The Unified Approach for Teletraffic Models to Convert Recursions for Global State Probabilities into Stable Form

Iversen, V.B. 1 and Stepanov, S.N. 2

1 COM Center
Building 345v, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark
vbi@com.dtu.dk
2 Sistema Telecom
125047, Moscow, 1st Tverskay-Yamskaya 5, Russia
stepanov@sistel.ru

Abstract. In this paper we describe a unified approach for teletraffic models to convert recursions for global state probabilities into stable form. At each step of recurrence we deal with normalised values of global state probabilities used for estimation of main stationary performance measures. This makes it suited for solving problems of dimensioning. In addition to being numerically stable, the main positive features of the suggested approach are its simplicity and small storage and computational requirements.

Keywords: teletraffic models, global state probabilities, stable recurrence, dimensioning.

1 Introduction

One of the main problems of an operator is to determine the volume of resources that are necessary for serving given input flows of demands with prescribed characteristics of QoS. As the resource is considered number of circuits, value of bandwidth represented by integer number of units, number of radio channels, and so on. The necessary volume is determined by comparison of suitably chosen performance measure with prescribed level of system functioning.

The general scheme of solving the formulated problem is as follows. In the first place it is necessary to construct the mathematical model that gives the required performance measure. Usually this is done in terms of queueing theory. Then based on numerical methods the algorithm is elaborated for estimating the performance measure. After performing the searching the necessary value of telecommunication volume is found. The numerical complexity of an approach is estimated by the numerical complexity for finding the performance measures for given input parameters multiplied by the number of times searching.

The significance of solving the formulated problem for optimal distribution of telecommunication resources has forced researchers to look for ways to decrease the amount of computational work. In this paper we described a unified approach for finding the desired solution and formulate the necessary conditions for its realization. We construct an
approach that allows us to find the solution in one single iteration. It is important to notice that the method described gives numerically stable algorithm because at each step it deals with normalised values of state probabilities calculated for subset of states when an arriving request for resource is blocked. The proposed method is a direct generalisation of the well-known recursion for Erlang’s B-formula. We revive the corresponding recurrence in Section 2. The general form of solution will be presented in Section 3. The application of the unified approach will be shown for a number of cases in Section 4 (where alternative and very simple solutions will be given for some already known results) and in Section 5 (where new solutions will be presented).

2 Recursions for Erlang’s B-formula

Let us consider a single link traffic model, where the link transmission capacity is represented by \( k \) basic bandwidth units. In the model we have one Poissonian flow of demands for connections with intensity \( \lambda \) which we for simplicity will name by calls. We assume that each call uses one bandwidth unit for the time of connection having mean value equal to one. To simplify the description of the model dynamics we assume that the duration of connection (service time) is exponentially distributed with parameter equal to one, but most of the obtained results are valid for general service time distributions.

Let \( i(t) \) be number of calls being served at time \( t \). The dynamics of the model are described by a Markov process \( r(t) \) having one component \( r(t) = i(t) \). The process \( r(t) \) takes values in the finite set of states \( S = \{0, 1, \ldots, k\} \) defined in accordance with link capacity. Let us by \( P(i) \) denote the unnormalised values of stationary probabilities of \( r(t) \). After normalisation, the value \( P(i) \) denotes the mean proportion of time when exactly \( i \) connections are established or \( i \) bandwidth units are occupied. From the system of state equations follows:

\[
P(i) = \frac{\lambda}{i} P(i - 1), \quad i = 1, 2, \ldots, k.
\]

(1)

Together with the normalising condition relation (1) gives the Erlang–B loss formula for estimation of the mean proportion of time when all \( k \) bandwidth units are occupied:

\[
E_k(\lambda) = \frac{\frac{\lambda^k}{k!}}{1 + \lambda + \frac{\lambda^2}{2!} + \cdots + \frac{\lambda^k}{k!}}.
\]

(2)

For dimensioning purposes (2) can be written in the form of recurrence:

\[
E_k(\lambda) = \frac{\lambda}{k} \cdot E_{k-1}(\lambda), \quad k = 1, 2, \ldots, \quad E_0(\lambda) = 1.
\]

(3)

The recursion (3) works with normalised values \( E_k(\lambda) \) so its implementation does not suffer from numerical problems such as overflow or loss of precision that very often occur for large values of \( k \). Using (3) consecutively for \( k = 1, 2, \ldots \), it is easy to find the value of link capacity \( k \) that is necessary for serving the given traffic flow with prescribed QoS. It is easy to see that the computational work for solving the problem equals computational work for estimating \( E_k(\lambda) \).
In the next section we construct an alternative scheme for proving (3). The result will be presented in general form. This approach is quite simple and can easily be generalised to a number of multi-rate models.

3 General case

Let us consider a model of a service system with \( k \) service units. Let \( p(i) \) be the mean proportion of time when exactly \( i \) service units are occupied. The main stationary performance measures of practical interest can be expressed as a function of \( p(i), i = 0, 1, \ldots, k \). Let parameter \( B \) denote a measure used for dimensioning. A traditional dimensioning problem related with system planning is formulated as follows:

For given input flow, find the minimum value of \( k \) satisfying the inequality:

\[
B \leq \pi, \tag{4}
\]

where \( \pi \) is a prescribed level of system performance. Typically it will be the proportion of calls lost or the proportion of time the system is blocked for call servicing. Usually only a limited number of \( p(i) \)'s are used in estimation of \( B \) and the number of \( p(i) \)'s used does not depend on the number of service units. For example, if each call of the input flow needs \( b \) units for its servicing then the proportion of time the system is blocked for call servicing can be estimated by the expression:

\[
B = p(k - b + 1) + p(k - b + 2) + \cdots + p(k).
\]

Later we assume that estimation of \( B \) for the traffic using \( k \) service units can be done by means of probabilities \( p(k - b + 1), p(k - b + 2), \ldots, p(k) \), where \( b \) is some integer number independent of \( k \).

To solve the problem (4) we usually proceed according to the following recursive scheme. Let us by \( P(i) \) denote unnormalised values of \( p(i), i = 0, 1, \ldots, k \), and suppose that the following relation is known for estimation of \( P(i) \):

\[
P(i) = \begin{cases} 
0 & \text{if } i < 0 \\
\alpha & \text{if } i = 0 \\
f(P(i - 1), P(i - 2), \ldots, P(i - m)) & \text{if } i = 1, 2, \ldots, k 
\end{cases} \tag{5}
\]

In (5) \( m \) is an integer. We suppose that its value is independent of the number of service units \( k \). In some cases \( m \) is just equal to the maximum number of service units needed for call servicing. The function \( f(\cdot) \) depends on the parameters of the model. Specific types of the function \( f(\cdot) \) will be presented in the next section where we will consider examples of teletraffic models. The parameter \( \alpha \) is any positive real number chosen as starting value of recurrence. Using (5) sequentially, we can express all \( P(i), i = 1, 2, \ldots, k \), by \( P(0) \). Then we find the normalising constant \( Q = \sum_{i=0}^{k} P(i) \) and the normalised values of state probabilities:

\[
p(i) = \frac{P(i)}{Q}, \quad i = 0, 1, \ldots, k. \tag{6}
\]

We suppose that for any choice of \( \alpha \) we obtain the same set of probabilities \( p(i), i = 0, 1, \ldots, k \), after normalisation. Next we calculate the performance measure \( B \) used in
criteria (4). If number of service units \( k \) does not satisfy (4), then the value of \( k \) is increased and we repeat the described actions once more.

The implementation of this procedure has at least two negative aspects. They are clearly observed for a large number of service units \( k \). For practically cases of interest almost all probability mass is located in states with large number of occupied service units. It means that states with a small number of occupied service units have very small probabilities of existence. The state with number of occupied service units equals to zero has the smallest probability different from zero. So when we start to express all \( P(i) \) through \( P(0) \) we very quickly experience problems of overflow. Another negative aspect is the increase of computational efforts. To solve the formulated dimensioning problem we need to perform a run of (5) for each value of \( k \) serving as a candidate for desired solution. Each time we need to start calculation from \( i = 0 \).

Let us formulate a condition on implementation of function \( f(\cdot) \) where we can solve this problem in one run as was done by recurrence (3) for service systems whose performance measures numerically are estimated by means of Erlang’s B-formula. we indicate when necessary the number of available service units by lower index for the corresponding set of probabilities. Let us assume that for any \( k > 1 \) the implementation of recurrence (5) for number of service units \( k - 1 \) and \( k \) and the same starting value gives sets of unnormalised probabilities \( P_{k-1}(0) = a, P_{k-1}(1), \ldots, P_{k-1}(k-1) \), and \( P_k(0) = a, P_k(1), \ldots, P_k(k-1), P_k(k) \) satisfying the following relations:

\[
P_k(i) = P_{k-1}(i), \quad i = 0, 1, \ldots, k - 1.
\]

Property (7) allows us to rewrite (5) in a form dealing with normalised probabilities.

Let us assume that we know the normalised set of probabilities \( p_{k-1}(0), p_{k-1}(1), \ldots, p_{k-1}(k-1) \) for number of service units \( k - 1 \). Let us assume that the number of service units is increased by one to \( k \). Then taking as starting value \( P_k(0) = p_k(0) \) and applying (5) we obtain according to property (7) the following set of unnormalised probabilities:

\[
P_k(i) = P_{k-1}(i), \quad i = 0, 1, \ldots, k - 1.
\]

For normalised probability \( P_k(k) \) we have:

\[
p_k(k) = \frac{P_k(k)}{P_k(0) + P_k(1) + \cdots + P_k(k)}
= \frac{P_k(k)}{p_k(0) + p_k(1) + \cdots + p_k(k-1) + P_k(k)}
= \frac{P_k(k)}{1 + P_k(k)}.
\]

Using (8) we rewrite (9) in the form:

\[
p_k(k) = \frac{f(p_{k-1}(k-1), p_{k-1}(k-2), \ldots, p_{k-1}(k-m))}{1 + f(p_{k-1}(k-1), p_{k-1}(k-2), \ldots, p_{k-1}(k-m))}.
\]
At this step we need to calculate the performance measure $B$ used for solving the dimensioning problem. Because the number of service units is increased by one we need to normalise probabilities $P_k(k - 1) = p_{k-1}(k-1)$, $P_k(k - 2) = p_{k-1}(k-2)$, ..., $P_k(k - b + 1) = p_{k-1}(k - b + 1)$, used for estimation of $B$ (the normalised value $P_k(k)$ has already been obtained by relation (10)). We can do this by means of equalities:

$$p_k(i) = \frac{p_{k-1}(i)}{1 + f(p_{k-1}(k-1), p_{k-1}(k-2), \ldots, p_{k-1}(k-j))}, \quad i = k-1, k-2, \ldots, k-b+1$$

(11)

If the obtained value of $B$ satisfies criteria (4), then the desired number of service units will be $k$. Otherwise it is necessary to increase the number of service units $k$ by one and make a new estimation of $B$ for $k + 1$. According to (8)-(10) to perform this step it is necessary to have normalised values of $P_k(k - 1)$, $P_k(k - 2)$, ..., $P_k(k - m)$. It is clear that these normalisations and (11) can be done in one run in the form:

$$p_k(i) = \frac{p_{k-1}(i)}{1 + f(p_{k-1}(k-1), p_{k-1}(k-2), \ldots, p_{k-1}(k-j))}, \quad i = k-1, k-2, \ldots, k-j$$

(12)

where $j = \max(b - 1, m)$.

A one–run algorithm giving at each step normalised values of state probabilities which are necessary for solving the problem of dimensioning based on (4) and (5) looks as follows:

**Step 1.** Let $p_0(0) = 1$.

**Step 2.** Let $j = \max(b - 1, m)$. For fixed $k = 1, 2, \ldots$, find normalised value of $p_k(i)$, $i = k, k - 1, \ldots, \max(k - j, 0)$, by using relations (10),(12)

$$p_k(k) = \frac{f(p_{k-1}(k-1), p_{k-1}(k-2), \ldots, p_{k-1}(k-j))}{1 + f(p_{k-1}(k-1), p_{k-1}(k-2), \ldots, p_{k-1}(k-j))};$$

(13)

$$p_k(i) = \frac{p_{k-1}(i)}{1 + f(p_{k-1}(k-1), p_{k-1}(k-2), \ldots, p_{k-1}(k-j))}, \quad i = k-1, k-2, \ldots, \max(k - j, 0)$$

**Step 3.** Here we calculate the performance measures, check the dimensioning criteria and either stop or continue the process of estimating the number of service units needed.

Fig. 1 shows the order of calculation of the normalised values of stationary probabilities in accordance with suggested approach. For the simplicity $j = 2$. We see that for fixed number of service units $k$ it is necessary to find only $\min(k, j) + 1$ normalised probabilities to be able to calculate the performance measure $B$ and proceed if necessary to the next step of recursion defined by (13).

Computational efforts for current number of service units $k$ is estimated by the total number of operations $N_f$ that are needed for estimation of function $f(\cdot)$ (once) with the number of operations $N_p$ that are needed for estimation of performance measure $B$ (once) and $j + 1$ operations of normalisation. So computational efforts to reach number of service units $k$ starting from 1 is estimated by $O\{(N_f + N_p + j + 1) k\}$. Storage requirements are estimated by $O\{j + 1\}$.

Because stable recursions are obtained directly from recursions for global state probabilities we can easily rewrite them into other forms in some cases more suitable for calculations than basic form. We consider two close modifications of the basic version of the formulated algorithm. Both allow to start stable recursions not from $k = 1$ but from
Let us consider the first modification of the basic algorithm. It is clear that we can start stable recursions from some intermediate value of \( k = k_0 \) if in some way (for example by means of (5)) we know normalised values of \( p_{k_0}(k_0), p_{k_0}(k_0 - 1), \ldots, p_{k_0}(k_0 - j) \). This follows directly from discussions previous stated algorithm (7)–(12) and Fig. 1. Let us for simplicity assume that \( k_0 \geq j \). Modified in such way the algorithm looks as follows

Step 1. Using (5) find normalised values \( p_{k_0}(k_0), p_{k_0}(k_0 - 1), \ldots, p_{k_0}(k_0 - j) \).

Step 2. Let \( j = \max(b - 1, m) \). For fixed \( k = k_0 + 1, k_0 + 2, \ldots \), find normalised value of \( p_k(i), i = k, k - 1, \ldots, k - j \), by using relations (10), (12):

\[
\begin{align*}
p_k(k) &= \frac{f(p_{k-1}(k-1), p_{k-1}(k-2), \ldots, p_{k-1}(k-j))}{1 + f(p_{k-1}(k-1), p_{k-1}(k-2), \ldots, p_{k-1}(k-j))}; \\
p_k(i) &= \frac{p_{k-1}(i)}{1 + f(p_{k-1}(k-1), p_{k-1}(k-2), \ldots, p_{k-1}(k-j))}, \quad i = k - 1, k - 2, \ldots, k - j.
\end{align*}
\]

Step 3. The same as formulated for basic version of algorithm.

The computational efforts are in accordance with modified versions of basic algorithm is estimated by \( O\{N_f + (N_B + j + 1)(1 - \frac{k_0}{B})\} k \}. Storage requirements are estimated by \( O\{j + 1\}. \)
Let us proceed to the second modification of the formulated algorithm. Now we suppose that the algorithm is applied for solving the dimensioning problems for large value of number of service units \( k \). In this case the distribution of probabilistic mass for global state probabilities has a specific form. Almost all probabilistic mass is located in the area very close to the state \((k)\), when all available service units are occupied.

Heuristics based on the analysis done for particular teletraffic models allows us to make the following suggestions. Let us choose some truncation level \( \ell \) and suppose that model functioning is located in global states \((\ell), (\ell+1), \ldots, (k)\). If the value of \( \ell \) is chosen so that \( p(\ell) \) is quite small then recurrence (5) for \( P(i) \) can rewritten in the form

\[
P^*(i) = \begin{cases} 
0 & \text{if } i < \ell \\
f(P^*(i-1), P^*(i-2), \ldots, P^*(i-m)) & \text{if } i = \ell + 1, \ell + 2, \ldots, k 
\end{cases} \tag{15}
\]

In (15) we use asterisk to show that \( P^*(i) \) gives approximate value of \( P(i) \). The accuracy of estimation should be studied separately for each specific model. We will show details on the example of one link multi-rate Poissonian model in Section 5.1. Now let us formulate the modified version of stable recurrence based on (15).

**Step 1.** Let \( p^*_i(\ell) = 1 \).

**Step 2.** Let \( j = \max(b-1,m) \). For fixed \( k = \ell + 1, \ell + 2, \ldots \), find normalised value of \( p^*_k(i), i = k, k-1, \ldots, \max(k-j, \ell) \), by using relations (10), (12) and (15):

\[
p^*_k(i) = \frac{f(p^*_{k-1}(k-1), p^*_{k-1}(k-2), \ldots, p^*_{k-1}(k-j))}{1 + f(p^*_{k-1}(k-1), p^*_{k-1}(k-2), \ldots, p^*_{k-1}(k-j))};
\]

\[
p^*_k(i) = \frac{p^*_{k-1}(i)}{1 + f(p^*_{k-1}(k-1), p^*_{k-1}(k-2), \ldots, p^*_{k-1}(k-j))}, \quad i = k-1, k-2, \ldots, \max(k-j, \ell).
\]

**Step 3.** The same as formulated for basic version of algorithm.

The computational efforts to reach number of service units \( k \) starting from \( \ell \) in accordance with modified versions of basic algorithm is estimated by \( O\{((N_f + N_B + j + 1) \cdot (k - \ell)) \}

Storage requirements are estimated by \( O(j + 1) \).

The results obtained are based on ideas from \([1],[2]\). If we implement the approach presented to specific teletraffic models we obtain numerically stable versions of recursive algorithms designed for estimation of model’s performance measures. For one link multi-rate models with product form solutions a number of numerically stable algorithms for calculation of main performance measures are known. \([3],[4],[5]\). If we compare the best results in this field \([4],[5]\) with algorithms obtained after usage of the suggested approach we can conclude that they have the same storage and computation requirement but here stable algorithms are derived much more simpler and by one scheme. All we need is just to take recurrence for global state probabilities in the form (5) and put it into relations (13) or it’s modifications (14), (16). This is shown in Sections 4.5.

The suggested approach has a number of other positive features. Firstly, the possibility of increasing the efficiency by truncation of the global state space is exploited. Another advantage is the possibility in unified way to treat a wide class of teletraffic models with/without product form solution. We illustrate this for couple of models in Sections 5.2-5.3, but first we show the usage of the approach for one link models with product form solutions.
4 One-link multi-service models with product form solutions

Let us consider a single link traffic model, where the link transmission capacity is represented by \( k \) basic bandwidth units, and let us suppose that we have \( n \) incoming Poisson flows of calls with intensities \( \lambda_s, s = 1, \ldots, n \). A call of \( s \)'th flow uses \( b_s \) bandwidth units for the time of connection. Without loss of generality we shall assume that the holding times all are exponentially distributed with the same mean value chosen to one, but it is known that the model considered is insensitive to the distribution of the holding time, and each flow may furthermore have individual mean holding times.

Let \( i_s(t) \) denote the number of calls of the \( s \)'th flow served at time \( t \). The model is described by an \( n \)-dimensional Markovian process of the type \( r(t) = \{i_1(t), i_2(t), \ldots, i_n(t)\} \) with state space \( S \) consisting of vectors \((i_1, \ldots, i_n)\), where \( i_s \) is the number of calls of the \( s \)'th flow being served by the link under stationary conditions. The state space \( S \) is defined as follows: \((i_1, \ldots, i_n) \in S, \ s = 1, \ldots, n, \ \sum_{s=1}^n i_s b_s \leq k \). Let us by \( P(i_1, \ldots, i_n) \) denote the unnormalised values of stationary probabilities of \( r(t) \). After normalisation, \( p(i_1, \ldots, i_n) \) denotes the mean proportion of time when exactly \( \{i_1, \ldots, i_n\} \) connections are established. Assume that for state \((i_1, \ldots, i_n)\) the value \( i \) denotes the total number of occupied bandwidth units \( i = i_1 b_1 + \cdots + i_n b_n \).

The process of transmission of \( s \)'th flow is described by blocking probability \( \pi_s, s = 1, \ldots, n \). Their formal definition through values of state probabilities are as follows (here and further, summations are for all states \((i_1, \ldots, i_n) \in S \) satisfying formulated condition, and by small characters we denote the normalised values of state probabilities):

\[
\pi_s = \sum_{i+b_s>k} p(i_1, \ldots, i_n). \tag{17}
\]

There are many algorithms for estimation of \( \pi_s \). All of them are based on product form relations valid for \( P(i_1, \ldots, i_n) \):

\[
P(i_1, \ldots, i_n) = P(0, \ldots, 0) \cdot \frac{\lambda_{i_1}^{i_1}}{i_1!} \cdot \frac{\lambda_{i_2}^{i_2}}{i_2!} \cdots \frac{\lambda_{i_n}^{i_n}}{i_n!}, \quad (i_1, \ldots, i_n) \in S.
\]

The most efficient calculation scheme for the model introduced is the recurrence algorithm first obtained in [6] and later also derived in [7],[8]. This algorithm exploits the fact that the performance measures (17) can be found if we for process \( r(t) \) know probabilities \( p(i) \) of being in the state where exactly \( i \) bandwidth units are occupied:

\[
p(i) = \sum_{i_1 b_1 + \cdots + i_n b_n = i} p(i_1, \ldots, i_n).
\]

The corresponding formulas are as follows:

\[
\pi_s = \sum_{i=k-b_s+1}^k p(i), \quad s = 1, 2, \ldots, n. \tag{18}
\]

The unnormalised values of \( P(i) \) are found by the recurrence:

\[
P(i) = \begin{cases} 
0 & \text{if } i < 0 \\
a & \text{if } i = 0 \\
\left(\frac{1}{a} \sum_{s=1}^n \lambda_s b_s P(i - b_s)\right) & \text{if } i = 1, 2, \ldots, k
\end{cases} \tag{19}
\]
This relation gives specific type of function \( f(\cdot) \) defined by (5) for the model considered. Using general framework formulated in Section 3 we obtain a one–run algorithm that at each step gives the normalised values of state probabilities which are necessary for solving the problem of dimensioning based on (18):

**Step 1.** Let \( p_0(0) = 1 \).

**Step 2.** Let \( b = \max_{0 \leq s \leq n} (b_s) \). For fixed \( k = 1, 2, \ldots \), find normalised value of \( p_k(i) \) using (13):

\[
p_k(k) = \frac{\frac{1}{k} \sum_{s=1}^{n} \lambda_s b_s p_{k-1}(k-b_s)}{1 + \frac{1}{k} \sum_{s=1}^{n} \lambda_s b_s p_{k-1}(k-b_s)};
\]

\[
p_k(i) = \frac{p_{k-1}(i)}{1 + \frac{1}{k} \sum_{s=1}^{n} \lambda_s b_s p_{k-1}(k-b_s)}, \quad i = k-1, k-2, \ldots, \max(k-b, 0)
\]

**Step 3.** Here we calculate the performance measures defined by (18), check the dimensioning criteria and either stop or continue the process of estimating the number of service units needed.

When implementing this version of recurrence algorithm we need to keep a vector of size \( O(b) \) in computer memory. Computational efforts are estimated by \( O((n+b)k) \). Multi-rate state-dependent models are treated in the same way. Due to shortage of place this part of investigations is omitted here.

5 Stable algorithms for some teletraffic models

5.1 Multi-rate Poissonian model with truncations

Let us consider the details of implementation the stable recurrence with truncation according scheme suggested in Section 3. The results will be presented for the one link multi-rate Poissonian model studied in Section 4.

Firstly, in exact way we derive the modified version of recurrence for global state probabilities suggested by relation (15). Let us choose some truncation level \( \ell \) and construct the auxiliary model to approximate the behaviour of the one-link multi-rate model considered in Section 4. The auxiliary model behaves as the initial model in states \( k = \ell, \ell + 1, \ldots, k \), except the cases when transition moves initial model to the states \( (i_1, \ldots, i_n) = (i_1b_1 + \cdots + i_nb_n < \ell \). In auxiliary model we prevent this transition by immediately putting a call of corresponding type into service (type of the call depends on the number of bandwidth units occupied by the call just finishing service). This fictitious call is served the same way as any call in the initial model.

The auxiliary model dynamic is described by a Markov process \( r^*(t) \) with the same components as \( r(t) \) used for description of initial model in Section 4. Process \( r^*(t) \) is defined on state space \( S^* \) that is smaller than \( S \) and consists of vectors \( (i_1, \ldots, i_n) \) with nonnegative components satisfying conditions: \( (i_1, \ldots, i_n) \in S^*, \ell \leq \sum_{s=1}^{n} i_s b_s \leq k \). Let us denote for \( r^*(t) \) by \( P^*(i) \) unnormalised probabilities of being in the global state \( (i) \) when exactly \( i \) bandwidth units are occupied. Probabilities \( P^*(i) \) will be served as estimate for corresponding probabilities \( P(i) \) of initial model.
In the same way as for the initial model we can prove that for the auxiliary model a product form solution is valid for stationary probabilities and corresponding recursion scheme for estimation of unnormalised value of $P^*(i)$

$$P^*(i) = \begin{cases} 
0 & \text{if } i < \ell \\
\alpha & \text{if } i = \ell \\
\frac{1}{i} \sum_{s=1}^{n} \lambda_s b_s P^*(i-b_s) & \text{if } i = \ell + 1, \ell + 2, \ldots, k
\end{cases} \tag{21}$$

As result in this particular case we prove the possibility to use the modification of recurrence (5) in the form suggested by relation (15) when dealing with deriving of stable recursion with truncation of global state space.

Using general framework formulated in Section 3 we obtain a one–run algorithm that gives at each step the normalised values of the model’s characteristics that are necessary for solving the dimensioning problem based on performance measures (18).

**Step 1.** Let $p^*_1(\ell) = 1$.

**Step 2.** Let $b = \max_{0 \leq s \leq n} (b_s)$. For fixed $k = \ell + 1, \ell + 2, \ldots$, find normalised value of $p^*_k(i)$, $i = k, k-1, \ldots, \max(k-b, \ell)$ by using relations (21)

$$p^*_k(k) = \frac{\frac{1}{k} \sum_{s=1}^{n} \lambda_s b_s p^*_{k-1}(k-b_s)}{1 + \frac{1}{k} \sum_{s=1}^{n} \lambda_s b_s p^*_{k-1}(k-b_s)}; \tag{22}$$

$$p^*_k(i) = \frac{p^*_{k-1}(i)}{1 + \frac{1}{k} \sum_{s=1}^{n} \lambda_s b_s p^*_{k-1}(k-b_s)}, \quad i = k-1, k-2, \ldots, \max(k-b, \ell)$$

**Step 3.** The same as formulated for basic version.

To illustrate the computational savings obtained when using stable recurrence with or without truncations of global state space we consider numerical example. In Table 1 relative efforts for three schemes of calculation are compared to reach value of $k = 100, 500, 1000, 5000, 10000$. Other model parameters are $n = 4$, $b_1 = 1$, $b_2 = 2$, $b_3 = 3$, $b_4 = 5$, $\lambda_s = \frac{k}{nb_s}$, $s = 1, 2, 3, 4$. In Table 1 the results presented in the column “Full search” are based on step-by-step usage of recurrence (19), the results presented in the column “Stable recurrence” are based on usage of stable recurrence in the form defined by (20), the results presented in the column “Stable recurrence & truncation” are based on usage of stable recurrence with truncation in the form defined by (22), the results presented in the column “Truncation level” give the value of $\ell$ that is chosen to provide relative error of one percent in estimating of weighted blocking $P$ defined for studied model as $P = \sum_{s=1}^{n} \lambda_s b_s \pi_s / \sum_{s=1}^{n} \lambda_s b_s$. For fixed value of $k$ computational efforts are normalised by efforts needed for usage of stable recurrence in the form defined by (20). The results show considerable savings of computational efforts.

### 5.2 One-link model with batch arrival demands

Let us consider a single link traffic model, where the link transmission capacity is represented by $k$ basic bandwidth units. In the model we have one Poissonian flow of batches
Table 1. Relative values of computational efforts to reach value $k$ for three schemes of calculation normalised by efforts needed for usage of stable recurrence in the form defined by (20)

<table>
<thead>
<tr>
<th>Number of service units $k$</th>
<th>Full search recurrence $k$</th>
<th>Stable recurrence $k$</th>
<th>Stable &amp; truncation $k$</th>
<th>Truncation level $\ell$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>50</td>
<td>1</td>
<td>0.37</td>
<td>63</td>
</tr>
<tr>
<td>500</td>
<td>250</td>
<td>1</td>
<td>0.18</td>
<td>410</td>
</tr>
<tr>
<td>1000</td>
<td>500</td>
<td>1</td>
<td>0.13</td>
<td>870</td>
</tr>
<tr>
<td>5000</td>
<td>2500</td>
<td>1</td>
<td>0.06</td>
<td>4700</td>
</tr>
<tr>
<td>10000</td>
<td>5000</td>
<td>1</td>
<td>0.042</td>
<td>9580</td>
</tr>
</tbody>
</table>

Table 1. Relative values of computational efforts to reach value $k$ for three schemes of calculation normalised by efforts needed for usage of stable recurrence in the form defined by (20)

for connections with intensity $\lambda$. Let us denote by $f_i$ the probability that batch size is $i$. Each call from the batch uses for its transmission only one bandwidth unit for exponentially distributed connection time with parameter equal to one. Let us suppose that index $i$ for $f_i$ varies from $1$ to $b$. If size of the batch exceeds the number of free bandwidth units then available bandwidth units are occupied by calls from the batch and excess calls are lost.

Let $i(t)$ be number of calls being served at time $t$. The dynamics of the model is described by a Markov process $r(t)$ having one component $r(t) = i(t)$. The process $r(t)$ takes values in the finite set of states $S = \{0, 1, \ldots, k\}$ defined in accordance with link capacity. Let us by $P(i)$ denote the unnormalised values of stationary probabilities of $r(t)$. After normalisation, the value $P(i)$ denotes the mean portion of time when exactly $i$ connections are established or $i$ bandwidth units are occupied. From the system of state equations it is easy to derive recurrence for $P(i)$:

$$P(i) = \begin{cases} 
0 & \text{if } i < 0 \\
1 & \text{if } i = 0 \\
\frac{1}{a} \left \{ P(i - 1) \sum_{s=1}^{b} f_s + \ldots + P(i - b) \sum_{s=b}^{b} f_s \right \} & \text{if } i = 1, 2, \ldots, k
\end{cases} \quad (23)$$

As the performance measure for solving dimensioning problem for this model can be taken traffic congestion $\pi_c$ defined as a fraction of offered traffic that is not carried

$$\pi_c = \frac{\lambda \sum_{s=1}^{b} f_s s - \sum_{i=1}^{k} p(i)i}{\lambda \sum_{s=1}^{b} f_s s}. \quad (24)$$

An alternative relation for carried traffic are obtained after summation of (23) over $i$ from 1 to $k$

$$\sum_{i=1}^{k} i p(i) = \lambda (1 - p(k)) \sum_{s=1}^{b} f_s + \lambda (1 - p(k) - p(k - 1)) \sum_{s=2}^{b} f_s + \ldots +$$

$$+ \lambda (1 - p(k) - p(k - 1) - \ldots - p(k - b + 1)) \sum_{s=b}^{b} f_s \quad (25)$$

The recurrence (23) and the definition of performance measure used for link dimensioning (24), (25) satisfy conditions formulated in Section 3. So we can implement a one-run three-step algorithm that gives at each step the normalised values of the model’s characteristics that are necessary for solving the dimensioning problem based on performance measure (24), (25).
6 Conclusions

In this paper we describe the general framework for constructing numerically stable versions of recursive algorithms designed for estimation of performance measures of different teletraffic models. The results obtained are based on ideas from [1],[2] and generalise for multi-rate models the well known recurrence for the Erlang–B formula for $k$ servers with Poissonian input of intensity $\lambda$. The fact that we at each step of recurrence are dealing with normalised values of state probabilities used for estimation of main stationary performance measures is very well suited for solving problems of dimensioning. In addition to being numerically stable, the main positive features of the suggested approach is its simplicity. All we need is just to take recurrence for global state probabilities in the form (5) and put it into relations (13) or it’s modifications (14), (16). Another advantage is the possibility to increase the efficiency of recursive algorithm by truncation of the state space used.

References