Optimal Clustering Structures for Hierarchical Topological Design of Large Computer Networks*

Leonard Kleinrock
Computer Science Department, University of California, Los Angeles, California 90024

Farouk Kamoun
Departement Informatique, Faculté des Sciences, Tunis, Tunisia

Large packet switching computer networks on the order of hundreds or thousands of nodes will soon emerge to handle the fast-growing demands in data communication and resource sharing among various information processing systems around the world. The network topology design problem has long been recognized as extremely complex and very quickly becomes unmanageable as the size of the network increases. Existing heuristic design procedures are quite efficient for the design of small to moderate-sized networks (25–75 nodes); however, they become very costly and even prohibitive when dealing with large networks. A design methodology based on the hierarchical clustering of the network nodes is presented in this paper in order to alleviate the computational cost involved in the design. More specifically, the emphasis is on the determination of a *clustering* structure which minimizes the computational cost of the design. Such a cost is assumed to have a polynomial growth with the number of nodes in the subnet to be designed. We present optimum results both for the number of clusters, number of superclusters, etc., and for the number of hierarchical levels. An expression for the average delay of a message in such a hierarchical network is also provided in terms of the average delays in the subnets composing the network. This decomposition leads to the design of smaller subnetworks for which we can utilize present design strategies.

1. INTRODUCTION

Computers have become essential in many of the daily operations of most public and private enterprises. Besides the scientific and/or data-management tasks handled within a computing facility, a rather tremendous volume of data transaction must be exchanged among different facilities whether or not they are located remotely. Moreover, the need for resource sharing of specialized hardware, software, data banks, etc.,

*This research was supported by the Advanced Research Projects Agency of the Department of Defense under contract MDA 903-77-C-0272.

Networks, Vol. 10 (1980) 221–248

0028-3045/80/0010-0221$02.80

has been long recognized [1]. As a result, distributed packet switching computer communication networks have come about to satisfy the ever-growing demands in data communication and resource sharing.

The success of the novel packet-switching technique in networks has been clearly demonstrated with the pioneering ARPANET [2-6]. This network, in operation since 1969, set the pace for the development of a multitude of other networks throughout the world (EPSS in England, CYCLADES and TRANSPAC in France, EIN and EURONET in Europe, DATAPAC in Canada, TELENET and AUTODIN II in the USA, etc.). It has also been the prototype for the developing data carrier industry (which some feel may become comparable in size to the telephone industry in a few years!).

A major network component is the communication subnet which is composed of the hardware and software necessary to carry data messages between host computers, terminal handlers, or other devices attached to the network.

Several issues arise with regard to the topological design and operational procedures for the communication subnet. Basically, the topological design is concerned with the location of the switching nodes, the selection of channel capacities and their geographical layout so as to satisfy the traffic and performance requirements with the least cost. Dynamic operational procedures control and direct the flow of messages in the network so as to avoid network congestion and to adapt to failures while maintaining a good network performance.

A great deal of work has been devoted to the development of efficient design and operational procedures. The principles of these techniques may be found in the appropriate references [7-13] and the bibliographies therein.

Existing procedures are quite suitable for the efficient design and operation of small to moderate-sized networks. Unfortunately, they become very costly (from a computational point of view) and in some instances infeasible if directly applied in the context of large networks [8, 9, 11, 14, 15]. Predictions indicate that, in fact, large networks of the order of hundreds (or possibly thousands) of nodes are soon to come. As a result, other design and operational procedures must be found which handle the large network case.

We address the issue of adaptive routing in large networks [16-18]. We observe that indeed the storage and channel capacity requirements due to routing become excessively costly as the number of nodes in the network becomes large. Hierarchical routing schemes based on the hierarchical clustering of the network nodes are there proposed to reduce the cost of routing. Those studies show a remarkable efficiency of optimally selected hierarchical routing schemes for large networks.

In this paper our interest lies in the development of topological design procedures for large networks. Here again, clustering is introduced to reduce the cost of design.

Several different formulations of the design problem related to the communication subnetwork can be found in the literature [9, 10, 13]. Generally, they correspond to different choices of performance measures, design variables, and constraints. Here, we select the following very general formulation:

Given: Node locations,
channel capacity options and costs
Minimize: Total communication cost

Over: Topology,
channel capacities,
routing
Subject to: Delay constraint,
reliability constraint,
traffic requirement.

In general, there are $2^{N(N-1)/2}$ possible topologies, where N is the number of nodes. Furthermore, capacities are available in discrete sizes. In addition, the constraints must be satisfied. This means that an *enormous* integer optimization problem must be solved. The nonlinearity of the time-delay functions [7, 10] and, in some cases, of the reliability measure [9] add more dimensions of complexity to the problem. A connectivity of 2 or 3 is often used as a reliability constraint; a higher connectivity may be required for large networks.

There exists no efficient technique for the exact solution of this topological design problem. Several heuristic procedures have been proposed and implemented. Among them, we mention the Branch X-change method [9], the cut saturation method [9], and the Concave Branch Elimination method [10, 13]. Typically they start with an initial topology over which they perform some alterations in the course of the optimization. Built into those procedures and inherent in the multicommodity nature of the flow is the determination of the shortest path between any pair of nodes in the network.* This operation requires between N^2 to N^3 operations and may be performed many times in the course of the optimization. The overall computational complexity corresponding to those heuristics is estimated to be on the order of N^3 to N^6 [9, 14].

For networks with more than a few hundred nodes, present procedures fail because of the enormous amount of computer time and storage needed to perform the suboptimization. Design procedures, based on hierarchical clustering, have been proposed [14, 15, 19, 20] to substantially reduce the computational cost and the storage requirement involved in the optimization step. Generally speaking, in a two-level hierarchical design, the nodes will be grouped into clusters and "gates" (special cluster-exchange nodes) will be selected from each cluster. Cluster subnets will be designed separately, and then a supernet of gates will be designed to connect the clusters together. The assumption is that nodes in the same cluster are most likely to be very few hops apart in either a nonhierarchical or hierarchical design. The approach described briefly above could be easily extended to more than two levels in the hierarchy (see Sec. 2).

The emphasis of this paper is on the determination of a clustering structure to be used in the design phase which minimizes the computational cost of the design. Such a cost is assumed to have a polynomial growth with the number of nodes in the subnet to be designed. We present optimum results both for the number of clusters, superclusters, etc., and for the number of hierarchical levels when the same design strategy (technology) is considered at all levels. Optimal clustering structures are also determined when different design strategies are considered, provided that the number

*Because of reliability requirements, tree networks are not considered in this study.

of levels is given. An expression for the average delay of a message in such a hierarchical network is also provided in terms of the average delays in the layer subnets composing the network. This decomposition should consequently lead to the design of smaller subnetworks for which we can utilize present design strategies. The problem of node assignment to clusters is not considered here; some nearness measures are, however, suggested.

Frank and others [9, 14] have shown from a feasibility study of a 1000 node network that, indeed, hierarchical structures are desirable for the design of large networks. They also posed the same questions concerning the optimal clustering structure but failed to answer them for the general case we consider (m levels).

2. HIERARCHICAL DESIGN OF COMPUTER NETWORKS: METHODOLOGY

The main idea behind the hierarchical design is to impose a decomposable structure on the design problem which will result in a set of smaller subproblems. In other words, we will introduce independence among subsets of design variables. The imposed independence will substantially reduce the set of feasible solutions and also, as a direct consequence, the computational cost. In doing so, there is the distinct risk of eliminating the optimal solution. Therefore, it is very important to seek "natural" decompositions.

Such decompositions will be realized through an m-level *h*ierarchical *c*lustering (MHC) of the set of nodes, based on some appropriate nearness measure. The MHC consists of grouping the network nodes (0th-level clusters) into 1st-level clusters, which in turn are grouped into 2nd-level clusters, etc. This operation continues in a bottom-up fashion until we group the $(m - 2)$nd-level clusters into $(m - 1)$st-level clusters whose union constitutes the mth-level cluster. The mth-level cluster is the highest level cluster, and as such, it includes all the nodes of the network. (This partitioning could also proceed in a top-down manner.) Because of the underlying MHC structure the m-level *h*ierarchical *t*opology design procedure will be denoted by MHT.

The assignment of nodes to clusters, clusters to superclusters, etc., may be realized through the application of various *ad hoc* clustering techniques [21-23]. These techniques generally use a nearness measure and try to determine natural groupings in terms of that measure. The nearness measures must take into account the cost of the different components of a communication network (switching nodes, channels, etc.), the traffic and reliability requirements, the delay of a message in the net, etc. Some examples of simple nearness measures s_{ij} are:

$$s_{ij} = 1/d_{ij} \quad \text{nearness between nodes } i \text{ and } j, \tag{i}$$

$$s_{ij} = \gamma_{ij}/(d_{ij})^{\epsilon}, \tag{ii}$$

where d_{ij} is the geographical distance between nodes i and j, γ_{ij} is the rate of traffic from i to j, and ϵ is some appropriate exponent.

Along with the hierarchical clustering of the nodes, we must select the gates (exchange nodes) for all clusters at all levels. The function of the gates from a given

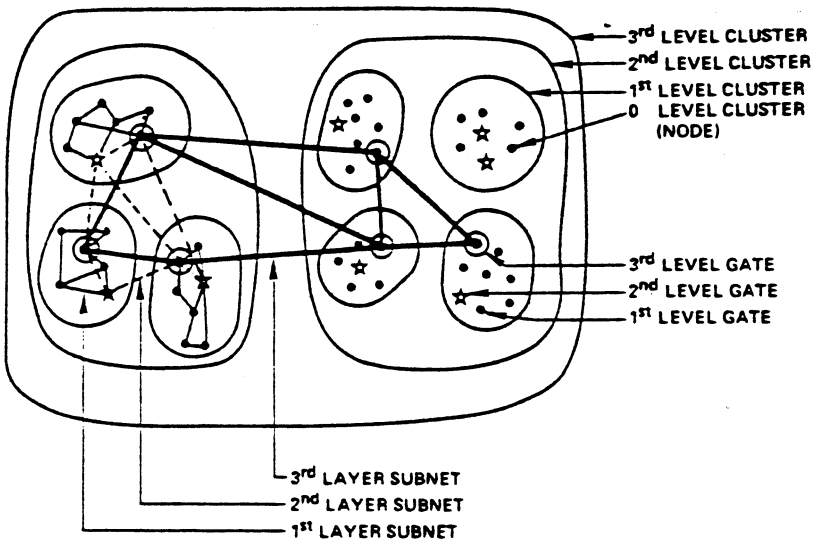

FIG. 1. A three-level hierarchical network.

cluster is to handle the traffic exchanged between the set of nodes in that cluster and those outside (Fig. 1). More specifically, the assumption underlying the flow of messages is as follows.

2.1 Flow Assumption

(a) Traffic between nodes in the same cluster, at any level, will only take paths which are *internal* to that cluster, i.e., paths contained in the corresponding local subnetwork.

(b) Traffic between nodes in different kth-level clusters ($k = 1, \ldots, m - 1$), but which belong to the same $(k + 1)$st-level cluster, will first be channeled via its local subnetwork to a $(k + 1)$st-level gate of the originating cluster; then it will take the $(k + 1)$st-layer subnetwork of gates to reach a $(k + 1)$st-level gate of the destination cluster, at which point it will be dispatched over the local subnetwork to finally reach the destination node. (This is a standard procedure in hierarchical networks.)

A kth-layer subnetwork is defined as a network connecting kth-level gates which belong to the same kth-level cluster.

Figure 1 illustrates some of the preceding definitions for a three-level hierarchical design.

Once the hierarchical classes are defined and the gates selected, then the previously developed network design techniques for moderate-sized networks may be used to design the layer subnetworks separately.

The direct application of the clustering techniques may lead to various nonoptimal cluster sizes, which will, in general, partially eliminate the computational gains obtained from optimal size clusters [23]. It is then important to determine those MHC structures which will minimize the computational cost incurred in the design phase of the MHT. In order to evaluate this cost, we make the following assumptions.

2.2 Computational Cost Assumption

(a) The computational cost incurred in the design of a kth-layer subnet connecting a set of n kth-level gates is equal to n^{α_k} ($k = 1, 2, \ldots, m$).

(b) The total computational cost involved in the total design is equal to the *sum* of the costs induced in the design of all the layer subnets.

The polynomial form of the computational cost is the one normally used [9] to characterize the computational complexity of most of the present design algorithms. The fact that different exponents α_k could be selected, depending on the level of the hierarchy, is provided to allow the modeling of the design of hierarchical networks where different technologies or design algorithms or both are considered at each level or group of levels. Clearly, the polynomial form with exponent exceeding unity represents a "loss of scale," and it is precisely this which we exploit by partitioning our design problem into a series of "smaller" problems.

2.3 Gate Assignment Rule

Given an integer valued vector $\beta = (\beta_1, \beta_2, \ldots, \beta_m)$, where $\beta_1 \triangleq 1$, and a selection rule, then starting at $k = 1$, select β_{k+1} $(k+1)$th-level gates among the set of kth-level gates of each kth-level cluster. Repeat this step sequentially until $k = m - 1$. A network node is considered to be a 1st-level gate (Fig. 1).

The choice of the vector β will be mainly related to the reliability constraint. If a K-connectivity constraint is to be imposed on the topology of the network, then the vector β must be such that $\beta_i \geq K$ for $i = 2, 3, \ldots, m$. This is obvious, since the set of the β_i ith-level gates of an $(i-1)$st-level cluster represents a cut set [24] for the other nodes in that cluster.

In summary, the main elements involved in the computational complexity of the MHT have been identified under a set of assumptions which, hopefully, retain the essential character of the class of hierarchical design procedures. We are now ready to state and solve the problem of finding an optimal clustering structure.

3. OPTIMAL CLUSTERING STRUCTURE

In this section we state and solve the optimal clustering structure problem.

3.1 Problem Statement

More definitions and notation are needed with regard to the hierarchical clustering structure. Any hierarchical classification scheme lends itself to a tree representation, as shown in Fig. 2 [25].

A *kth-level cluster*, C_k, is defined recursively as a set of $(k-1)$th-level clusters. It corresponds to a node (vertex) at level k in a tree representation.

A kth-level cluster is identified, similar to the Dewey notation, by a vector of predecessors, $i_{k+1} = (i_m, i_{m-1}, \ldots, i_{k+1})$. The notation $C_k(i_{k+1})$ will be used when there is a need to identify C_k. The index i_m indicates the $(m-1)$st-level cluster, say $C_{m-1}(i_m)$, to which C_k belongs; i_{m-1} indicates the $(m-2)$nd level cluster in $C_{m-1}(i_m)$ to which C_k belongs, etc.

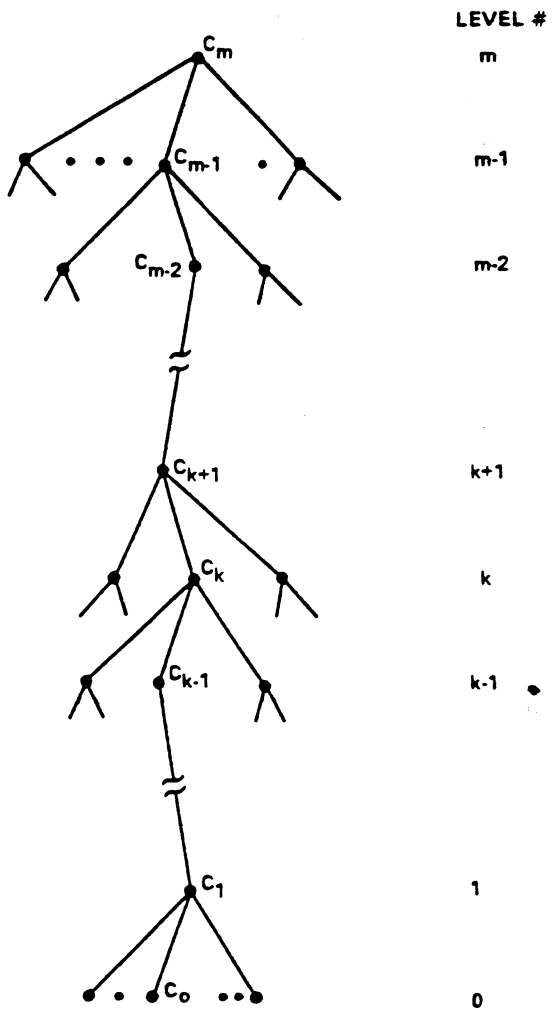

FIG. 2. Tree structure of the hierarchical clustering.

Note that a leaf in the tree representation corresponds to a node (0th-level cluster) in the network.

The *degree* of a kth-level cluster, C_k, is defined as the number of $(k-1)$st-level clusters included in C_k. It also indicates the downward degree of the corresponding node in the tree. We denote by $n_k(i_{k+1})$ the degree of $C_k(i_{k+1})$; we also define $\mathbf{n}_k \triangleq \{n_k(i_{k+1})\}$, where i_{k+1} is the vector of degrees of all the kth-level clusters. Moreover, we let $\mathbf{n} = (n_1, n_2, \ldots, n_m)$ be the degree vector. Finally, S will denote the set of nodes and N its size.

We are now ready to derive expressions for the size constraint, the gate constraint, and the computational cost.

The summation of the degrees of all the 1st-level clusters gives the total number of

nodes in the network (i.e., the total number of leaves in the tree structure). Hence,

$$N = \sum_{i_m=1}^{n_m} \cdots \sum_{i_k=1}^{n_k(i_{k+1})} \cdots \sum_{i_2=1}^{n_2(i_3)} n_1(i_2). \tag{1}$$

Equation (1) will generally serve as a constraint over the choice of the optimal degree vector n, and it will be referred to as the *size constraint*. To illustrate these definitions, we give an example of a two-level hierarchical clustering in the Appendix.

Let $g_k(i_{k+1})$ denote the number of kth-level gates in cluster $C_k(i_{k+1})$; it corresponds to the size of the kth-layer subnet which connects all the $(k-1)$st-level clusters of $C_k(i_{k+1})$. Because of the gate assignment rule, the following relation exists between degrees and sizes:

$$g_k(i_{k+1}) = \beta_k n_k(i_{k+1}). \tag{2}$$

Note that for $k = 1$, the set of 1st-level gates of a 1st-level cluster is simply the set of all the network nodes contained in that cluster (Fig. 1), hence $g_1(i_2) = n_1(i_2)$, which explains the convention $\beta_1 = 1$. For the above gate selection scheme to be feasible, the number of kth-level gates of a kth-level cluster C_k, must be no less than β_{k+1}, i.e., $g_k(i_{k+1}) \geq \beta_{k+1}$ $\forall k = 1, \ldots, m-1$. This relation, combined with Eq. (3), will henceforth introduce a constraint, to be referred to as the *gate constraint*, over the choice of the degree vector n characterizing the MHC structure.

Gate Constraint

$$n_k(i_{k+1}) \geq \beta_{k+1}/\beta_k, \qquad \forall\, k = 1, 2, \ldots, m-1, \qquad \forall\, i_{k+1}. \tag{3}$$

From the above considerations, the computational cost of the design of a kth-layer subnet is equal to $[g_k(i_{k+1})]^{\alpha_k} = [\beta_k n_k(i_{k+1})]^{\alpha_k}$. This expression, summed over all the kth-level clusters, represents the computational cost incurred in the design of all the kth-layer subnets, which we denote by $G_k(m, \mathrm{n}, \alpha, \beta)$ or simply G_k:

$$G_k(m, \mathrm{n}, \alpha, \beta) = \sum_{i_m=1}^{n_m} \sum_{i_{m-1}=1}^{n_{m-1}(i_m)} \cdots \sum_{i_{k+1}=1}^{n_{k+1}(i_{k+2})} [\beta_k n_k(i_{k+1})]^{\alpha_k},$$
$$G_m(m, \mathrm{n}, \alpha, \beta) = (\beta_m n_m)^{\alpha_m}. \tag{4}$$

Note that for $k = m$, by convention, we will set the multiple summation to unity.

The total computational cost is

$$G(m, \mathrm{n}, \alpha, \beta) = \sum_{k=1}^{m} G_k(m, \mathrm{n}, \alpha, \beta). \tag{5}$$

Finally, we have the following:

Problem Statement

$$\begin{aligned} \text{Given: } & N, \alpha, \beta \\ \text{Minimize: } & G(m, \mathbf{n}, \alpha, \beta) \quad \text{[See Eq. (5)]} \\ \text{Over: } & m \text{ and } \mathbf{n} \\ \text{Subject to: } & \text{size constraint} \quad \text{[See Eq. (1)]}, \\ & \text{gate constraint} \quad \text{[See Eq. (3)]}, \\ & m \text{ positive integer variable}, \\ & \mathbf{n} \text{ vector of positive integer variables.} \end{aligned} \tag{6}$$

3.2 General Solution

The precise problem expressed above is a nonlinear integer programming problem which has not been solved in its entirety. In order to make progress toward a solution, *we choose to relax some of the constraints, namely, the integer and gate constraints.* It is also necessary to temporarily freeze the variable m. The relaxation of the gate constraint is not of too much consequence, since it will be shown that for practical values of β, α, N, m, the optimal solution will satisfy that constraint. The added relaxation of the integer constraint will lead to a lovely analytic solution when m is given. (For some particular properties of the vectors α and β, an analytical solution is found for the optimal m.) The resulting real-valued solution is of considerable importance for at least two reasons: (1) the analytic study of its behavior, with respect to the variables m, α, β, will provide us with insight as to how much computational gain can be obtained through the application of the MHT design procedure and into the choice of the appropriate clustering structure. (2) A suboptimal integer solution could be directly obtained from the real-valued solution (thereby providing an upper bound on the cost). As a consequence of the relaxation of the integer constraint, one question arises as to what is the meaning of a discrete summation where the upper variable is not an integer. As an example, Eq. (A2) becomes meaningful only if n_2 is integer or if all $n_1(i_2)$'s are equal, say to n_1, in which case the summation becomes $n_2(n_1)^{\alpha_1}$. In fact, the solution of the optimization problem will show that clusters of the same level must be of the same degree; hence, all the summations in Eqs. (1) and (5) will become meaningful *a posteriori*!

Proposition 1. Given m, the number of levels in the hierarchy, and assuming that $\alpha_i > 1$ for all $i = 1, \ldots, m$, then the optimal clustering structure is such that:

(a) All clusters at the same level, $k = 1, \ldots, m$, are composed of an *equal* number of lower level clusters (i.e., all nodes at the same level in the tree representation are of equal degree). The optimal degree vector reduces to an m-dimensional vector, $\mathbf{n} = (n_1, n_2, \ldots, n_m)$ whose components are the solution of the following set of difference equations:

$$n_k(\mathrm{i}_{k+1}) = n_k \qquad \forall\, k = 1, \ldots, m \text{ and } \mathrm{i}_{k+1},$$

$$n_1 = \frac{N}{n_2 n_3 \cdots n_m},$$

$$n_k = \left[\frac{\prod_{i=1}^{k-1} (\alpha_i - 1)}{\alpha_k (\beta_k)^{\alpha_k}} \frac{B_k}{D_k} \left[\prod_{i=k+1}^{m} n_i \right]^{-\prod_{i=1}^{k-1} \alpha_i / D_k} \right]^{D_k / D_{k+1}}, \quad k = 2, 3, \ldots, m, \tag{7}$$

where

$$D_k = \sum_{j=2}^{k} \left(\prod_{i=j}^{k-1} \alpha_i \right) \left(\prod_{i=1}^{j-2} (\alpha_i - 1) \right) \qquad \forall\, k \geqslant 1, \tag{8}$$

and

$$\left[\frac{B_k}{D_k} \right]^{D_k} = \prod_{j=2}^{k-1} \left[\alpha_j^{-\alpha_j D_j} \left[\frac{\prod_{i=1}^{j-1} (\alpha_i - 1)}{(\beta_j)^{\alpha_j}} \right]^{-\prod_{i=1}^{j-1} (\alpha_i - 1)} \right]^{\prod_{i=j+1}^{k-1} \alpha_i} (N)^{\prod_{i=1}^{k-1} \alpha_i}, \qquad \forall k \geq 2. \tag{9}$$

(b) With this optimum solution, the minimum computational cost is

$$G(m, \alpha, \beta) = B_{m+1}. \tag{10}$$

The proof is given in [16]; the special $m = 2$ case is treated in the Appendix.

As a consequence of Proposition 1, the optimal size vector also reduces to an m-dimensional vector, $\mathbf{g} = (g_1, g_2, \ldots, g_m)$, whose components are given by

$$g_k = \beta_k n_k, \qquad k = 1, \ldots, m. \tag{11}$$

The combination of Eqs. (8) and (9), at $k = m + 1$, will provide an explicit expression for the minimum computational cost, $G(m, \alpha, \beta)$. [In what follows the shorter notation G and $G(m)$ will also be used.] The next natural step to take is to perform the *minimization of* G with respect to m. This may easily be done numerically when m is restricted to a certain range of integer values. However, for some particular choices of α and β, we can derive an *analytic* expression for the global optimum. Let us first consider the case where $\alpha_1 \leqslant 1$.

Proposition 2. Under the condition, $0 < \alpha_1 \leqslant 1$, the global optimum solution of our problem is achieved for $m = 1$, i.e., no clustering is required.

The proof is given in the Appendix. We now proceed with some special cases.

3.3 Optimal Clustering Structure with Uniform Design Strategy and Gate Assignment

This section deals with the special case where the same design procedure is used at all levels of the hierarchy. Also, an equal number of gates is selected from clusters at all levels. That is,

$$\begin{aligned} \alpha_k &= \alpha \quad \forall\, k \geq 1, \quad \alpha > 1, \\ \beta_k &= \beta \quad \forall\, k \geq 2, \quad \beta_1 \triangleq 1. \end{aligned} \tag{12}$$

As a result, much simpler expressions are obtained for $\mathbf{n}$ and G at optimality, given m. Furthermore, it is possible, analytically, to find the optimum number of levels in the hierarchy. At global optimality, all the layer subnets must be of equal size. This surprising result has the intuitive explanation given below.

Corollary 1. Given m, the number of levels in the hierarchy, and assuming* that $\alpha_i = \alpha$ for $i = 1, 2, \ldots, m$, that $\alpha > 1$, and that $\beta_i = \beta$ for $i = 2, 3, \ldots, m$, then the optimal clustering structure is such that

$$\begin{aligned} n_1 &= \beta \frac{\alpha}{\alpha - 1} \left[\left(\frac{\alpha - 1}{\alpha} \right)^m \frac{N}{\beta} \right]^{(\alpha-1)^{m-1}/D_{m+1}} \\ n_k &= \frac{\alpha}{\alpha - 1} \left[\left(\frac{\alpha - 1}{\alpha} \right)^m \frac{N}{\beta} \right]^{(\alpha-1)^{m-k}\alpha^{k-1}/D_{m+1}}, \quad k = 2, \ldots, m. \end{aligned} \tag{13}$$

With this solution, the minimum computational cost is

$$G(m, \alpha, \beta) = D_{m+1} \left[\left(\beta \frac{\alpha}{\alpha - 1} \right)^{\alpha(\alpha-1)D_m} \left(\frac{(\alpha - 1)^{(\alpha-1)^m}}{\alpha^{\alpha^m}} \right)^{m-1} N^{\alpha^m} \right]^{1/D_{m+1}}, \tag{14}$$

where

$$D_k = \alpha^{k-1} - (\alpha - 1)^{k-1} \quad \text{for } k \geq 1. \tag{15}$$

The proof [16] is a direct consequence of Proposition 1. From the above we make the following observations.

(1) With an optimal m-level structure, the computational cost is reduced from the order of N^α to $N^{[\alpha^m/(\alpha^m-(\alpha-1)^m)]}$. As an example, in Figs. 3 (and 5), we show this reduction; e.g., when $N = 100$ and $m = 10$, we gain seven orders of magnitude. (Note that the value at $m = 1$ corresponds to the nonhierarchical cost of N^α.)

(2) *Size vector:* From Eqs. (11) and (12),

$$g_1 = n_1, \quad g_k = \beta n_k, \quad k = 2, 3, \ldots, m.$$

*$\alpha \leq 1$ is treated in [16].

Substituting Eq. (13) into the above expression, we get an expression for g_k, valid for $k = 1, 2, \ldots, m$.

(3) *Gate constraint:* To satisfy the gate constraint, Eq. (3), the vector $\mathbf{n}$ must be such that

$$n_1 \geq \beta, \quad n_k \geq 1, \quad k = 2, \ldots, m$$

This condition will always be satisfied if

$$\left(\frac{\alpha - 1}{\alpha}\right)^m \frac{N}{\beta} \geq 1, \quad \text{i.e., } m \leq \ln \frac{N}{\beta} \Big/ \ln \frac{\alpha}{\alpha - 1} \tag{16}$$

Below, we show that the region of interest for m will effectively correspond to the above condition.

Proposition 3 (Global optimum). Under the conditions of Eq. (12) and for m a real variable, the global optimum† clustering structure is achieved for a number of levels

$$m_* = \ln \frac{N}{\beta} \Big/ \ln \frac{\alpha}{\alpha - 1} \tag{17}$$

and a degree vector n^*

$$\begin{aligned} n_1^* &= \beta \frac{\alpha}{\alpha - 1}, \\ n_k^* &= \frac{\alpha}{\alpha - 1}, \quad k = 2, 3, \ldots, m_*. \end{aligned} \tag{18}$$

The corresponding minimum computational cost is

$$G_*(\alpha, \beta) = \left(\frac{N}{\beta} - 1\right) \beta^\alpha \frac{\alpha^\alpha}{(\alpha - 1)^{\alpha - 1}}. \tag{19}$$

The proof results directly from the minimization of $G(m, \alpha, \beta)$, which we differentiate with respect to m [16].

Now for some comments regarding the global minimum solution.

1. *Cost:* At global optimality the computational cost is reduced to the order of N. (This is to be compared to the nonhierarchical cost of N^α.)

2. *Size vector:* At global optimality, the size vector is such that

$$g_k^* = \beta \frac{\alpha}{\alpha - 1}, \quad k = 1, 2, \ldots, m_* \tag{20}$$

†"Global optimum" refers to the optimum solution with respect to m. Also, * indicates values at global optimality.

Equation (20) indicates that at optimality, all the layer subnets are of equal size, which depends only on α and β. The explanation of this very simple and interesting property is given below.

3. *Gate constraint:* From Eq. (20), since $\alpha > 1$,

$$g_k^* > \beta, \quad k = 1, 2, \ldots, m.$$

Hence, the gate constraint, Eq. (3), is always satisfied at optimality. Moreover, as noticed in Eq. (16), the optimal solution, given $m \leq m_*$, will also satisfy the gate constraint.

4. *Number of layer subnets:* Equation (19) may be rewritten as

$$G_* = \left(\frac{N}{\beta} - 1\right)(\alpha - 1)\left(\beta \frac{\alpha}{\alpha - 1}\right)^{\alpha}. \tag{21}$$

From Eq. (21), we conclude that the number of layer subnets is

$$NL_* = \left(\frac{N}{\beta} - 1\right)(\alpha - 1) \tag{22}$$

each of size $\beta\alpha/(\alpha - 1)$ and each of computational cost $\beta[\alpha/(\alpha - 1)]^{\alpha}$. The above results could also be derived by counting the number of clusters in the tree structure (Fig. 2).

5. *Cost distribution:* For practical purposes, $N/\beta \gg 1$; then from Eq. (19) we have $G_* \simeq \alpha(N/n_1^*)[\beta\alpha/(\alpha - 1)]^{\alpha}$. Let G_1^* be the computational cost of designing the 1st-layer subnet; then since N/n_1^* is the total number of 1st-layer subnets, we get

$$G_* \simeq \alpha G_1^*,$$

which says that the design of the 1st-layer subnets represents approximately $1/\alpha$ of the total computational cost.

6. *Behavior of the optimal tree structure with respect to α:* We will assume that $\beta = 1$. Let us define a *regular tree* of degree K as a tree whose nodes are all of equal downward degree K, except for the leaves (downward degree zero) [25]. As an example, a binary tree is a regular tree with degree 2.

The global optimum solution given in Eq. (18) becomes, for $\beta = 1$,

$$n_k^* = \frac{\alpha}{\alpha - 1}, \quad k = 1, 2, \ldots, m_*.$$

We are interested in the set of α's which yield integer solutions, i.e., which correspond to regular tree structures. If K is the degree of such trees, then α must be such that

$$\alpha/(\alpha - 1) = K \Longleftrightarrow \alpha = K/(K - 1)$$

Moreover, if a regular tree of degree K is composed of m_* levels, then it contains K^{m_*} leaves. Consequently, there is a one-to-one correspondence between the set of regular trees of degree K ($K \geq 2$ integer) whose number of levels is m_* (integer ≥ 2) and the global optimal solutions, where $\alpha = K/(K - 1)$ and $N = K^{m_*}$ for $K = 2, 3, \ldots, \infty$. Notice that the above set of α's is contained in the interval (1, 2) of real values i.e., $1 < \alpha \leq 2$. Also, $\alpha = 2$ corresponds to a *binary tree* representation.

7. *Irreducibility*. The simplicity of the solution [Eq. (20)] obtained at global optimality leads us to consider a more intuitive approach toward its derivation based on irreducibility considerations.

Let us define an *irreducible* set of nodes as a set which is such that no computational gain can be obtained through the application of the MHT for the design of the corresponding communication network. By contradiction, it is obvious that at global optimality, each layer subnet to be designed must correspond to an irreducible set of nodes. Let q be the size of an irreducible set. Then from Eqs. (1) and (2), we get $n_* = q$, $n_k = q/\beta$, $k = 2, \ldots, m$, and $q = \beta(N/\beta)^{1/m}$. Replacing n_k by its value in Eq. (5), we arrive at an expression for the cost G in terms of m whose minimum is attained for $m = m_*$ [16].

Note that we assumed the size of an irreducible set to be unique. This is justified *a posteriori*, since an irreducible set must be such that $m_* = 1$; hence its size N_0 is a solution of

$$\ln \frac{N_0}{\beta} \Big/ \ln \frac{\alpha}{\alpha - 1} = 1,$$

which yields a unique solution.

Further properties of the optimal solution and a numerical study are provided in Sec. 4.

3.4. Uniform Design Strategy: Variable Gate Assignment

This section deals with the case where $\alpha_k = \alpha \ \forall k$ and where β_k is arbitrary for each k.

The purpose is to model design requirements whereby different reliability constraints† are imposed on the design of the layer subnets, depending on their level in the hierarchy. However, the same design procedure is applied at all levels (i.e., $\alpha_k = \alpha$).

Corollary 2. Under the conditions of Proposition 1, and assuming that all the α_k's are equal, the optimal solution is, for $k = 1, 2, \ldots, m$,

†In [16] we derive some partial results related to global optimality when β_k is of the form $\beta_k = \gamma^{k-2}\beta$, $k \geq 2$. We also study the special case of a proportional gate assignment. In this scheme the number of kth-level gates to be selected is proportional to the number of $(k - 1)$st-level gates from which they are selected. The solution was found to be degenerate.

$$n_k = \frac{\alpha}{\alpha - 1}\left[\left(\frac{\alpha - 1}{\alpha}\right)^m N\right]^{(\alpha-1)^{m-k}\alpha^{k-1}/D_{m+1}}$$

$$\times \left[\frac{\prod_{i=k+1}^{m} (\beta_i)^{\alpha^{m+k-i}(\alpha-1)^{i-k-1}} \prod_{i=1}^{k-1} (\beta_i)^{\alpha^{k-i}(\alpha-1)^{m-k+i-1}}}{\beta_k^{\alpha D m}}\right]^{1/D_{m+1}} . \quad (23)$$

With this optimal assignment, the minimum computational cost is

$$G = D_{m+1}\left[\left(\prod_{i=2}^{m} \beta_i^{\alpha^{m+1-i}(\alpha-1)^{i-1}}\right)\left(\frac{\alpha}{\alpha - 1}\right)^{\alpha(\alpha-1)Dm}\right.$$

$$\left. \times \left(\frac{(\alpha - 1)^{(\alpha-1)^m}}{\alpha^{\alpha^m}}\right)^{m-1} N^{\alpha^m}\right]^{1/D_{m+1}} . \quad (24)$$

4. VARIATIONS AND LIMITING BEHAVIOR OF THE OPTIMAL SOLUTION WITH RESPECT TO THE DESIGN VARIABLES: UNIFORM DESIGN AND GATE ASSIGNMENT

The behavior of the computational cost and the degree vector at optimality, given m, will now be studied with respect to m. This study is restricted to the uniform design strategy and gate assignment case, i.e., $\alpha_k = \alpha \ \forall k$ and $\beta_k = \beta \ \forall k \geq 2$. Of importance is the fact that the minimum cost at a given m, $G(m)$ (also denoted by G), converges *fairly fast* to its minimum value, versus m, and remains very close to that optimum value as m grows to infinity. A similar phenomenon will characterize the behavior of the n_k's versus m. We will also notice that as α goes to infinity, the optimal solution, given m, is such that all layer subnets must be of equal size; such a property was, for finite α, only true at global optimality, i.e., $m = m_*$.

The variations of G_* and m_* with respect to α, β, N do not disclose any remarkable property, except that m_* rapidly reaches its asymptotic value as α becomes greater than 2.

In what follows, we will restrict the study to the practical situation where

$$\beta \geqq 1, \quad N/\beta > 1. \quad (25)$$

4.1 Behavior of the Optimal Solution and Objective Function, Given m

$G(m)$ versus m

Section 3.3 showed that as m increases from zero to infinity, $G(m)$ decreases, reaching its minimum at $m = m_*$ [note $m_* > 0$ because of Eq. (25) and $\alpha > 1$], and then increases. The limit of $G(m)$, as m goes to infinity is

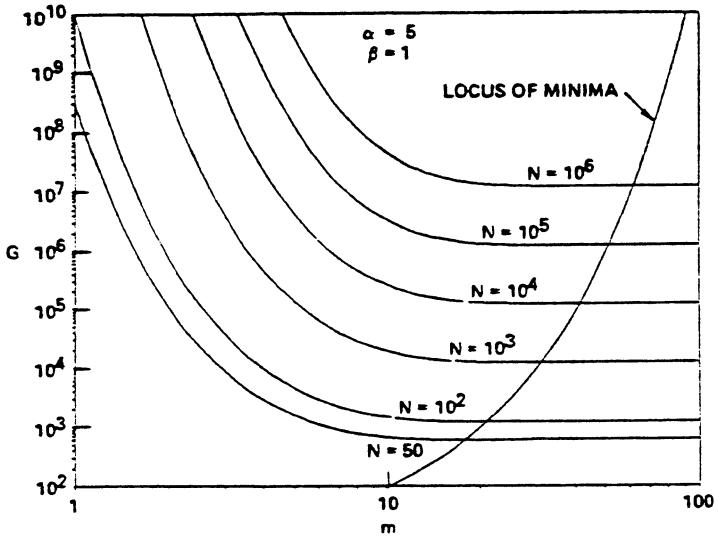

FIG. 3. Minimum computational cost $G(m, \alpha, \beta)$ given m; $\alpha = 5, \beta = 1$.

$$G_\infty \triangleq \lim_{m \to \infty} G(m) = \beta^{\alpha-1} \frac{\alpha^\alpha}{(\alpha - 1)^{\alpha-1}} N.$$

Note that $G_\infty - G_* = \beta^\alpha [\alpha^\alpha/(\alpha - 1)^{\alpha-1}]$ is independent of N and that the normalized difference $(G_\infty - G_*)/G_* = \beta/(N - \beta)$ goes to zero as N grows.

In what follows, two sets of figures (Figs. 3-6), each corresponding to a specific value of the pair (α, β), $\{(5, 1)\ (5, 3)\}$, will be shown. In each set, the functions $G(m)$

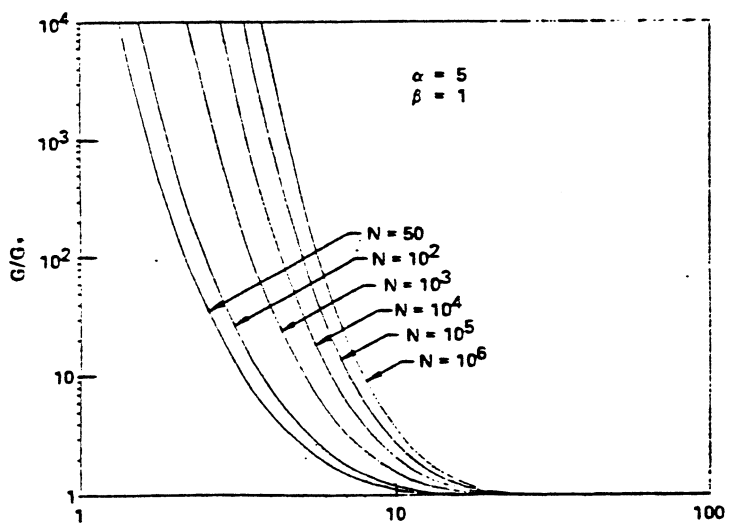

FIG. 4. Ratio of computational cost at optimality given m and at global optimality G/G_*; $\alpha = 5, \beta = 1$.

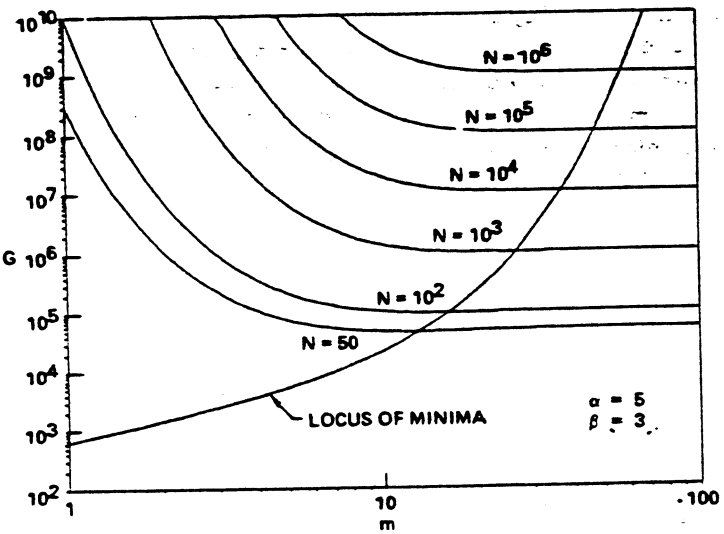

FIG. 5. Minimum computational cost $G(m, \alpha, \beta)$ given m; $\alpha = 5, \beta = 3$.

and $G(m)/G_*$ are plotted with respect to m for several values of N, $N = \{50, 10^2, 10^3, 10^4, 10^5, 10^6\}$. The curves $G(m)$ versus m illustrate the initially decreasing, then slightly increasing and asymptotic behavior of the optimal computational cost for a fixed m. By comparing $G(m)$ to N^α, we are able to appreciate the enormous computational gains obtained through the application of the MHT.

The curves $G(m)/G_*$ versus m illustrate a "clamping" effect whereby, once $G(m)$ reaches its minimum value of $m = m_*$, Eq. (17), it will appear as if it remains indef-

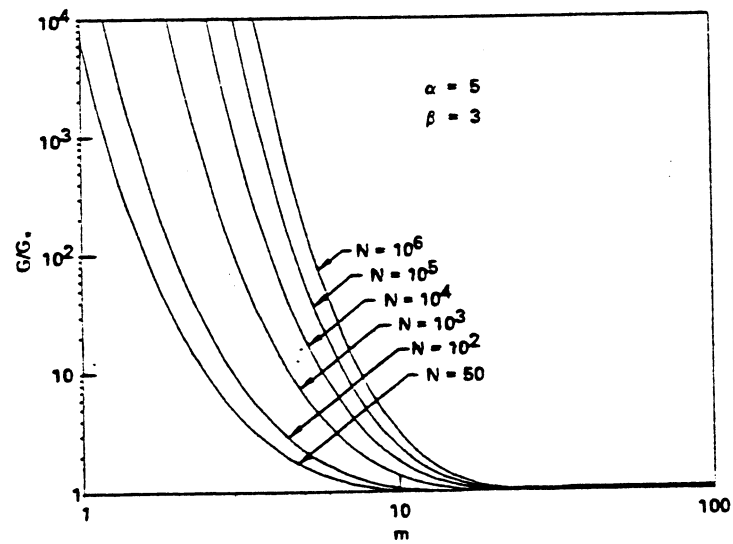

FIG. 6. Ratio of computational cost at optimality given m and at global optimality G/G_*; $\alpha = 5, \beta = 3$.

initely at that value. They also illustrate the fairly fast convergence of $G(m)$ toward a value close to the minimum, for a value of m relatively smaller than m_*. This indicates that we may actually obtain most of the computational gains with hierarchical structures whose number of levels (m) is *much smaller* than the optimal ones (m_*). Finally, the two sets of figures are presented in order to indicate the effect of the design parameters α and β on the behavior of the aforementioned functions.

n_k versus m

Differentiating Eq. (13) with respect to m, we find, after simplification,

$$\frac{dn_k}{dm} = n_k \frac{(\alpha - 1)^{m-k} \alpha^{m+k-1}}{D_{m+1}^2} \left(\ln \frac{\alpha}{\alpha - 1}\right) \left[\left(\frac{\alpha - 1}{\alpha}\right)^m - 1 - \ln \frac{N}{\beta} + m \ln \frac{\alpha}{\alpha - 1}\right].$$

For $\alpha > 1$ and $m \geq 0$, dn_k/dm has a unique root m_0 which is *independent* of k. Also for $m < m_0$ the derivative is negative, and conversely, for $m > m_0$ it is positive. Consequently, m varies from zero to infinity

(i) all n_k's, $k < m_0$, decrease, reach a minimum, and then increase toward their limits.

(ii) all n_k's, $k > m_0$, increase toward their limits.

The limit of n_k for k fixed, as m goes to infinity, is

$$\lim_{m \to \infty} n_1 = \beta \frac{\alpha}{\alpha - 1},$$

$$\lim_{m \to \infty} n_k = \frac{\alpha}{\alpha - 1}, \qquad k \geq 2, k \textit{ fixed}.$$

If we let k vary, more particularly if $k = m$, then

$$\lim_{m \to \infty} n_m = 0.$$

Moreover, $\ln n_m$ has as an asymptote, a straight line whose equation is

$$-\frac{m}{\alpha} \ln \frac{\alpha}{\alpha - 1} + \ln \frac{\alpha}{\alpha - 1} \left(\frac{N}{\beta}\right)^{1/\alpha}$$

As can be seen in Fig. 7, the behavior of n_k versus m, for $k \leq m_*$, exhibits a "clamping" effect as previously described. Moreover, this phenomenon is quite remarkable here, since the limit of n_k, as m goes to infinity, is equal to the value of n_k at global optimality. In Fig. 7 we see our old result that $n_k = \alpha/(\alpha - 1)$ for $k > 1$ at $m = m_*$. This study of how n_k behaves as m deviates from its optimal value at m_* indicates that as long as $m > m_*$, then most of the n_k take on that value which they would have at optimality anyway [i.e., $\alpha/(\alpha - 1)$]. Some of the n_k (in the vicinity of $k \approx m$) deviate considerably from $\alpha/(\alpha - 1)$, but as we saw in Figures 3-6, this hardly affects the cost G.

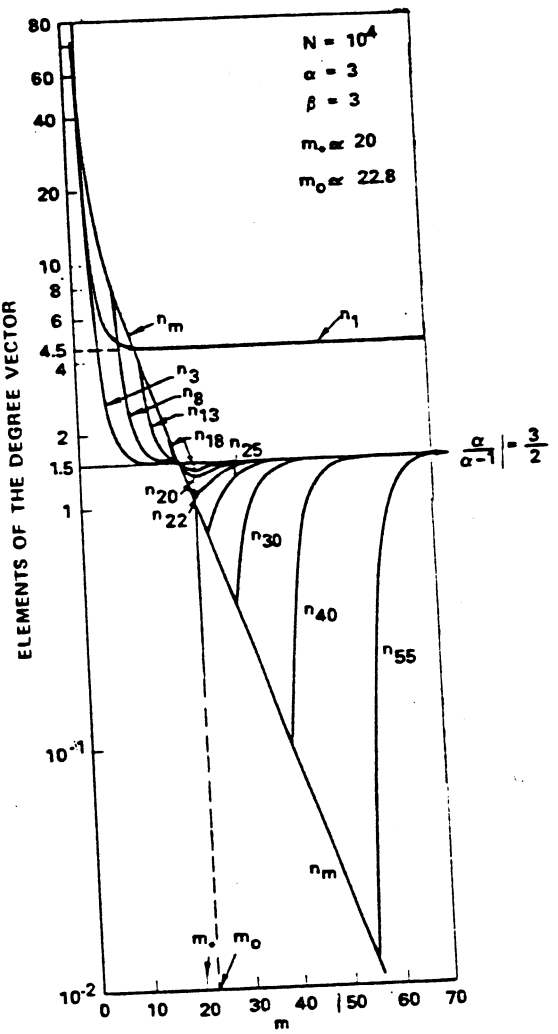

FIG. 7. Variations of the optimal degrees with respect to m.

4.2 Behavior of G_* and m_* with Respect to α, β, N

G_* versus α

Differentiating Eq. (19) with respect to α, we find

$$\frac{dG_*}{d\alpha} = \left(\frac{N}{\beta} - 1\right) \beta^{\alpha} \frac{\alpha}{(\alpha - 1)^{\alpha - 1}} \ln \beta \frac{\alpha}{\alpha - 1}.$$

Under the conditions of Eq. (25) and $\alpha > 1$, G_* is an increasing function of α. As α goes to infinity, G_* will be asymptotic to the expression $e\beta^{\alpha}(\alpha - 1/2)[(N/\beta) - 1]$.

If $\beta = 1$ the asymptote is a straight line, which implies that at the limit, G_* will show

a *linear* growth with α. This is surprising, since the original computational cost is proportional to the α *power* of the number of nodes.

G_* versus N

Equation (19) shows that G_* varies linearly with N. It is also important to note that the optimal hierarchical structure reduces the computational cost from the order of N^α steps to the order of N steps!

G_* versus β

Differentiating Eq. (19) with respect to β, we arrive at

$$\frac{dG_*}{d\beta} = \frac{\alpha^\alpha}{(\alpha - 1)^{\alpha-1}} \beta^{\alpha-2} [(\alpha - 1)N - \alpha\beta].$$

Consequently, G_* is an increasing and then a decreasing function, as β varies from 1 to N [for $N > \alpha/(\alpha - 1)$ and under the conditions of Eq. (25)]. Equating the derivative to zero, we find

$$\beta_0 = \frac{\alpha - 1}{\alpha} N. \tag{26}$$

Substituting Eq. (26) into Eq. (19), we arrive at the maximum value of G_* (with respect to β), $G_*(\beta = \beta_0) = N^\alpha$. This maximum cost corresponds to a design with *no* hierarchical structure. This last equation checks with the fact that for a set of α, β, N, satisfying Eq. (26), the optimal number of levels, m_*, is equal to unity [Eq. (17)], i.e., for such a set, a nonhierarchical design is optimal.

m_* versus N and β

From Eq. (17), we see that m_* varies as the logarithm of N/β. This logarithmic behavior appears to be characteristic of hierarchical structures [16, 17, 27].

m_* versus α

Differentiating Eq. (17) with respect to α, we find

$$\frac{dm_*}{d\alpha} = \frac{\ln (N/\beta)}{\alpha(\alpha - 1)[\ln (\alpha/(\alpha - 1))]^2}.$$

Consequently, for $\alpha > 1$, m_* is an increasing function of α. The limiting values of m are

$$\alpha \to 1^+ \Rightarrow \begin{cases} m_* \to 0, \\ \dfrac{dm_*}{d\alpha} \to \infty, \end{cases}$$

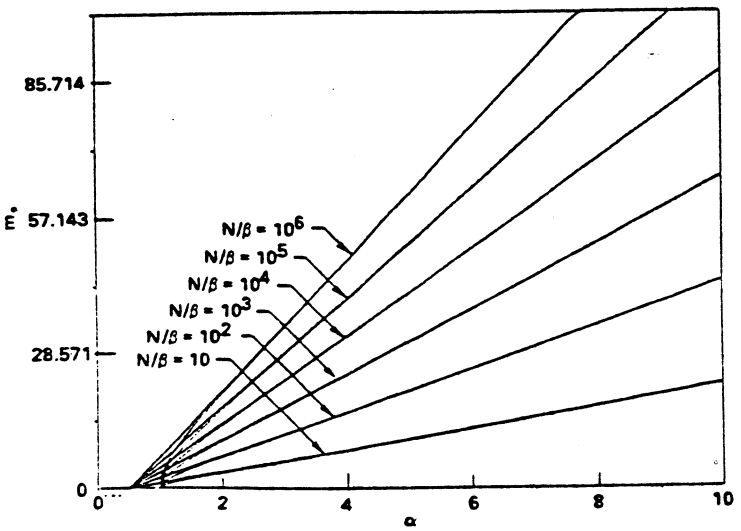

FIG. 8. Optimal number of levels in the hierarchical design.

$$\alpha \longrightarrow +\infty \Rightarrow \begin{cases} m_* \longrightarrow +\infty, \\ \dfrac{dm_*}{d\alpha} \longrightarrow \ln (N/\beta). \end{cases}$$

The asymptote, as α goes to infinity, is the straight line $m_* = (\alpha - 1/2) \ln (N/\beta)$. Figure 8 shows the plots of m_* versus α for several values of N/β. Note that m_* rapidly reaches its asymptotic value as α becomes greater than 2. Also, $m_* = 1$ for $\alpha = (N/\beta)/(N/\beta - 1)$, which means that for $1 \leqslant \alpha \leqslant (N/\beta)/(N/\beta - 1)$, *no partitioning* is required; furthermore, a loss will be incurred if we try to do so. This was seen to be true also for $0 \leqslant \alpha \leqslant 1$, where no partitioning is also optimal.

4.3 Limiting Behavior of the Optimal Degree Vector, Given m, with Respect to α

From Eqs. (13) and (15), we obtain

$$\alpha \longrightarrow 1^+ \Rightarrow \begin{cases} n_k \longrightarrow +\infty, & \text{for all } k\text{'s}, k \neq m, \\ n_m \longrightarrow 0. \end{cases}$$

In other words, since the limit of n_m is zero, for any given m, $m \geqslant 2$, then no clustering must be allowed. This confirms the earlier result obtained when looking at the behavior of m_* with respect to α. Also,

$$\alpha \longrightarrow +\infty \Rightarrow \begin{cases} n_1 \longrightarrow \beta \left(\dfrac{N}{\beta}\right)^{1/m}, \\ n_k \longrightarrow \left(\dfrac{N}{\beta}\right)^{1/m}, \quad k = 2, \ldots, m. \end{cases} \tag{27}$$

Thus,

$$\alpha \to +\infty \Rightarrow g_k \to \beta \left(\frac{N}{\beta}\right)^{1/m}, \quad k = 1, 2, \ldots, m.$$

Consequently, as α goes to infinity, the optimal solution, given any m, is such that all layer subnets are of equal size. Such a property was found to be true for finite α, only at global optimality. Note the same result holds true when the β_k's are different; in [16] we show that

$$\lim_{\alpha \to +\infty} n_k = \frac{1}{\beta_k} \left[N \prod_{i=1}^{m} \beta_i\right]^{1/m}, \quad k = 1, 2, \ldots, m.$$

This terminates our study of the optimal solution with a uniform design strategy and gate assignment. In [16], further extensions are presented. Namely, the optimal solution G is compared with two intuitive solutions. The R-solution (R stands for root) corresponds to a regular tree representation, i.e., all degrees, at all levels, are equal:

$$r \triangleq n_k, \quad k = 1, 2, \ldots, m.$$

The Q-solution models the situation where all layer subnets are of equal size:

$$q \triangleq g_k, \quad k = 1, 2, \ldots, m.$$

The properties of these three solutions are as follows [16]:

1. All three solutions possess an optimal number of levels, which is the same for G and Q (namely, m_*).
2. For large values of m, the G-solution is asymptotic to a constant, whereas the Q and R solutions asymptotically grow linearly with m. In other words, for $m \geq m_*$, G, contrary to Q and R, is not sensitive to m.
3. The G and Q solutions meet, for any α, at $m = m_*$ and, for $\alpha \to \infty$, at any given m.
4. The Q and R solutions are identical for $\beta = 1$.
5. As expected, G is always smaller than or equal to Q and R, and for $\beta \neq 1$, Q is certainly better than R for values of m in the neighborhood of m_*, but not necessarily outside.

Finally, a heuristic algorithm has also been developed [16] in order to generate an integer suboptimal solution from the optimal real-valued solution.

So far we have determined optimal structures for the hierarchical design of computer networks. A few questions remain unstudied, namely, the actual assignment of nodes to clusters, the decomposition of global variables in terms of secondary variables related to the different levels in the hierarchy, etc. The decomposition of the average delay is performed in Sec. 5.

5. DELAY EXPRESSION FOR HIERARCHICAL NETWORKS

Given a network with an m-level hierarchical structure, the problem is to express the total average delay in the network in terms of the average delays in the layer subnets composing the network. We assume that part (a) of Proposition 1 is true, i.e., all kth-level clusters are composed of an equal number of $(k-1)$st-level clusters n_k, $(k = 1, \ldots, m)$. The n_k's are not necessarily optimal. We also assume that all kth-layer subnets induce the same average delay, T_k, over the traffic. This last assumption will usually appear as a design constraint, and as such, it is a reasonable one.

The traffic in hierarchical networks may be divided into m classes. *Class k traffic* is defined as the traffic between pairs of nodes which belong to the same kth-level cluster but not to any lower level clusters. If we define τ_k as the average delay incurred by class k traffic and γ_{ij} as the average message rate from source i to destination j, then

$$T = \sum_{k=1}^{m} \frac{\Gamma_k}{\Gamma} \tau_k, \tag{28}$$

where Γ is the total external traffic, i.e., $\Gamma = \Sigma_{ij}\gamma_{ij}$, T is the average delay of a message in the network [7], and Γ_k is the total class k traffic.

From the Flow Assumption we know that class k traffic, when going from its origin to its destination, has to go up through $k-1$ layers to the kth common layer, then down $k-1$ layers.

Consequently,

$$\tau_k = 2 \sum_{i=1}^{k-1} T_i + T_k, \quad k \geq 1. \tag{29}$$

Substituting Eq. (29) into Eq. (28), we arrive at

$$T = \frac{1}{\Gamma} \sum_{k=1}^{m} \left(\Gamma_k + 2 \sum_{i=k+1}^{m} \Gamma_i \right) T_k, \tag{30}$$

which is an expression for T in terms of the T_k's.

Γ_k may also be evaluated using the flow assumption, but in general, it will yield too complicated an expression. If we assume a *uniform traffic* pattern, i.e., $\gamma_{jk} = \gamma \ \forall \ (j, k)$, then

$$\Gamma_k = N(n_k - 1) \left(\prod_{i=1}^{k-1} n_i \right) \gamma, \quad k \geq 1. \tag{31}$$

Substituting Eq. (30) into Eq. (31), after some algebra we arrive at the result stated in Proposition 4.

Proposition 4. Under the above assumptions, the total average delay in a hierarchical network is

$$T = \frac{1}{N-1}\left[\left(N - \prod_{k=1}^{m-1} n_k\right) T_m + \sum_{k=1}^{m-1}\left(2N - (n_k+1)\prod_{i=1}^{k-1} n_i\right) T_k\right].$$

As an example,† let $N = 10^3$, $n_1 = n_2 = n_3 = 10$; then $T = 0.9T_3 + 1.82T_2 + 1.99T_1$.

6. SUMMARY

In this paper we studied the major aspects related to the hierarchical design of large computer networks. The focus was primarily on the determination of a certain clustering structure of the set of nodes to be used in the design phase. Optimal clustering structures were determined so as to minimize the computational cost required in the design phase. The general solution (i.e., different design strategies and gate assignment from one level to another) was derived when the number of hierarchical levels m is fixed. The global optimum solution was obtained with the more uniform case, whereby the same design strategy and gate assignment are used at all levels. The global optimum solution is such that all layer subnets are of equal size. Such a peculiar property was found to have an intuitive explanation. Furthermore, we mentioned the one-to-one correspondence of the global optimum solution for $\alpha = K/(K-1)$ with some regular trees of downward degree K.

Perhaps the most significant result of this work was to show in the case $\alpha_k = \alpha$, $\beta_k = \beta$ that the cost of design reduced from N^α (for a nonhierarchical design) to a cost simply proportional to N. We conjecture this to be true for more general cases as well.

We were also able to decompose the average message delay in a hierarchical network in terms of the average delays in the layer subnets composing the network.

Finally, it appears that the general methodology and decomposition models developed here for the design of large nets may be directly applicable or extended to more general large systems where some sort of decomposition must be introduced to alleviate the difficulty in analysis, design, or evaluation.

APPENDIX: PROOFS OF PROPOSITIONS

Proof of Proposition 1

The general proof consists of showing that when all other variables are fixed, the $n_1(\cdot)$'s must all be equal (to n_1). Then, n_1 is replaced by its optimal expression in the objective function, and the same operation is repeated for the degrees at the next level, and so forth, until all levels are exhausted. Here we consider a two-level hierarchical clustering composed of n_2 1st-level clusters. Let i_2 ($i_2 = 1, 2, \ldots, n_2$) denote an arbitrary 1st-level cluster, and $n_1(i_2)$ be the corresponding number of nodes, then

$$N = \sum_{i_2=1}^{n_2} n_1(i_2). \tag{A1}$$

†A similar expression, for that specific example, $T = T_3 + 2T_2 + 2T_2$ is given in [19], however, no generalization is shown there.

The computational design cost of the 1st-layer subnets is

$$G_1 = \sum_{i_2=1}^{n_2} [n_1(i_2)]^{\alpha_1}. \tag{A2}$$

There is also a unique 2nd-layer subnet connecting all 2nd-level gates. Each 1st-level cluster contributes β_2 2nd-level gates, thus $G_2 = [\beta_2 n_2]^{\alpha_2}$. The total cost of G is then equal to

$$G = G_1 + G_2. \tag{A3}$$

We now proceed with the proof.

First n_2 is fixed. Solve the problem with respect to $n_1(i_2), i_2 = 1, 2, \ldots, n_2$.

The objective function Eq. (A3), being a sum of power functions $[n_1(i_2)]^{\alpha_1}$, which are convex in the region $\{n_1(i_2) > 0$, for all i's$\}$, is itself a convex function in that region (recall $\alpha_1 > 1$). Let us take the Lagrangian:

$$L(\mathrm{n}_1, \lambda) = \sum_{i_2=1}^{n_2} [n_1(i_2)]^{\alpha_1} - \lambda \left[\sum_{i_2=1}^{n_2} n_1(i_2) - N \right],$$

where $\lambda \geqslant 0$.

Note that we discarded the constant term $(\beta_2 n_2)^{\alpha_2}$ and the positivity constraint, which we will check *a posteriori*. In the region $\{\mathrm{n}_1 > 0\}$, the Lagrangian is a convex function, and if the solution of $\nabla L_1 = 0$, where ∇ denotes the gradient operator [26], is such that $\mathrm{n}_1 > 0$, then it must be optimal:

$$\nabla L = 0 \Rightarrow \frac{\partial L}{\partial n_1(i_2)} = 0 \qquad \forall\, i_2 = 1, \ldots, n_2.$$

Hence,

$$\alpha_1 [n_1(i_2)]^{\alpha_1 - 1} - \lambda = 0 \Rightarrow n_1(i_2) = \left(\frac{\lambda}{\alpha_1}\right)^{1/(\alpha_1 - 1)} \qquad \forall\, i_2 = 1, \ldots, n_2,$$

which means that all $n_1(i_2)$'s are equal, and therefore, from the size constraint,

$$n_1(i_2) = n_1 = N/n_2,$$

which is greater than zero. The rest is an optimization over the single variable n_2, whose solution satisfies Eq. (7) where m is set equal to 2. The corresponding minimal computational cost is given by Eq. (10) where m is also set equal to 2.

Proof of Proposition 2

By contradiction, assume that $m > 1$; then, similar to the proof of Proposition 1 [16], let us fix m ($m > 1$, integer) and all degrees, $n_k(\mathrm{i}_{k+1})$, $(k = 2, \ldots, m)$, and solve

the reduced problem with respect to $n_1(i_2)$'s. i.e.,

$$\text{min: } G_1 = \sum_{i=1}^{NC_1} [n_1(i)]^{\alpha_1} \qquad \text{[see Eq. (4)]},$$

$$\text{over: } n_1(i),$$

$$\text{s.t.: } \sum_{i=1}^{NC_1} n_1(i) = N \qquad \text{[see Eq. (1)]},$$

$$n_1(i) \geqslant 0 \text{ real variable.}$$

where NC_1 denotes the number of 1st-level clusters, and $n_1(i)$ denotes the size of an arbitrary 1st-level cluster. Since $\alpha_1 \leqslant 1$, then $[n_1(i)]^{\alpha}$ is a concave function; hence, the objective function above, G_1, is a concave function. Also, the set of *feasible* vectors, $\mathbf{n}_1 = \{n_1(i)\}$ is a bounded convex polyhedron whose vertices correspond to vectors with all zeros except for one component equal to N, i.e., $(N, 0, 0, \ldots, 0)$ and its permutations. As a result, we are faced with the optimization of a concave function over a bounded convex polyhedron, whose optimal solution is well known to be at a vertex. At any vertex, $G_1 = N^{\alpha}$; consequently, it is the minimum. Notice that this result is true for any NC_1, i.e., for any $n_k(i_{k+1})$ $(k = 2, \ldots, m)$. In other words, for any vector $\mathbf{n}$ satisfying the size constraint, Eq. (1), $G_1 \geqslant N^{\alpha_1}$, which, combined with Eq. (5), gives

$$G(m, \mathbf{n}, \alpha, \beta) \geqslant N^{\alpha_1} + \sum_{k=2}^{m} G_k(m, \mathbf{n}, \alpha, \beta) \qquad \forall\, \mathbf{n}, \text{ feasible.}$$

Hence,

$$G(m, \mathbf{n}, \alpha, \beta) > N^{\alpha_1} \qquad \forall\, \mathbf{n}, \text{ feasible.}$$

Consequently, the optimal solution to our problem (as stated at the end of Sec. 3.2) for integer $m > 1$, must also satisfy the above inequality which is a contradiction, since, if $m = 1$, $G(m, \mathbf{n}, \alpha, \beta) = N^{\alpha_1}$.

References

[1] L. G. Roberts, "Multiple Computer Networks and Intercomputer Communications," Proceedings of the ACM Symposium on Operating Systems, Gatlinburg, TN, October 1967.

[2] S. Carr, S. Crocker, and V. Cerf, "HOST-HOST Communication Protocol in the ARPA Network," *AFIPS Nat. Comput. Conf. Expo. Conf. Proc.*, 36, 589-597 (1970).

[3] H. Frank, I. T. Frisch, and W. Chou, "Topological Considerations in the Design of the ARPA Computer Network," *AFIPS Nat. Comput. Conf. Expo. Conf. Proc.*, 36, 581-587 (1970).

[4] F. Heart, R. Kahn, S. Ornstein, W. Crowther, and D. Walden, "The Interface Message Processor for the ARPA Computer Network," *AFIPS Nat. Comput. Conf. Exp. Conf. Proc.*, 36, 551-567 (1970).

[5] L. Kleinrock, "Analytic and Simulation Methods in Computer Network Design," *AFIPS Nat. Comput. Conf. Expo. Conf. Proc.*, 36, 569–579 (1970).
[6] L. G. Roberts, and B. D. Wessler, "Computer Network Development to Achieve Resource Sharing," *AFIPS Nat. Comput. Conf. Expo. Conf. Proc.*, 36, 543–549 (1970).
[7] L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay*, McGraw-Hill, New York, 1964 (out of print); reprinted by Dover, New York, 1972.
[8] G. L. Fultz, "Adaptive Routing Techniques for Message Switching Computer-Communication Networks," UCLA-ENG-7252, School of Engineering and Applied Science, University of California, Los Angeles, July 1972.
[9] H. Frank, and W. Chou, "Topological Optimization of Computer Networks," *Proc. IEEE*, 60(11), 1385–1397 (1972).
[10] L. Kleinrock, *Queueing Systems, Vol. II. Computer Applications*, Wiley-Interscience, New York, 1976.
[11] J. M. McQuillan, "Adaptive Routing Algorithms for Distributed Computer Networks," Report No. 2831, Bolt Beranek and Newman, Inc., Cambridge, MA, May 1974.
[12] D. W. Davies, and D. Barber, *Communication Networks for Computers*, Wiley, New York, 1973.
[13] M. Gerla, "The Design of Store-and-Forward (S/F) Networks for Computer Communications," UCLA-ENG-7319, School of Engineering and Applied Science, University of California, Los Angeles, January 1973.
[14] H. Frank, M. Gerla, and W. Chou, "Issues in the Design of Large Distributed Computer Communication Networks," in *Proceedings of the National Telecommunication Conference*, Atlanta, GA, November 1973, IEEE, New York, pp. 37A-1–37A-8.
[15] M. Gerla, W. Chou, and H. Frank, "Computational Considerations and Routing Problems for Large Computer Communication Networks," *Proceedings of the National Telecommunication Conference*, Vol. 2, Atlanta, GA, November 1973, IEEE, New York, pp. 2B-1–2B-11.
[16] F. Kamoun, *Design Considerations for Large Computer Communication Networks*, UCLA-ENG-7642, Computer Science Department, School of Engineering and Applied Science, University of California, Los Angeles, March 1976.
[17] L. Kleinrock, and F. Kamoun, "Hierarchical Routing for Large Networks; Performance Evaluation and Optimization," *Comput. Networks*, 1, 155–174 (1977).
[18] F. Kamoun, and L. Kleinrock, "Stochastic Performance Evaluation of Hierarchical Routing for Large Networks," *Comput. Networks*, 3, 337–353 (1979).
[19] Network Analysis Corporation, "The Practical Impact of Recent Computer Advances on the Analysis and Design of Large Scale Networks," First Semiannual Technical Report, Glen Cove, NY, May 1973.
[20] G. J. Coviello, and R. D. Rosner, "Cost Considerations for a Large Data Network," *International Computer Communication Conference Proc.*, 289–294, Stockholm, August 1974.
[21] M. R. Anderberg, *Cluster Analysis for Applications*, Academic, New York, 1973.
[22] R. O. Duda, and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
[23] B. W. Kernighan, and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," *Bell Syst. Tech. J.*, 49, 291–307 (1970).
[24] F. Harary, *Graph Theory*, Addison-Wesley, Reading, MA, 1972.
[25] D. E. Knuth, *The Art of Computer Programming*, Vol. 1, Addison-Wesley, Reading, MA, 1969.
[26] W. I. Zangwill, *Nonlinear Programming, A Unified Approach*, Prentice-Hall, Englewood Cliffs, NJ, 1969.

[27] C. K. Chow, "An Optimization of Storage Hierarchies," *IBM J. Res. Dev.*, 18(3), 194–203 (1974).
[28] G. H. Dysart, and N. D. Georganas "NEWCLUST: An Algorithm for the Topological Design of Two-Level, Multidrop Teleprocessing Networks," *IEEE Trans. Commun.*, 26(1), 55–62 (1978).

Received September 1976
Revised October 1979