CHAPTER 8

Dimensioning Hierarchical Networks

Much work has been carried out on the optimal dimensioning of hierarchical networks for a single demand matrix [1,2,3,4]. Efficient numerical methods allow the calculation to be made for large long distance and metropolitan networks comprising hundreds of nodes. Because each method currently in use was developed by a different organization, these techniques are often described in ways that hide their common structure; in fact, they are fundamentally similar.

Starting from the mathematical program corresponding to the optimal dimensioning problem, we derive the standard ECCS equations of Truitt [1] and those of Pratt [4]. By setting these equations in a more general context, we see how to extend them to the multihour case. We also discuss other formulations of the dimensioning problem with different grade-of-service constraints.

8.1 The ECCS Method for Single-Hour Dimensioning

Current dimensioning methods for telephone networks originated with Truitt [1] on the ECCS method. These results were later improved by Pratt [4], who extended Truitt's simple triangular network to the double-sector tandem case. At the same time, Rapp [3] gave a simple characterization of direct, high-usage, and final trunk groups that could be used in manually designing small- to medium-scale networks. All of this work led to the derivation of numerous computer implementations by virtually all postal administrations or telephone operating companies in the industrialized world for the design of their telephone networks. Most of these methods are still in use today, forming basic dimensioning tools for the new networks.

We now present a general nonlinear optimization model of hierarchical dimensioning. We derive, under suitable assumptions, the ECCS equations of Truitt and the double-sector tandem equations of Pratt, showing how these equations lead to a technique applicable to arbitrary networks. The presentation is based on a general nonlinear programming formulation, differing somewhat from the traditional method, where the only variables are the sizes of the high-usage groups. This approach not only helps us better understand the theoretical foundation of the method, but also helps justify the simplifying assumptions that make it so efficient. In addition, it provides a basis for studying multihour dimensioning of hierarchical networks, as well as generalizing dimensioning under end-to-end grade-of-service constraints.

Problem Definition

In this section, we consider the classical case where the grade-of-service constraints are expressed as blocking constraints on final groups. It is useful to define a special notation for trunk groups:

$\{J\}$ = The index set of final trunk groups.

$\{I\}$ = The index set of high-usage trunk groups.

B_j = The blocking probability on the j^{th} final. This is a function $B(A, N)$ of the traffic offered to the link and the number of trunks. A simple case is the Erlang B function for Poisson traffic.

$C'_s = dC_s/dN_s \geq 0$.

$\overline{B}_j$ = The maximum blocking allowed on the j^{th} final.

Unless otherwise specified, an index i or j refers to a high-usage or final trunk group, respectively. The optimal dimensioning problem is then

$$\begin{aligned} \min_N \; z(\mathbf{N}) &= \sum_s C_s(N_s) \\ B_j(\mathbf{N}) &\leq \overline{B}_j \quad j \in \{J\} \qquad (\mathbf{u}) \\ N_s &\geq 0 \qquad (\mathbf{v}) \end{aligned} \tag{8.1}$$

where the minimization is carried over the reals, and $\mathbf{u}$ and $\mathbf{v}$ are the vectors of multipliers corresponding to the blocking and positivity constraints, respectively. The Lagrange function is given by

$$\mathcal{L}(\mathbf{N}, \mathbf{u}, \mathbf{v}) = \sum_s C_s(N_s) + \sum_j u_j(B_j - \overline{B}_j) - \sum_s v_s N_s,$$

and the first-order Kuhn-Tucker conditions are

$$C'_i + \sum_{j \in \{J\}} u_j \frac{\partial B_j}{\partial N_i} - v_i = 0 \quad i \in \{I\} \tag{8.2}$$

$$C'_j + \sum_{m \in \{J\}} u_m \frac{\partial B_m}{\partial N_j} - v_j = 0 \quad j \in \{J\} \tag{8.3}$$

$$u_j(B_j - \overline{B}_j) = 0 \quad j \in \{J\} \tag{8.4}$$

$$v_s N_s = 0 \quad \forall s \in \{I\} \cup \{J\} \tag{8.5}$$

The equations for the final trunk groups and the high-usage groups are written separately. These equations take a simple form under assumptions that generally hold for public telephone networks.

Optimality Equations for the Final Trunk Groups

Assume $N_j \neq 0$, $j \in \{J\}$, which is necessary if we have a hierarchical routing. If the optimal size of a final group turns out to be zero, the hierarchy has not been well chosen: Either it should be changed, or we should accept a nonoptimal solution, with the final of nonzero size. Then, at the optimal solution, $v_j = 0$ and Eqs. (8.3) and (8.4) become

$$C'_j + \sum_m u_m \frac{\partial B_m}{\partial N_j} = 0$$

$$u_j(B_j - \overline{B}_j) = 0$$

We also know that, at the optimal solution, the blocking constraints will be tight — a direct consequence of the facts that the costs are nondecreasing and that the optimization is done over the reals, as well as of the form of the constraints, where there is exactly one variable N_j for each constraint. Thus we need not worry about Eq. (8.4), which will be automatically satisfied. This fact is crucial for the efficiency of the method described here. We now make the second simplifying assumption,

$$\frac{\partial B_m}{\partial N_j} = \frac{\partial B_j}{\partial N_j} \delta_{m,j}, \tag{8.6}$$

which is reasonable if the blocking is low on the finals. Under these conditions, we can solve Eq. (8.3) explicitly, computing the value of the multipliers at optimality as follows:

$$u_j = -\frac{C'_j}{\partial B_j / \partial N_j}. \tag{8.7}$$

Thus the important conclusion follows that, under the assumptions of tight constraints at the optimal solution and of Eq. (8.6), the multipliers can be computed explicitly; they disappear from the problem.

Optimality Equations for High-Usage Groups

Using Eq. (8.7), Eq. (8.2) becomes

$$v_i = C'_i - \sum_{j \in \{J\}} \frac{C'_j}{\partial B_j / \partial N_j} \frac{\partial B_j}{\partial N_i}. \tag{8.8}$$

Equation (8.8) forms a set of simultaneous nonlinear equations, one for each high-usage group; each solution yields a stationary point of (8.1). Before

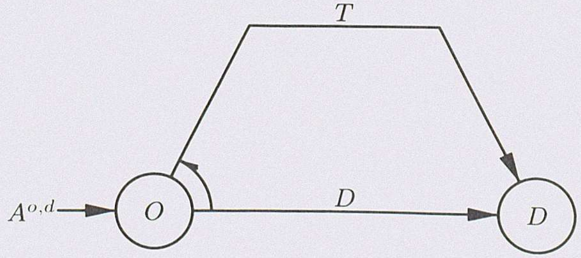

Figure 8.1 Two-Link Network

discussing general solution methods, we look at some simple cases, recovering the classical ECCS equations for the two-path and square networks. At first we will assume that $N_i > 0,\ i \in \{I\}$, which implies that $\mathbf{v} = \mathbf{0}$ and permits us to ignore Eq. (8.5). Although not a valid assumption for realistic networks, this simplifies the theory; we shall see later how to handle the cases where $N_i = 0$.

The ECCS Method for Simple Networks

Equation (8.1) is a nonlinear optimization with differentiable functions. A number of general-purpose algorithms and computer implementations are available to solve this problem [5]. These methods are not used here for a variety of reasons, most importantly because of the very large size of the problems that must be resolved, even for moderately large networks. Specialized, extremely efficient techniques have been developed to take advantage of the special features of the dimensioning problem.

While general-purpose nonlinear programming techniques rely on *descent* methods, these special techniques attempt to find a solution to the optimality conditions for the high-usage case (8.8) *directly*. The following equation uses the blocking constraints to size the finals:

$$B_j(N_j) = \overline{B}_j.$$

We show how the technique is carried out for some simple networks, deriving the standard equations proposed in the literature.

Two-Link Network. The first case is a two-node, two-link network with alternate routing; calls blocked on the direct D link are offered to the tandem T link [1] (see Fig. 8.1). Following the usual method, we assume that the cost functions are linear (although, with suitable modifications, all that follows could equally be stated for differentiable cost functions).

$$C(N_D) = C_D N_D$$
$$C(N_T) = C_T N_T$$

and

$$N_i > 0$$

Eq. (8.8) becomes

$$C_D = \frac{C_T}{\partial B_T / \partial N_T} \frac{\partial B_T}{\partial N_D}.$$

Let

A_D = The total traffic offered to the direct link. This is the first-offered traffic.

a_T = The total traffic offered to the tandem link. This is the sum of the first-offered traffic plus the overflow from the direct link.

$\hat{a}_D$ = The overflow traffic from the direct link.

$\overline{a}_D$ = The total traffic carried on the direct link.

Then

$$\frac{\partial B_T}{\partial N_D} = \frac{\partial B_T}{\partial a_T} \frac{\partial \hat{a}_D}{\partial N_D},$$

and Eq. (8.8) becomes

$$\frac{C_D}{C_T} = \frac{\partial B_T / \partial a_T}{\partial B_T / \partial N_T} \frac{\partial \hat{a}_D}{\partial N_D},$$

where $\partial B_T / \partial a_T$ means that the partial derivative of B_T is evaluated at a_T, the current value of the total link-offered traffic. Because a_T is the sum of the first-offered and overflow traffic, the partial derivative depends on the size of the direct group. We now use the fact that, at the optimal solution, the blocking constraint on the final is tight. For this group, there are three parameters B, a, and N that are linked through the traffic model, in this case the Erlang B function

$$B = E(a, N).$$

Any variation of one parameter will produce a variation of at least one of the other two parameters. We can write the following differential equation to express this relation:

$$dB = \frac{\partial E}{\partial a} da + \frac{\partial E}{\partial N} dN. \tag{8.9}$$

Because we know that, at the final solution, the blocking on the final is tight, we must compensate for any variation in the traffic offered to the link by a variation of the number of circuits such that $dB = 0$. Replacing this condition in Eq. (8.9), we get

$$\frac{\partial B_T / \partial a_T}{\partial B_T / \partial N_T} = -\left(\frac{dN_T}{da_T}\right)_{\overline{B}_T}.$$

The right-hand side denotes the variation in the number of trunks required to maintain the blocking at $\overline{B}_T$ for a small variation of total offered traffic. This yields the optimality equation

$$\frac{C_D}{C_T} = -\left(\frac{\partial \hat{a}_D}{\partial N_D}\right)\left(\frac{dN_T}{da_T}\right)_{\overline{B}_T}. \tag{8.10}$$

This is a nonlinear equation in which the right-hand side has two terms that depend on the single variable N_D. The solution yields the optimal value for the number of trunks on the direct route. We can reformulate Eq. (8.10) by defining the marginal capacity of the final β_T:

$$\frac{1}{\beta_T} \triangleq \left(\frac{dN_T}{da_T}\right)_{\overline{B}_T}. \tag{8.11}$$

We also define the marginal occupancy of the high-usage group:

$$H \triangleq -\left(\frac{\partial a_D}{\partial N_D}\right) = \left(\frac{\partial \overline{a}_D}{\partial N_D}\right), \tag{8.12}$$

which represents the rate of change of the traffic carried on the direct link with respect to the number of circuits on this link for a fixed value of the offered traffic. Finally, if we let $CR = C_T/C_D$, the equilibrium condition becomes

$$\frac{\beta}{CR} = H, \tag{8.13}$$

where CR is called the *cost ratio* for obvious reasons. In the North American literature, this equation is generally written

$$ECCS = \frac{\gamma}{CR}, \tag{8.14}$$

with

$$ECCS \triangleq H$$
$$\gamma \triangleq \beta$$

Hence the name of the method. The name can be understood from Eq. (8.14) as follows. Traffic was traditionally measured by counting the number of busy circuits on a group during periods of 100 seconds. Thus a convenient unit of traffic was the number of calls present during these 100 seconds, which became the number of CCS, each CCS representing 100 call seconds. The term ECCS means *economic* CCS. From Eq. (8.12), we see that it is the amount of traffic that can be carried on the direct route for a small increment in the capacity of this route. For integer sizes, ECCS can be defined as the amount of additional traffic that can be carried on the last trunk added to the route. Similarly, Eq. (8.11) defines the additional amount of traffic that can be carried on the final for a small increment of capacity, leaving the blocking unchanged (for

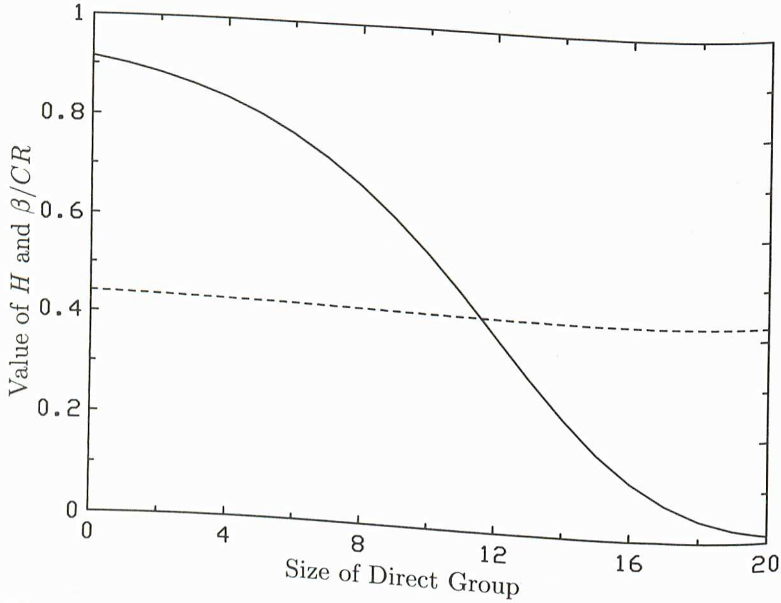

Figure 8.2 Representation of ECCS Equation. $A_T = 20$, $A_D = 10$, $CR = 2$, and $B = 0.01$

the final, we assume that carried $\approx$ offered). The equilibrium equation (8.13) defines the optimal size of the direct group as the number of circuits such that the traffic carried on the last circuit of the direct route (the *economical* CCS) is the same as the traffic carried on the last circuit of the alternate route, scaled by the ratio of provisioning costs on the two routes. Equation (8.10) can be rewritten

$$C_D \left(\frac{\partial N_D}{\partial \bar{a}_D} \right) = C_T \left(\frac{\partial N_T}{\partial a_T} \right)_{\overline{B}_T} .$$

That is, the optimal group size is the one where the cost of routing a small increment of traffic on the direct route is the same as the cost of routing it over the alternate route at the same blocking level. This characterization of the optimal group size is another manifestation of Wardrop equilibrium conditions for transportation networks and marginal-cost equilibrium condition for nonlinear multicommodity flows.

The optimal value for N_D is computed from Eq. (8.13), a nonlinear equation in the variable N_D. Although there are many numerical methods for solving nonlinear equations, in practice the following technique is used. Let β be constant, for example, $\beta = 28$ CCS/circuit. Given the value of CR, the

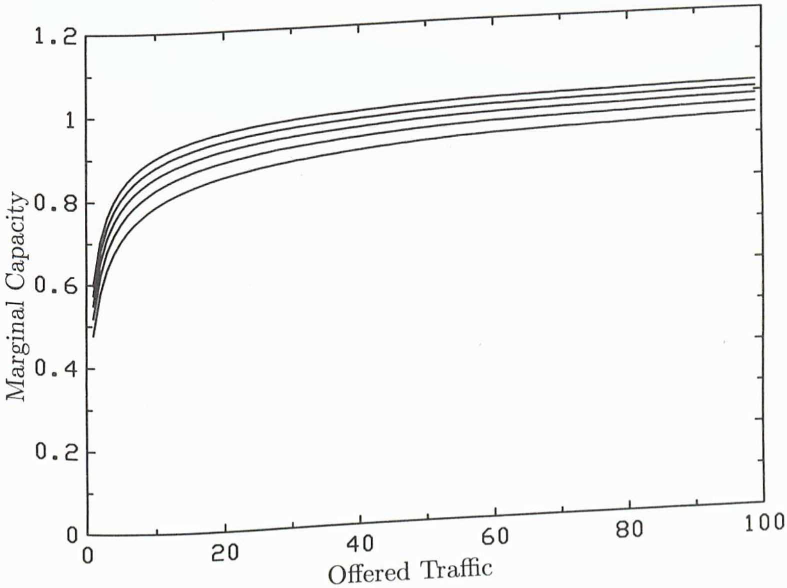

Figure 8.3 Value of Marginal Capacity. Blocking from 0.01 to 0.05

ratio of costs on the direct path to costs on the overflow path, the only term containing N_D is H. We can therefore solve the problem simply by examining a graph of $\partial \overline{a}_D / \partial N_D$ for A_D equal to the known value of the first-offered traffic. Figure 8.2 shows a plot of the two functions H and β as functions of N_D for typical values of traffic. It is clear that β is a very slowly varying function of N_D and that it can safely be taken as a constant while the optimality equations are solved. The particular value that must be chosen depends more on the traffic offered to the final than on N_D, as can be seen from Fig. 8.3. Nevertheless, the assumption of constant β is often sufficient to give satisfactory values for the direct group.

The ECCS equation has been derived under the assumption that $N_D > 0$. Such a solution will occur whenever $\beta/CR < 1$. If this is not so, Eq. 8.13 has no solution. In this case, we can see that the value $N_D = 0$ is a solution of the Kuhn-Tucker equation (8.13). We have, from Eq. (8.8),

$$\begin{aligned} v_D &= C_D - C_T \frac{\partial B_T / \partial N_D}{\partial B_T / \partial N_T} \\ &= C_D - C_T \frac{H}{\beta_t}. \end{aligned} \tag{8.15}$$

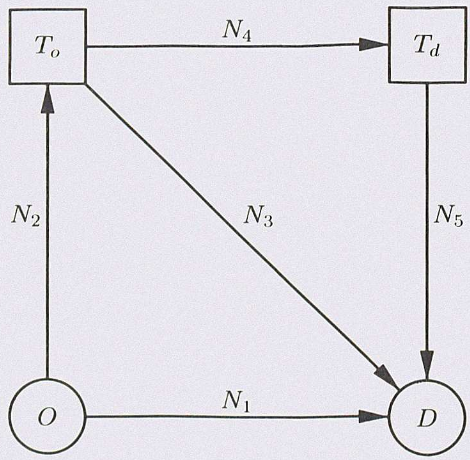

Figure 8.4 Double-Sector Tandem Network

It is not difficult to show that, if $N_D = 0$, then the right-hand side is non-negative, and that the value $N_D = 0$ satisfies the Kuhn-Tucker equation (see Problem 8.3).

Double-Sector Tandem Network. The second simple case is the familiar double-sector tandem network, with two tandem and two lower-level nodes [4] (see Fig. 8.4). Here too, we assume that $N_i > 0$ at the optimal solution. We write the first component of Eq. (8.8), the gradient in the direction N_1:

$$\frac{\partial B_j}{\partial N_1} = \frac{\partial B_j}{\partial a_j}\frac{\partial a_j}{\partial \hat{a}_1}\frac{\partial \hat{a}_1}{\partial N_1}.$$

A new term appears in the form of $\partial a_j / \partial \hat{a}_1$, the variation of traffic offered to the final with respect to a variation of overflow from the direct. Because intermediate group 3 stands between these two groups, we expect this derivative to depend somehow on the parameters of group 3. Using Eq. (8.7), Eq. (8.8) becomes

$$C'_1 = \sum_j C'_j \frac{\partial B_j / \partial N_j}{\partial B_j / \partial a_j}\frac{\partial a_j}{\partial \hat{a}_1}\frac{\partial \hat{a}_1}{\partial N_1}. \tag{8.16}$$

Using the fact that the constraints are tight at the optimum, we can write for each final

$$-\frac{\partial B_j / \partial a_j}{\partial B_j / \partial N_j} = \left(\frac{dN}{da}\right)_{\overline{B}}$$
$$= \frac{1}{\beta_j}.$$

Assuming low losses on the finals, we get

$$\begin{aligned}\frac{\partial a_j}{\partial \hat{a}_1} &= \frac{\partial a_j}{\partial \hat{a}_3}\frac{\partial \hat{a}_3}{\partial \overline{a}_2}\frac{\partial \overline{a}_2}{\partial \hat{a}_1}, \quad j = 4,5 \\ &= \frac{\partial a_j}{\partial \hat{a}_3}\frac{\partial \hat{a}_3}{\partial a_3} \\ &= \frac{\partial a_j}{\partial \hat{a}_3}\gamma_3,\end{aligned}$$

where

$$\gamma_3 \triangleq \frac{\partial \hat{a}_3}{\partial a_3}, \tag{8.17}$$

where γ_3 is called the marginal overflow and should not be confused with the quantity γ used in the traditional form of the ECCS equation (8.14). Similarly,

$$\frac{\partial a_2}{\partial \hat{a}_1} = 1.$$

Again we define the marginal occupancy of the high-usage group,

$$H_1 = -\frac{\partial \hat{a}_1}{\partial N_1}, \tag{8.18}$$

and Eq. (8.16) becomes

$$C_1' = H_1 \left\{ \left[\frac{C_5'}{\beta_5} + \frac{C_4'}{\beta_4} \right] \gamma_3 + \frac{C_2'}{\beta_2} \right\}, \tag{8.19}$$

which is the form in [4]. The equation for the other high-usage group N_3 can be derived in the same manner,yielding

$$C_3' = H_3 \left[\frac{C_4'}{\beta_4} + \frac{C_5'}{\beta_5} \right]. \tag{8.20}$$

(See Problem 8.1.) We have reduced the dimensioning problem to finding the solution of two *simultaneous* nonlinear equations (8.19) and (8.20) in two unknowns N_1 and N_3. Although we could solve the equations by any general-purpose method for nonlinear systems, we use instead a special relaxation technique that relies on their structure. Note that $H_1 = H_1(N_1)$ and, similarly, that $H_3 = H_3(N_3)$; that is, the marginal occupancies depend only on the parameters of the high-usage group being dimensioned if the traffic offered to the group is known. If we take all the βs and γs of Eqs. (8.19) and (8.20) to be constants, then the two equations *separate*, and the values of N_1 and N_3 can be computed by solving each equation independently. An iterative procedure can be used to adjust the coefficients after each sizing until convergence is achieved — which in turn suggests the general relaxation method discussed below.

These optimality equations are derived under the assumption that, at the optimal solution, we have $N_i > 0$. Clearly this need not be the case; we should examine how the equations change without this assumption. This point is closely related to the nature of the dimensioning equations for the high-usage group, as given by Eqs. (8.19) and (8.20). We know that the $H(N_i)$ function is a monotone-decreasing function of N_i and that $H(0) \leq 1$. For the particular choice of cost ratios, the equation may well turn out not to have any solution. The traditional procedure when this occurred was to take $N_i = 0$ as the solution of the equation, using the heuristic argument that the cost ratio is so low that it is not worth using the direct route, and that all the traffic should go on the final. In fact, this argument can be given a rigorous basis; it is left as an exercise (Problem 8.3) to show that, if the dimensioning equation has no solution, then the value $N_i = 0$ satisfies the optimality conditions for the problem reformulated without the assumption that $N_i > 0$ at the optimal solution.

One final remark is in order concerning solutions at zero. Because the group does not exist, it obviously cannot be used, indicating in turn the best routing to use in the network. In other words, although we may have assumed initially that the routing used the group in question, the group may be uneconomical to put in the network, in which case our assumption about the routing was clearly wrong. In this sense, we can say that the results of dimensioning can affect the routing decisions taken for the network. This is about as far as the matter goes, however, since the presence of a high-usage group may be economically justified for the assumed routing, but could equally well not be for some other routing. In other words, solving the dimensioning problem usually does not give much information concerning the routing, contrary to what is sometimes stated in the literature [6].

General ECCS Method

For a general hierarchical network, Eq. (8.8) represents a set of simultaneous nonlinear equations whose solution yields the optimal value of the high-usage trunk group sizes. The finals are dimensioned from the grade-of-service constraints, given that these constraints are tight at the optimal solution.

These equations must be solved for all high-usage groups. They are coupled because of the terms $\partial B_j / \partial N_i$, whose evaluation is also the most time-consuming part of the calculation. We give here a recursive procedure for this computation, using the fact that the groups are ordered, at least to the accuracy required for dimensioning. This, incidentally, yields a relaxation technique for solving the equations.

First note that the derivatives are computed for fixed group sizes. Group j and group i can therefore be coupled only through overflow or carried traffics. That is, a small perturbation of the size of group i will induce a variation of overflow and carried traffics that will propagate toward j via the groups actually used to go from i to j by all traffic components offered to i. As a consequence,

$$\frac{\partial B_j}{\partial N_i} = 0 \quad \forall j \mid \mu(j) \leq \mu(i).$$

Consider now the case where j is after i. Let

$$\begin{aligned} B_j &= E(a_j, N_j) \\ a_j &= \sum_k a_j^k \end{aligned}$$

where a_j^k is the traffic offered to link j that came along the k^{th} path from link i; we call this a component of the traffic offered to the link. In what follows, we assume that all traffic streams offered to a group experience the same blocking. Thus B_j is a function of the *total* traffic offered to the link, not of the particular composition of this flow. We can write

$$\begin{aligned} \frac{\partial B_j}{\partial N_i} &= \frac{\partial B_j}{\partial a_j} \frac{\partial a_j}{\partial N_i} \\ &= \frac{\partial B_j}{\partial a_j} \sum_k \frac{\partial a_j^k}{\partial N_i}. \end{aligned}$$

The computation is reduced to evaluating the derivative of each component of the traffic offered to j with respect to N_i. This computation can be done recursively, looking at all the paths from i to j in the influence graph. Let $i = m_0, m_1, m_2, \ldots, j = m_j$ be some such path for component k. We have

$$\frac{\partial a_{m_1}^k}{\partial N_i} = \begin{cases} a_i^k \dfrac{\partial B_i}{\partial N_i} & \text{if } i \text{ overflows on } m_1 \\ -a_i^k \dfrac{\partial B_i}{\partial N_i} & \text{if } i \text{ is carried on } m_1 \end{cases}$$

$$\frac{\partial a_{m_\ell}^k}{\partial N_i} = \begin{cases} \dfrac{\partial a_{m_{\ell-1}}^k}{\partial N_i} \left[B_{m_{\ell-1}} + a_{m_{l-1}}^k \dfrac{\partial B_{m_{\ell-1}}}{\partial a_{m_{\ell-1}}^k} \right] & \text{if } m_{\ell-1} \text{ overflows on } m_\ell \\ \dfrac{\partial a_{m_{\ell-1}}^k}{\partial N_i} \left[1 - B_{m_{\ell-1}} - a_{m_{l-1}}^k \dfrac{\partial B_{m_{\ell-1}}}{\partial a_{m_{\ell-1}}^k} \right] & \text{if } m_{\ell-1} \text{ is carried on } m_\ell \end{cases} \tag{8.21}$$

The calculation is clearly recursive, the order proceeding from i to j along all the paths between these two nodes of the influence graph. Let

$$G_{i,m_1}^k = \frac{\partial B_i}{\partial N_i}$$

$$G_{m_\ell}^k = \begin{cases} \left[B_{m_\ell} + a_{m_{l-1}}^k \dfrac{\partial B_{m_\ell}}{\partial a_{m_\ell}^k} \right] & \text{if } m_\ell \text{ overflows on } m_\ell \\ 1 - \left[B_{m_\ell} + a_{m_{l-1}}^k \dfrac{\partial B_{m_\ell}}{\partial a_{m_\ell}^k} \right] & \text{if } m_\ell \text{ is carried on } m_\ell \end{cases}$$

Note that G does not depend directly on the component, except that the arc of the influence graph is an overflow or a carry arc. This arc is a parameter only of the link since it involves the current total link-offered traffic, as well as derivatives of the blocking function with respect to this traffic. We can write

$$\frac{\partial a_j^k}{\partial N_i} = a_i^k \prod_{s=0}^{j} G_{m_s}^k$$
$$= a_i^k \frac{\partial B_i}{\partial N_i} \prod_{s=1}^{j} G_{m_s}^k.$$

The derivative of the blocking is then

$$\frac{\partial B_j}{\partial N_i} = \frac{\partial B_j}{\partial a_j} \sum_k a_i^k \frac{\partial B_i}{\partial N_i} \prod_{s=1}^{j} G_{m_s}^k$$
$$= \frac{\partial B_j}{\partial a_j} \frac{\partial B_i}{\partial N_i} a_i \sum_k \frac{a_i^k}{a_i} \prod_{s=1}^{j} G_{m_s}^k$$
$$= -\frac{\partial B_j}{\partial a_j} H_i(N_i) \sum_k \frac{a_i^k}{a_i} \prod_{s=1}^{j} G_{m_s}^k. \tag{8.22}$$

We can thus calculate the derivatives with respect to N_i recursively by scanning all the links from i that lead to j. Because the same components are used for all j downward from i, the information can be stored; only a single pass is required for calculating all the descendants.

The optimality equations (8.8) can be rewritten, assuming $N_i > 0$,

$$0 = C_i' - \sum_j C_j' \frac{\partial B_j / \partial a_i}{\partial B_j / \partial N_j} H_i(N_i) \Gamma_{i,j}(\mathbf{N})$$
$$= C_i' - H_i(N_i) \sum_j \frac{C_j'}{\beta_j} \Gamma_{i,j}(\mathbf{N}), \tag{8.23}$$

where

$$H_i(N_i) = -\left(\frac{\partial \hat{a}_i}{\partial N_i}\right)$$
$$= -a_i \frac{\partial B_i}{\partial N_i}$$
$$\Gamma_{i,j} = \sum_k \frac{a_i^k}{a_i} \prod_{s=1}^{j} G_{m_s}^k \tag{8.24}$$

Eq. (8.23) can be solved by a relaxation method similar to the one used for the simple two-path network. Assuming that the $\Gamma_{i,j}$ and β_j are constants independent of the value chosen for the N_is, Eq. (8.23) separates into independent nonlinear equations in a single variable. An iterative procedure is used to find the correct values for the $\Gamma_{i,j}$s and β_js. The relaxation algorithm is as follows:

1. The β_j and $\Gamma_{i,j}$ are known, either from the previous iteration or from some initial estimate. Making them fixed separates the system (8.8) into independent equations, one for each high-usage group.
2. Using Eq. (8.23), calculate the value of H_i for each direct link. Knowing these quantities as well as the first-offered traffic to these links allows us to calculate the number of circuits N_i that must be installed according to the current values of the $\Gamma_{i,j}$.
3. Knowing the values of N_i and the first-offered traffic, calculate the total traffic offered to each overflow link.
4. Knowing the value of traffic offered to the overflow links, calculate new values for the β_j and the $\Gamma_{i,j}$.
5. If the new values differ from the old ones by more than a preset limit, return to step 1, using these new values of the β_j and the $\Gamma_{i,j}$. Otherwise, a stationary point has been found.

We call an ECCS method any dimensioning method that attempts to solve the Kuhn-Tucker conditions directly by such a relaxation procedure. As we see later, similar techniques can be used in different contexts; the efficiency obtained on these problems should be comparable to that of the standard ECCS method explained here. In practice, the procedure can be modified in several ways:

1. Most important, the traffic must be updated frequently during an iteration, every time the size of a high-usage group is changed. Otherwise the relaxation procedure often will not converge.
2. The procedure can be speeded up by taking the βs and Γs constant, reducing the relaxation procedure to a single iteration. Even if these terms are not assumed to be constant, they converge very quickly to a stationary value, so that it is seldom necessary to iterate more than two or three times.
3. Because solutions must be in integers, trunk sizing is often done for integer N only. This may speed up the calculation of the trunk groups.

A word should be said about traffic models used in the dimensioning. Traffic flows within a hierarchical network can be broadly classified into two categories. The high-usage direct groups receive external flows only; for these links, the Poisson model is accurate. The finals, however, receive traffic that has been blocked somewhere else in the network. Thus the traffic offered to these groups is highly peaked; the Poisson model is *not* adequate to compute their blocking. The ECCS method can easily be modified to work with a two-moment model for traffic throughout the network. The simpler technique of using Poisson models for high-usage and intermediate groups, and a two-moment model for the finals, may be adequate in many situations, avoiding the larger storage requirements of the full two-moment model.

Summary

The relaxation procedure described here can be applied to the dimensioning of very large hierarchical networks, with hundreds of nodes. This impressive efficiency is the consequence of four features of the problem:

1. For a given value of the group sizes, traffic flows are not computed iteratively, but in a single pass over all the links. This is because the links can be ordered, which in turn is a consequence of *both* the routing and the assumption of low blocking on the finals.

2. The method works directly on the Kuhn-Tucker equations, and is more like a Newton method for solving nonlinear equations than a traditional descent method for nonlinear optimization.

3. The grade-of-service constraints are expressed on a link-by-link basis, facilitating the separation of the Kuhn-Tucker equations during the procedure.

4. The constraints are tight at the solution, which, in conjunction with the low blocking on finals, allows the multipliers to be eliminated from the Kuhn-Tucker equations.

As we shall see, these characteristics are not always present when more complex routings are used or when more than one traffic demand is considered. Nevertheless, the ECCS method can still be used in the latter case and large gains in efficiency can be expected over other methods proposed so far.

8.2 Multihour Dimensioning

The dimensioning methods described in Section 8.1 are based on the existence of a single matrix representing the traffic for between all the offices. It is known, however, that traffic demand is not a stationary process and that the average traffic flow between two offices can vary substantially. Within metropolitan networks, variations occur during the day because of the different periods of peak demand between the city core and the suburbs. A similar daily effect is present in international traffic or in long distance networks spanning different time zones, as in North America. Finally, multiple traffic demands may vary seasonally because of large population movements during holidays and vacation periods.

A simplistic solution consists of either (1) ignoring the effect, dimensioning the network based on a single, presumably representative, traffic matrix, or (2) taking the effect into account, dimensioning the network for a matrix made of the peak values of each individual matrix. The first method does not guarantee an adequate grade of service in all periods, while the second provides adequate service but at a much higher cost than necessary.

The technique of multihour dimensioning specifically takes into account these variations in the offered load. The basic idea is to utilize the fact that demand peaks do not occur at the same time between all pairs of offices, thus reducing the number of circuits that must be provided but still maintaining an adequate grade of service in all time periods. The potential economies depend on the existence of alternate routing to efficiently utilize network capacity during off hours; trunk savings might be greater than otherwise the case for fixed routing if the routing rules were modified as the demand varies over time. This topic is discussed elsewhere in the chapters dealing with dynamic and adaptive routings (Chapters 7 and 9). For now, we restrict ourselves to the case of fixed hierarchical alternate routing.

This section is divided as follows. First we give a precise mathematical programming formulation of the multihour-dimensioning problem. Then we review several types of solution methods, the first of which is so-called *equivalent methods*, where the multihour problem is reduced to a single problem, or perhaps a sequence of single-hour problems. Then we examine two methods that deal directly with the multihour problem: (1) a primal method that requires the techniques of nondifferentiable optimization and (2) a dual method that generalizes the multihour case of the single-hour ECCS method (see Section 8.1).

Problem Definition

The mathematical programming formulation of multihour dimensioning, quite similar to Eq. (8.1), can be written

$$\begin{aligned} &\min_N \sum_i C_i(N_i) \\ &B_j^t(\mathbf{N}) \le \overline{B}_j^t, \quad j \in \{F\}, \quad t = 1, \ldots T \qquad (\mathbf{u}^t) \\ &N_i \ge 0 \qquad (\mathbf{v}) \end{aligned} \tag{8.25}$$

where

A_t = The traffic matrix for period t.

$\overline{B}_j^t$ = The grade of service constraint on the j^{th} final in period t.

$B(\mathbf{N})^t$ = The blocking function on the finals in period t. This function depends implicitly on the traffic matrix A_t, a dependence made explicit when needed by writing the function as $B_j^t(A_t, \mathbf{N})$.

$\mathbf{u}, \mathbf{v}$ = The Kuhn-Tucker multipliers for the blocking and positivity constraints, respectively.

Note two points about this formulation. First, Eq. (8.25) differs from Eq. (8.1) by the *addition* of constraints. This means that the solution of the multihour problem *cannot* be less costly than the solution of a single-hour problem with any one of the A_t as the traffic matrix. Claims to the effect that multihour methods are lower cost than single-hour methods mean that the single-hour problem has a different matrix from that of any one of the matrices used for multihour dimensioning.

Second, the grade of service can be made period dependent — a very reasonable assumption if rates are period dependent, the usual case in telephone networks. After all, if one is paying $x\%$ less during weekends, it is quite reasonable that the grade of service during that period should be somewhat poorer that during weekdays, when the full rate is applied.

Equivalent Single-Hour Methods

Because the ECCS method is an extremely efficient way to dimension hierarchical networks for a single-traffic matrix, it is natural to try to use it in solving the multihour problem. This has been done is two different ways. The first, called the *sizing-up* method, solves a sequence of problems, each with the correct demand matrix, but with additional constraints on the group sizes [7]. Sizing up works as follows:

1. Rank the A_t in some predetermined order.

2. Solve a sequence of T single-hour problems, each of the form

$$\min_{N_s^{(t)}} \sum_s C_s(N_s^t)$$

$$B_j^t(A_t, N^{(t)}) \leq \overline{B}_j^t$$

and with the additional constraints

$$N_s^{(t)} \geq N_s^{(t-1)} \geq 0.$$

The method yields a sequence of nondecreasing values N^t; the solution to (8.25) is given by N^T. The network thus produced meets the grade-of-service constraints in all periods and for the appropriate matrix. Furthermore, it is linear in T, probably as good as could be expected in terms of computational requirements. On the negative side, the final solution strongly depends on the ordering of the matrices, and it is not known how far the solution is from the true optimum of (8.25). Published results [7] indicate that the method may "overdimension" networks — the gains obtained by using it over the usual cluster busy-hour technique explained next are lower than those obtained by the more accurate methods of Elsner and of Girard, Lansard, and Liau. Beshai, Pound, and Horn claim that the sizing-up method gives comparable savings on other networks, but no such evidence has yet been presented.

Another kind of approximate method is based on computing a matrix A that is equivalent to the set of matrices A_t in that it somehow captures the important effects of noncoincidence, and then to use this matrix with the single-hour ECCS method. Two approaches of this type have been devised: (1) the so-called *cluster busy-hour method* and (2) the *equivalent-matrix method* of Rapp [8]. Both methods assume that the grade of service is the same in all periods, ensuring that the saturation period is the period when the largest traffic is offered to the final, which is not the case if the grades of service are different.

The cluster busy-hour method is a way of choosing a period t that will be used in a single-hour ECCS method to dimension the network [9]. Because the constraints are expressed in term of blocking on the finals, the significant period in which to do the dimensioning should be when the traffic offered to a final is greatest. This concept is captured by the notion of a cluster, which is, for an originating office i, the set of all (i, m) high-usage groups and the (i, T_i) final. The cluster busy-hour is defined as the period when the total cluster traffic is greatest, or, more precisely, the period when the total carried traffic on the high-usage groups plus the total traffic offered to the (i, T_i) final is greatest. If t_i is the busy hour for a cluster originating at i, the equivalent traffic matrix has row $\mathbf{a}^i = \mathbf{a}^i_{t_i}$. The equivalent matrix is then used in the single-hour ECCS method to dimension the network.

Because the method does not guarantee that the grade of service is met in all periods, a further sizing-up stage is sometimes required to increase the capacity of the finals to meet the actual demand in off-hours. Given that the sizes of the high-usage groups are left unchanged during this operation, this poses no difficulty. The method is currently used to dimension telephone networks for more than one traffic matrix. It can also be extended to take into account both legs of the alternate route, in which case it is called the *significant-hours* method [9].

Rapp's method [8] can be best understood by considering the dimensioning of a high-usage group i, j within a two-level network with tandem T. This