END TO END BLOCKING IN TELEPHONE NETWORKS: A NEW ALGORITHM

Jean-Pierre GUERINEAU and Jacques LABETOUILLE

Centre National d'Etudes des Télécommunications
CNET/PAA/ATR 38 Rue du Général Leclerc
92131 Issy Les Moulineaux, FRANCE

ABSTRACT

In this paper, we present a new method for calculating point-to-point traffic stream probabilities within a telephone network. This method takes into account the baulking effects by adding virtual capacities to some of the trunk groups.

The paper firstly reviews other existing methods for calculating point-to-point grades of service and then describes the new algorithm. It concentrates on the differences between the new algorithm and the existing ones and presents a comparison of results.

This new method appears to give good results and is easy to implement. It can be easily introduced in a planning tool for calculating networks in which significant losses are accepted.

1. INTRODUCTION

In the design of telephone networks, planners minimize the cost under constraints of grade of service to obtain alternative routing tables and trunk group sizes. Different parameters have been chosen for use as grade of service criteria such as the loss probability on last choice trunk groups [1], and more recently other methods have been developed taking into account the point-to-point losses [2]-[3].

The method of [1] uses a simple model of the network to compute the loss probability [4] and the algorithm is not time consuming. [4] gives good results when losses are low and this is the case when the planner designs the network under normal conditions. When designing private networks for which high losses have to be accepted, errors become significant and this phenomenon also occurs in the case of link failures and concentration of traffic in one direction.

The method of [2] uses a simplified algorithm to compute the end-to-end loss probability and can be used for large networks. The algorithm of [3] takes into account the blocking probability on a path and not only on a single trunk group, this means that a call blocked on a trunk group of a path has to be removed from the previous trunk groups of its path. This algorithm being based on an iterative method, the computer time is very long which prevents its application to large networks. So several methods have been used to analyse the end-to-end loss probabilities of telephone networks, but their complexity is too high to implement them in a planning tool. We present in this paper, a new algorithm more precise than that described in [2] and which can be easily introduced in a planning tool.

Firstly, we present a brief survey of existing analysis techniques and then we describe our method. Finally, we give results on several networks and we compare them with exact solutions or simulations and other approximate methods.

2. NETWORK MODEL

In most of the methods, some assumptions have been used for network analysis and a certain number of them are the same in all methods.

• Originating traffics are Poisson.
• Call holding times are independent with exponential distribution function.
• Network is in statistical equilibrium.
• Each exchange has full access to outgoing trunks.
• Call set-up times are negligible.
• Blocked calls are cleared and do not repeat attempt.
• The occupancy distributions of trunk groups are independent.

Some methods take into account the dependence of trunk groups [5] but these methods can be used as references to test validity of approximate methods. They are too complex to deal with large networks and we will not discuss them in this paper.

The main differences between traffic models are the following:

• depending on the method, streams of overflow traffic are described by their moments (one, two or first three moments) and can be modelled as renewal processes [6] or with "equivalent random theory" [7].

• streams of carried traffic are modelled as Poisson processes or as smooth traffics [8].

• some of the methods distinguish the individual streams by apportioning the total overflow.
3. PRESENTATION OF METHODS

3.1. The Simplest Model

Firstly, we present the method of [4] which applies Erlangs formula and the equivalent random theory. It does not take into account the fact that a call which is carried on a given link may be blocked further along its path and so it ignores the effects of balking (this assumption is valid when the losses are low). The method is well suited to hierarchical networks in which the list of trunk groups can be ordered and the traffic offered on a given link \((i,1)\) can be obtained easily. Then carried and overflow traffic are computed and offered to corresponding trunk groups. So, the main advantage is that the list of trunk groups is explored only once. The point-to-point loss probability is obtained by scanning the list of trunk groups in reverse order. The main assumption made here is that the trunk groups are independent.

It can be seen that it is not necessary to perform any iteration, so this algorithm is fast and can be applied to large networks. In the case of high losses, errors become significant due to the simplifications in the model. A certain number of models have therefore been developed to improve results, by considering losses on a path and they are discussed below.

3.2. Use of a Reduced Offered Traffic

The various models presented in this section take into account the fact that a call going through a trunk group could be blocked further on the path. In general, these methods consider the traffic offered to a trunk group \((i,k)\) to be of the form

\[
t = \sum_{j \in \xi} t_{ij} - \tau
\]

where \(\xi\) is a set of all destination nodes to which calls are sent from \(i\) via \(k\) and \(\tau\) is a traffic corresponding to the blocked traffic on the rest of the paths in which \((i,k)\) is included. The methods differ mainly in the way they compute \(\tau\) and also in the number of moments taken into account.

Katz [9] was the first to introduce this method. To represent routing schemes, a route tree generation (see example in Fig. 1) is build.

\[
\text{Fig. 1. Example of route tree generation}
\]

In the paper, a load assignment procedure is considered where the mean and variance of offered parcel are adjusted. An adjustment factor is required to account for the increased availability of links seized by calls which fail to reach their destination. The effect of the adjustment is to reduce the offered loads to effective values from which link blocking probabilities may be computed. This adjustment factor is obtained empirically from simulation results on a certain number of networks. Different formulae are employed for the link overflow parameter calculations, depending on the variance to mean ratio of the link offered load. An iteration procedure is performed to obtain point-to-point loss probabilities in which the initial values of these probabilities are set to zero and in each subsequent iteration, calculations are performed to obtain improved estimates of the traffic parameters. These calculations use the network routing plan and the traffic parameters computed in the previous iteration. The main disadvantage of this method is the use of a correction factor which is obtained empirically from simulation results.

The paper [10], using similar criteria, proposes analytic procedure to obtain the adjustment factor. In the paper two philosophies are considered for calculating the traffic offered to a group. In the first one, the traffic offered to a trunk group is the traffic carried by the route plus the traffic lost because of the trunk group congestion, this is the hypothesis of Katz' paper. In the second one, the traffic offered to the trunk group is the traffic carried by the groups of the path except the group under examination. By comparison with simulations, it is shown that the first philosophy generally overestimates losses and the second one underestimates them.

Two papers are presented by Manfield and Downs [11]-[12] in which the main difference lies in the number of moments used, in the first one, one moment is used and in the second three moments are used to describe the overflow traffic, carried traffic is modelled as Poisson processes. A method of [13] is used to represent uniquely the routing scheme for a network with a given route plan and arbitrary form of route control. Imaginary loss nodes are introduced in the route tree in order to distinguish the ways in which calls are lost in the system. Fig. 2 shows an example.

\[
\text{Fig. 2. Example of augmented route tree for step-by-step route control}
\]
The paper considers a general link model as a set of limited availability servers. Considering just the case of a single traffic stream offered to a link, a call which arrives and finds a free server is rejected with a constant baulking probability $b$ according to an independent Bernoulli trial. The offered traffic is decomposed into two processes, one corresponding to a reduced offered traffic $a^* = a(1-b)$, where the other of intensity $a^*$ is not accepted on the link. The algorithm is decomposed into two phases. In the first, assuming that the link blocking probabilities and the overflow characteristics of each link are known, the link offered traffics are derived. An important point is that the carried traffic of each link due to one stream of path offered traffic is numerically the same. The second phase consists of link traffic segregation. With the one moment method, results are easy to obtain, but with a three moment method, the carried and overflow traffic belonging to each individual offered stream have to be determined, each stream being modelled as a renewal process by a moment match technique.

3.3. Use of a Fictitious Offered Traffic

In the paper of F. Le Gall and J. Bernussou [14], a one moment model is considered which incorporates the definition of fictitious offered traffic which enables one to take into account the deviation of smooth and peaked traffic from the Poisson. A differential equation can be written for the mean carried traffic in transient state and it can be viewed as a fluid equation. An approximation is made to define a fictitious instantaneous traffic, so that the Erlang formula can be applied even in the transient state. A generalisation is made for networks which require the knowledge of network structure and the knowledge of the routing scheme at each node. A numerical integration of the differential equations until the steady state is reach can be performed. It may be preferable to resolve directly the system for the steady state using relaxation techniques.

4. PRESENTATION OF OUR METHOD

4.1. Aim Objective

When loads are high, the method of [4] is to simple and errors become significant. We can see what happens on a simple example shown in Fig.3.

![Fig.3. Simple example](image_url)

We consider a traffic stream which uses links $(i,k)$ and $(k,j)$. In a simple model like one presented in [4], a call carried on link $(i,k)$ and blocked on $(k,j)$ will still be considered as carried by $(i,k)$. So, it can be deduced that $p_{a_e}$ trunks are occupied on $(i,k)$ for calls which were rejected by $(k,j)$, $(p_e$, being the blocking probability of trunk group $(k,j)$ and $a_e$ the carried traffic on link $(i,k)$ which is offered to $(k,j)$). So the blocking probability on $(i,k)$ is certainly overestimated by this simple model.

In our method, we propose to take account of calls blocked on a later part of the path. This is done by considering that these calls occupy fictitious trunks on $(i,k)$, so we increase the size of the trunk group by $x$ trunks where $x=p_{a_e}a_e$. In computing the blocking probabilities, we will have to consider non-integer number of trunks, so adapted formulae should be used.

4.2. Description of the Method

- **Model of network.**

  Consider the assumptions presented in paragraph 2 in which traffic streams are represented by the first two moments. Overflow traffic streams are computed using equivalent random theory and the superposition of carried traffic streams is modelled as a Poisson process. The total offered traffic to a link is the superposition of a Poisson process, and if this link receives overflow traffic, of a renewal process, so using the method of [15], we can decompose the different traffic streams to obtain individual loss probabilities.

- **Description of the algorithm.**

  The algorithm is divided in two phases:

  1. Knowing the sizes of links, point-to-point losses are computed using the method developed in [4].

  2. Knowing efficiency rates, additional virtual values are computed.

  Let us consider a given trunk group $(i,k)$ noted $m : e$ represents the calculated carried traffic on $m$. A part of this traffic will be blocked further downstream, and let $x_m$ be this lost traffic. We will increase the size of trunk group $m$ by this same value $x_m$ as in the simple example of Fig.3.

  Let $x$ be the set of streams offered to link $m$.

  Let $L \subseteq$ be the set of links occupied by the stream $\tau$, downstream of $k$.

  Let $p_{n,\tau}$ be the loss probability of the stream $\tau$ on link $n$.

  Let $t_{n,\tau}$ be the offered traffic of the stream $\tau$ on link $n$.

  We have:

  \[ x = \sum_{\tau \in L \cap \mathbb{N}} (\sum_{p \in T} p \cdot t) \quad (1) \]

  Let $e_{k,\tau}$ be the efficiency rate for each pair of nodes and for each stream.
carried traffic of stream \( \tau \) from \( k \) to destination of \( \tau \)

\[
e_k,\tau = \frac{\text{offered traffic of stream } \tau \text{ from } k \text{ to destination of } \tau}{m,\tau}
\]

We also have:

\[
x = \sum_{m,\tau} \left( \sum_{k,\tau} p,\tau \right) \cdot \left( 1-e,\tau \right) \cdot t,\tau \cdot m,\tau
\]

(2)

It can be seen that when computing losses on a given link, we have to store in memory the offered traffic and the lost traffic corresponding to each stream. In the same way, when computing the point-to-point loss probabilities, we have to store the efficiency rate for each stream.

If we use exactly the same model as in [4] without apportioning the individual traffic streams, the above formulae become simpler: losses and efficiency rates do not depend upon individual streams, so we have:

\[
x = \sum_{m,\tau} \left( \sum_{n,\tau} p,\tau \right) \cdot \left( 1-e,\tau \right) \cdot t,\tau \cdot m,\tau
\]

(3)

\[
x = \sum_{m,\tau} \left( \sum_{k,\tau} (1-p,\tau) \cdot (1-e,\tau) \right) \cdot t,\tau
\]

(4)

In (4), \( e_k \) depends on the origin \( k \) and the extremity of the stream, but does not depend on the stream itself.

We have supposed that the network can be ordered. So by scanning the list of links only once, we obtain loss probabilities on each link. In the same way, scanning the list in reverse order we can compute the efficiency rate for each stream and for each node. Then virtual additional sizes are computed for each link, using equation (2). The algorithm then iterates and stops when the differences between loss probabilities of two consecutive iterations are smaller than a given value.

We have checked the convergence of the method for different networks and under several operating conditions: normal load, small overloads, high overloads. Convergence is obtained generally within two or three iterations. To check the validity of the method, we have changed the starting conditions of the iteration by considering some non-zero, initial values of \( x_i \). In every example, the method converges on the same solution, so we can assume that most of networks could be processed with our method.

Finally, the method can be summarized in the flowchart shown in Fig 4.

5. RESULTS AND COMPARISONS

In this paragraph, we present some results obtained on different networks. First, we compare our method with results presented by Manfield and Downs [12] on the same network and we discuss the differences obtained. Then, we give results on a small network with different loads and we compare results with those obtained by a simulation method. Finally, we test the algorithm on a network composed of links in series and we compare with exact results; the main interest here is to show where the results differ from those of the exact model and to analyse the results to forecast the types of errors due to the method.
The number of trunks in each link is indicated adjacent to the link. A and E are origin exchanges, B is the unique destination exchange, C and D are tandem exchanges. It is assumed that there is also external traffic arriving at C and D via B. This traffic represents traffic coming from other parts of the overall network. We assume this traffic is Poisson since it must already have been carried on other links. The alternative routes from the various exchanges are indicated with arrows (an arrow indicates the overflow).

From A to B, we have two overflow links: respectively A-C and A-D. From D to B, we have one overflow link D-C and from E to B, one overflow link E-D.

We compare our method with Manfield and Downs [12], for normal condition in Table 1, and for overload condition in Table II.

<table>
<thead>
<tr>
<th>Stream</th>
<th>Offered Traffic</th>
<th>G.O.S (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Proposed Method</td>
<td>Method of [12]</td>
</tr>
<tr>
<td></td>
<td>1st Step</td>
<td>Final Results</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-B</td>
<td>20</td>
<td>3.9</td>
</tr>
<tr>
<td>C-B</td>
<td>8</td>
<td>7.1</td>
</tr>
<tr>
<td>D-B</td>
<td>6</td>
<td>5.2</td>
</tr>
<tr>
<td>E-B</td>
<td>16</td>
<td>5.2</td>
</tr>
</tbody>
</table>

Table I - Example 1 - Normal condition

<table>
<thead>
<tr>
<th>Stream</th>
<th>Offered Traffic</th>
<th>G.O.S (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-B</td>
<td>24</td>
<td>13</td>
</tr>
<tr>
<td>C-B</td>
<td>8</td>
<td>14.6</td>
</tr>
<tr>
<td>D-B</td>
<td>8</td>
<td>23</td>
</tr>
<tr>
<td>E-B</td>
<td>20</td>
<td>18.6</td>
</tr>
</tbody>
</table>

Table II - Example 2 - Overload condition

In example 1, we obtain the same results as Manfield and Downs. With our method we obtain final results at second iteration. Compared with simulation results, we have a good precision, except for traffic stream D-B where the relative error on the traffic lost is of 25%. It can be seen that only one extra iteration improves the results of [4], which correspond to the first iteration. In example 2, it can be seen that errors are of the same order of magnitude as in Manfield and Downs, and compared with simulations results the maximum relative error, obtained on stream C-B, has a value of 14%. The final result is obtained with 3 iterations and compared to [4], the precision is greatly increased. If we consider the total traffic lost in the network, the grade of service is 14.1% with their method. So, for this example, our method overestimates the total grade of service, the method of [12] underestimates it. In this example, it was not necessary to split into individual streams because on a given link, we have either a Poisson process or an overflow process but not both.

5.2 Second Network

We will present here results on a small network with a mixing of Poisson and overflow traffic on some links, so it will be possible to check the improvement due to the splitting of streams. The network is represented on Fig 6.

Fig 6. Second Network

Traffic are offered from origin exchanges 1-2-3 to exchange 4. Moreover, two streams are offered from nodes 2 and 3 to the outside of the network, the loss probabilities of these streams on the outside are respectively $q_1$ and $q_2$. The alternative routes form the various exchanges are indicated with arrows. We have checked our method with different values of traffic and different sizes of trunk groups. The method has been performed with and without splitting of streams to evaluate the improvement due to the splitting. As reference, a simulation method of QNAP [16] has been used. For example 3 (see Table III), we have compared our method with that of [14]. For examples 3, 4, 5, we have $q_1=q_2=0.1$.

<table>
<thead>
<tr>
<th>Size of links</th>
<th>Stream</th>
<th>Offered Traffic</th>
<th>G.O.S (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4:4</td>
<td>1-4</td>
<td>5</td>
<td>15.2</td>
</tr>
<tr>
<td>1-6:3</td>
<td>2-4</td>
<td>6</td>
<td>15.1</td>
</tr>
<tr>
<td>6-4:4</td>
<td>3-4</td>
<td>4</td>
<td>28.5</td>
</tr>
<tr>
<td>2-4:5</td>
<td>2-out</td>
<td>4</td>
<td>29.5</td>
</tr>
<tr>
<td>2-5:6</td>
<td>side</td>
<td>3</td>
<td>27.6</td>
</tr>
<tr>
<td>5-4:4</td>
<td>3-out</td>
<td>3</td>
<td>27.6</td>
</tr>
<tr>
<td>5-6:3</td>
<td>side</td>
<td>3</td>
<td>27.6</td>
</tr>
<tr>
<td>5-5:8</td>
<td>side</td>
<td>3</td>
<td>27.6</td>
</tr>
</tbody>
</table>

Table III - Example 3 - G.O.S. results
Results for example 4 are shown in table IV.

<table>
<thead>
<tr>
<th>Size of links</th>
<th>Stream</th>
<th>Offered Traffic</th>
<th>G.O.S (%)</th>
<th>QNAP</th>
<th>Our method without splitting</th>
<th>Our method with splitting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4:5</td>
<td>1-4</td>
<td>5</td>
<td>10.6</td>
<td>11</td>
<td>10.9</td>
<td></td>
</tr>
<tr>
<td>1-6:3</td>
<td>2-4</td>
<td>4</td>
<td>9.3</td>
<td>7.8</td>
<td>9.1</td>
<td></td>
</tr>
<tr>
<td>6-4:3</td>
<td>3-4</td>
<td>4</td>
<td>38</td>
<td>41.1</td>
<td>40.7</td>
<td></td>
</tr>
<tr>
<td>2-4:5</td>
<td>2-out</td>
<td>4</td>
<td>24.4</td>
<td>31.6</td>
<td>30.2</td>
<td></td>
</tr>
<tr>
<td>5-4:3</td>
<td>3-out</td>
<td>3</td>
<td>29</td>
<td>33.7</td>
<td>33.9</td>
<td></td>
</tr>
<tr>
<td>5-6:2</td>
<td>side</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-5:6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table IV - Example 4 - G.O.S results

In example 5, we consider the same network, and we add two streams representing traffics coming from the outside of the network: a stream of 5 to 4 carried on links 5-6 and 6-4; a link from 6 to 4 carried on link 6-4.

<table>
<thead>
<tr>
<th>Size of links</th>
<th>Stream</th>
<th>Offered Traffic</th>
<th>G.O.S (%)</th>
<th>QNAP</th>
<th>Our method without splitting</th>
<th>Our method with splitting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4:4</td>
<td>1-4</td>
<td>5</td>
<td>21.6</td>
<td>21.1</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>1-6:3</td>
<td>2-4</td>
<td>6</td>
<td>16.5</td>
<td>14.9</td>
<td>16.5</td>
<td></td>
</tr>
<tr>
<td>6-4:3</td>
<td>3-4</td>
<td>4</td>
<td>33.2</td>
<td>36.2</td>
<td>36.3</td>
<td></td>
</tr>
<tr>
<td>2-4:5</td>
<td>5-4</td>
<td>1.2</td>
<td>48</td>
<td>56.7</td>
<td>52.4</td>
<td></td>
</tr>
<tr>
<td>2-5:6</td>
<td>6-4</td>
<td>1.2</td>
<td>43.5</td>
<td>39.9</td>
<td>39.5</td>
<td></td>
</tr>
<tr>
<td>5-4:4</td>
<td>2-out</td>
<td>4</td>
<td>27.8</td>
<td>32</td>
<td>29.3</td>
<td></td>
</tr>
<tr>
<td>5-6:3</td>
<td>side</td>
<td>3</td>
<td>25.6</td>
<td>26.2</td>
<td>26.2</td>
<td></td>
</tr>
<tr>
<td>3-5:7</td>
<td>3-out</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table V - Example 5 - G.O.S results

The results obtained for tables III, IV, V show a good precision compared to exact results of QNAP. Splitting of streams leads to some improvements of G.O.S, but even without splitting we obtain correct results. The convergence is obtained after 2 or 3 iterations. Also for this example, we decrease error by 50% from the first iteration to the second or third iteration.

5.3. Links in series

We consider the system composed of two links in series of Fig. 7. We have three traffic streams:

- a stream 1 carried on links $N_1$ and $N_2$ of intensity $a_{12}$
- a stream 2 carried on link $N_1$ of intensity $a_1$
- a stream 3 carried on link $N_2$ of intensity $a_2$

In this example when losses are high, effects due to the dependence of links become significant and we have checked our method to analyse the errors occurring on the G.O.S. Our results have been compared with exact values obtained from QNAP. $N_1$ and $N_2$ have been set to small values to compute the Markov system but conclusions can be generalized.

In table VI, we summarize results obtained for example 6, with $N_1=N_2=3$.

<table>
<thead>
<tr>
<th>$a_{12}$</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>Flow</th>
<th>Flow</th>
<th>Flow</th>
<th>G.O.S (%)</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 0 1</td>
<td>1</td>
<td>0</td>
<td>45</td>
<td>-</td>
<td>-</td>
<td>51.9</td>
<td>38.6</td>
</tr>
<tr>
<td>3 1 0</td>
<td>45</td>
<td>45</td>
<td>-</td>
<td>45</td>
<td>51</td>
<td>40.3</td>
<td></td>
</tr>
<tr>
<td>3 0 0</td>
<td>34.6</td>
<td>-</td>
<td>-</td>
<td>34.6</td>
<td>-</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>3 0.5 0.5</td>
<td>45</td>
<td>33.6</td>
<td>33.6</td>
<td>51.2</td>
<td>31.3</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>1.5 0 0.5</td>
<td>21</td>
<td>20.8</td>
<td>-</td>
<td>26.7</td>
<td>19.4</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>1.5 0 0.25</td>
<td>13.4</td>
<td>-</td>
<td>-</td>
<td>21.2</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>1.5 0.25 0.25</td>
<td>20.5 17.8</td>
<td>17.8</td>
<td>26.7</td>
<td>14.9</td>
<td>13.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table VI - G.O.S - results for links in series

It can be seen that when $a_1=a_2=0.5$, the loss probability of stream 1 is overestimated and the losses on streams 2 and 3 are underestimated. In the case of smaller load, we obtain the same kind of results.
The above considerations can enable us to forecast the types of errors in networks. In table V, example 5, we have obtained results which could be forecasted by the above considerations: stream 5-4 is overestimated and stream 6-4 is underestimated, which is consistent with results obtained in table VI.

Using the method of [8] to model smooth traffics, we have applied our method to examples of table VI. We have not found a significant improvement in every case and so, we have decided to keep the original model.

We have also applied our method to examples given in [5] in which to compare the robustness of our method when traffics and number of trunks are modified and we have obtained roughly the same results as Katz [9] and Manfield and Downs [12].

6. CONCLUSION

The method presented in this paper can be applied to hierarchical networks or to networks in which trunk groups can be ordered, with a step-by-step routing control.

Comparisons with other methods show an equivalent precision over a wide range of network types. The use of virtual capacities to take into account the baulking effects is easy to implement and leads to a method which is robust and converges rapidly. An important advantage is the possibility to introduce this algorithm for calculation of point-to-point loss probabilities in a planning tool; the computing time is doubled for the calculation of the grade of service but the improvement is important and can justify the use of the method.

REFERENCES