INSTITUTE OF
ACTUARIAL SCIENCE
&
ECONOMETRICS

REPORT AE 1/2004

Call Packing Bounds for Overflow Queues

N.M. van Dijk
E. van der Sluis

University of Amsterdam

CALL PACKING BOUNDS FOR OVERFLOW QUEUES

NICO M. VAN DIJK AND ERIK VAN DER SLUIS

University of Amsterdam

Abstract

Finite queueing loss systems are studied with overflow. For these systems there is no simple analytic expression for the loss probability or throughput. This paper aims to prove and promote easily computable bounds as based upon the so-called call packing principle. Under call packing a standard product form expression is available.

It is proven that call packing leads to a guaranteed upper bound for the loss probability. In addition, an analytic error bound for the accuracy is derived which also leads to a secure lower bound. The call packing bound is also proven to be superior to the standard Erlang bound.

Numerical results seem to indicate that the call packing bound is a substantial improvement over the Erlang bound and a quite reasonable first order and secure upper bound approximation. The results thus seem to support a practical usefulness as well as further extension.

1. Introduction

Overflow or threshold routing mechanisms are most natural in a variety of practical queueing systems, most notably in manufacturing, telecommunications, computer networking and call centers.

A classical and simple telecommunication example is that of a simple switch which allows calls to be rerouted to a second server (link) group when all lines of the primary server (link) group are occupied (see Figure 1).

AMS 2000 subject classification. Primary 60K25; secondary 60J27, 60J99, 90B22.
Keywords and phrases. Call centers, queueing, call packing, overflow systems.

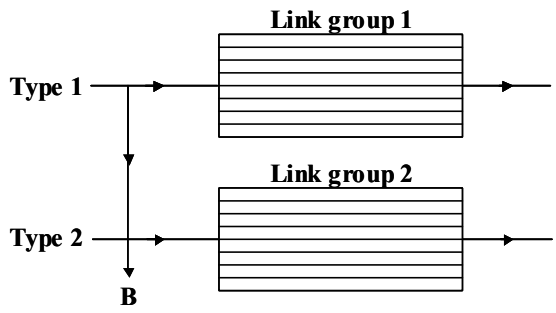

Figure 1: Rerouting of calls between two server groups.

In present day technology this mechanism is still most actual such as in *circuit switched telecommunication networks* (see Figure 2).

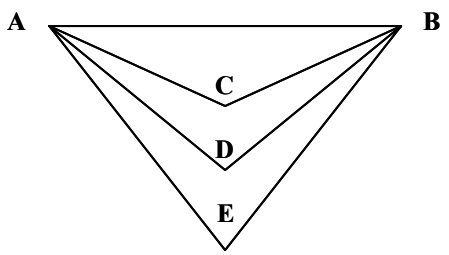

Figure 2: Circuit switched telecommunication network.

Here a long-distance communication, say between A and B, is bypassed via C when there are no direct links available. Or otherwise, via D, or otherwise … and so on. In this way the actual loss probability for not being able to find a free connection directly, is eventually kept to a minimum, say less than 1%.

Another application area for overflow mechanisms which is of significant practical interest is that of *call centers*. Here separately located call center locations, often for reasons of

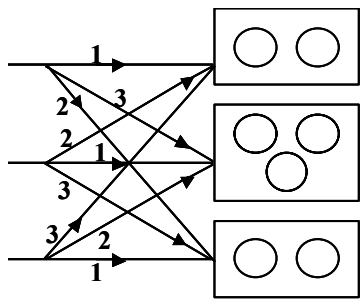

Figure 3: Call Center Skill based routing.

employment and activities, are pooled to form one virtual call center by overflow mechanisms. Similarly, within one call center one may have different server (agent) groups with specified primary skills but between which overflow may take place for a second or third skill allocation in situations of excess demand (*skill based routing*).

Motivation

Unfortunately, even for the most simple overflow example as in Figure 1, there is no simple analytic expression such as for the joint steady state distribution of busy servers and related performance measures. For example, the overflow stream from group 1 is known to be hyper-exponential (cf. [27]). And when both groups are finite, no closed form expression for the throughput of successfully completed type 1 or type 2 calls is known.

As a first approximation, the first group can be regarded as a standard Erlang loss queue. For the loss probability of type 1 calls this directly leads to a rough upper bound. (This will be referred to as the Erlang bound in section 4.2.)

As a second more accurate approximation for this loss probability, the second group can also be regarded as a standard Erlang loss system with the loss rate of the first group added. (This will be referred to as the Erlang loss approximation in section 5.) However, there is no guarantee for its accuracy and it can be expected to provide a lower bound while an upper bound for the loss probability seems of more practical interest.

Third, a standard more practical approach is to use simulation. Indeed, at present dedicated call center simulation packages are widely available (e.g. [28]).

In this paper, another approach is followed by providing a simple *bound* as based upon a so-called *call packing* (CP) principle. Under call packing a standard type product form solution is available. This principle has long been known in classic teletraffic theory (e.g. [3], [5]). However, so far it has been disregarded as it was and is still generally perceived to be unrealistic. (In fact, it has become realistic in present day telecommunication structures.)

In what follows, in contrast, the call packing principle will be suggested in order to guarantee rough but simple and secure performance bounds such as for quick but guaranteed performance evaluation purposes. The expressions used are not new and have most likely

been used in practical teletraffic engineering situations for approximative purposes (e.g. [2], [3], [5], [8], [11], [12] and references therein). However, as of yet, no formal justification seems to be available.

Results

More precisely, as a first step towards more complex practical structures, the simple but generic case of the two-server group as in Figure 1 will be studied. By applying an error bound theorem, adopted from earlier research (by one of the authors) (e.g. [18]) it will be shown how the computation of the loss probability (or throughput) under the call packing principle leads to

i) easily computable and guaranteed upper (and lower) bounds and approximations,
ii) analytic error bounds for the accuracy.

The technicalities of this error bound theorem application are of interest in themselves as it also concludes stochastic comparison results. Moreover, it seems promising for extension to other measures and more complex overflow structures. Numerical results are included which seem to support a practical usefulness of the CP-bounds as:

- a secure upper bound and improvement over the Erlang bound and approximation
- a quite reasonable first order approximation
- to provide a secure lower bound.

Outline

In section 2, we first present the model of interest and argue why it is not solvable. Next, the call packing principle is explained as a natural modification under which the system does become solvable. In section 3, a general error bound result is presented. The actual results to be considered as new are presented in section 4 along with numerical support in section 5. First, in section 4.1 the error bound result is applied in order to compare the original system of interest with the call packing computation as well as to establish an error bound for its accuracy. In essence, the results follow from Lemma 4.1. The lengthy technical details of its proof are of interest in itself and for extensions. Next, in section 4.2 and 4.3 it is also used to show that the CP-bound improves the standard Erlang bound respectively to prove another bound that can also be used (as presented before in the literature). Finally, in section 5 the

quality of the CP-bound, a comparison with other bounds and approximations and the practical value of the CP-bound is investigated and argued by numerical illustration.

2. Model and call packing bounds

Consider the generic model as depicted in Figure 1 with two server groups of N_1 and N_2 servers and two types of incoming calls at Poisson rates λ_1 and λ_2 at each of the two server groups respectively. The call durations are exponential with parameters μ_1 and μ_2 for calls of type 1 and type 2 respectively.

When all N_1 servers are busy a type 1 call is overflowed to the second server group. When the second server group is fully occupied overflowed type 1 calls or incoming type 2 calls are directly rejected (and lost). Once a type 1 call is overflowed to the second group it completes its call at this group.

Clearly, the first server group can be analyzed as a standard M / M / N multi-server queue. However, in order to evaluate the throughput of type 1 calls, we necessarily need the joint steady state distribution of type 1 and type 2 calls at the second server group.

No closed form

Unfortunately, this distribution cannot be provided in closed analytic form unless a so-called notion of station balance per type of call can be verified (cf. [22], [23]). This notion however is violated as outlined below. To this end, denote by $\boldsymbol{n} = (m_1, o_1, n_2)$ the state with

m_1: type 1 calls (non-overflowed) at server group 1;
o_1: type 1 calls (overflowed) at server group 2;
n_2: type 2 calls at server group 2.

For example, assume that $N_1 = 10$ and consider the state $(m_1, o_1, n_2) = (8, 2, 0)$ with 8 (type 1) calls at group 1 and 2 calls (also of type 1) at group 2. (Note that this state can arise). Then station balance is necessarily violated at server group 2 as

the rate out of this state due to a departure at group 2, which would lead to state (8, 1, 0), is positive,

while

the rate into this state due to an arrival at group 2, which would then have occurred in state (8, 1, 0), is 0.

Instead, an arrival of an incoming type 1 call in state (8, 1, 0) would have led to state (9, 1, 0) rather than to state (8, 2, 0).

Product form modification

Intuitively, this rate inconsistency can be repaired or avoided in two ways:

i) Whenever $m_1 < N_1$: also stop (interrupt) the service of type 1 calls at the second group
ii) By letting type 1 calls be switched back to the first group once a group 1 server becomes available ($m_1 < N_1$).

Both modifications are easily shown to exhibit a closed product form solution. A proof of this product form statement for the first modification will be given later on in section 4.3, while for the second it is argued below by equations (2.2). Alternatively, in [6] and [7] also a different (third) product form modification has been provided by simply regarding the primary and overflow group as completely decoupled with arrival rate $\lambda_1+\lambda_2$ at the overflow group. (in fact, in these references the service times are server rather than type dependent.) As will be argued in section 4.3, for the problem of interest this third modification will lead to the same results as by the first modification.

The first (and thus also equivalent third) modification has already been suggested and studied before in the literature (cf. [6], [7], [20]) in order to argue an upper bound, as based upon this product form, for the loss probability of type 1 calls. However, the second modification seems more natural or at least can in general be expected to be more accurate as an approximation, as will also be supported by numerical results in section 5. We will therefore focus our primary attention on the second modification, while also paying some more attention to the first and third in section 4.3 and 5.

In fact, the mechanism of the second modification has long been known in teletraffic literature under the name of *call packing* (e.g. [3], [5]). In what follows, this system will therefore be referred to as the *call packing* (CP) system.

Call Packing Bounds

For the call packing system and with c a normalizing constant, the following product form steady-state distribution applies, as shown below by (2.2).

$$\bar{\pi}(m_1, o_1, n_2) = c \frac{1}{n_1!}\left(\frac{\lambda_1}{\mu_1}\right)^{n_1} \frac{1}{n_2!}\left(\frac{\lambda_2}{\mu_2}\right)^{n_2} \quad \text{with } n_1 = m_1 + o_1 \qquad \text{for all } \boldsymbol{n} \in \bar{S} \text{ with} \tag{2.1}$$

$$\bar{S} = \left\{ \boldsymbol{n} = (m_1, o_1, n_2) \;\middle|\; \begin{array}{ll} m_1 \le N_1 & \text{if } o_1 = 0 \\ m_1 = N_1 & \text{if } o_1 > 0, \text{ and} \\ o_1 + n_2 \le N_2. & \end{array} \right\}$$

As based upon the so-called notion of station balance (see [23]) or by writing out the global balance equations, the verification of this product form under the call packing principle is established by checking the following local balance equations for each of the three components separately and for all $\boldsymbol{n} = (m_1, o_1, n_2) \in \bar{S}$:

$$\begin{aligned} \bar{\pi}(m_1, o_1, n_2)\; m_1 \mu_1 1_{\{m_1 > 0\}} &= \bar{\pi}(m_1 - 1, o_1, n_2) \lambda_1 1_{\{m_1 > 0\}} \\ \bar{\pi}(N_1, o_1, n_2)\; (N_1 + o_1)\, \mu_1 1_{\{o_1 > 0\}} &= \bar{\pi}(N_1, o_1 - 1, n_2) \lambda_1 1_{\{o_1 > 0\}} \\ \bar{\pi}(m_1, o_1, n_2)\; n_2 \mu_2 1_{\{n_2 > 0\}} &= \bar{\pi}(m_1, o_1, n_2 - 1) \lambda_2 1_{\{n_2 > 0\}}. \end{aligned} \tag{2.2}$$

Remark 2.1 One may note that the state description $\boldsymbol{n} = (m_1, o_1, n_2)$, which is necessarily required for the original system, can be reduced to $\boldsymbol{n} = (n_1, n_2)$ for the CP-system. However, as we will have to compare the original and CP-system (see the proof of Theorem 4.1) a common state notation is required.

The throughputs $\bar{\boldsymbol{T}}_1$ of completed type 1 calls and $\bar{\boldsymbol{T}}_2$ of completed type 2 calls are now easily calculated by

$$\begin{aligned} \bar{\boldsymbol{T}}_1 &= \sum\nolimits_{(m_1, o_1, n_2)} \bar{\pi}(m_1, o_1, n_2) \mu_1 (m_1 + o_1) \\ \bar{\boldsymbol{T}}_2 &= \sum\nolimits_{(m_1, o_1, n_2)} \bar{\pi}(m_1, o_1, n_2) \mu_2 n_2. \end{aligned} \tag{2.3}$$

Bounds

Clearly, the accuracy of $\bar{\boldsymbol{T}}_1$ and $\bar{\boldsymbol{T}}_2$ as approximations for the throughputs $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ of the original system of interest may depend on the size of N_1 and N_2 and the traffic ratios. But at

least, one may intuitively expect that call packing provides a guaranteed (lower bound for the) throughput and guaranteed (upper bound for the) loss probability as it possesses its own servers as much as possible (first) before using servers for other call types. (In this respect it could be regarded as 'fair'.) Such bounds can be of practical interest in order to guarantee a sufficiently large throughput or a sufficient small loss percentage of type 1 calls (such as for setting norms or for dimensioning).

It would thus be of interest to show that

$$\begin{aligned} \bar{\boldsymbol{T}}_1 &\le \boldsymbol{T}_1 \qquad & (\boldsymbol{L}_1 \le \bar{\boldsymbol{L}}_1) \\ \bar{\boldsymbol{T}}_2 &\ge \boldsymbol{T}_2 \qquad & (\boldsymbol{L}_2 \ge \bar{\boldsymbol{L}}_2). \end{aligned}$$

where the value $\boldsymbol{L}$ represents the loss probability as resulting from

$$\boldsymbol{T}_i = \lambda_i(1-\boldsymbol{L}_i) \qquad (i=1,2). \tag{2.4}$$

But as of yet, despite this intuition, no formal proof seems to be available. In fact, one can provide counterintuitive examples (such as in section 4.2), which can show the opposite on a sample path basis. A formal proof will therefore be of interest. More importantly, it will be included as a side result in Theorem 4.1 in combination with an analytic error bound for $\left|\bar{\boldsymbol{T}}_1 - \boldsymbol{T}_1\right|$ and thus also $\left|\bar{\boldsymbol{L}}_1 - \boldsymbol{L}_1\right|$.

In order to do so, in the next section we first provide a general framework by which we can simultaneously provide a formal proof for the bounds and establish an analytic error bound.

3. Error bound result

This section will present an error bound theorem as adopted from [25].

Consider a continuous-time Markov chain (CTMC) with countable state space S and with transition rate matrix $\boldsymbol{Q} = \boldsymbol{q}(i,j)$ with $\boldsymbol{q}(i,j)$ the transition rate for a transition from state i to j. For convenience, we assume this chain uniformizable, that is, for some finite constant B:

$$\sum_{j \ne i} \boldsymbol{q}(i,j) \le B \qquad \text{for all } i. \tag{3.1}$$

Then, by virtue of this boundedness assumption, the CTMC can also be evaluated as a discrete-time Markov chain (DTMC) with one-step transition matrix, where $h = 1 / B$: (e.g. see [4], [19], [26]):

$$\boldsymbol{P} = \boldsymbol{I} + h\boldsymbol{Q} \qquad \text{or more detailed}$$
$$\boldsymbol{P}(i,j) = \begin{cases} h\ \boldsymbol{q}(i,j) & j \neq i \\ 1 - h\sum_{k\neq i}\boldsymbol{q}(i,k) & j = i \end{cases} \tag{3.2}$$

Most notably, we can compute average performance measures $\boldsymbol{G}$ of the CTMC by means of this DTMC as outlined below. To this end, consider some *reward rate* function $r(i)$ that incurs a reward $r(i)$ per unit of time as long as the system is in state i. Then the cumulative reward over time t when starting in state i at time 0, is given by

$$\boldsymbol{V}_t(i) = \int_0^t \left\{ \sum_j \boldsymbol{P}_s(i,j)\ r(j) \right\} ds$$

where $\boldsymbol{P}_s(i,j)$ represents the transition probability over time s when starting in state i at time 0 to be in state j at time s, and where the integral is assumed to be well defined as under standard conditions. Then, under natural ergodicity and irreducibility conditions of the CTMC, also the average expected reward $\boldsymbol{G}$ is well defined and independent of the initial state as defined by:

$$\boldsymbol{G} = \lim_{t\to\infty}\ \boldsymbol{V}_t(i)\ /\ t \qquad (i \in S) \tag{3.3}$$

Then, by virtue of the uniformization, we can also evaluate $\boldsymbol{G}$ by:

$$\boldsymbol{G} = \lim_{k\to\infty}\ h^{-1}\boldsymbol{V}_k(i)/k \qquad (i \in S), \text{ where } \boldsymbol{V}_0(\ ) \equiv 0 \text{ and} \tag{3.4}$$

$$\boldsymbol{V}_{k+1}(i) = h\,r(i) + \sum_j \boldsymbol{P}(i,j)\boldsymbol{V}_k(j) \qquad (k = 0,1,2,...)\ (i \in S) \tag{3.5}$$

where $\boldsymbol{V}_k$ can be regarded as the expected cumulative reward function over k steps, each of time length h, while a reward $h\ r(i)$ is incurred per step when the system is in state i. Here it is noted that the time factor h could be deleted in $\boldsymbol{G}$ and $\boldsymbol{V}_k$, but it is included not only for its natural interpretation of uniformization, but also for convenience in proof steps that will follow. (In fact, those of Lemma 4.1.)

The advantage of the latter discrete-time version over the continuous-time formulation is the recursive structure of the reward function $\boldsymbol{V}_k$, which will enable us to inductively proof bounds in Lemma 4.1 later on. This in turn will allow us to establish error bounds when computing average reward functions $\boldsymbol{G}$ and $\bar{\boldsymbol{G}}$ when comparing two CTMC's,

$$\begin{cases} \boldsymbol{G} \text{ corresponding to } (S, \boldsymbol{Q}, r) \\ \bar{\boldsymbol{G}} \text{ corresponding to } (\bar{S}, \bar{\boldsymbol{Q}}, \bar{r}) \end{cases} \qquad \text{where } \bar{S} \subseteq S$$

To this end, the following result is adopted from [25]. Herein, $\{\bar{\pi}(i) \mid i \in \bar{S}\}$ represents the steady state distribution of $(\bar{S}, \bar{\boldsymbol{Q}}, \bar{r})$.

Result 3.1 Suppose that for some function γ(.) at $\bar{S}$ and all $i \in \bar{S}$:

$$0 \le \bar{r}(i) - r(i) + \sum_j \left[\bar{\boldsymbol{q}}(i,j) - \boldsymbol{q}(i,j)\right]\left[\boldsymbol{V}_k(j) - \boldsymbol{V}_k(i)\right] \le \gamma(i) \tag{3.6}$$

Then

$$0 \le \bar{\boldsymbol{G}} - \boldsymbol{G} \le \sum_i \bar{\pi}(i)\gamma(i) \tag{3.7}$$

Proof. Immediate by combining Theorem 2.1 and 2.2 from [25].

Remark 3.1 Note that the 0-lower bound in (3.6) implies a comparison result. For such results also the stochastic comparison approach (e.g. see [9], [13], [15], [16], [17]) is known. The Markov reward approach as reflected by Result 3.1, however, differs from the stochastic comparison approach in two ways:

i) It also leads to error bounds (as its major objective)

ii) It may establish comparison results for which a stochastic comparison approach by sample path comparison may fail. (e.g. see [10], [18]).

As such, the Markov reward approach can be seen as an extension of the stochastic comparison approach. As a major drawback, however, it is more complex and it requires Markovian or exponential structures.

The stochastic comparison approach in contrast is generally easier to apply and is **not** restricted to Markovian or exponential type structures. This can be a substantial advantage

when dealing for instance with non-exponential queueing systems. (For a possible extension of the Markov reward approach to the non-exponential case see Remark 4.2.) However, with the exception of special situations (see e.g. [10]), the stochastic comparison (or related sample path) approach, does **not** lead to analytic bounds when comparing two systems.

For a somewhat more extensive discussion of these two approaches and their differences see [25].

Remark 3.2 In section 4, we will apply Result 3.1 in order to establish analytic error bounds for the inaccuracy introduced by analyzing the overflow model under the call packing modification.

Generally, the essential step for this application is to establish bounds for the difference terms $\boldsymbol{V}_k(j)-\boldsymbol{V}_k(i)$. This step can be quite complicated and depends on the system of interest. For the present application, it is established by Lemma 4.1.

Remark 3.3 Note that the actual value of B does not appear in (3.6) and (3.7). The value B can therefore be chosen arbitrarily large satisfying (3.1).

4. Results

By applying the general error bound result from section 3, in this section we will show that expression (2.1) as based upon the call packing assumption leads to guaranteed bounds for the throughput or loss probability. Moreover, an intuitively appealing analytic error bound will be established.

Numerical support is presented in section 5 to give an indication of the accuracy of the error bounds for practical purposes.

4.1 CP-bound: Proof and error bound

We will focus at the throughput (or loss probability) of type 1 calls, as performance measure of primary interest. To this end, in the setting of section 3, we identify a state i with a state

$$\boldsymbol{n} = (m_1, o_1, n_2)$$

and we use the reward rate

$$r(\boldsymbol{n}) = (m_1 + o_1)\,\mu_1. \tag{4.1}$$

Furthermore, (with Remark 3.3 kept in mind) note that the boundedness assumption (3.1) is verified by

$$B = \lambda_1 + \lambda_2 + (N_1 + N_2)\mu_1 + N_2\mu_2 \qquad \text{and let } h = B^{-1}$$

We adopt all notation from section 3 and use upper bars for the system under call packing. The essential step to apply Result 3.1 is the technical Lemma 4.1 that will be given below. The technical details of its proof which will follow after Remark 4.1 are rather complex and lengthy and are of interest in itself as well for extension. This lemma leads to the following result.

Theorem 4.1 We have

$$\overline{\boldsymbol{T}}_1 \le \boldsymbol{T}_1 \text{ and} \tag{4.2}$$

$$\left|\overline{\boldsymbol{T}}_1 - \boldsymbol{T}_1\right| \le \overline{\pi}(n_1 > N_1)\, N_1\mu_1 \left(\frac{\lambda_1 + \mu_2}{\mu_2}\right) \tag{4.3}$$

Proof. With $\overline{q}$ and q the transition rates of the overflow system with and without call packing respectively, and for any $\boldsymbol{n} \in \overline{S} = \{(m_1, o_1, n_2) \mid m_1 \le N_1,\ m_1 = N_1 \text{ if } o_1 > 0,\ o_1+n_2 \le N_2\}$, we have

$$\begin{aligned}
&\sum\nolimits_{\boldsymbol{n}'} \left[\overline{q}(\boldsymbol{n}, \boldsymbol{n}') - q(\boldsymbol{n}, \boldsymbol{n}')\right]\left[\boldsymbol{V}_k(\boldsymbol{n}') - \boldsymbol{V}_k(\boldsymbol{n})\right] \\
&\quad = 1_{\{o_1>0,\ m_1=N_1\}}\, N_1\mu_1 \left[\boldsymbol{V}_k(N_1, o_1 - 1, n_2) - \boldsymbol{V}_k(N_1 - 1, o_1, n_2)\right] \\
&\quad = 1_{\{o_1>0,\ m_1=N_1\}}\, N_1\mu_1\, \Delta_k^{1,2}(N_1 - 1, o_1 - 1, n_2).
\end{aligned} \tag{4.4}$$

Lemma 4.1 below and Result 3.1 complete the proof of (4.2) and (4.3) for the throughputs $\overline{\boldsymbol{T}}_1$ and $\boldsymbol{T}_1$. □

Lemma 4.1 With $C_1 = (1 + \lambda_1 / \mu_2)$ and $C_2 = \lambda_1 / \mu_2$

$$0 \le \Delta_t^1(\boldsymbol{n}) \quad = \boldsymbol{V}_t(m_1+1, o_1, n_2) - \boldsymbol{V}_t(\boldsymbol{n}) \le C_1 \tag{4.5}$$

$$0 \le \Delta_t^2(\boldsymbol{n}) \quad = \boldsymbol{V}_t(m_1, o_1+1, n_2) - \boldsymbol{V}_t(\boldsymbol{n}) \le C_1 \tag{4.6}$$

$$0 \geq \Delta_t^3(\boldsymbol{n}) \quad = \boldsymbol{V}_t(m_1, o_1, n_2+1) - \boldsymbol{V}_t(\boldsymbol{n}) \geq -C_2 \tag{4.7}$$

$$0 \leq \Delta_t^{2,3}(\boldsymbol{n}) = \boldsymbol{V}_t(m_1, o_1+1, n_2) - \boldsymbol{V}_t(m_1, o_1, n_2+1) \leq C_1 \tag{4.8}$$

$$0 \geq \Delta_t^{1,2}(\boldsymbol{n}) = \boldsymbol{V}_t(m_1+1, o_1, n_2) - \boldsymbol{V}_t(m_1, o_1+1, n_2) \geq -C_1 \tag{4.9}$$

Remarks 4.1

1. Before presenting the proof of Lemma 4.1 some remarks are in place with respect to the various inequalities, or rather the lower and upper bounds in (4.5)-(4.9) and the order in which these inequalities and bounds will be proven.

   The proof of Theorem 4.1 only uses inequality (4.9). However, writing out the exact recurrent expression for $\Delta_t^{1,2}(\boldsymbol{n})$ (see (4.20)) will also lead to an expression of the form $\Delta_t^{2,3}(\boldsymbol{n})$ which in turn leads to the expressions $\Delta_t^3(\boldsymbol{n})$, $\Delta_t^2(\boldsymbol{n})$ and $\Delta_t^1(\boldsymbol{n})$. In order to also establish estimates 0, C_1 and C_2 in the right directions, it therefore turns out to be more natural and convenient to eventually prove (4.9) in the order of proving (4.5)-(4.9).

2. It is also noted beforehand that

$$\begin{aligned} \Delta_t^{1,2}(\boldsymbol{n}) &= \Delta_t^1(\boldsymbol{n}) - \Delta_t^2(\boldsymbol{n}) \\ \Delta_t^{2,3}(\boldsymbol{n}) &= \Delta_t^2(\boldsymbol{n}) - \Delta_t^3(\boldsymbol{n}) \end{aligned} \tag{4.10}$$

   As a consequence, (4.8) and (4.9) do not directly follow from (4.5)-(4.7) due to the opposite signs in (4.10).

Proof of Lemma 4.1. Recall that Lemma 4.1 purely concerns the original overflow model. The proof will follow by induction in k. Clearly, (4.5)-(4.9) hold for $t = 0$. Let (4.5)-(4.9) hold for $t \leq k$. Then we need to verify (4.5)-(4.9) also for $t = k+1$. We will do so subsequently for each inequality separately. Herein, to obtain the throughput of type 1 calls we recall the reward rate (4.1).

Equation (4.5) for $t = k+1$.

First note that we only need to consider states (m_1, o_1, n_2) with $m_1 < N_1$. By writing out all transitions in state (m_1, o_1, n_2) as according to (3.2) and (3.5) we find:

$$
\begin{aligned}
\boldsymbol{V}_{k+1}(m_1,o_1,n_2) = {} & h\,(m_1+o_1)\mu_1 \\
& + h\,m_1\mu_1 1_{\{m_1>0\}}\boldsymbol{V}_k(m_1-1,o_1,n_2) \\
& + h\;o_1\mu_1 1_{\{o_1>0\}}\boldsymbol{V}_k(m_1,o_1-1,n_2) \\
& + h\;n_2\mu_2 1_{\{n_2>0\}}\boldsymbol{V}_k(m_1,o_1,n_2-1) \\
& + h\,\lambda_1\boldsymbol{V}_k(m_1+1,o_1,n_2) \\
& + h\,\lambda_2 1_{\{o_1+n_2<N_2\}}\boldsymbol{V}_k(m_1,o_1,n_2+1) \\
& + h\,\lambda_2 1_{\{o_1+n_2=N_2\}}\boldsymbol{V}_k(m_1,o_1,n_2) \\
& + \left[1-h\,m_1\mu_1-h\,o_1\mu_1-h\,n_2\mu_2-h\lambda_1-h\lambda_2\right]\boldsymbol{V}_k(m_1,o_1,n_2)
\end{aligned}
\tag{4.11}
$$

and similarly for state (m_1+1, o_1, n_2)

$$
\begin{aligned}
\boldsymbol{V}_{k+1}(m_1+1,o_1,n_2) = {} & h\,(m_1+1+o_1)\mu_1 \\
& + h\,m_1\mu_1\boldsymbol{V}_k(m_1,o_1,n_2) + h\,\mu_1\boldsymbol{V}_k(m_1,o_1,n_2) \\
& + h\;o_1\mu_1 1_{\{o_1>0\}}\boldsymbol{V}_k(m_1+1,o_1-1,n_2) \\
& + h\;n_2\mu_2 1_{\{n_2>0\}}\boldsymbol{V}_k(m_1+1,o_1,n_2-1) \\
& + h\,\lambda_1 1_{\{m_1+1<N_1\}}\boldsymbol{V}_k(m_1+2,o_1,n_2) \\
& + h\,\lambda_1 1_{\{m_1+1=N_1;\,o_1+n_2<N_2\}}\boldsymbol{V}_k(N_1,o_1+1,n_2) \\
& + h\,\lambda_1 1_{\{m_1+1=N_1;\,o_1+n_2=N_2\}}\boldsymbol{V}_k(N_1,o_1,n_2) \\
& + h\,\lambda_2 1_{\{o_1+n_2<N_2\}}\boldsymbol{V}_k(m_1+1,o_1,n_2+1) \\
& + h\,\lambda_2 1_{\{o_1+n_2=N_2\}}\boldsymbol{V}_k(m_1+1,o_1,n_2) \\
& + \left[1-h\,(m_1+1)\mu_1-h\,o_1\mu_1-h\,n_2\mu_2-h\lambda_1-h\lambda_2\right]\boldsymbol{V}_k(m_1+1,o_1,n_2)
\end{aligned}
\tag{4.12}
$$

Now in order to subtract (4.11) from (4.12), it would be convenient to collect terms pair-wise with the same transition probabilities. To this end, we make the following steps. In the right hand side of (4.11) we add the term (which is equal to 0)

$$
h\,\mu_1\boldsymbol{V}_k(m_1,o_1,n_2) - h\,\mu_1\boldsymbol{V}_k(m_1,o_1,n_2)
$$

and we rewrite the term with coefficient $h\,\lambda_1$ as:

$$
\begin{aligned}
h\,\lambda_1\boldsymbol{V}_k(m_1+1,o_1,n_2) = {} & h\,\lambda_1 1_{\{m_1+1<N_1\}}\boldsymbol{V}_k(m_1+1,o_1,n_2) \\
& + h\,\lambda_1 1_{\{m_1+1=N_1;o_1+n_2<N_2\}}\boldsymbol{V}_k(N_1,o_1,n_2) \\
& + h\,\lambda_1 1_{\{m_1+1=N_1;o_1+n_2=N_2\}}\boldsymbol{V}_k(N_1,o_1,n_2).
\end{aligned}
$$

Then by subtracting (4.11) from (4.12), pair-wise by coefficients and where, for the sake of clarity, we keep terms in that are equal to 0, we find

$$\begin{aligned}
\Delta^1_{k+1}(m_1,o_1,n_2) &= \boldsymbol{V}_{k+1}(m_1+1,o_1,n_2)-\boldsymbol{V}_{k+1}(m_1,o_1,n_2)\\
&= h\,\mu_1\\
&+h\,m_1\mu_1 1_{\{m_1>0\}}\Delta^1_k(m_1-1,o_1,n_2)\\
&+h\;o_1\mu_1 1_{\{o_1>0\}}\Delta^1_k(m_1,o_1-1,n_2)\\
&+h\;n_2\mu_2 1_{\{n_2>0\}}\Delta^1_k(m_1,o_1,n_2-1)\\
&+h\,\mu_1\left[\boldsymbol{V}_k(m_1,o_1,n_2)-\boldsymbol{V}_k(m_1,o_1,n_2)\right]\\
&+h\,\lambda_1 1_{\{m_1+1<N_1\}}\Delta^1_k(m_1+1,o_1,n_2)\\
&+h\,\lambda_1 1_{\{m_1+1=N_1;\,o_1+n_2<N_2\}}\Delta^2_k(N_1,o_1,n_2)\\
&+h\,\lambda_1 1_{\{m_1+1=N_1;\,o_1+n_2=N_2\}}\left[\boldsymbol{V}_k(N_1,o_1,n_2)-\boldsymbol{V}_k(N_1,o_1,n_2)\right]\\
&+h\,\lambda_2 1_{\{o_1+n_2<N_2\}}\Delta^1_k(m_1,o_1,n_2+1)\\
&+h\,\lambda_2 1_{\{o_1+n_2=N_2\}}\Delta^1_k(m_1,o_1,n_2)\\
&+\left[1-hm_1\mu_1-h\mu_1-ho_1\mu_1-hn_2\mu_2-h\lambda_1-h\lambda_2\right]\Delta^1_k(m_1,o_1,n_2)
\end{aligned} \tag{4.13}$$

By using the induction hypothesis for $t = k$:

$$\Delta^1_k(m_1,o_1,n_2)\geq 0$$
$$\Delta^2_k(m_1,o_1,n_2)\geq 0$$

and cancelling the 0-terms, we thus directly verify by (4.13): $\Delta^1_{k+1}(m_1,o_1,n_2)\geq 0$. Conversely, substitute the induction hypothesis for $t = k$:

$$\Delta^1_k(m_1,o_1,n_2)\leq C_1$$
$$\Delta^2_k(m_1,o_1,n_2)\leq C_1$$

and note that the term $h\,\mu_1\left[\boldsymbol{V}_k(m_1,o_1,n_2)-\boldsymbol{V}_k(m_1,o_1,n_2)\right]=0$ compensates for the first additional term $h\mu_1$, where we can also use that

$$h\,\mu_1\leq h\,\mu_1\,C_1,$$

in order to use the fact that all coefficients (as transition probabilities) sum up to 1. Then we also conclude from (4.13):

$$\Delta^1_{k+1}(m_1, o_1, n_2) \le C_1.$$

We have thus proven (4.5) for $t = k+1$.

Equation (4.6) for $t = k+1$.

We proceed as above for the induction step for (4.6), but now leave the cumbersome intermediate steps to the reader. First, by writing out all transitions to obtain $\boldsymbol{V}_{k+1}(m_1, o_1, n_2)$ and $\boldsymbol{V}_{k+1}(m_1, o_1+1, n_2)$ and by comparing these transitions pair-wise, where again some of these terms will have to be rewritten, we finally obtain the following expression. Herein, as before, for the clarity of the derivation and the arguments that will follow, some of the terms are equal to 0, but are left in.

$$\begin{aligned}
\Delta^2_{k+1}(m_1, o_1, n_2) = {} & h\mu_1 \\
& + h\, m_1 \mu_1 1_{\{m_1>0\}} \Delta^2_k(m_1-1, o_1, n_2) \\
& + h\, o_1 \mu_1 1_{\{o_1>0\}} \Delta^2_k(m_1, o_1-1, n_2) \\
& + h\, n_2 \mu_2 1_{\{n_2>0\}} \Delta^2_k(m_1, o_1, n_2-1) \\
& + h\mu_1 \left[\boldsymbol{V}_k(m_1, o_1, n_2) - \boldsymbol{V}_k(m_1, o_1, n_2)\right] \\
& + h\, \lambda_1 1_{\{m_1<N_1\}} \Delta^2_k(m_1+1, o_1, n_2) \\
& + h\, \lambda_1 1_{\{m_1=N_1; o_1+n_2+1<N_2\}} \Delta^2_k(N_1, o_1+1, n_2) \\
& + h\, \lambda_1 1_{\{m_1=N_1; o_1+n_2+1=N_2\}} \left[\boldsymbol{V}_k(N_1, o_1+1, n_2) - \boldsymbol{V}_k(N_1, o_1+1, n_2)\right] \\
& + h\, \lambda_2 1_{\{o_1+n_2+1<N_2\}} \Delta^2_k(m_1, o_1, n_2+1) \\
& + h\, \lambda_2 1_{\{o_1+n_2+1=N_2\}} \left[\boldsymbol{V}_k(m_1, o_1+1, n_2) - \boldsymbol{V}_k(m_1, o_1, n_2+1)\right] \\
& + \left[1 - h(m_1+o_1+1)\mu_1 - hn_2\mu_2 - h\lambda_1 - h\lambda_2\right] \Delta^2_k(m_1, o_1, n_2)
\end{aligned} \tag{4.14}$$

Again, note that the term with coefficient $h\mu_1$ is equal to 0 and vanishes, by which we compensate for the first additional term $h\mu_1$, while all coefficients sum up to 1. In addition, also the term with coefficient $h\,\lambda_1 1_{\{m_1=N_1; o_1+n_2+1=N_2\}}$ is equal to 0. Furthermore, the one but last term can be written as:

$$h\lambda_2 1_{\{o_1+n_2+1=N_2\}} \Delta^{2,3}_k(m_1, o_1, n_2)$$

By substituting the lower and upper bound from the induction hypotheses (4.6) and (4.8) for $t = k$, we thus verify (4.6) for $t = k+1$.

Equation (4.7) for $t = k+1$.

Similarly, by writing out all transitions in the states (m_1, o_1, n_2+1) and (m_1, o_1, n_2) and noting that the reward rates in both states are equal, by steps as under (4.5) we finally find:

$$\begin{aligned}
\Delta^3_{k+1}(m_1,o_1,n_2) = {} & h\, m_1\mu_1 1_{\{m_1>0\}}\Delta^3_k(m_1-1,o_1,n_2) \\
& + h\, o_1\mu_1 1_{\{o_1>0\}}\Delta^3_k(m_1,o_1-1,n_2) \\
& + h\, n_2\mu_2 1_{\{n_2>0\}}\Delta^3_k(m_1,o_1,n_2-1) \\
& + h\,\mu_2\left[\boldsymbol{V}_k(m_1,o_1,n_2)-\boldsymbol{V}_k(m_1,o_1,n_2)\right] \\
& + h\,\lambda_1 1_{\{m_1<N_1\}}\Delta^3_k(m_1+1,o_1,n_2) \\
& + h\,\lambda_1 1_{\{m_1=N_1;\, o_1+n_2+1<N_2\}}\Delta^3_k(N_1,o_1+1,n_2) \\
& + h\,\lambda_1 1_{\{m_1=N_1;\, o_1+n_2+1=N_2\}}\left[\boldsymbol{V}_k(N_1,o_1,n_2+1)-\boldsymbol{V}_k(N_1,o_1+1,n_2)\right] \\
& + h\,\lambda_2 1_{\{o_1+n_2+1<N_2\}}\Delta^3_k(m_1,o_1,n_2+1) \\
& + h\,\lambda_2 1_{\{o_1+n_2+1=N_2\}}\left[\boldsymbol{V}_k(m_1,o_1,n_2+1)-\boldsymbol{V}_k(m_1,o_1,n_2+1)\right] \\
& + \left[1-h(m_1+o_1)\mu_1-hn_2\mu_2-h\mu_2-h\lambda_1-h\lambda_2\right]\Delta^3_k(m_1,o_1,n_2)
\end{aligned} \tag{4.15}$$

By canceling the 0-terms as before, substituting the upper bound 0 from (4.7) and the lower bound 0 from (4.8), both for $t = k$, and noting that the one but last term can be rewritten as:

$$-h\lambda_1 1_{\{m_1=N_1;o_1+n_2+1=N_2\}}\Delta^{2,3}_k(m_1,o_1,n_2)$$

we directly verify from (4.15):

$$\Delta^3_{k+1}(\boldsymbol{n}) \le 0.$$

Conversely, by substituting the lower bound from (4.7) for $t = k$, which we denote by $-C_2$, and the upper bound from (4.8) for $t = k$, which we denote by C_1, and noting that in expression (4.15) the terms with coefficient $h\mu_2$ and with coefficient $h\,\lambda_2 1_{\{o_1+n_2+1=N_2\}}$ have cancelled, also the lower bound of (4.7) for $t = k+1$ is verified provided

$$h\lambda_1\left[C_1-C_2\right] \le h\mu_2 C_2$$

as guaranteed by

$$C_1 \le \left[\frac{\lambda_1 + \mu_2}{\lambda_1} \right] C_2 \tag{4.16}$$

Equation (4.8) for $t = k+1$.

By writing out all transitions in the states (m_1, o_1+1, n_2) and (m_1, o_1, n_2+1), we find:

$$\begin{aligned}
\Delta_{k+1}^{2,3}(m_1, o_1, n_2) = h\mu_1 & \\
& + h\, m_1 \mu_1 1_{\{m_1>0\}} \Delta_k^{2,3}(m_1 - 1, o_1, n_2) \\
& + h\, o_1 \mu_1 1_{\{o_1>0\}} \Delta_k^{2,3}(m_1, o_1 - 1, n_2) \\
& + h\, n_2 \mu_2 1_{\{n_2>0\}} \Delta_k^{2,3}(m_1, o_1, n_2 - 1) \\
& + h\, \mu_1 \left[\boldsymbol{V}_k(m_1, o_1, n_2) - \boldsymbol{V}_k(m_1, o_1, n_2 + 1) \right] \\
& + h\, \mu_2 \left[\boldsymbol{V}_k(m_1, o_1 + 1, n_2) - \boldsymbol{V}_k(m_1, o_1, n_2) \right] \\
& + h\, \lambda_1 1_{\{m_1<N_1\}} \Delta_k^{2,3}(m_1 + 1, o_1, n_2) \\
& + h\, \lambda_1 1_{\{m_1=N_1; o_1+n_2+1<N_2\}} \Delta_k^{2,3}(m_1, o_1 + 1, n_2) \\
& + h\, \lambda_1 1_{\{m_1=N_1; o_1+n_2+1=N_2\}} \Delta_k^{2,3}(m_1, o_1, n_2) \\
& + h\, \lambda_2 1_{\{o_1+n_2+1<N_2\}} \Delta_k^{2,3}(m_1, o_1, n_2 + 1) \\
& + h\, \lambda_2 1_{\{o_1+n_2+1=N_2\}} \Delta_k^{2,3}(m_1, o_1, n_2) \\
& + \left[1 - h m_1 \mu_1 - h o_1 \mu_1 - h\mu_1 - h n_2 \mu_2 - h\mu_2 - h\lambda_1 - h\lambda_2 \right] \Delta_k^{2,3}(m_1, o_1, n_2)
\end{aligned} \tag{4.17}$$

Again, by substituting the estimate 0 from the hypotheses (4.8), (4.6) and (4.7) for $t = k$, also the right hand side from (4.17) is estimated from below by 0. By substituting the estimates C_1 from (4.6) and (4.8) and C_2 from (4.7) for the expression:

$$\left[\boldsymbol{V}_k(m_1, o_1, n_2) - \boldsymbol{V}_k(m_1, o_1, n_2 + 1) \right] = -\Delta_k^3(m_1, o_1, n_2)$$

the right hand side is also bounded from above by

$$h\mu_1 + C_1 + h\mu_1 \left[C_2 - C_1 \right] \le C_1$$

provided

$$C_1 \geq C_2 + 1 \tag{4.18}$$

By combining (4.16) and (4.18) and choosing $C_1 = (\lambda_1+\mu_2) / \mu_2$ and $C_2 = \lambda_1 / \mu_2$, the proof of (4.8) is thus completed for $t = k+1$.

Equation (4.9) for $t = k+1$.

Finally, to verify (4.9), as before we write out all transitions in state (m_1+1, o_1, n_2) and (m_1, o_1+1, n_2) to find:

$$\begin{aligned}
\Delta_{k+1}^{1,2}(m_1,o_1,n_2) = {} & h\, m_1\mu_1 1_{\{m_1>0\}}\Delta_k^{1,2}(m_1-1,o_1,n_2) \\
& + h\, o_1\mu_1 1_{\{o_1>0\}}\Delta_k^{1,2}(m_1,o_1-1,n_2) \\
& + h\, n_2\mu_2 1_{\{n_2>0\}}\Delta_k^{1,2}(m_1,o_1,n_2-1) \\
& + h\,\mu_1\left[\boldsymbol{V}_k(m_1,o_1,n_2)-\boldsymbol{V}_k(m_1,o_1+1,n_2)\right] \\
& + h\,\mu_1\left[\boldsymbol{V}_k(m_1+1,o_1,n_2)-\boldsymbol{V}_k(m_1,o_1,n_2)\right] \\
& + h\,\lambda_1 1_{\{m_1+1<N_1\}}\Delta_k^{1,2}(m_1+1,o_1,n_2) \\
& + h\lambda_1 1_{\{m_1+1=N_1;\, o_1+n_2+1<N_2\}}\left[\boldsymbol{V}_k(N_1,o_1+1,n_2)-\boldsymbol{V}_k(N_1,o_1+1,n_2)\right] \\
& + h\lambda_1 1_{\{m_1+1=N_1;\, o_1+n_2+1=N_2\}}\left[\boldsymbol{V}_k(N_1,o_1+1,n_2)-\boldsymbol{V}_k(N_1,o_1+1,n_2)\right] \\
& + h\,\lambda_2 1_{\{o_1+n_2+1<N_2\}}\Delta_k^{1,2}(m_1,o_1,n_2+1) \\
& + h\,\lambda_2 1_{\{o_1+n_2+1=N_2\}}\left[\boldsymbol{V}_k(m_1+1,o_1,n_2+1)-\boldsymbol{V}_k(m_1,\,o_1+1,\,n_2)\right] \\
& + \left[1-h(m_1+1)\mu_1-h(o_1+1)\mu_1-hn_2\mu_2-h\lambda_1-h\lambda_2\right]\Delta_k^{1,2}(m_1,o_1,n_2)
\end{aligned} \tag{4.19}$$

which by cancelling all 0-terms can be rewritten as:

$$\begin{aligned}
\Delta_{k+1}^{1,2}(m_1,o_1,\,n_2) = {} & h\, m_1\mu_1 1_{\{m_1>0\}}\Delta_k^{1,2}(m_1-1,o_1,n_2) \\
& + h\, o_1\mu_1 1_{\{o_1>0\}}\Delta_k^{1,2}(m_1,o_1-1,n_2) \\
& + h\, n_2\mu_2 1_{\{n_2>0\}}\Delta_k^{1,2}(m_1,o_1,n_2) + h\mu_1\Delta_k^{1,2}(m_1,o_1,n_2) \\
& + h\,\lambda_1 1_{\{m_1+1<N_1\}}\Delta_k^{1,2}(m_1+1,o_1,n_2) + h\,\lambda_2 1_{\{o_1+n_2+1<N_2\}}\Delta_k^{1,2}(m_1,o_1,n_2+1) \\
& + h\lambda_2 1_{\{o_1+n_2+1=N_2\}}\left[\boldsymbol{V}_k(m_1+1,o_1,n_2+1)-\boldsymbol{V}_k(m_1,\,o_1+1,\,n_2)\right] \\
& + \left[1-h(m_1+1)\mu_1-h(o_1+1)\mu_2-hn_2\mu_2-h\lambda_1-h\lambda_2\right]\Delta_k^{1,2}(m_1,o_1,\,n_2)
\end{aligned} \tag{4.20}$$

Now note that the one but last term in the right hand side of (4.20) can be rewritten as

$$h\lambda_2 1_{\{o_1+n_2+1=N_2\}}\left[\boldsymbol{V}_k(m_1+1,o_1,n_2+1)-\boldsymbol{V}_k(m_1+1,\,o_1,\,n_2)+\right.$$
$$\left.\boldsymbol{V}_k(m_1+1,\,o_1,\,n_2)-\boldsymbol{V}_k(m_1,\,o_1+1,\,n_2)\right] \qquad (4.21)$$
$$= h\lambda_2 1_{\{o_1+n_2+1=N_2\}}\left[\Delta_k^3(m_1+1,o_1,n_2)+\Delta_k^{1,2}(m_1,o_1,n_2)\right]$$

By induction hypothesis and substitution we thus directly conclude by (4.20) and (4.21):

$$\Delta_{k+1}^{1,2}(\boldsymbol{n}) \le 0.$$

The induction for (4.9) is now completed by noting

$$\Delta_{k+1}^{1,2}(m_1,o_1,n_2) = \Delta_{k+1}^{1}(m_1,o_1,n_2) - \Delta_{k+1}^{2}(m_1,o_1,n_2) \ge -C_1 \qquad (4.22)$$

as by (4.5) and (4.6) for $t = k+1$ as already proven above. By recalling the induction, the proof of Lemma 4.1 is hereby completed. □

Remarks 4.2

1. (Comparison result)

The comparison result 4.1, even though intuitively obvious, appears not to have been proven formally before. This comparison result might also be provable by sample path comparison along the lines such as in [6], [7], [10], [13], [16]. However, in the present setting the comparison result (4.2) directly follows as a side result for proving the error bound result (4.3), as based upon Lemma 4.1

2. (Non-exponential case)

The product form expression (2.1) can be shown to be *insensitive*, that is to remain valid also for non-exponential call durations with means μ_i^{-1}, $i = 1, 2$, (e.g. [1], [5], [14], [22], [23]). It can therefore be expected and it is conjectured that Theorem 4.1 also applies for the non-exponential case. However, as the notational and technical details will be quite complex (e.g. cf. [21]), the present paper is restricted to the exponential case. A proof for this conjecture remains of interest.

3. (Waiting facilities)

As an extension of practical interest, also waiting facilities can be thought of. In this case, more notational complexity and a more complicated product form modification will be

required (e.g. see [2], [24]). Related comparison and error bound results can be expected, but have not yet been investigated.

4. (Other measures)

Other performance measures could have been considered, such as a mean delay, queue length or steady state tail probability. The actual error bound result (as based upon a technical lemma like Lemma 4.1) however, will depend on the specific form of r. As the throughput and loss probability are of primary natural interest, we have restricted our investigation to these measures. Similar results though can be expected along the same lines for these other measures.

5. (Other overflow structures)

In line with 3, also more complex overflow structures may be considered such as arising in call centers. In this respect, the results of this paper should be seen as just a first step to provide formal support for practical bounds and approximations. Further research in this direction seems certainly of interest.

4.2 CP-bound: Comparison with Erlang bound

A simple other upper bound for the loss probability of the original overflow system is provided by Erlang's loss expression:

$$\boldsymbol{L}_E = \overline{\boldsymbol{L}}_1 = \frac{1}{N_1!}\left(\frac{\lambda_1}{\mu_1}\right)^{N_1}\left[\sum_{k=0}^{N_1}\frac{1}{k!}\left(\frac{\lambda_1}{\mu_1}\right)^{k}\right]^{-1} \tag{4.23}$$

as corresponding to the system in which the groups are completely separated without overflow from group 1 to group 2. We refer to this system as the *Erlang system* (as opposed to the *call packing* system) and to (4.23) as the *Erlang bound*.

(Here, no proof at all is needed for the Erlang expression to provide an upper bound as an overflow call in the original system can in no way interfere anymore with group 1.)

This Erlang bound is standardly used to provide a rough first order approximation for the order of the exact loss probability $\boldsymbol{L}$. As intuitively appealing though, in all numerical

examples (as will be presented in section 5) the call packing bound appears to improve this Erlang bound as upper bound for the loss probability. The improvement appears to be substantial in the natural situation that the overflow group 2 is not highly congested by itself. (Otherwise, overflow wouldn't make much sense in the first place.) More precisely, in this natural situation the call packing bound and the exact loss probability appear to be rather close. It thus seems appealing and practically useful to use the call packing bound and not the Erlang bound. However, no formal justification seems to be available that the CP-bound always improves the Erlang bound. In contrast, at sample path basis, a 'counterintuitive' example is easily constructed, as shown below.

Counterintuitive example

Consider an overflow system with one primary server (line) and one overflow server (line). Suppose both servers to be idle at time 0. See Figure 4 for the arrival times and call duration of 6 calls. Under call packing the second type 1 call is first served at the overflow server before being switched back to the primary server. After switching, a type 2 call arrives so that both servers become busy and three type 1 calls are lost. In the Erlang system, in contrast, these calls are served while only one type 1 call is lost. In other words, in this example the average loss probability of type 1 calls for the Erlang system is smaller than for the CP system.

arrival time	call duration type 1	type 2
0	2	
1	9	
2		9
3	2	
6	2	
9	2	

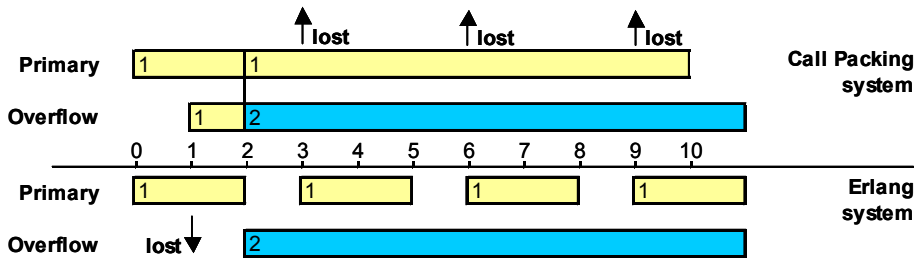

Figure 4: Lost calls in a call packing system versus Erlang system.

A formal proof is thus required. To this end, we will proceed as in section 4.1 as based upon section 3. To compare the Erlang and CP-bound we need to be able to compare the underlying Markov chains. To do so, for the Erlang system we consider the joint state description for both the first (primary) and second (overflow) group. For the call packing system, in contrast, note that it suffices to keep track of the number of type 1 and type 2 calls. The state description can thus be denoted by $\boldsymbol{n} = (n_1, n_2)$ with n_i the number of type i calls, i =

1, 2. In the setting of section 3, let the Erlang system be associated with the transition rates $\bar{\boldsymbol{q}}$ as well as its corresponding quantities be denoted with an upper bar, while (as in contrast to section 4.1) the call packing system with transition rates $\boldsymbol{q}$ and its quantities without an upper bar. Now first note, as essentially required in section 3, that

$$\begin{aligned}&\bar{S} \subseteq S \text{ with}\\ &\bar{S} = \{(n_1, n_2) | n_1 \le N_1,\, n_2 \le N_2\}\\ &S = \{(n_1, n_2) | (n_1 - N_1)^+ + n_2 \le N_2\}\end{aligned}$$

Theorem 4.2 With $\bar{\boldsymbol{T}}_1$ and $\boldsymbol{T}_1$ the throughput of type 1 calls for the *Erlang* and call packing (CP) system respectively, and $\bar{\boldsymbol{L}}_1$ and $\boldsymbol{L}_1$ the corresponding loss probabilities, we have

$$\bar{\boldsymbol{T}}_1 \le \boldsymbol{T}_1 \quad (\bar{\boldsymbol{L}}_1 \ge \boldsymbol{L}_1) \tag{4.24}$$

Proof. For any $\boldsymbol{n} \in \bar{S} \subseteq S$:

$$\begin{aligned}&\sum\nolimits_{\boldsymbol{n}'} \left[\bar{q}(\boldsymbol{n}, \boldsymbol{n}') - q(\boldsymbol{n}, \boldsymbol{n}')\right]\left[\boldsymbol{V}_k(\boldsymbol{n}') - \boldsymbol{V}_k(\boldsymbol{n})\right] = \\ &\quad 1_{\{n_1 = N_1; n_2 < N_2\}} N_1 \mu_1 \left[\boldsymbol{V}_k(N_1, n_2) - \boldsymbol{V}_k(N_1 + 1, n_2)\right]\end{aligned} \tag{4.25}$$

Lemma 4.2 below and Result 3.1 now complete the proof. □

Lemma 4.2

$$\Delta_t^1(\boldsymbol{n}) = \boldsymbol{V}_t(n_1 + 1, n_2) - \boldsymbol{V}_t(n_1, n_2) \ge 0 \tag{4.26}$$

$$\Delta_t^2(\boldsymbol{n}) = \boldsymbol{V}_t(n_1, n_2 + 1) - \boldsymbol{V}_t(n_1, n_2) \le 0 \tag{4.27}$$

$$\Delta_t^{1,2}(\boldsymbol{n}) = \boldsymbol{V}_t(n_1 + 1, n_2) - \boldsymbol{V}_t(n_1, n_2 + 1) \ge 0 \tag{4.28}$$

Proof. Again, the proof will be based on induction. Clearly, (4.26), (4.27) and (4.28) hold for $t = 0$. Let (4.26), (4.27) and (4.28) hold for $t \le k$. Recall that we consider the call packing system. Then, by steps similar to those in the proof of Lemma 4.1 to derive (4.13), the following relations can be derived for $\Delta_{k+1}^1(\boldsymbol{n})$, $\Delta_{k+1}^2(\boldsymbol{n})$ and $\Delta_{k+1}^{1,2}(\boldsymbol{n})$. Herein, for the clarity of their derivation, some of the terms which are indeed equal to 0 are kept in.

$$
\begin{aligned}
\Delta^1_{k+1}(n_1,n_2) = {} & h\mu_1 \\
& + h\, n_1\mu_1 1_{\{n_1>0\}} \Delta^1_k(n_1-1,n_2) \\
& + h\, n_2\mu_2 1_{\{n_2>0\}} \Delta^1_k(n_1,n_2-1) \\
& + h\, \mu_1 \left[\boldsymbol{V}_k(n_1,n_2) - \boldsymbol{V}_k(n_1,n_2) \right] \\
& + h\, \lambda_1 1_{\{(n_1-N_1)^+ + n_2 + 1 < N_2\}} \Delta^1_k(n_1+1,n_2) \\
& + h\, \lambda_1 1_{\{(n_1-N_1)^+ + n_2 + 1 = N_2\}} \left[\boldsymbol{V}_k(n_1+1,n_2) - \boldsymbol{V}_k(n_1+1,n_2) \right] \\
& + h\, \lambda_2 1_{\{(n_1-N_1)^+ + n_2 + 1 < N_2\}} \Delta^1_k(n_1,n_2+1) \\
& + h\, \lambda_2 1_{\{(n_1-N_1)^+ + n_2 + 1 = N_2\}} \Delta^{1,2}_k(n_1,n_2) \\
& + \left[1 - h(n_1+1)\mu_1 - hn_2\mu_2 - h\lambda_1 - h\lambda_2 \right] \Delta^1_k(n_1,n_2)
\end{aligned}
\tag{4.29}
$$

By substituting the induction hypotheses (4.26), (4.27) and (4.28) for $t = k$, we hereby directly verify $\Delta^1_{k+1}(n_1,n_2) \geq 0$. (I.e. (4.26) for $t = k$+1.) Similarly,

$$
\begin{aligned}
\Delta^2_{k+1}(n_1,n_2) = {} & h\, n_1\mu_1 1_{\{n_1>0\}} \Delta^2_k(n_1-1,n_2) \\
& + h\, n_2\mu_2 1_{\{n_2>0\}} \Delta^2_k(n_1,n_2-1) \\
& + h\mu_2 \left[\boldsymbol{V}_k(n_1,n_2) - \boldsymbol{V}_k(n_1,n_2) \right] \\
& + h\, \lambda_1 1_{\{(n_1-N_1)^+ + n_2 + 1 < N_2\}} \Delta^2_k(n_1+1,n_2) \\
& + h\, \lambda_1 1_{\{(n_1-N_1)^+ + n_2 + 1 = N_2\}} \left[-\Delta^{1,2}_k(n_1,n_2) \right] \\
& + h\, \lambda_2 1_{\{(n_1-N_1)^+ + n_2 + 1 < N_2\}} \Delta^2_k(n_1,n_2+1) \\
& + h\, \lambda_2 1_{\{(n_1-N_1)^+ + n_2 + 1 = N_2\}} \left[\boldsymbol{V}_k(n_1,n_2+1) - \boldsymbol{V}_k(n_1,n_2+1) \right] \\
& + \left[1 - hn_1\mu_1 - h(n_2+1)\mu_2 - h\lambda_1 - h\lambda_2 \right] \Delta^2_k(n_1,n_2)
\end{aligned}
\tag{4.30}
$$

Again, by substituting the induction hypotheses (4.27) and (4.28) (note the minus sign) for $t = k$, we directly verify: $\Delta^2_{k+1}(n_1,n_2) \leq 0$. (I.e. (4.27) for $t = k$+1.) Finally,

$$
\begin{aligned}
\Delta_{k+1}^{1,2}(n_1,n_2) = {} & h\,\mu_1 \\
& + h\,n_1\mu_1 1_{\{n_1>0\}}\Delta_k^{1,2}(n_1-1,n_2) \\
& + h\,n_2\mu_2 1_{\{n_2>0\}}\Delta_k^{1,2}(n_1,n_2-1) \\
& + h\,\mu_1\left[\boldsymbol{V}_k(n_1,n_2)-\boldsymbol{V}_k(n_1,n_2+1)\right] \\
& + h\,\mu_2\left[\boldsymbol{V}_k(n_1+1,n_2)-\boldsymbol{V}_k(n_1,n_2)\right] \\
& + h\,\lambda_1 1_{\{(n_1-N_1)^+ + n_2+1<N_2\}}\Delta_k^{1,2}(n_1+1,n_2) \\
& + h\,\lambda_2 1_{\{(n_1-N_1)^+ + n_2+1<N_2\}}\Delta_k^{1,2}(n_1,n_2+1) \\
& + h(\lambda_1+\lambda_2) 1_{\{(n_1-N_1)^+ + n_2+1=N_2\}}\left[\boldsymbol{V}_k(n_1+1,n_2)-\boldsymbol{V}_k(n_1,n_2+1)\right] \\
& + \left[1-h(n_1+1)\mu_1 - h(n_2+1)\mu_2 - h\lambda_1 - h\lambda_2\right]\Delta_k^{1,2}(n_1,n_2)
\end{aligned}
\tag{4.31}
$$

Here, first note that the fourth term in the right hand side of (4.31) can be written as $h\mu_1\left[-\Delta_k^2(n_1,n_2)\right]$, the fifth as $+h\mu_2\Delta_k^1(n_1,n_2)$ and the one but last as $h(\lambda_1+\lambda_2)1_{\{(n_1-N_1)^+ + n_2+1=N_2\}}\Delta_k^{1,2}(n_1,n_2)$. Then, again by substituting the induction hypotheses (4.26), (4.27) and (4.28) for $t = k$, we also directly verify: $\Delta_{k+1}^{1,2}(n_1,n_2) \le 0$. (I.e. (4.28) for $t = k+1$.) The induction completes the proof. □

Remark 4.3 (Error bound) As in Theorem 4.1 and Lemma 4.1 also an error bound could be concluded of the form

$$
\left|\overline{\boldsymbol{T}}_1 - \boldsymbol{T}_1\right| \le \overline{\pi}(n_1 = N_1)\,\lambda_1\left(1+\frac{\lambda_1}{\mu_2}\right)
$$

However, as this error bound does effectively provide just a zero lower bound for the loss probability it has not been included in Theorem 4.2.

4.3 Stop bound

Another upper bound for the loss probability of the original overflow system could be obtained by the (first) modification, already mentioned in section 2, to ensure a product form:

> Stop the service of overflowed type 1 calls at the second group when capacity at the first group becomes available ($m_1 < N_1$).

The state description $\boldsymbol{n} = (m_1, o_1, n_2)$ will remain the same and with $\bar{\pi}(\boldsymbol{n})$ the steady state distribution under this modification, again a product form can be concluded, as based upon the so-called notion of station balance (see [23]) or by writing the global balance equations, by verifying the local balance equations for each component separately:

$$\begin{aligned}
\bar{\pi}(m_1,o_1,n_2)\, m_1\mu_1 1_{\{m_1>0\}} &= \bar{\pi}(m_1-1,o_1,n_2)\lambda_1 1_{\{m_1>0\}} \\
\bar{\pi}(m_1,o_1,n_2)\, o_1\, \mu_1 1_{\{m_1=N_1;\, o_1>0\}} &= \bar{\pi}(N_1,o_1-1,n_2)\lambda_1 1_{\{m_1=N_1;\, o_1>0\}} \\
\bar{\pi}(m_1,o_1,n_2)\, n_2\mu_2 1_{\{n_2>0\}} &= \bar{\pi}(m_1,o_1,n_2-1)\lambda_2 1_{\{n_2>0\}}.
\end{aligned} \tag{4.32}$$

which lead to the product form:

$$\bar{\pi}(m_1,o_1,n_2) = c\frac{1}{m_1!}\left(\frac{\lambda_1}{\mu_1}\right)^{m_1}\frac{1}{o_1!}\left(\frac{\lambda_1}{\mu_1}\right)^{o_1}\frac{1}{n_2!}\left(\frac{\lambda_2}{\mu_2}\right)^{n_2}. \tag{4.33}$$

Remark 4.4

In fact, with m_2 the total number of calls at the second group, by simple summation these probabilities lead

$$\bar{\pi}(m_1,m_2) = c\frac{1}{m_1!}\left(\frac{\lambda_1}{\mu_1}\right)^{m_1}\frac{1}{m_2!}(\rho)^{m_2} \quad \text{with } \rho = \frac{\lambda_1}{\mu_1}+\frac{\lambda_2}{\mu_2}. \tag{4.34}$$

These in turn coincide exactly to the joint steady state distribution for the (third) modification as suggested in [6], [7] and [13], by considering the overflow group as an independent group with arrival rate $\lambda_1+\lambda_2$ and with mean service time:

$$\frac{\lambda_1\mu_1^{-1}+\lambda_2\mu_2^{-1}}{\lambda_1+\lambda_2}. \tag{4.35}$$

We will briefly refer to this (first) modification the *Stop-system*. Clearly, it seems intuitively obvious that this modification leads to an upper bound for the loss probability of type 1 calls. This can be proven rather directly by sample path comparison if there are no type 2 calls. However, in the presence of type 2 calls this will be more complicating as the effect for type 1 calls and type 2 calls can be conflicting as also illustrated in the proof of Lemma 4.1 (e.g.

see (4.20) and (4.21)). For the special case that $\mu_1 = \mu_2$ its proof can be concluded directly from [7] or [13].

Along the lines of this paper, however, a formal proof for the present case, also with two types of calls and call dependent means, is obtained directly.

To this end, let the stop-system be associated with the transition rates $\bar{\boldsymbol{q}}$ as well as its corresponding quantities be denoted with an upper bar, while the original system as in section 4.1 is associated with transition rates $\boldsymbol{q}$ and its quantities without an upper bar. Now first note, that

$$\bar{S} = S \text{ with}$$
$$S = \{(m_1, o_1, n_2) \mid m_1 \le N_1;\ o_1 + n_2 \le N_2\}$$

Theorem 4.3 With $\bar{\boldsymbol{T}}_1$ and $\boldsymbol{T}_1$ the throughput of type 1 calls for the stop and original system respectively, and $\bar{\boldsymbol{L}}_1$ and $\boldsymbol{L}_1$ the corresponding loss probabilities, we have

$$\bar{\boldsymbol{T}}_1 \le \boldsymbol{T}_1 \quad (\bar{\boldsymbol{L}}_1 \ge \boldsymbol{L}_1) \tag{4.36}$$

Proof. For any state $\boldsymbol{n} \in \bar{S} = S$:

$$\begin{aligned} &\textstyle\sum_{\boldsymbol{n}'} \left[\bar{q}(\boldsymbol{n}, \boldsymbol{n}') - q(\boldsymbol{n}, \boldsymbol{n}')\right]\left[\boldsymbol{V}_k(\boldsymbol{n}') - \boldsymbol{V}_k(\boldsymbol{n})\right] \\ &\quad = 1_{\{m_1 < N_1;\ o_1 > 0\}} o_1 \mu_1 \left[\boldsymbol{V}_k(m_1, o_1, n_2) - \boldsymbol{V}_k(m_1, o_1 - 1, n_2)\right] \\ &\quad = 1_{\{m_1 < N_1;\ o_1 > 0\}} o_1 \mu_1 \Delta_k^2 (m_1, o_1 - 1, n_2) \end{aligned} \tag{4.37}$$

Relation (4.6) from Lemma 4.1 and Result 3.1 now complete the proof. □

Remark 4.5

The natural question might come up which of the two bounds is superior: the CP-bound or the stop-bound as based on this stop protocol. Intuitively, the stop protocol reduces the throughput quite seriously. Generally, it might therefore be thought of as being less accurate. Indeed, this will be supported by numerical results (see Figure 5) for rather natural situations. Nevertheless, in rather unnatural extremely overloaded overflow situations by type 2 calls, it might not be true. An ordering of the CP-bound and Stop-bound can therefore not be provided.

Remark 4.6

By the same steps as in Theorem 4.2 or 4.3, or by expression (4.34), it is also easily shown that the loss probability under the stopping protocol ($\bar{\boldsymbol{L}}_1$ in this section) improves the Erlang bound $\boldsymbol{L}_E$ (see (4.23)).

5. Numerical results

In this section, we provide some numerical results in order to show the improvement of the CP-bound over the Erlang bound to investigate, the quality with respect to the exact value and to illustrate the potential of the analytic error bound.

As the throughput and loss probability of type 1 calls are directly related by (2.4) while the loss probability seems more distinctive as a measure of quality, we prefer to present the numerical results for loss probabilities. The results of Theorem 4.1, 4.2 and 4.3 can then be expressed by

$$(\boldsymbol{L}_{CP} - \delta)^+ \le \boldsymbol{L} \le \min\{\boldsymbol{L}_{CP}, \boldsymbol{L}_S\} \le \boldsymbol{L}_E \quad \text{with} \quad \delta = \bar{\pi}(n_1 > N_1)\frac{N_1\mu_1}{\lambda_1}\left(\frac{\lambda_1 + \mu_2}{\mu_2}\right). \tag{5.1}$$

with $\bar{\pi}(\ )$ as by (2.1) and where $\boldsymbol{L}_{CP}$, $\boldsymbol{L}_S$, $\boldsymbol{L}_E$ and $\boldsymbol{L}$ denote the loss probability for the Call Packing, Stop-, Erlang- and original overflow system respectively.

Call Packing bound versus Erlang bound

First we investigate overflow systems that can be expected in natural situations: a heavily loaded primary server group (efficient use of first group) combined with an overflow group with low utilization (e.g. 40%), so that overflow makes sense. By varying the arrival rate of type 1 calls (from 20% to 100% of the capacity of group 1) Figure 5 shows the effect of overflow for the loss probability as opposed to the loss probability of just the primary station, that is the Erlang bound. Here the loss probability of the original system is obtained by simulation. The Erlang bound progressively overestimates the loss probability. The CP-bound in contrast progressively improves the Erlang bound. In fact, the loss probabilities of the original and the call packing system remain in the same order of magnitude with the CP-bound as a secure upper bound. In addition, also the stopping bound is included as based on

the modification discussed in section 4.3, as also suggested in [20] and which appears to be equivalent to the modification provided in [6], [7], [13].

Experiment 1	
λ_1	-
μ_1	1
N_1	10
λ_2	4
μ_2	1
N_2	10

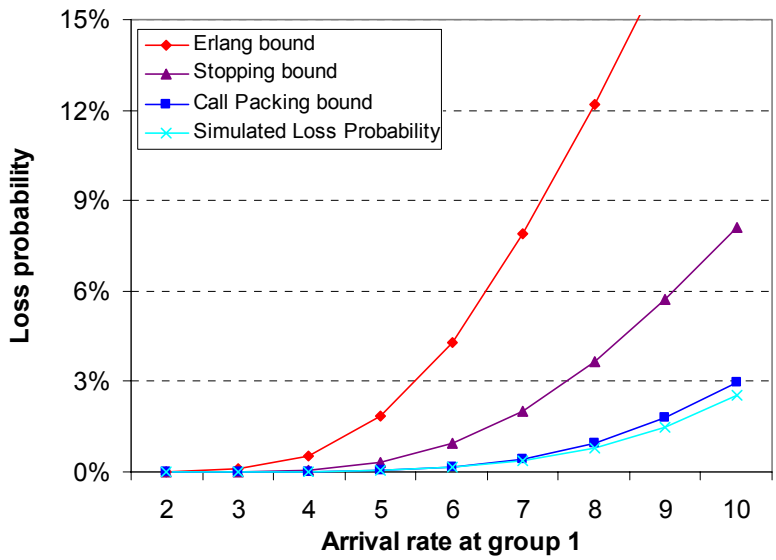

Figure 5: Results of experiment 1.

To further examine the effect of the overflow group we increased the workload of the second group to 60, 80 and 100% and repeated the same experiment. The results are summarized in Figure 6. The results show that the CP-bound remains to provide a considerable improvement over the Erlang bound also for highly occupied overflow groups

Experiment 2	
λ_1	-
μ_1	1
N_1	10
λ_2	6, 8, 10
μ_2	1
N_2	10

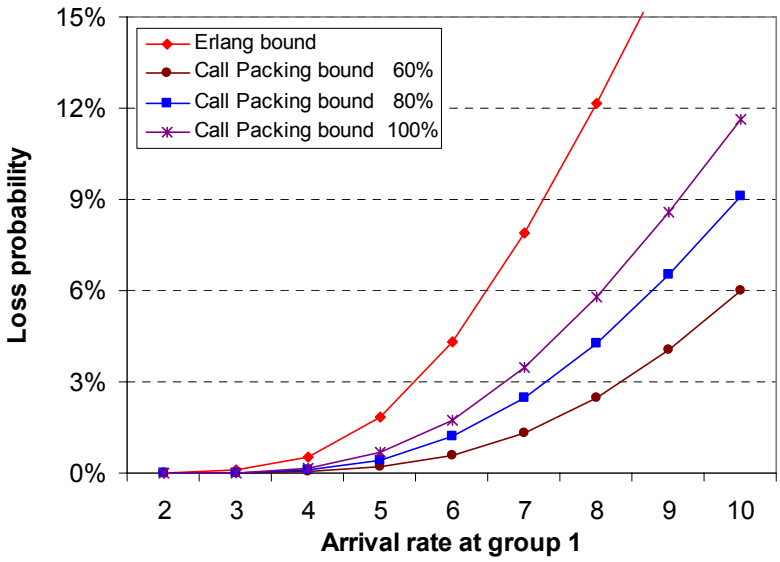

Figure 6: Results of experiment 2.

Error bound

But obviously as illustrated by Figure 6, the discrepancy between the CP-bound and the Erlang bound becomes smaller the more occupied the overflow group. For that reason we also created an extremely busy (400% workload) overflow group, while varying the workload of the primary group by changing the number of servers.

As could be expected, in this case the CP-bound and the Erlang bound are reasonably close (see Figure 7) and both provide a secure upper bound. However, the call packing system also provides a lower bound as based upon the error bound result in Theorem 4.1 as presented by (5.1).

Experiment 3	
λ_1	0.1
μ_1	0.01
N_1	-
λ_2	40
μ_2	1
N_2	10

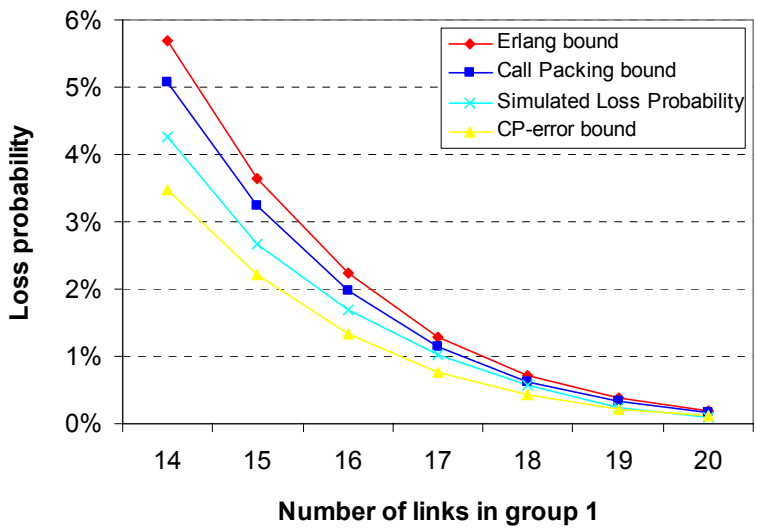

Figure 7: Results of experiment 3.

Call Packing bound versus Exact loss probability

Finally, to further investigate the quality of the CP-bound we examined systems in which both groups are highly congested (the primary group up to 150%; the overflow group 133%). In both systems (the original and call packing) a type 1 call thus has a high probability of finding both groups fully occupied. In the call packing system an overflowed call is switched back as soon a server in the primary group becomes available. As a result, the primary group remains heavily occupied. In the original system in contrast, the overflowed call is completed in the overflow group while keeping free capacity exclusively for type 1 calls. This suggests that the original system performs substantially better than the call packing system.

Call Packing Bounds

Nevertheless, as illustrated in Figure 8 even in these situations the CP-bound appears to perform reasonably well. In combination with Figure 5 and Figure 7 the CP-bound thus seems to provide a quite reasonable first order approximation overall.

Experiment 4	
λ_1	-
μ_1	0.1
N_1	4
λ_2	40
μ_2	10
N_2	3

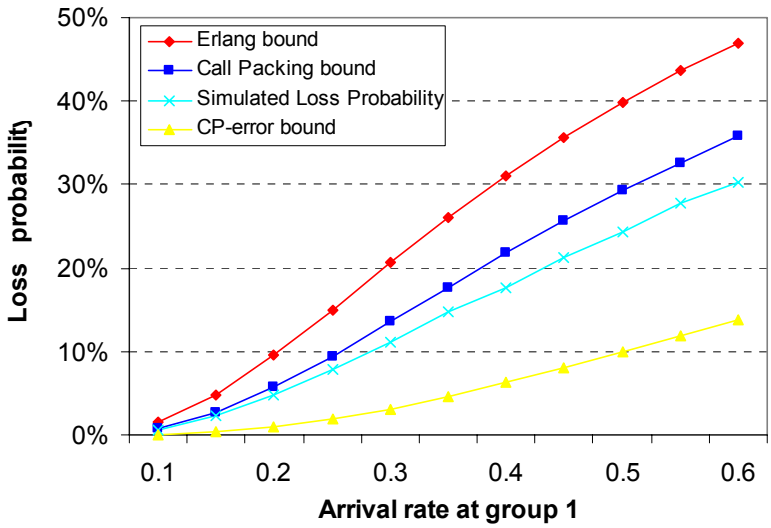

Figure 8: Results of experiment 4.

CP bound and Erlang loss approximation

In line with the latter statement, as another 'standard' approximation also the Erlang loss expressions $\boldsymbol{B}_1$ for both the first group (with arrival rate λ_1) and $\boldsymbol{B}_2$ the second group (with arrival rate $\lambda_2 + \boldsymbol{B}_1\,\lambda_1$) can be combined, leading to an approximation of the form

$$\boldsymbol{B}_1 \cdot \boldsymbol{B}_2$$

Also this approximation turns out to be reasonably accurate as illustrated by Figure 9. However, there is no formal guarantee at all for the approximation to be accurate nor to provide a secure (lower or upper) bound for the loss probability. In fact, intuitively it will lead to a lower bound (as also reflected by Figure 9). As an upper rather than lower bound for a loss probability will generally be more important (such as for setting service norms or dimensioning purposes) in the first place, the CP-bound, which also seems to establish the same order of accuracy as the Erlang approximation, thus seems to be more practical as a secure upper bound approximation.

Experiment 5	
λ_1	-
μ_1	1
N_1	10
λ_2	8
μ_2	1
N_2	10

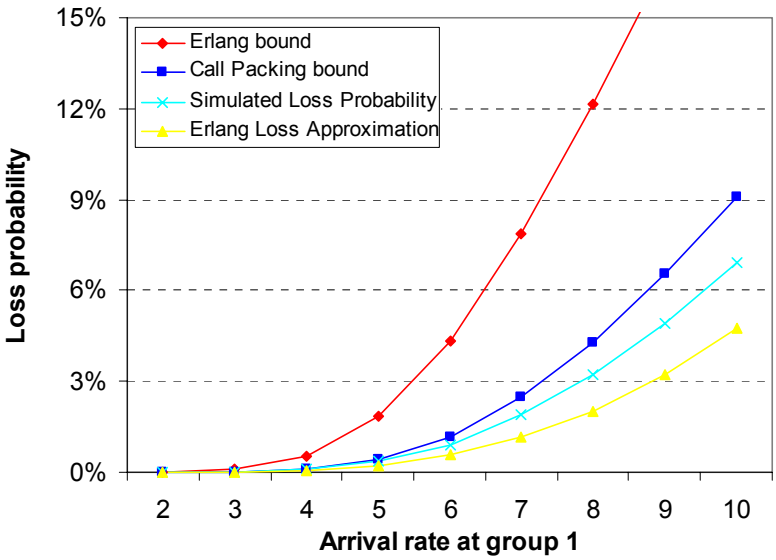

Figure 9: Results of experiment 5.

Figure 9 provides an illustrative summary of the upper bounds, as proven in sections 4.1, 4.2 and 4.3, as well as the loss approximation.

References

[1] BARBOUR, A.D. (1976). Networks of queues and the method of stages. *Adv. Appl. Prob.* **8** 584-591.

[2] BEREZNER, S.A., A.E. KRZESINSKI and P.G. TAYLOR (1998). A Product-Form "Loss Network" with a Form of Queueing, *J. Appl. Probab.* **34** 1075–1078.

[3] BERRY, L.T.M. and W. HENDERSON (1989). Some Exact results in Performance Analysis of Alternative Routing Communications Networks, *A.T.R.* **23** 35-42.

[4] GRASSMANN, W. (1990). Finding Transient Solutions in Markovian Event Systems through Randomization. In *Numerical Solutions of Markov Chains*, (ed. W.J. Stewart), 375-395. Marcel Dekker.

[5] HENDERSON, W. AND P.G. TAYLOR (1988). Alternative Routing Networks and Interruptions, *Proceedings of the 12th International Teletraffic Conference*, Torino.

[6] HORDIJK, A. and A. RIDDER (1987). Stochastic Inequalities for an Overflow Model. *J. Appl. Prob.* **24** 696-708.

[7] HORDIJK, A. and A. RIDDER (1988). Insensitivity Bounds for the Stationary Distribution of Non-reversible Chains. *J. Appl. Prob.* **25** 9-20.

[8] KELLY, F.P. (1991). Loss Networks, *Ann. Appl. Probab.* **1** 319-378.

[9] MASSEY, W.A. (1987). Stochastic orderings for Markov processes on partially ordered spaces, *Math. Oper. Res.* **12** 350-367.

[10] MIYAZAMA, M. and N.M. VAN DIJK (1997). A Note on Bounds and Error Bounds for Non-Exponential Batch Arrival Systems, *Probab. Engrg. Inform. Sci.* **11** 189-201.

[11] MOLINA, E.C. (1931). Appendix to: "Interconnection of Telephone Systems – Graded Multiple" by R.I. Wilkinson. *Bell System Technical Journal* **10** 271-283.

[12] PALLANT, D.L. and P.G. TAYLOR (1995). Modelling Handovers in Cellular Mobile Networks with Dynamic Channel Allocation, *Oper. Res.* **43** 33–42.

[13] RIDDER, A. (1987). Stochastic Inequalities for Queues. Ph.D. Thesis, University of Leiden.

[14] SCHASSBERGER, R. (1978). The insensitivity of stationary probabilities in networks of queues. *Adv. Appl. Prob.* **10** 357-378.

[15] SHAKED, M. and J.G. SHANTHIKUMAR (1994). *Stochastic orders and their applications*. Academic Press, San Diego.

[16] SONDERMAN, D. (1979). Comparing multi-server queues with finite waiting rooms, II: Different number of servers, *Ann. Appl. Probab.* **11** 448-455.

[17] STOYAN, D. (1983). *Comparison Methods for Queues and Other Stochastic Models*. Wiley, New York.

[18] TAYLOR, P.G. and N.M. VAN DIJK (1998). Strong Stochastic Bounds for the Stationary Distribution of a Class of Multicomponent Models, *Oper. Res.* **46** 665–674.

[19] TIJMS, H.C. (1994). *Stochastic Models: An Algorithmic Approach*. Wiley, Chichester.

[20] VAN DIJK, N.M. (1987). Simple and Insensitive Bounds for a Grading and an Overflow Model. *Oper. Res. Lett.* **6** 73-76.

[21] VAN DIJK, N.M. (1988). A Formal Proof for the Insensitivity of Simple Bounds for Finite Multi-Server Non-Exponential Tandem Queues. *Stochastic Process. Appl.* **27** 261-277.

[22] VAN DIJK, N.M. (1990). Mixed parallel processors with interdependent blocking. *Appl. Stochastic Models and Data Analysis* **6** 85-100.

[23] VAN DIJK, N.M. (1993). *Product forms for Queueing Networks: A system's Approach*. Wiley.

[24] VAN DIJK, N.M. (1994). Product forms for Flexible Manufacturing Systems. Research report, unpublished.

[25] VAN DIJK, N.M. (1997). Bounds and error bounds for queueing networks. *Ann. Oper. Res.* **79** 295-319.

[26] VAN DIJK, N.M. and K. SLADKÝ (2000). A Note on Uniformization for Dynamic Non-negative Systems, *J. Appl. Probab*. **37** 329–341.

[27] VAN DOORN, E.A. (1984). On the overflow process from a finite Markovian queue. *Performance Eval.* **4** 233-240.

[28] Winter Simulation Conference 2003, http://www.wintersim.org/

UNIVERSITY OF AMSTERDAM
FACULTY OF ECONOMICS AND ECONOMETRICS
ROETERSSTRAAT 11
1018 WB AMSTERDAM
THE NETHERLANDS
E-MAIL: N.M.vanDijk@uva.nl
H.J.vanderSluis@uva.nl
URL: http://www.uva.fee.nl/ke/sluis