1. INTRODUCTION

At the origin of queueing theory is Erlang's loss system to analyze the call congestion in a telephone exchange. It is well-known (cf. Cohen [1]) that the blocking probability depends only on the expected service time. This phenomenon is a special case of the following insensitivity property.

It is encountered in a number of basic queueing models and it says: "The stationary distribution depends on the service time distribution only through their means".

In practice telephone systems known as gradings, have several instead of one trunk group. The appropriate queueing model then has several queues. In the most simple case queues $Q_1$ and $Q_2$ and customers or calls who find all servers of $Q_1$ busy, are offered to $Q_2$. It turns out that for this overflow model the insensitivity property is lost.

For computing the blocking probability it is mostly assumed that the service time distributions are exponential. Although in practice they often are non-exponential. Even for the exponential model it is not simple to compute the call congestion. There does not exist a closed form expression for the stationary distribution.

Queues or more generally networks of queues fall in two classes. The first class consists of those models which have a closed form expression for the stationary distribution. It is usual to call these the product form queueing models. The second class of nonproduct form models are in general much more difficult to analyze. Erlang's loss model belongs to the first class, whereas the overflow model is of the second type.

The queueing models of product form have all a kind of partial balance. In Hordijk and Van Dijk [7] the jobmark process is introduced. The state is then
defined as the union of the marks of the jobs present in the system. These marks summarize all relevant characteristics of the jobs. The appropriate partial balance in this setting becomes job local balance: "The probability flow out of a state due to a transition of a certain job is equal to the probability flow into that state due to a transition of that same job".

The job local balance property is necessary and sufficient for the insensitivity property (cf. Hordijk [3] or Whittle [11]). The problem in verifying whether this partial balance property holds is that the stationary distribution is unknown. That is the reason that in [7] the notion of adjoint process is used. For a network of queues we get an adjoint process by making the routing reversible. For the overflow model the adjoint process is equal to the original.

The usefulness of adjoint processes follows from the following theorem, "The job local balance property holds if and only if adjoint processes are reversible". Moreover, the stationary distributions of original and adjoint processes are in that case the same.

It is well known that the reversibility of a process can be verified through the Kolmogorov property, "The product of the transition rates in a cycle of states in one direction is equal to the product of the transition rates in the reversed direction". It follows from this property that the stationary probabilities have a closed form expression. If we denote states with the generic notation $E$ (a set of job-marks), $q(E_i,E_j)$ the transition rate from state $E_i$ to $E_j$ and $P(E)$ the stationary probability of state $E$ then it reads,

\[
P(E) = P(E_0) \frac{q(E_0,E_1)}{q(E_1,E_0)} \frac{q(E_1,E_2)}{q(E_2,E_1)} \cdots \frac{q(E_k,E)}{q(E,E_k)}
\]

for $E_0$ an arbitrarily chosen reference state (e.g. the empty system) and any path of states such that the denominators are positive. Product form queueing models have a stationary distribution of the form (1) with possibly the rate $q$ replaced by $q_A$ the rate of an adjoint process. These models satisfy the job local balance property and are insensitive.

Unfortunately, the majority of queueing models are of the second class. The idea of the methodology discussed in this paper is to bound the original model between an upper and lower model which both are of product form. These bounds are then insensitive and we may expect that the bounds remain valid for non-exponential service time distributions.
This method requires ingenuity to find suitable upper and lower models. In specific models one can analyze in which situations local balance fails. Quite often then natural ways to restore local balance become obvious. The paper Van Dijk [9] gives a clear review of this approach applied to a delay system, an overflow model, systems with breakdowns and finite tandem queues. For further research in this direction by Van Dijk see the references of [9]. In Hordijk and Ridder [5] the overflow model is analyzed, also for non-exponential service times. A general method to construct an upper (lower) model is obtained by taking the supremum (infimum) of the right hand side of (1) over all simple paths or a suitable chosen subclass of paths. This approach is studied in Hordijk and Ridder [6]. In Ridder [8] we find an extensive study on the method. Besides thorough mathematical proofs for the stochastic inequalities also many specific models are studied, specially overflow and communication models. In this overview paper we intend to give an exposition of the bounds for overflow models (section 2) and a computer communication model (section 3).

2. OVERFLOW MODELS

We consider a service system of m queues or facilities say \( Q_1, Q_2, \ldots, Q_m \). The number of servers in facility \( Q_i \) is \( s_i \). In gradings \( s_i = 1 \) for all \( i \). There are \( k \) different types of jobs. The arrival process of type \( j \) customers is Poisson with parameter \( \lambda_j \). An arriving job of type \( j \) will first search for a free server at \( Q_j \). If there is no free server at \( Q_j \) it tries to find a free server at queue \( Q_i \). In this order the job will search a list of queues depending on its type number \( j \), say \( L_j = (i_1, i_2, \ldots, i_{n_j}) \). If all servers of these queues are busy the job or call is lost and will leave the system without service. The service time distribution of a type \( j \) job at \( Q_i \) has a distribution \( F_{ij} \) with mean \( \mu_{ij} \). If \( F_{ij} \) is not varying with \( i \) or \( j \) then the overflow model is called job respectively server dependent. For telephone systems with several trunk groups and rerouting of blocked calls this overflow model is of interest.

The stationary distribution of the overflow is not of product form and therefore cannot be (easily) computed. In [5] the server dependent overflow model with \( m = k = 2 \) and \( L_1 = (1,2) \) and \( L_2 = (2) \) is studied. In this model there are two queues \( Q_1 \) and \( Q_2 \) with \( s_1 \) respectively \( s_2 \) servers and arriving demands at \( Q_1 \) who find all servers of \( Q_1 \) busy are offered to \( Q_2 \). At \( Q_1 \) and \( Q_2 \) the arrival processes of type 1 respectively type 2 customers have intensities \( \lambda_1 \) and \( \lambda_2 \). The service time distribution at \( Q_1 \) is \( F_1 \) with mean \( \mu_1 \). Let us explain why for this model the job local balance property does not hold. If in a certain state a job of type 1 is being served at \( Q_2 \) while at \( Q_1 \) there are free servers then
the probability flow out of that state due to a transition of our job is \( \nu_2 \). The probability flow into the state due to our job is zero because there are free servers at \( Q_1 \), hence there is no job local balance in this state. There are several modifications for which the job local balance property does hold. In [5] we assumed for the lower and upper model that in every state the arrival stream at \( Q_2 \) of type 1 jobs is Poisson with intensities \( \lambda_1 \) respectively \( \lambda_2 \).

It is clear that these modifications satisfy job local balance and hence the stationary distributions have product form and are insensitive.

The loss probabilities \( L_1 \) and \( L_2 \) for type 1 respectively type 2 jobs become \( B(s_1,a_1)B(s_2,a_2) \) and \( B(s_2,a_2) \) with \( B(s,a) \) the loss probability in the Erlang loss system with \( s \) servers and offered load \( a \) (see Cooper [2]). For the upper and lower model we take \( a_1 = \frac{\lambda_1}{\nu_1} \) and \( a_2 = \frac{(\lambda_1+\lambda_2)}{\nu_2} \) respectively \( a_2 = \frac{\lambda_2}{\nu_2} \).

There exist tables of the \( B(s,a) \)-function and also efficient recursive computing schemes. So these bounds can be calculated rather easily for large values of \( s_1 \) and \( s_2 \) and therefore may be useful in practice. The bounds do not depend on the service time distribution so they are insensitive. The true loss probabilities are not insensitive, they vary with the service time distribution. Therefore we can not expect that the insensitive bounds are very tight. They are useful because they provide an interval which contains the true loss probabilities for all service time distributions.

<table>
<thead>
<tr>
<th>( \lambda_2 = 0 )</th>
<th>( \lambda_2 = 5 )</th>
<th>( \lambda_2 = 10 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_1 )</td>
<td>( L_2 )</td>
<td>( L_1 )</td>
</tr>
<tr>
<td>0.00609</td>
<td>0.01221</td>
<td>0.05605</td>
</tr>
<tr>
<td>0.00382</td>
<td>0.00714</td>
<td>0.05357</td>
</tr>
<tr>
<td>0.01029</td>
<td>0.01677</td>
<td>0.05945</td>
</tr>
<tr>
<td>0.00667</td>
<td>0.01049</td>
<td>0.05667</td>
</tr>
<tr>
<td>( L(p) )</td>
<td>0.00033</td>
<td>0.00185</td>
</tr>
</tbody>
</table>

In table 1 we have listed the loss probabilities for \( s_1 = s_2 = 5 \), \( \lambda_1 = 7.5 \) and various values of \( \lambda_2 \). In the rows 1 to 4 the true loss probabilities \( L_1 \) and \( L_2 \) for type 1 respectively type 2 jobs are given for different service time distributions. The rows 5 and 6 give \( \bar{L}_1 \) and \( \bar{L}_2 \) the lower respectively the upper bound. Notice that a good approximation is obtained by taking the average value of the lower and upper bound. It turns out that the obvious approximation \( L(p) \) given in row 7, by taking \( p\lambda_1 + \lambda_2 \) (with \( p = B(s_1,a_1) \) the loss probability of \( Q_1 \)) as intensity for the type 2 arrival process, gives a poor approximation. See [5] for further details.

Although the upper and lower bounds are intuitively obvious, it is by no means
easy to show their validity for all service time distribution. In [5] a rigorous proof is given for all distributions with decreasing failure rate. In [8] and [10] the proof for the pure overflow model ($\lambda_2 = 0$) is given. However, a mathematical proof for the case $\lambda_2 > 0$ is still lacking.

In [9] and [10] other modifications to restore job local balance are given. For the upper bound it is assumed that whenever a job completes its service at $Q_2$ in a state for which one of the servers of $Q_1$ is free then it starts a complete new service. For the lower bound various modifications are used.

In [6] a general method to obtain product form modifications is given. The method can be applied to any jobmark process. We need a partial ordering on the state space. For the general overflow model with queues $Q_1, \ldots, Q_m$ which is server dependent the state is the $m$-tuple of the number of jobs at the various queues. The vector ordering is then the appropriate partial ordering. For exponential service times the process is assumed to be monotone.

A reversible Markov chain can be constructed whose stationary distribution is stochastically larger. The stationary distribution of this chain is obtained by taking in formula (1) for $E_0$ the state with no jobs and for arbitrary state $E$ the maximum of the right hand side over all shortest paths from $E_0$ to $E$. Similarly the minimum is taken for a stationary distribution which is stochastically smaller.

This upper and lower model in stochastic order then provide bounds for the loss probabilities, say $L^*_1$ and $L^*_2$. The following inequalities can be shown (see [6]),

$$L_1 \leq \bar{L}_1 \leq L^*_1 \quad \text{and} \quad L_2 \leq L^*_2 \leq \bar{L}_2.$$ 

From numerical results can be concluded that it is worthwhile to compute different modifications and to use the tightest bounds. In very simple models the various bounds of Hordijk and Ridder and Van Dijk are the same. Mostly they differ and their quality vary with the system parameters.

The bounds obtained through reversible Markov chains are again insensitive. However the mathematical proof that they are valid for all service time distributions is far from being complete. In [8] they are shown to be valid for hyperexponential distributions. In the monograph [8] analogous results for the general server dependent overflow are obtained. Also job dependent models are studied there.

COMPUTER COMMUNICATION MODEL

In [4] the performance of a computer system called Doom (Decentralized Object Oriented Machine) is analyzed. The machine is developed for parallel computing. The architecture consists of a large number of identical computers, connected
to each other through a network of links, according to some graph topology. In the execution of a program subprocesses are running on the separate computers. They communicate through messages, which are sent through the links. Due to physical constraints it is not possible to construct a fully connected network. It is therefore important to compute the performance of the Doom machine under various graph topologies such as ring, chordal ring, binary cube.

For the precise description of the queueing network associated with the packet switching of messages in the Doom machine we refer to [4]. It turns out that in the communication processors there are multiserver queueing models with smallest workload discipline. A server stands for a link and has his own queue. An arriving job is assigned to that server whose total workload in the queue and service is the smallest. The stationary distribution of the number of jobs at the various servers does not have a product form.

The following product form modifications have been used in [4]. In the first model we replace the shortest workload discipline by the discipline in which jobs select their server in an aselect way. It can be shown that the average number of service completions during any time period will be smaller in this modified model. Hence the approximation gives a larger average time delay. It gives an upper bound for this performance measure. In the second modification all the servers are available for all jobs. This gives a smaller average time delay and provides a lower bound. Simulation results show that in most cases the lower bound gives a better approximation. For numerical results we refer to [4].

REFERENCES


