IMPROVED TRUNK ENGINEERING ALGORITHM FOR HIGH-BLOCKING HIERARCHICAL NETWORKS

R. N. RAO
Bell Telephone Laboratories, Inc., Holmdel, New Jersey, U. S. A.

ABSTRACT
In high-blocking networks, the increased efficiencies of the trunk groups necessitate a more accurate evaluation of the economics of alternate routing. This paper presents a new algorithm that high-blocking networks which computes these increased efficiencies and evaluates configurations where traffic on some groups is not automatically alternate routed. Use of this algorithm has resulted in trunking cost savings of 5 to 10 percent.

INTRODUCTION
M. Eisenberg and W. Elsner (1,7) have shown that multihour engineering provides significant savings by taking advantage of the noncoincidences in the busy hours of the switch-to-switch traffic parcels. This paper describes an extension of their algorithms to include high-blocking networks. Studies at Bell Laboratories by R. Dayem and myself indicated that further savings could be obtained by more accurate evaluation of high-blocking effects on trunk group efficiencies and alternate route economies. In our study the high-blocking objective is around 10 percent blocking on the final trunk groups as compared to 1 percent blocking on the U.S. Public Telephone Network. Therefore under the higher blocking objective, direct groups can become so efficient that, even with moderate loads, it may be more economical to full group. On the other hand, it may be more economical to alternate route a large percentage of traffic because of the increased efficiencies of the groups in the alternate route. The problem then is to economically size the interswitch trunks in a two-level hierarchical network based on multihour load sets to meet a given switch-to-switch blocking objective. The algorithm described here, which has been designed for corporate networks with high-blocking objectives, addresses this problem.

It evaluates the economics of full grouping and alternate routes traffic only from those groups that prove uneconomical to full group. Also, the algorithm computes the marginal capacity of every group in the alternate route, as a function of the mean and the variance of the offered load and the design blocking probability. This function is derived from the relationships developed by R. I. Wilkinson (2) and is given in Appendix A. The increase in the marginal capacity because of a high-blocking objective can be significant. For example, a trunk group of 20 circuits at 1 percent blocking has a marginal capacity of about 27 CCS (0.75 Erlang) but at 10 percent blocking, the trunk group has a marginal capacity near 36 CCS (1 Erlang), assuming a peakedness of 1.5.

The algorithm is an extension of the multihour trunk engineering concepts developed by M. Eisenberg (1) to apply to high-blocking networks. The Eisenberg algorithm partitions the high-usages groups in a network into several sets called subclusters. It then, applies a mathematical model for economic alternate routing to each subcluster to determine the sizes of the groups in it. After all the subclusters are considered, the final groups are sized for the maximum of the resultant hourly offered loads.

The new algorithm takes a similar approach to Eisenberg but partitions the high-use groups into just two sets and uses a more accurate mathematical model to size the groups in each of the sets. The new model uses computed values for the marginal capacity of the groups in the alternate route. Also during the search for the minimum cost solution, the new algorithm computes the resultant busy hour load for every group in the alternate route. On the other hand, both the Eisenberg (1) and the Elsner (7) algorithms use a given constant value for the marginal capacities of all the groups in the alternate route. These algorithms assume the resultant busy hour for all but one group in the alternate route. Furthermore, they assume that it is economical to alternate route the traffic on every high-use group.

The new algorithm, however, evaluates the alternative of full grouping for those high-use groups that do not indicate potential savings from full grouping. The Eisenberg-Elsner algorithms obtain the final solution by a single pass through all the subclusters. The new algorithm repeats the procedure as many times as necessary to accurately consider the mutual dependencies of the two subproblems arising from the interactions of the traffic parcels offered to the two sets. Thus, the new algorithm obtains a sequence of solutions improving in cost until convergence is established. It must be pointed out that the Eisenberg-Elsner algorithms were developed specifically for low-blocking, large networks, such as the U. S. Public Telephone Network. Many of the modifications made in the new algorithm may not be significant in those networks.

The algorithm presented in this paper has been implemented in the network synthesis module of the Enhanced Network Administration System (3), which is a set of mechanized tools developed at Bell Laboratories for assisting A.T.&T. Long Lines in administering new corporate network services offered by the Bell System. The use

* A high-use groups in a hierarchical network connects a lower level switch to a switch that is not the home (upper level) switch for the lower level switch (Figure 1).

† A final group connects either two upper level switches or a lower level switch to its home switch (Figure 1).
of the algorithm has resulted in a 5- to 10-
percent saving in trunking costs compared with
that of the Elsner algorithm. The new algo-
rithm produces these savings through increased
network utilization and a more equitable dis-
tribution of overall switch-to-switch blocking.

In the next section we discuss the trunk
engineering problem and provide an overview of
the algorithm. We, then, describe the model as
it relates to the economic advantages of full
grouping and alternate routing. Application of
the algorithm to two corporate networks is then
discussed. And, finally, some improvements to
the algorithm are described.

THE PROBLEM AND AN OVERVIEW OF THE ALGORITHM

The design problem is to size the interswitch
trunk groups in a given 2-level hierarchical
network to minimize the cost of the interswitch
trunks. We assume that the cost of a trunk
group is the product of its size and unit cost
and that all the unit costs are given. The cost
of switching is assumed to be included in
the unit cost of the trunk groups. The cost
minimization is subject to the constraint that
the switch-to-switch traffic offered to them
is not to exceed a given design objective value.

If the 2-level hierarchy of the switches is
specified, a sequence of permissible alternate
routes for each switch-to-switch traffic parcel
can be determined. Figure 1 shows a 2-level
hierarchical network with 2 upper level and 3
lower level switches. The figure also shows the
sequence of alternate routes for some traffic parcels. Figure 1 also shows the vari-
ous terms used in this paper to describe the
group trunk in a 2-level hierarchical network.

LEGEND:  
\[ \Delta \] UPPER LEVEL SWITCH 
\[ \bigcirc \] LOWER LEVEL SWITCH 
\[ \rightarrow \] INTERMEDIATE HIGH-
USAGE GROUP 
\[ \rightarrow \] PRIMARY INTERREGION
HIGH-USAGE GROUP 
\[ \rightarrow \] PRIMARY INTRAREGION
HIGH-USAGE GROUP

Table: Alternate Routing Table

<table>
<thead>
<tr>
<th>TRUNK GROUP</th>
<th>COST PER UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, B, C, D</td>
<td>3</td>
</tr>
<tr>
<td>E, F, G, H</td>
<td>4</td>
</tr>
</tbody>
</table>

The computation of the overall switch-to-switch
blocking probability for a traffic parcel in a
hierarchical network is quite complex. There-
fore it becomes impractical to enforce an upper
bound on this blocking probability in an iter-
ative design procedure. Instead, we impose a
constraint on the blocking probabilities on the
final trunk groups. Since a path consisting of
final trunk groups forms the last-choice route
for all traffic parcels, the new constraint is
related to the original one. The use of this
blocking-on-finals criterion reduces the com-
plexities of the design problem considerably.
It also provides a basis for comparing the other
groups, such as the Eisenberg-Elsner
algorithms, which use such a criterion. Use of
the blocking-on-finals criterion does require a
service evaluation capability, such as S. K.
Katz's evaluator (4) at the end of the design
process. This capability ensures that the ori-
ginal blocking objectives are met.

The algorithm examines solutions iteratively,
reducing the cost at each step until no further
changes appear to be economical. At the end of
each iteration, the marginal costs of the back-
ground loads are computed for each trunk
group. These figures, then, are used to guide
the next iteration. To start the iteration
process, the algorithm obtains an initial
solution by using a simple multihour version of
the ECCS method (5).

The algorithm examines the high-usage trunk
groups in two stages. In the first stage, all
primary interregion high-usage groups are con-
sidered. These groups have two alternate
groups for service offering ordered to them.

The sizing of groups in both stages of the
algorithm consists of determining which groups
are to be full grouped and minimizing a non-
linear cost function to determine the sizes of
the other groups. We use a multihour version
of a criterion developed at Bell Laboratories
to determine candidates for full grouping.
However, since the criterion is empirical, we
systematically evaluate the economics of full
grouping for each of the candidates and full
group a candidate only if it proves economical
to do so. This evaluation consists of deter-
mining the estimated total trunking costs with
and without full grouping the candidate. The
mathematical model for economic alternate rout-
ing is used for determining the estimated
trunking costs of high-usage groups that are
not full grouped.

At the end of the second stage, we have a new
set of sizes for all the high-usage groups.
These sizes are used to determine the resultant
hourly offered loads on the finals. The finals
are sized for the maximum of the hourly offered
loads. The cost of the new complete solution
is compared to the cost of the solution at the
beginning of the iteration to determine if
further iterations are indicated.

In every iteration, the solution obtained at
the end of the previous iteration (for the initial
solution) is used to determine the mar-
ginal capacities of the alternate routes. Also,
the previous solution is used to compute the
background loading on finals and the
intermediate high-usage groups. These back-
ground loads are used in the model. Thus, if
the solution obtained at the end of the itera-
tion is the same as the previous solution, an
additional iteration will result in the same
solution since all the parameters in the model
have remained unchanged. The algorithm stops
when an iteration results in no additional sav-
ings in trunking costs.

ITC-9

RAO-2
CRITERION FOR FULL GROUPING

R. Dayem at Bell Laboratories developed an empirical criterion for determining whether a high-usage group should be full grouped. His analysis indicated that if it is economical to full group a high-usage group when \( A \), the offered load in Erlangs, exceeds \( 0.4P(R-1) \) where \( P \) is the design blocking on the final trunk group and \( R \) is the ratio of the cost of a path on the alternate route to the cost of a circuit on the high-usage group. It is assumed here that the alternate route, although it is made up of two or more trunk groups, has a single significant busy hour and that \( A \) is the load in that hour. We seek a multihour version of the above criterion to assist the algorithm in identifying a subset of high-usage groups as potential candidates for full-grouping.

For simplicity, assume that the alternate route consists of two trunk groups with busy hours \( t_1 \) and \( t_2 \). Let \( \Lambda(t_1) \) and \( \Lambda(t_2) \) be the load offered to the high-usage group during the hours \( t_1 \) and \( t_2 \), respectively. Then one version of the above criterion would be to use for \( A \) the minimum of \( \Lambda(t_1) \) and \( \Lambda(t_2) \). Let min-test denote this criterion. A less conservative version for \( A \) would be the maximum of \( \Lambda(t_1) \) and \( \Lambda(t_2) \). Let average test denote this version. Clearly, if a trunk group passes the min-test, it must pass the average test. The procedure used in the two stages for evaluating the savings of full-grouping is described in the following paragraphs.

For every high-usage group, we determine whether it passes the min-test, or the average-test, or neither. Based on the results of the tests, we partition the high-usage groups into three sets: \( S_1 \), \( S_2 \), and \( S_3 \). A high-usage group is put in set \( S_1 \) if it passes the min-test, in set \( S_2 \) if it passes only the average test, and in set \( S_3 \) if it passes neither test. First, we use the mathematical model of economic alternate routing to determine the least cost, \( C_1 \), of economically alternate routing the traffic on all the high-usage groups. Next, we determine the least cost, \( C_2 \), of alternate routing the traffic on the high-usage groups in set \( S_2 \) and \( S_3 \). The groups in \( S_1 \) are full grouped during the determination of \( C_2 \).

Let \( F_1 \) denote the cost of full grouping the groups in \( S_1 \). If \( C_1 > C_2 + F_1 \), we conclude that it is economical to full group all the groups in \( S_1 \) and we proceed to evaluate individually the groups in \( S_2 \) for full grouping. If \( C_1 \leq C_2 + F_1 \), we conclude that the groups in \( S_1 \) need to be evaluated individually for full grouping, and we move the groups in \( S_1 \) to the second set \( S_2 \). Next we evaluate the groups in \( S_2 \) individually. That is, we full group a group in \( S_2 \) and determine the cost of economically alternate routing the other groups in \( S_2 \) and in \( S_1 \). If the total estimated cost shows a reduction, we move this group to set \( S_1 \) and consider it economical to full group. Otherwise, that group is moved to set \( S_3 \). In this manner all the groups in \( S_2 \) are examined and moved to either \( S_1 \) or \( S_3 \). Thus, at the end of this process \( S_1 \) contains the groups that have proved economical to full group, and \( S_2 \) is an empty set. The sizes of the groups in \( S_3 \) have been determined by the optimization model.

THE MATHEMATICAL MODEL FOR ECONOMIC ALTERNATE ROUTING

Let \( N = \{1,2,\ldots,n\} \) denote a given set of high-usage groups that needs to be sized for economic alternate routing. Let \( M = \{1,2,\ldots,m\} \) denote the set of high-usage groups that we wish to design alternate routes to the traffic offered from the groups in \( N \) can be offered. In stage 1, the set \( M \) consists of finals and intermediate high-usage groups and set \( N \) consists of primary interregion high-usage groups. In stage 2, the set \( N \) consists of intermediate high-usage groups and intraregion high-usage groups and the set \( M \) consists of finals.

While sizing the groups in \( N \), we assume that the groups in \( M \) are like finals, i.e., the traffic offered to them is not alternate routed. This assumption is also made in the Eisenberg-Elsner algorithms and is not strictly correct since the intermediate high-usage groups do overflow in turn to finals. We will return to this aspect in the section on "Further Improvements."

Let \( L(i) \) and \( H(i) \) denote the numbers of the switches connected by the ith high-usage group in the set \( M \) and set \( N \) of usage groups, respectively. Let \( a_i(h) \) and \( a_i(OV) \) denote the offered load on the ith high-usage group from \( L(i) \) to \( H(i) \), and from \( H(i) \) to \( L(i) \), respectively, in hour \( h \) for \( h = 1,2,\ldots,NH \). Let \( \delta(h) = a_i(h) + a_i(OV) \). Let \( C_i, X_i \) be the unit cost and the size (to be determined) of the ith high-usage group trunk.

Let \( B(x,a) \) denote the blocking function for a trunk group of size \( x \) with an offered load of an Erlang. This is the Erlang B function if the load is Poisson distributed; otherwise it denotes the blocking computed by the Equivalent Random method (6), assuming the variance of the load is known. Let \( A_i(h), \ j = 1,2,\ldots,m \) denote the background load on the \( j \)th trunk group in \( N \) in hour \( h \). The background load is the load excluding the overflow from the groups in \( N \).

Let \( d_j \) be the unit cost of the \( j \)th trunk group in \( M \). Let \( y_j \) be the marginal capacity of the \( j \)th trunk group in \( M \) computed using the previous solution and the function given in Appendix A. Let \( W_j = 1 \) if the \( j \)th group in \( M \) is part of the first alternate route for the traffic from \( L(i) \) to \( H(i) \) for \( i \in N \). Let \( \gamma_j \) be defined for \( H(i) \) to \( L(i) \) traffic.

The estimated cost of serving the traffic on the groups in \( N \) can be written as follows:

$$\text{Total Estimated Cost} = \sum_{i=1}^{n} C_i X_i + \sum_{j=1}^{m} d_j \text{Max} \left\{ A_j(h) + OV_j(h) \right\} \gamma_j$$

where

$$OV_j(h) = \sum_{i=1}^{NH} \left\{ a_{i1}(h)W_{ij}^{(1)} + a_{i2}(h)W_{ij}^{(2)} \right\}$$

Note that \( \sum_{j=1}^{m} C_i X_i \) is the cost of having

$$X_1,X_2,\ldots,X_n \text{ trunks, and}$$

is the estimated cost of the \( j \)th trunk group in \( M \). It is assumed that this trunk group will have to be designed to provide no more than the design blocking in all hours. Furthermore, it is assumed that the size of the group trunk is proportional to the mean of the maximum total load. The linear proportionality constant is \( \lambda_j \), which is computed using the design blocking objective and the mean and the variance of the maximum total load offered to the group in the previous solution. The cost function \( (1) \) is similar to the one used in the Eisenberg-Elsner algorithms except for \( W_j, W_i \) and \( y_j \)'s. The objective is to determine nonnegative integral
values for $x_1, x_2, \ldots, x_n$ such that the total estimated cost is a minimum.

W. Elsner (7) pointed out that the cost function in [1] is a convex function of the $x_i$'s if they are considered as continuous variables. However, the function is not differentiable at those values of the $x_i$'s where the maximum in [1] occurs for two or more hours. The method proposed by Elsner was a steepest-descent gradient method and required computation of the direction of steepest descent at every step. Since the function is not differentiable, finding the steepest descent direction at a point where three or more hours determine the maximum in [1] for one or more values of $i$ is itself a nonlinear programming problem.

The number of terms containing the maximum function, namely $m$, in [1] determines to a large extent the difficulty in solving the problem. In the implementation of Eisenberg-Elsner algorithms, $m$ was taken as 1. For $m > 1$ the steepest descent technique becomes an increasingly difficult method to use. We resort to a coordinate search method, i.e., successive one-dimensional searches are conducted until no further improvements are possible. The drawback with this method is that the solution obtained may not be the optimal solution. But an iterative approach should reduce the probability of converging to a nonoptimal point.

The above model is used numerous times in the new algorithm in both stages. In the stage 2 optimization, the model can be simplified since there is no need to consider the directionality of the offered traffic. This is because the overflow routes in both the directions are the same (i.e., $W_i = W_{ij}$).

**APPLICATION**

Two corporate networks were engineered using the new algorithm and the Elsner algorithm with various representative values for the design blocking on the finals. NET1 consisted of 12 switches with 2 upper level switches. NET2 consisted of 6 switches with 2 upper level switches. Table 1 contains the details of the results obtained by the two algorithms. In each network we defined the hour with the largest total interswitch load as the network busy hour. The last column in Table 1 contains the achieved average switch-to-switch blocking probability in the network busy hour.

<table>
<thead>
<tr>
<th>NETWORK</th>
<th>SWITCHES ON FINAL LOWER LEVEL</th>
<th>ALGORITHM USED</th>
<th>HIGH-USAGE CIRCUITS</th>
<th>COST INDEX</th>
<th>FINAL SWITCHES</th>
<th>CIRCUITS</th>
<th>COST INDEX</th>
<th>TOTAL SWITCHES</th>
<th>CIRCUITS</th>
<th>AVG. BLOCKING %</th>
</tr>
</thead>
<tbody>
<tr>
<td>NET1</td>
<td>10</td>
<td>ELNSER ALG</td>
<td>1232</td>
<td>72</td>
<td>573</td>
<td>900</td>
<td>900</td>
<td>1800</td>
<td>100</td>
<td>2%</td>
</tr>
<tr>
<td></td>
<td>2 ELNSER ALG</td>
<td>1089</td>
<td>98</td>
<td>326</td>
<td>491</td>
<td>140</td>
<td>140</td>
<td>280</td>
<td>100</td>
<td>0.5%</td>
</tr>
<tr>
<td>NET2</td>
<td>5 ELNSER ALG</td>
<td>1103</td>
<td>79</td>
<td>372</td>
<td>906</td>
<td>128</td>
<td>128</td>
<td>256</td>
<td>100</td>
<td>0.1%</td>
</tr>
<tr>
<td></td>
<td>1 ELNSER ALG</td>
<td>1115</td>
<td>79</td>
<td>372</td>
<td>906</td>
<td>128</td>
<td>128</td>
<td>256</td>
<td>100</td>
<td>0.1%</td>
</tr>
<tr>
<td></td>
<td>6 NEW ALG</td>
<td>255</td>
<td>82</td>
<td>430</td>
<td>720</td>
<td>113</td>
<td>113</td>
<td>226</td>
<td>100</td>
<td>0.6%</td>
</tr>
<tr>
<td></td>
<td>5 NEW ALG</td>
<td>281</td>
<td>89</td>
<td>414</td>
<td>714</td>
<td>116</td>
<td>116</td>
<td>232</td>
<td>100</td>
<td>0.6%</td>
</tr>
<tr>
<td></td>
<td>4 NEW ALG</td>
<td>325</td>
<td>91</td>
<td>414</td>
<td>714</td>
<td>116</td>
<td>116</td>
<td>232</td>
<td>100</td>
<td>0.6%</td>
</tr>
<tr>
<td></td>
<td>3 NEW ALG</td>
<td>355</td>
<td>90</td>
<td>414</td>
<td>714</td>
<td>116</td>
<td>116</td>
<td>232</td>
<td>100</td>
<td>0.6%</td>
</tr>
</tbody>
</table>

Table 1 - Comparison of the New Algorithm With the Elsner Algorithm

The new algorithm produces a much higher average blocking than the Elsner algorithm but still meets the design blocking objective. Figures 2 and 3 show the distribution of the switch-to-switch blocking achieved in NET1 and NET2, respectively, by the two algorithms. Again, the new algorithm produces higher blocking than the Elsner algorithm. For a high-blocking objective (10 percent) the new algorithm did full group a large portion of the high-usage groups.

![Figure 2 - Net 1: 10 Percent Blocking-on-Final Objective Switch-to-Switch Blocking in the Network Busy Hour](image1)

![Figure 3 - Net 2: 5 Percent Blocking-on-Final Objective Switch-to-Switch Blocking in the Network Busy Hour](image2)
For example, in NET1 at 10 percent blocking on the final trunk groups, 21 out of 55 high-usage groups were full grouped and 3 high-usage groups were not equipped i.e., their sizes were zero.

FURTHER IMPROVEMENTS

As indicated earlier, the algorithm does not guarantee that the switch-to-switch blocking will be below a desired value. Closer analysis of the algorithm showed that an important effect was being ignored in [1] when used in stage 2. An example will explain this. Referring to Figure 1, the traffic offered to interregion trunk group AB in the direction A to B overflows first to AD. If carried on AD, it is then offered to DB. However if it gets blocked on AD, it then overflows to AC, and if carried, is then offered to CB, and not to CD as assumed by the model (in stage 2). Only the blocked traffic on CB is offered to CD. Similarly, some of the traffic offered from B to A ends up using DA, which is not accounted for in the algorithm. As a result, the actual loads offered to the intermediate high-usage groups like AD and CB can be different from what the model assumes they are. This difference can result in undersizing some of the intermediate high-usage groups and oversizing some of the final groups (like CD).

In order to consider the impact of A to B overflow on trunk group CB and of B to A overflow on DA, it is necessary to know the link-blocking probabilities on AD, AC, CB, and BD. But when AB is being sized by the algorithm, these groups are not yet sized. This difficulty is resolved by using the link-blocking probabilities computed based on the previous solution. At the end of the current iteration these values are updated. The model of [1] is rewritten using the link-blocking probabilities.

Consider the expression for \( OV_j(h) \) in [1], where \( OV_j(h) \) denotes the mean of the total amount of overflow traffic offered to the jth trunk group in the set N in hour \( h \). We compute this by summing overflows resulting from those high-usage groups in set N that can offer overflow to the jth trunk group in N. But if the jth trunk group in N happens to be a second or third link in the alternate route for the overflow traffic, then the actual amount of overflow offered to it gets reduced by the blocking on the preceding links. Furthermore, if the overflow happens to be from an interregion high-usage group, the traffic blocked on a preceding link gets alternate routed and offered to another intermediate high usage group. All these effects can be included in [1] by using the link-blocking probabilities on the trunk groups in the previous solution as an estimate of those in the current solution.

The details of the implementation have been omitted here but the improvement has been successfully incorporated into the computer program implementing the trunk engineering algorithm. This has resulted in further savings (~1-2 percent) in trunking costs because a few of the figures were oversized earlier. Also, with this improvement it is possible to guarantee switch-to-switch blocking probabilities below the design objective since they can be computed accurately using the link-blocking values.

ACKNOWLEDGMENT

The algorithm was coded and tested mostly by Mr. G. W. Mulcahy of Bell Laboratories. I also wish to acknowledge the guidance provided by Mr. A. L. Kairo of Bell Laboratories.

REFERENCES


APPENDIX A

Computation of the Marginal Capacity

R. I. Wilkinson (2) provides approximate functional relations for the number of trunks needed, \( N \), to maintain a blocking probability of \( P \) when offered a load of mean \( A \) Erlangs with peakedness \( Z \) (i.e., the variance = \( A^2Z \)). They are as follows:

\[
N = DA + EZ - Fe^{-2(4)} - G \quad (1)
\]

\[
.12Z + .2 \leq A \leq 1.7Z + 4.5
\]

\[
P \leq .04
\]

\[
N = HA + iZ + J(AZ)^{1/2}/2 + \frac{KA}{2.5} + L \quad (11)
\]

for \( A \geq 1.7Z + 4.5 \)

\[
P \leq .10
\]

The above formulae are suggested for \( 1 \leq Z \leq 10 \) and \( 2 \leq N \leq 250 \). Numerical values for the coefficients D through L are given for \( P = .01, .02, .03, .05, .07, \) and .10.

R. Dayem developed curve fits to the coefficients in these equations so that they could be used for all values of \( P \) in the range .01 to .10. Differentiating the expressions for \( N \) with respect to the offered load \( A \) and ignoring higher order terms gives the expressions for the reciprocal of \( \rho \), the marginal capacity (defined in the footnote on page 1) as follows:

\[
\gamma^{-1} = 0.9764 - 1.112P
\]

\[
+ (-0.0934-0.4903 \log P)^{1/2}/A
\]

\[
+ (0.01515 \log P + 0.03891)/(2.5+2)
\]

for

\[
Z \geq 1, 0 < P < 0.10, A \geq 1.7Z + 4.5
\]
Also,
\[ \gamma^{-1} = 0.9206 - 0.3225 \log P \]
\[ - (0.1950 + 2.205 \log P)e^{-A-Z} \]
for
\[ Z \geq 1, \ 0.12Z + 0.2 \leq A \leq 1.7Z + 4.5, \ P \leq 0.04. \]

For \( P > 0.10 \), R. Dayem found that \( \gamma^{-1} = (1-P) \)
provides a fairly accurate approximation.

Since all the above expressions are approximate, the marginal capacities computed from them were checked with actual values computed from traffic tables for \( 1 \leq Z \leq 2, \ 0.01 \leq P \leq 0.10, \) and \( 0.5 \leq A \leq 100, \) and resulted in less than 5-percent error in most cases.