geq 0\}$ ($k = 1, 2, \ldots, N$), 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 $Y_k$:

$$p_k(i, j) = \lim_{t \to \infty} P[I(t) = i, J_k(t) = j], \hspace{1cm} i = 0, 1, \ldots, 2^N - 1, \hspace{1cm} j = 0, 1, \ldots. \hspace{1cm} (5)$$

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

$$L_k = \sum_{j=1}^{2^N-1} \sum_{i=0}^{2^N-1} p_k(i, j). \hspace{1cm} (6)$$

3 Queue size distributions

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

(a) from state $(i, j)$ to state $(i', j)$, where $i'$ is a configuration with either one more, or one fewer operative server;

(b) from state $(i, j)$ to state $(i, j + 1)$, if the routing probability to queue $k$ in configuration $i$, $q_k(i)$, is non-zero;

(c) from state $(i, j)$ to state $(i, j - 1)$, if $j > 0$ and server $k$ is operative in configuration $i$. 

5
The balance equations for $Y_k$ are best written in vector and matrix form. Define the (row) vector of equilibrium probabilities of all states with $j$ jobs in queue $k$:

$$v_k(j) = [p_k(0,j), p_k(1,j), \ldots, p_k(2^N - 1,j)] , \quad j = 0, 1, \ldots \ .$$

(7)

Let $A = (a_{i,i'}) (i, i' = 0, 1, \ldots, 2^N - 1)$ be the matrix of instantaneous transition rates corresponding to transitions (a). If in configuration $i$ the subset of operative servers is $\sigma$, and in $i'$ it is $\sigma + \{\ell\}$, for some server $\ell$, then $a_{i,i'} = \eta_{i\ell}$; similarly, if in $i'$ the configuration is $\sigma - \{\ell\}$, for some server $\ell$, then $a_{i,i'} = \xi_{i\ell}$. It is also useful to introduce the diagonal matrix, $D_A$, whose $i$'th diagonal element is the $i$'th row sum of $A (i = 0, 1, \ldots, 2^N - 1)$.

Let $B_k$ be the diagonal matrix whose $i$'th diagonal element is equal to $\lambda q_k(i)$; these elements are the instantaneous transition rates corresponding to transitions (b). Also, let $C_k$ be the diagonal matrix whose $i$'th diagonal element is equal to $\mu_k$ if server $k$ is operative in configuration $i$, and 0 otherwise; these are the instantaneous transition rates corresponding to transitions (c).

When $j > 0$, the vectors (7) satisfy the balance equations

$$v_k(j)(D_A + B_k + C_k) = v_k(j)A + v_k(j - 1)B_k + v_k(j + 1)C_k , \quad j = 1, 2, \ldots \ .$$

(8)

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

$$v_k(0)(D_A + B_k) = v_k(0)A + v_k(1)C_k \ .$$

(9)

In addition, all probabilities must sum up to 1:

$$\sum_{j=0}^{\infty} v_k(j)e = 1 ,$$

(10)

where $e$ is a column vector with $2^N$ elements, all of which are equal to 1.

The above equations can be solved by several methods. Perhaps the best approach is to use spectral expansion (see [9, 10]). Rewrite (8) in the form

$$v_k(j)Q_{k,0} + v_k(j + 1)Q_{k,1} + v_k(j + 2)Q_{k,2} = 0 , \quad j = 0, 1, \ldots \ ,$$

(11)

where $Q_{k,0} = B_k$, $Q_{k,1} = A - D_A - B_k - C_k$ and $Q_{k,2} = C_k$. This is a homogeneous vector difference equation of order 2, with constant coefficients. Associated with it is the characteristic matrix polynomial, $Q_k(z)$, defined as

$$Q_k(z) = Q_{k,0} + Q_{k,1}z + Q_{k,2}z^2 .$$

(12)
Denote by \( z_{k,\ell} \) and \( \psi_{k,\ell} \) the generalized eigenvalues and left eigenvectors of \( Q_k(z) \). These quantities satisfy
\[
\psi_{k,\ell} Q_k(z_{k,\ell}) = 0, \quad \ell = 1, 2, \ldots, d,
\]
where \( d = \text{degree} \{ \text{det}(Q_k(z)) \} \).

The eigenvalues do not have to be simple, but it is assumed that if \( z_{k,\ell} \) has multiplicity \( r \), then it has \( r \) linearly independent left eigenvectors. This is invariably observed to be the case in practice. Under that assumption, any solution of (11) is of the form
\[
v_k(j) = \sum_{\ell=1}^{d} x_{k,\ell} \psi_{k,\ell} z_{k,\ell}^j, \quad j = 0, 1, \ldots,
\]
where \( x_{k,\ell} (\ell = 1, 2, \ldots, d) \), are arbitrary (complex) constants.

Moreover, since only normalizeable solutions are acceptable, if \( |z_{k,\ell}| \geq 1 \) for some \( \ell \), then the corresponding coefficient \( x_{k,\ell} \) must be set to 0. Numbering the eigenvalues of \( Q_k(z) \) in increasing order of modulus, the spectral expansion solution of equation (11) can be written as
\[
v_k(j) = \sum_{\ell=1}^{c} x_{k,\ell} \psi_{k,\ell} z_{k,\ell}^j, \quad j = 0, 1, \ldots,
\]
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 \( Q_k(z) \) have always been observed to be simple, real and positive.

Substituting (15), for \( j = 0 \) and \( j = 1 \), into (9), yields a set of homogeneous linear equations for the unknown coefficients \( x_{k,\ell} \). There are \( 2^N - 1 \) independent equations in this set (rather than \( 2^N \)) because the generator matrix of the Markov process is singular. A further, non-homogeneous equation is provided by (10), which now becomes
\[
\sum_{\ell=1}^{2^N} x_{k,\ell} \psi_{k,\ell} e^{z_{k,\ell}} = 1.
\]

These equations can be solved uniquely for the coefficients \( x_{k,\ell} \), if \( c = 2^N \). This turns out to be the case when (3) is satisfied. Indeed, the ergodicity condition is equivalent to the requirement that \( Q_k(z) \) has exactly \( 2^N \) eigenvalues strictly inside the unit disk.
Having determined the coefficients \( x_{k,\ell} \), the average number of jobs in queue \( k \) is obtained by substituting (15) into (6):

\[
L_k = \sum_{\ell=1}^{2^N} \frac{x_{k,\ell} z_{k,\ell} \Psi_{k,\ell} \beta}{(1 - z_{k,\ell})^2}.
\] (16)

4 Evaluation of scheduling strategies

In order to reduce the number of parameters that have to be given values when defining the routing strategy, we shall evaluate and compare several strategies based on a single routing vector, \( q = (q_1, q_2, \ldots, q_N) \). In each case, the optimization problem is to choose the elements of that vector so as to minimize the average response time, given by (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 [1, 8, 14]):

\[
L_k = \frac{\lambda q_k [(\xi_k + \eta_k)^2 + \xi_k \mu_k]}{\xi_k + \eta_k - \lambda q_k (\xi_k + \eta_k)}.
\] (17)

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 \( \sigma \), and that subset is non-empty, send jobs to node \( k \) only if \( k \in \sigma \), with probability proportional to \( q_k \):

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

If \( \sigma \) 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
\( \sigma \), an incoming job can be directed to any node \( k \) for which \( k \leq m \) or \( k \in \sigma \), or both, with probability

\[
q_k(\sigma) = \frac{q_k}{\sum_{l \in \{1, 2, \ldots, m\} \cup \sigma} q_l}, \quad (k \leq m) \lor (k \in \sigma).
\]

Figure 2: Average response time as a function of the job arrival rate.

\( N = 3, \mu_1 = 150, \mu_2 = 170, \mu_3 = 190, \xi_1 = 20, \xi_2 = 30, \xi_3 = 40, \eta_1 = \eta_2 = \eta_3 = 50 \)
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 \(\sigma\), and \(\sigma \neq \emptyset\), an incoming job is directed to node \(k\), only if \(k \in \sigma\), with probability proportional to \(q_k\):

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

If \(\sigma\) is empty, the job is sent to one of the first \(m\) nodes, with probability

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

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 routing 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 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 routing 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 outperforms 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 routing vector constant, it is optimized for each value of \(\lambda\), then the corresponding plots
do not cross: fixed(1) becomes uniformly better than fixed(2), which in turn becomes better than fixed.

![Graph](image)

Figure 3: Average response time in a 2-node system, as a function of the routing vector \((q, 1-q)\). \(\lambda = 50, \mu_1 = 150, \mu_2 = 100, \xi_1 = \xi_2 = 1, \eta_1 = \eta_2 = 10\)

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 routing.

Figure 3 concerns a 2-node system where the routing vector, \((q, 1 - q)\), is varied on the range \(0 \leq q \leq 1\) (remember that that vector is used in making routing 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 routing vector.

(c) For the fixed and two selective strategies, the best routing 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 in (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 shows the performance of a 5-node system as a function of the job arrival rate, under an approximately optimal routing vector. For each value of \(\lambda\), 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 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 routing vector is employed.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4}
\caption{Optimised average response time as a function of the job arrival rate. $N = 5$, $\mu_1 = 150$, $\mu_2 = 160$, $\mu_3 = 170$, $\mu_4 = 180$, $\mu_5 = 190$, $\xi_1 = \xi_2 = \xi_3 = \xi_4 = \xi_5 = 50$, $\eta_1 = 50$, $\eta_2 = 60$, $\eta_3 = 70$, $\eta_4 = 80$, $\eta_5 = 100$}
\end{figure}

A numerical search for the optimal routing 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

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

Figure 5: Optimised average response time as a function of the job arrival rate.

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

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

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 routing vector be

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

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 5, the performance of the above heuristic is compared to that of the optimal routing vector (which does depend on \(\lambda\)), 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 routing strategy. The servers have the same breakdown and repair characteristics (about 90\% operative), but different speeds. The average response time is plotted against \(\lambda\).

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 routing vector can be very large. Unfortunately, the fine performance of the heuristic under the selective routing 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.

We combine the attempt to improve the heuristic with that of 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 \(\lambda_{i1}\), and are served at rate \(\mu_i\). During inoperative periods, distributed exponentially with mean \(1/\eta_i\), jobs arrive in a Poisson stream at rate \(\lambda_{i0}\), and the service rate is 0. For a given strategy and routing 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

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

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

where the probabilities \(p_{\sigma}\) are given by (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 routing strategy, the approximation coincides with the exact solution. The two arrival rates are then equal: $\lambda_{i1} = \lambda_{i0} = \lambda q_i$.

Figure 6: Exact and approximate solutions of average response time with optimal routing vector for different strategies. $N = 5$, $\mu_1 = 150$, $\mu_2 = 160$, $\mu_3 = 170$, $\mu_4 = 180$, $\mu_5 = 190$, $\xi_1 = \xi_2 = \xi_3 = \xi_4 = \xi_5 = 50$, $\eta_1 = 50$, $\eta_2 = 60$, $\eta_3 = 70$, $\eta_4 = 80$, $\eta_5 = 100$

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

\[
L_i = \frac{\eta_i \lambda_{i1} + \xi_i \lambda_{i0} + \frac{\xi_i}{\xi_i + n} \lambda_{i0} (\mu_i + \lambda_{i0} - \lambda_{i1})}{\eta_i \mu_i - \eta_i \lambda_{i1} - \xi_i \lambda_{i0}}.
\] (18)

Note that if \( \lambda_{i0} = 0 \), i.e. if node \( i \) does not accept jobs while broken, then (18) reduces to the standard result for the average queue size in an \( M/M/1 \) queue with parameters \( (\lambda_{i1}, \mu_i) \).

The following optimization procedure is now suggested: for a given strategy, find the routing vector which minimizes the approximate average response time, \( W_{\text{approx}} \). The search for that vector is considerably facilitated by the ease of computing \( W_{\text{approx}} \).

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 optimization is practically indistinguishable from that obtained by optimizing 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 routing vector. A comparison between the exact and approximate values of \( W \), in the context of a 5-node system under several routing strategies, is illustrated in Figures 6 and 7. In figure 6, the fixed(\( 4 \)) and the selective strategies are evaluated for different values of \( \lambda \) and the corresponding optimal routing vector. In all cases, the approximation underestimates the exact response time. 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 not exceed 10%.

Figure 7 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.

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 $O(N^{2^N})$.

Figure 7: Exact and approximate solutions of average response time, with optimised routing vector, as a function of job arrival rate. $N = 5$, $\lambda = 350$, $\mu_1 = 150$, $\mu_2 = 160$, $\mu_3 = 170$, $\mu_4 = 180$, $\mu_5 = 190$, $\xi_1 = \xi_2 = \xi_3 = \xi_4 = \xi_5 = 50$, $\eta_1 = 50$, $\eta_2 = 60$, $\eta_3 = 70$, $\eta_4 = 80$, $\eta_5 = 100$

This is a large computational effort even for systems of 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 Joint distribution for $N = 2$

It has already been pointed out that the problem of determining the steady-state joint distribution of the numbers of jobs at two nodes is of mainly theoretical interest. Moreover, the methods for solving two-dimensional Markov processes by reduction to boundary value problems are quite well known (see [3, 2]). We shall therefore present the analysis of the case $N = 2$ only in outline, omitting many of the details. On the other hand, this model has some unusual features which justify the inclusion of such an outline.

A two-node system has four possible configurations: both servers broken, server 2 operative and server 1 broken, server 1 operative and server 2 broken, both servers operative. These configurations will be numbered 0, 1, 2 and 3, respectively. To simplify the notation, assume that the selective routing policy with routing vector $(q_1, q_2)$ is employed (other strategies are treated similarly). Thus, in configurations 0 and 3, jobs are directed to node $k$ with probability $q_k$ ($k = 1, 2$), while in configurations 1 and 2 they go to the only operative server with probability 1.

Denote by $p_i(j_1, j_2)$ the steady-state probability that the configuration is $i$, and there are $j_1$ jobs in queue 1 and $j_2$ jobs in queue 2. Introduce the generating functions

$$g_i(x, y) = \sum_{j_1=0}^{\infty} \sum_{j_2=0}^{\infty} p_i(j_1, j_2)x^{j_1}y^{j_2}, \quad i = 0, 1, 2, 3.$$  \hspace{1cm} (19)

It is a simple matter to transform the balance equations of the Markov process into four equations involving these generating functions:

$$\alpha(x, y)g_0(x, y) = \xi_2 g_1(x, y) + \xi_1 g_2(x, y),$$  \hspace{1cm} (20)

$$\beta(y)g_1(x, y) = \eta_2 yg_0(x, y) + \xi_1 g_3(x, y) - \mu_2(1-y)g_1(x, 0),$$  \hspace{1cm} (21)

$$\gamma(x)g_2(x, y) = \eta_1 xg_0(x, y) + \xi_2 g_3(x, y) - \mu_1(1-x)g_2(0, y),$$  \hspace{1cm} (22)
and
\[ R(x, y)g_3(x, y) = -\mu_1 y(1 - x)g_3(0, y) - \mu_2 x(1 - y)g_3(x, 0) \]
\[ + \eta_1 x y g_1(x, y) + \eta_2 x y g_2(x, y) , \tag{23} \]
where
\[ \alpha(x, y) = \lambda q_1(1 - x) + \lambda q_2(1 - y) + \eta_1 + \eta_2 , \]
\[ \beta(y) = \lambda y(1 - y) - \mu_2(1 - y) + \xi_2 y + \eta_1 y , \]
\[ \gamma(x) = \lambda x(1 - x) - \mu_1(1 - x) + \xi_1 x + \eta_2 x , \]
\[ R(x, y) = \lambda q_1 x y (1 - x) + \lambda q_2 x y (1 - y) - \mu_1 y (1 - x) - \mu_2 x (1 - y) + (\xi_1 + \xi_2) x y . \]

The first step of the solution is to use (20), (21) and (22) in order to eliminate \( g_0(x, y) \), and express \( g_1(x, y) \) and \( g_2(x, y) \) in terms of \( g_3(x, y) \) and the two 'boundary' functions \( g_1(x, 0) \) and \( g_2(0, y) \). This yields two relations of the form
\[ D(x, y)g_k(x, y) = a_k(x, y)g_1(x, 0) + b_k(x, y)g_2(0, y) + c_k(x, y)g_3(x, y) , \tag{24} \]
for \( k = 1, 2 \), where \( D(x, y), a_k(x, y), b_k(x, y) \) and \( c_k(x, y) \) are known.

Now consider (23), and suppose for the moment that \( g_1(x, y) \) and \( g_2(x, y) \) are known. Then in the right-hand side of that equation there are two unknown boundary functions, \( g_3(x, 0) \) and \( g_3(0, y) \). These can be obtained by exploiting the fact that whenever \( R(x, y) = 0 \) and \( g_3(x, y) \) is finite, the r.h.s. of (23) must vanish (for a similar development see [11]). There is an algebraic curve \( y = \varphi_1(x) \), which satisfies \( R(x, \varphi_1(x)) = 0 \), such that when \( x \) is on a certain closed contour (in this case the circle \( |x| = \sqrt{\mu_1/(\lambda q_1)} \)), \( \varphi_1(x) \) is real. Using that curve, the problem of determining \( g_3(x, 0) \) is reduced to one of finding a function which is analytic in the interior of a closed contour and satisfying on that contour a specific boundary condition. That is a boundary value problem, whose solution is given in an integral form.

In the same way, by using an algebraic curve \( x = \varphi_2(y) \), with \( R(\varphi_2(y), y) = 0, y \) on a closed contour and \( \varphi_2(y) \) real, the determination of \( g_3(0, y) \) is reduced to the solution of another boundary value problem.

Thus, if \( g_1(x, y) \) and \( g_2(x, y) \) are known on the two curves \( y = \varphi_1(x) \) and \( x = \varphi_2(y) \), the function \( g_3(x, y) \) can be determined for all \( x \) and \( y \). There are also a couple of unknown constants that appear in the solution (e.g., \( g_3(1, 0) \) and \( g_3(0, 1) \)). These are obtained from the known marginal probabilities.
A similar approach applies to the solution of (24). If \( g_3(x, y) \) is known on two curves, \( y = \psi_1(x) \) and \( x = \psi_2(y) \), which satisfy \( D(x, \psi_1(x)) = 0 \) and \( D(\psi_2(y), y) = 0 \) respectively, then \( g_1(x, y) \) and \( g_2(x, y) \) can be determined for all \( x \) and \( y \) by solving two (different) boundary value problems.

The above discussion suggests an iterative procedure for finding all unknown functions:

1. Make an initial guess for the values of \( g_3(x, y) \) on the curves \( y = \psi_1(x) \) and \( x = \psi_2(y) \);
2. Compute \( g_1(x, y) \) and \( g_2(x, y) \) on the curves \( y = \varphi_1(x) \) and \( x = \varphi_2(y) \);
3. Compute \( g_3(x, y) \) on the curves \( y = \psi_1(x) \) and \( x = \psi_2(y) \);
4. Repeat steps 2 and 3 until the differences between two consecutive iterates for \( g_1(x, y) \), \( g_2(x, y) \) and \( g_3(x, y) \) becomes sufficiently small.

Thus the solution of the model is reduced, via the boundary value problems, to determining the fixed point of three functional equations. If the ergodicity condition is satisfied and the iterative procedure converges to valid generating functions, then it must provide the unique solution of the Markov process. We do not have a proof of convergence. As is often the case with fixed-point solution methods, the only way of assessing the feasibility and efficiency of the approach is by numerical experimentation. That is outside the scope of this paper.

6 Generalizations

The solution methodology described in section 3 can be applied to more general models involving routing 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.
It would be easy to modify the selective and selective-m strategies 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 (1) and from the distributions of the processes $Y_k$.

Another possible generalization 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 routing decisions, do not depend on how many jobs are present at other queues, the analysis would proceed as in section 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.

7 Conclusions

The type of 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 optimize the splitting of the input stream among the nodes, under different routing policies. However, such an optimizations 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 routing 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.

Acknowledgement

This work was carried out in connection with the Basic Research projects QMIPS (Quantitative Methods In Parallel Systems), and PDCS 2 (Predictably Dependable Computer Systems), funded by the European Union. The first author is supported by a CASE scholarship and is sponsored by British Telecom.

References


