Topological Design of Computer Communication Networks

Bezalel Gavish
Owen Graduate School of Management
Vanderbilt University
Nashville, Tennessee, 37203

The topological design of computer communication networks is a complex process which consists of selecting the set of locations in which Network Control Processors (NCP's) will be placed, deciding on the set of backbone links to connect the NCP's, linking end user nodes to NCP's and deciding on the set of routes which support communications between communicating end users. This problem was handled in the past by breaking the overall problem into subproblems which are individually solved by heuristics. The quality of the solutions generated by this heuristics are hard to evaluate since the optimal solutions to this problems are not known. This paper develops a nonlinear formulation of the topological design problem, a Lagrangean relaxation of the problem is developed. The Lagrangean solutions provide lower bounds on the optimal solutions, those are further improved using subgradient optimization procedures. Heuristics are developed for generating feasible solutions to the problem, the quality of solutions generated by this heuristics are compared to the lower bounds generated by the Lagrangean procedures on a set of test cases.

1. Introduction

Computer communication networks offer significant economic advantages to the organizational units which implement and use them. Gavish [18] has shown the existence of self-propelling incentive mechanisms which have a tendency to increase the size and scope of computer networks. Technological innovations lead to a continuing decline in the cost of providing communication services and to the introduction of new types of services (Voice-mail, FAX, Video services, Videotext, Electronic Data Interchange...). The number and range of applications supported by communication based computer systems have been increasing at an accelerated rate. AT&T's annual reports show an increase in annual data communications-related revenues from $500 million in 1971 to $6 billion in 1980. The annual growth rate of this segment of the industry in the last decade was over 20 percent per year.

A variety of computer networks such as SNA[29], BNA[32] and DECNET[11] architectures, TELNET[46], TTYMNET[47], TRANSPAC[10], CSNET[8], BITNET[31], AIS/Net 1000[1] and DATAPAC[7] are being offered by computer and communication providers. The size and complexity of existing networks range from a few nodes to hundreds of nodes in the backbone network covering national and international communications, supporting from a few end users to millions of users on a single network. All indications point to a further increase in the number as well as in the size and complexity of such networks.

A problem faced by designers and managers of computer communication networks is: What should be the overall configuration of the network. This involves selecting the set of nodes for placement of backbone network control processors (NCP's), deciding on the set of links which interconnect the selected backbone nodes, selecting the set of links connecting end users to the backbone network and the set of routes to be used by communicating end users so as to insure an acceptable performance level at a minimum cost. In contrast to previously published research, we assume that the network topology is not given a priori. External traffic requirements are given, and messages in the network follow static, non-bifurcated routes. The motivation in concentrating on this routing strategy is that it is common to most operational networks. Moreover, simulation results in [25] suggest that, for medium to large scale networks at steady state, there is no significant difference between the delays induced in the network by good static and adaptive routing.

Static routing policies are implemented by providing each pair of communicating nodes in the network with an ordered set of routes out of which at session initiation the first available route is chosen, (see [2]). In this paper we concentrate on the choice of the primary route, i.e., the most "favorable" one among a given set of candidate routes. Once an initial topology has been selected, a more detailed analysis can be undertaken in order to determine the exact set of routes and link capacities to be used in the final design.

Almost all previously published research assumes that NCP locations are a priori given and deal with the capacity assignment problem and routing problem as two separate problems. When the capacity assignment is considered [7, 35, 36], the routing policy, and therefore the flow on each link, is assumed to be known, and the least costly feasible capacity assignment is selected. Conversely, in the literature dealing with the flow assignment problem, both for the static [9, 13, 14, 21, 26, 39] and for the adaptive [3, 15, 34, 37, 40, 43, 44, 47, 48, 49] case, a feasible routing strategy which minimizes some performance measure (generally expressed in terms of the average delay per message in the network) is selected for a given capacity assignment.

The literature focusing on the combined problem of capacity and flow assignment (CFA) is very limited. In [37] the authors incorporate the heuristic methods for capacity assignment developed in [36] into a more general procedure. Using several initial flow assignments as starting points, the procedure iterates between the cost minimizing capacity assignment algorithms and a flow assignment phase in which a measure of the average delay is minimized.

0073-1129/89/0000/0770$01.00 © 1989 IEEE
until a local optimum is reached. In addition, a priority assignment scheme is also considered. Using a similar iterative approach, Gerla and Kleinrock present in [28] four heuristic methods for solving the CFA problem based on their flow deviation algorithm [13]. Dealing with the two subproblems independently is often inappropriate. Due to the close interplay between the capacity value of a link and the delay incurred by a given flow on that link, Gavish and Neuman in [22,23] combine the two problems into a single model and develop a solution procedure which solves this combined problem.

The problem addressed in this paper is combinatorial in nature. To illustrate its computational complexity, consider a small network with $n$ nodes assuming $n(n-1)$ pairs of communicating backbone nodes. The number of bidirectional links in this graph is bounded by $n(n-1)/2$. There are two ways in which the complexity can be illustrated. One is to assume that $2^n$ combinations of NCP locations have to be considered. For each one of them, a routing and capacity assignment problem has to be solved and the best solution among those alternatives has to be selected, each one of this subproblems is quite difficult to solve on its own right. Another possibility is to directly evaluate the $2^{n(n-1)/2}$ potential topologies stemming from enumerating and evaluating all possible subsets of backbone links (and nodes) in the design, for each combination the optimal routing problem has to be solved. For both cases the time complexity is too high to yield acceptable algorithms.

The remainder of the paper is organized as follows: in Section 2 the problem is formulated as a nonlinear integer programming problem. Section 3 presents the Lagrangean relaxation of the problem and the method for solving the Lagrangean subproblem, followed in Section 4 by a description of heuristics used to generate feasible solutions to the problem. Numerical results of preliminary computational experiments are presented in Section 5. In Section 6 we show how the basic formulation can be extended to accommodate additional constraints.

2. Mathematical Formulation of the Problem

A majority of all previously published models for designing computer communication networks, concentrate on the selection of links to connect a predetermined set of NCP's and, as a secondary consideration, the selection of primary routes between all communicating entities in the network. The model developed in this paper also helps the designer to select the locations (from a candidate set) in which NCP's will be placed.

It is assumed that the network designer or manager has identified a set of locations that are candidates for NCP placement. Formation of the candidate set might be based on practical considerations such as; site ownership, accessibility, security or closeness to centers of activity. The model trades investment in the network capacity to the quality of service provided to the average network user as measured by the expected end to end network delay. The objective of the model, is to select a subset of those locations for actual placement of NCP's. The model simultaneously selects the set of links which will connect those NCP's and the set of primary routes to support the network traffic.

The networks considered here have to support a large user community. The criteria used for selecting the preferred design among all design alternatives are; Overall network costs, which consist of two major cost components. First are the NCP, link and traffic carrying charges. Those are traded versus the cost of delay imposed on messages carried by the network. Higher investments in direct network capabilities (node and link capacities, higher connectivity, lower network diameter and lower Hop count) reduce the average end to end delays that messages experience, thus reducing the average user delay costs. On the other hand, if an average user is willing to tolerate longer delays, the investment in node and line costs can be reduced. Users having special response time requirements, can be later handled by using appropriate priority and charging schemes for different classes of users and by fine tuning the routing decisions.

To develop the mathematical model a distinction is made between two sets of network nodes. One set consists of end user nodes, the second set is the set of backbone network nodes. The set of end user nodes consists of the set of locations in which the network users reside, from which messages originate and to which they have to be transferred. Those include locations of terminals, personal computers and mainframes. Such locations are typically looked upon in the networking literature as being extraneous to the backbone network. The traffic generated by end user nodes is supported by local access networks who handle their transfer to and from the backbone network. A network may contain from a few to hundreds of thousands of end users. Taking into consideration the practical complexity of the problem, it is unrealistic to represent each one of the end users as a separate node. Using aggregation techniques end user nodes can be clustered into clusters that are within a geographic proximity.

In contrast to most earlier published research on topological design of computer networks, it is assumed here that end user nodes are given as input to the design process: Their number, location and intensity of traffic generated by them are not affected by the topological design of the backbone network. Since the network topology and the subset of selected NCP's is not known in advance, the
specific NCP node to which each one of the end users will be connected depends on the subset of locations selected for NCP placement during the design process. A second set of nodes used in modeling the backbone network are nodes which are internal to the backbone network and are candidates for NCP placement. The identification of a candidate set of nodes for NCP placement is not a simple task. However, by taking into consideration: The high likelihood that in an optimal design, NCP’s will be placed close to end user concentrations of activity; Security and privacy considerations limit such candidate node locations to places controlled by the organization providing the networking services. Significant limits are put on the choice of places that can be selected as candidates for NCP placement. Assuming that NCP’s and the locations in which they are placed do not come for free, it is expected that, in an optimal design, NCP’s will be placed only in a subset of the candidate locations (The numerical results in Section 5 support this observation).

Local access networks can have complex topologies which might be composed of a combination of tree, loop, concentrators and a variety of line types. Those are difficult to model and solve even for centralized networks or for a priori given backbone network (see [17, 19]). In order to simplify the formulations and solution procedures, it is assumed that clusters of end users are connected to the network directly via dedicated lines.

Since communications originate and terminate at end user locations, a route supporting communication between two end users has to start and terminate in the two locations. A route supporting communication between two end user nodes consists of an ordered sequence of links: a link connecting the source user node to an NCP in the backbone network, a sequence of intranetwork links leading from this NCP to the NCP connected to the destination node, and a dedicated link connecting this NCP to the destination end user location. The formulation has to ensure that for each communicating pair of end user nodes, a path supporting them has been selected, and that all links and NCP’s on that path have been installed.

Let Π be the index set of all communicating origin destination pairs in the network. \( p, p' \in \Pi \) is the index of the \( p \)th communicating pair, in order to simplify the exposition, the term commodity is used to represent a specific pair of communicating nodes that have to be supported by the network. \( S_p \) is the index set of routes which can support the traffic generated by commodity \( p \). i.e. they start at the source node of commodity \( p \) and terminate at the destination node of the same commodity. The network design model has to ensure that for every commodity \( p \) a route is selected (out of \( S_p \)) to support the traffic generated by this commodity, and that all the links and NCP’s used by those paths are actually selected by the model.

In order to formulate the problem, we define the following sets:

- \( \Pi \) - is the index set of all commodities.
- \( S_p \) - is the subset of routes that are candidates to support commodity \( p, p \in \Pi \).
- \( R \) - the set of all candidate routes in the network, \( R = \bigcup_{p \in \Pi} S_p \).
- \( I \) - the index set of candidate locations in which NCP’s can be placed (only a subset of those locations will be actually used in the final design).
- \( K \) - the index set of end user locations. This set is given in advance and is independent of the final topology selected for the network.
- \( L_r \) - the index set of links used by route \( r \).
- \( L \) - the set of candidate links which can be used in the network, \( L = \bigcup_{r \in R} L_r \).

For every end user node \( k, k \in K \), we are given the amount of traffic generated by this node to each one of the commodities that originate from this node. \( \lambda_p \) is the message intensity for commodity \( p \). In theory, up to \( |K| - 1 \) commodities can originate from each end user location. In practice, for a typical user the majority of his communications are with a small subset of network users, thus only a small subset of the \((|K|(|K| - 1))/2\) commodities have to be defined. To simplify the model and analysis it is assumed that all messages have an exponential distribution of message length with a mean of \( 1/\mu \). The independence assumption [33] is used to model the queuing delay in the network.

The following set of decision variables are used in formulating the model:

- \( Z_{i} \) - is a binary variable which is equal to one if an NCP is placed in location \( i \). It is zero otherwise.
- \( Y_{ij} \) - is equal to one if a link is established between locations \( i \) and \( j \). It is zero otherwise. Links can be established between NCP’s or, between end user nodes and NCP’s.
- \( X_r \) - is equal to one if route \( r, r \in R \) is selected to support the appropriate commodity.

Using the above definitions, the number of messages carried on link \((i,j)\) per unit of time, is given by \( \sum_{r \in R} X_r \delta_{ij} \), where \( \delta_{ij} \) is an indicator function, it is equal to one if link \((i,j)\) appears in route \( r \), and is zero otherwise. \( \lambda_r \) is the amount of traffic (messages/packets per second) generated by the commodity which is supported by route \( r \). It should be noted that for a given set of candidate routes and links, \( \delta_{ij} \) is a given parameter and is not a decision variable.

The objective of the design is to balance the overall investment in the network, versus the cost of delay imposed on network users. Higher investments in network capacity
lead to lower delays (and costs). The design attempts to minimize the sum of the two costs. It is composed of the following cost components:

a. NCP and network node associated costs, those include the hardware and software costs of an NCP, site preparation and maintenance costs. \( C_i \) is the present value of all those costs for site \( i \). Using the above notation those costs are equal to \( \sum_{i \in I} C_i Z_i \).

b. Link-related costs. Those are costs associated with setting up and maintaining the links that interconnect the NCP’s, and NCP’s to end user locations. \( S_{ij} \) is the cost of connecting locations \( i \) and \( j \). Those costs are captured by \( \sum_{(i,j) \in L} F_{ij} U_{ij} \) where \( L \) is the set of potential NCP to NCP links, and the set of links connecting end user nodes to NCP’s.

c. Traffic transfer costs. The cost of transferring messages over the links from their origination points to their destinations, \( U_{ij} \) is the cost per unit of traffic carried over link \((i,j)\) and \( F_{ij} \) is the amount of traffic per unit of time carried on that link. The traffic transfer costs are given by \( \sum_{(i,j) \in L} F_{ij} U_{ij} \) where \( L \) is the set of potential NCP to NCP links, and the set of links connecting end user nodes to NCP’s.

d. Costs associated with the delay encountered by messages carried over the network. Those are given by \( \sum_{(i,j) \in L} F_{ij} D/(Q_{ij} - F_{ij}) \) where \( Q_{ij} \) is the capacity of link \((i,j)\) and \( D \) is the unit cost of message delay in the network. By varying the value of \( D \) we can change the tradeoffs between direct network costs and the delays imposed on network users.

Not all of the above cost components may be relevant to a particular design. When some of this cost components are irrelevant, they can be set equal to zero.

The objective of the design is to minimize the present value of overall system costs, which are given by:

\[
Z = \text{Min} \left\{ \sum_{i \in I} C_i Z_i + \sum_{(i,j) \in L} Y_{ij} S_{ij} + \sum_{(i,j) \in L} F_{ij} U_{ij} + D \sum_{(i,j) \in L} F_{ij} D/(Q_{ij} - F_{ij}) \right\}
\]

Subject to:

- \( Y_{ij} \leq Z_i \) \( \forall i \in I, (i,j) \in L \) (2.2)
- \( Y_{ij} \leq Z_j \) \( \forall j \in I, (i,j) \in L \) (2.3)
- \( X_r \leq Y_{ij} \) \( \forall (i,j) \in L_r \) (2.4)
- \( \sum_{r \in S} X_r = 1 \) \( \forall p \in \Pi \) (2.5)
- \( F_{ij} = \sum_{r \in S} X_r \lambda_r \delta_{ij} / \mu \) \( \forall (i,j) \in L \) (2.6)
- \( Z_i, X_r, Y_{ij} = 0 \) or 1 \( \forall i \in I, (i,j) \in L, r \in R \) (2.7)

The constraints in (2.2, 2.3) ensure that a link may exist between locations \( i \) and \( j \) only if NCP’s have been placed in those two locations, or an NCP has been placed in one of those locations and the other location corresponds to an end user location. The constraints in (2.4) guarantee that a route can be established only if all the links through which it passes have been selected by the model. From (2.5) follows that exactly one primary route has to be selected for each commodity that is supported by the network. This assumes a non bifurcated flow. In addition to the above constraints, technological considerations might restrict the amount of traffic that can be handled by NCP’s and links. The simplest one to incorporate is the total flow through a link, letting \( F_{ij} \) be the flow on link \((i,j)\). The flow on a link can not exceed its capacity, thus

\[
F_{ij} \leq Q_{ij} \quad \forall (i,j) \in L
\]

In addition to the above constraints, due to technical and physical constraints, limits may be imposed on the amount of traffic that NCP’s can handle. Those extensions are discussed in Section 6.

3. The Lagrangean Relaxation

The network design problem given by (2.1-2.7) is a nonlinear combinatorial optimization problem which is NP-complete. In this section we develop a Lagrangean relaxation of the problem which provides lower bounds on the optimal solution value. Using a subgradient optimization procedure, this bound can be increased to provide a tight bound on the optimal value. The difference between the lowest feasible solution value and the highest lower bound is an upper bound on the error introduced by selecting this feasible solution.

In order to develop the Lagrangean problem, the optimization problem is reformulated as

\[
Z = \min \left\{ \sum_{i \in I} C_i Z_i + \sum_{(i,j) \in L} Y_{ij} S_{ij} + \sum_{(i,j) \in L} F_{ij} U_{ij} + \sum_{(i,j) \in L} D F_{ij} D/(Q_{ij} - F_{ij}) \right\}
\]

subject to:

- \( Y_{ij} \leq Z_i \) \( \forall i \in I, (i,j) \in L \) (3.2)
- \( Y_{ij} \leq Z_j \) \( \forall j \in I, (i,j) \in L \) (3.3)
- \( \sum_{r \in S} X_r = 1 \) \( \forall p \in \Pi \) (3.4)
- \( F_{ij} \geq \sum_{r \in S} X_r \lambda_r \delta_{ij} / \mu \) \( \forall (i,j) \in L \) (3.5)
- \( F_{ij} \geq Q_{ij} Y_{ij} \) \( \forall (i,j) \in L \) (3.6)
- \( F_{ij} \geq 0 \) \( \forall (i,j) \in L \) (3.7)

The equality in (3.5) can be replaced by inequality due to the fact that the objective function is strictly increasing
with \( F_i \) and therefore in any extremal solution, \( F_i \) will always attain its lowest possible value. This change is significant in the Lagrangean relaxation, since it restricts the multiplier values leading to faster convergence and to good multiplier and lower bound values.

The Lagrangean problem is generated by multiplying the constraints in (3.5) by a vector of multipliers \( \alpha_i \) and adding them to the objective function. The following Lagrangean problem is obtained.

**Problem \( P(\alpha) \)**

\[
L(\alpha) = \min \left\{ \sum_{i \in I} c_i z_i + \sum_{(i,j) \in L} (y_{ij} s_{ij} + f_{ij} u_{ij} + \frac{DF_{ij}}{Q_{ij} - F_{ij}}) \right. \\
+ \sum_{(i,j) \in L} \alpha_i f_{ij} - \sum_{r \in R} \alpha_r \lambda_r x_r / \mu \left. \right\}
\]

subject to: (3.2-3.4), (3.6-3.8).

It is well known from optimization theory (Geoffrion [24]) that for any vector of multipliers, \( L(\alpha) \) is a lower bound on the objective function value of the original problem, i.e., \( L(\alpha) \leq \mathcal{Z} \quad \forall \alpha \leq 0 \). We are interested in obtaining the tightest possible lower bound, i.e., in the multipliers vector \( \alpha^* \), which satisfies \( L(\alpha^*) = \max_{\alpha} \{ L(\alpha) \} \). Any computational procedure for approximating \( \alpha^* \) relies on our ability to quickly solve Problem \( P(\alpha) \). The above relaxation allows us to separate the Lagrangean problem into subproblems which are easier to solve.

For any given vector \( \alpha \), the Lagrangean problem can be decoupled into two problems such that \( L(\alpha) = L_1(\alpha) + L_2(\alpha) \) where:

**Subproblem \( P_1(\alpha) \)**

\[
L_1(\alpha) = \min \left\{ \sum_{i \in I} c_i z_i + \sum_{(i,j) \in L} (y_{ij} s_{ij} + \alpha_i u_{ij}) f_{ij} \right. \\
+ \sum_{(i,j) \in L} \frac{DF_{ij}}{Q_{ij} - F_{ij}} \left. \right\}
\]

Subject to: (3.2, 3.3), (3.6-3.8)

**Subproblem \( P_2(\alpha) \)**

\[
L_2(\alpha) = \min \left\{ \sum_{(i,j) \in L \cup R} -\alpha_i \lambda_r x_r / \mu \right. \\
\left. \right\}
\]

subject to: (3.4) and \( X_r = 0 \) or 1 \( \forall r \in R \).

Subproblem \( P_1(\alpha) \) is a nonlinear combinatorial optimization problem. It can be further simplified by taking advantage of the following observation: Given a solution \( \bar{Z}_i, \bar{Y}_{ij} \) to subproblem \( P_1(\alpha) \), the optimization over the \( F_{ij} \) variables is separable over links. When \( Y_{ij} \) equals zero, from (3.6) it follows that \( \bar{F}_{ij} \) equals zero. When \( \bar{Y}_{ij} \) equals one, \( \bar{F}_{ij} \) is derived by the following procedure:

Let \( \bar{F}_{ij} \) be the optimal solution to the following problem:

\[
\bar{F}_{ij} = \min \left\{ F_{ij} (u_{ij} + u_{ij}) + \frac{DF_{ij}}{Q_{ij} - F_{ij}} \right\}
\]

subject to:

\[
0 \leq \bar{F}_{ij} < Q_{ij}
\]

This is a nonlinear optimization problem whose optimal solution \( \bar{F}_{ij} \) is given by

\[
\bar{F}_{ij} = \begin{cases} 
Q_{ij} - \sqrt{-\frac{DQ_{ij}}{\alpha_i - \bar{F}_{ij}}} & \text{if } -\frac{DQ_{ij}}{\alpha_i - \bar{F}_{ij}} < Q_{ij} \\
0 & \text{otherwise.}
\end{cases}
\]

Substituting \( \bar{F}_{ij} \) in (3.9) leads to the optimal value of \( V_{ij} \).

The constraints in Subproblem \( P_2(\alpha) \) are such that \( F_{ij} \) equals zero when \( Y_{ij} \) equals zero, and is equal to \( \bar{F}_{ij} \) when \( Y_{ij} \) equals one. Therefore, without loss in generality, subproblem \( P_2(\alpha) \) can be rewritten as:

\[
L_2(\alpha) = \min \left\{ \sum_{i \in I} c_i z_i + \sum_{(i,j) \in L} y_{ij} \bar{s}_{ij} \right. \\
\left. \right\}
\]

subject to:

\[
Y_{ij} \leq Z_i \quad \forall \text{ } (i,j) \in L \\\nY_{ij} \leq Z_j \quad \forall \text{ } (i,j) \in L \\\nZ_i, Y_{ij} = 0 \text{ or } 1 \quad \forall \text{ } (i,j) \in L
\]

where \( \bar{s}_{ij} = s_{ij} + V_{ij} \).

This is a linear combinatorial optimization problem which can be solved by relaxing the integrality constraints, solving the corresponding linear program and using a branch and bound procedure in order to enforce the integrality conditions. The following property is used to further reduce the size of the problem to be solved:

For every link \( (i,j) \) with a cost \( s_{ij} \geq 0 \) without loss in generality the corresponding \( Y_{ij} \) value can be set to zero.

Using this property, it is possible to reduce the size of the optimization subproblem. The cost coefficients in the remaining problem satisfy the following conditions: \( s_{ij} < 0 \quad \forall (i,j) \in L \) and \( c_i \geq 0 \quad \forall i \in I \), where \( I \) and \( L \) are the set of nodes and links in the reduced subproblem. In Gavish[16] we show how to efficiently solve the reduced subproblem.

Solving \( L_2(\alpha) \) is much simpler. Subproblem \( P_2(\alpha) \) can be rewritten as:

\[
L_2(\alpha) = \min \left\{ \sum_{i \in I} \sum_{r \in R} \sum_{(i,j) \in L} -\alpha_i \lambda_r x_r / \mu \right. \\
\left. \right\}
\]
subject to:

$$\sum_{r \in S_p} X_r = 1 \quad \forall p \in \Pi$$
$$X_r = 0 \text{ or } 1 \quad \forall r \in R.$$  

This subproblem can be further decomposed into |\Pi| subproblems, one for each commodity where the p-th subproblem is:

$$L^*_p(\alpha) = \min \left\{ \sum_{r \in S_p} a_r X_r \right\}$$

subject to:

$$\sum_{r \in S_p} X_r = 1$$
$$X_r = 0 \text{ or } 1 \quad \forall r \in S_p$$

where $$a_r = \sum_{b \in \Pi} -a_{b} \mu \delta_{ij} / \mu.$$  

This subproblem is solved by setting $$X_r = 1$$ for that index b, b \in S_p, that satisfies:

$$a_b = \min \{a_r\}$$

$$L^*_p(\alpha) = a_b, \text{ and } L_0(\alpha) = \sum_{p \in \Pi} L^*_p(\alpha).$$

The number of routes possible for each commodity is an exponential function of the network size, in order to accommodate larger size networks and reduce storage and time requirements. It is possible to use an implicit representation of the sets S_p and keep in memory only one route for each commodity, if better routes exist they are identified and generated when needed by a route generation procedure, similar to the one used in Gavish and Altinkemar[20] and replace the candidate route in S_p.

The function $$L(\alpha)$$ is a nondifferentiable function over \alpha and therefore classical optimization procedures cannot be applied for solving the problem \max \{L(\alpha)\}. Several procedures have been suggested in the literature for approximating \alpha*. Those include dual ascent, multiplier adjustment, column generation and subgradient optimization procedures. Due to our past positive computational experience with subgradient optimization, we have adopted it for this investigation.

The subgradient optimization procedure[30] is an iterative procedure where, in its p-th iteration, we are given \{\alpha_{ij}^p\}. Using those multiplier values, the Lagrangean problem is solved, obtaining $$L(\alpha^p), X^p, Y^p, Z^p$$ and $$\lambda^p$$. Based on this solution, the multipliers used in the next iteration are computed using the formula

$$\alpha_{ij}^{p+1} = \alpha_{ij}^p - t_p \gamma_{ij}^p$$

where $$\gamma_{ij}^p$$ are the subgradient directions given by:

$$\gamma_{ij}^p = F_{ij}^p - \sum_{r \in R} X_r^p \lambda_r^p \delta_{ij} / \mu$$

where $$t_p$$ is a step size. It has been shown by Poljack [41] that $$L(\alpha^p)$$ converges to $$L(\alpha^*)$$ when the sequence $$t_p$$ satisfies the following conditions: $$\lim t_p \to 0$$ and $$\sum_{p} t_p \to \infty$$. Unfortunately, such a sequence is not practical for a computational procedure and therefore $$t_p$$ is approximated by the following formula:

$$t_p = \frac{Z - L(\alpha^p)}{||\gamma_{ij}^p||^2}$$

where Z is an overestimate on $$L(\alpha^*)$$ and $$\gamma_{ij}$$ is a scalar restricted to $$0 < \gamma_{ij} < 2$$. Since $$L(\alpha^*) \leq Z(\alpha^*)$$, any feasible solution to the topological design problem can be used as Z. In the computational procedure we use the minimum of those values as Z.

The subgradient optimization procedure consists of the following steps:

Step 1. Initialization:

a. Using an appropriate heuristic, compute an initial value for Z.

b. Select an initial value for $$\alpha^0$$.

c. Set $$\alpha^* = \alpha^0, p = 0, ITR = 0$$ (improvement counter), $$IMT = 0$$ (iteration counter), $$L(\alpha^0) = 0, \delta_p = 2$$.

Step 2. Solving the Lagrangean problem:

a. Set $$IMP = IMP + 1$$


c. Update the parameters:

a. If $$L(\alpha^p) > L(\alpha^*)$$ set $$L(\alpha^*) = L(\alpha^p), \alpha^* = \alpha^p, IMP = 0$$.

b. If $$X^p, Y^p, Z^p, T^p$$ are feasible for Problem P and have a cost lower than Z, replace Z with this cost.

c. If $$IMP = IMPLIMIT$$, set $$\delta_p = \delta_p / 2$$, $$\alpha^p = \alpha^*, IMP = 0$$, go to step 2.

d. Test for termination conditions (one of the following conditions):

$$\delta_p < \varepsilon_1, \text{or } ||\gamma_{ij}^p|| < \varepsilon_2, \text{or } t_p < \varepsilon_3, \text{or } (Z(\alpha^p))/Z < \varepsilon_4, \text{or } ITR > ITRLIMIT.$$  

If a termination condition is satisfied, terminate. Otherwise, go to step 4.

Step 4. Update the multipliers:

a. Set $$\alpha_{ij}^{p+1} = \min \{0, \alpha_{ij}^p + t_p \gamma_{ij}^p(\alpha^p)\}$$

b. Set $$p = p + 1$$

c. Set $$ITR = ITR + 1$$, go to step 2.

4. Heuristics for Generating Feasible Solutions

The Lagrangean relaxation procedure described in the previous section, provides a lower bound on the unknown optimal solution value of the network design problem. For this problem, it is difficult to provide an a priori theoretical bound on the gap between the optimal solution and the best Lagrangean value. In order to be able to empirically evaluate the viability of this approach, several heuristics have been developed, they generate feasible solutions to
the problem. The gap between the best feasible solution generated and the best Lagrangean value is an estimate on the distance of this feasible solution to the optimal value on the one hand, and on the gap between the Lagrangean value and the optimal solution.

Three types of heuristics have been developed and tested. A heuristic which is based primarily on feasibility considerations, a greedy type heuristic which uses an NCP and link drop procedure, and heuristics that are based on partial enumeration of the (exponential) solution space. In what follows we present a short description of the heuristics, followed in section 5 by computational results obtained using this heuristics and the Lagrangean procedure.

2. Greedy Drop Type Heuristic: The heuristic begins from a solution in which an NCP is placed in each candidate location, a complete graph connects this NCP’s forming the backbone network, end-user nodes are linked to the closest NCP’s and a flow deviation algorithm based heuristic is used to determine the initial routing of messages from end users to their destinations through the backbone network.

In the improvement phase of the heuristic, the heuristic calculates the cost of the above solution and examines what will be the impact of removing one of the NCP’s from the backbone network (redundant links are removed, end users are reassigned and the Flow deviation algorithm is used to reroute the network flow). Each NCP is considered for removal and the one leading to the highest improvement in the objective function is removed leading to a new base solution. This procedure is repeated until no further reduction in the system costs is obtained.

3. Partial Enumeration Based Heuristic: This heuristic is based on the observation that if we consider all NCP placement combinations in which k NCP’s are assigned to k locations (out of all possible |I| candidate locations) and solve this case to optimality, then by varying k from 1 to the number of candidate locations all possible solutions to the problem have been examined, clearly the least costly among this solutions is the optimal solution to the overall problem. Examining all the solutions in which k out of |I| candidate locations are selected for NCP placement has a time complexity of \(O(B|I|^k)\) where B is the time complexity for a routing and capacity assignment problem. This function grows very fast with k and therefore could be applied only to very small or very large (close to |I|) values of k. In what follows the highest value of k is denoted by \(k_{\text{max}}\).

Algorithm 3 \((k_{\text{max}})\)

a. Let \(\text{Best} = \infty\); \(k = 1\).

b. Examine all possible combinations of placing k NCP’s in k of the \(|I|\) candidate locations for NCP placement. For each combination find the best end-user assignment to NCP’s, the subset of links to be used to link the selected NCP’s and the routing strategy in the backbone network. For each possible assignment compute the cost of this solution.

c. Let \(Z(k)\) be the value of the lowest cost solution among all the solutions examined in Step b.

d. If \(Z(k) < \text{Best}\), let \(\text{Best} = Z(k)\) and save the solution.

e. If \(k = k_{\text{max}}\) stop, otherwise set \(k = k + 1\) and go to Step b.

Clearly Algorithm 3 is quite limited to very small values of \(k_{\text{max}}\) (3 or 4 for most practical problems). For large scale problems in which more then a few NCP locations are needed in order to generate low cost feasible and effective solutions, it is possible to use algorithm 3 as a building block for a modified algorithm (algorithm 3A) which can accommodate more then a few NCP’s.

5. Computational Results

In order to test the effectiveness of the Lagrangean relaxation and heuristic procedures the algorithms described in sections 3 and 4 have been programmed in Fortran. A number of preliminary computational tests have revealed a clear superiority of the Partial Enumeration Based Heuristic. It consistently produced lower cost networks and at comparable or lower amounts of computing times. In all tested cases the best solution generated by the heuristics were within fifteen percent of the lower bounds produced through the subgradient optimization procedure.

The test data for the experiments consisted of problems varying from fifty to two hundred end user centers and from ten to forty nodes that are candidates for NCP placement. The link cost structure used in the experiments was identical to the one used in Gavish and Neuman [22] and Gavish and Altinkemar [20] for testing their algorithms for solving the Routing and Capacity assignment problem. The node locations have been selected from a uniform distribution over a square of dimensions 2100. The air distances between these points were used as link distances in this experiments. The number of communicating pairs is a parameter which is set by the user for each case. Since the network link capacities have been set to 50Kbps for the backbone links and 9600 Baud for links connecting end user nodes to the backbone network. The number of commodities that can start at a node is limited not to generate a total load which exceeds the link capacity. This bound is equal to the link capacity divided by; The mean message length times arrival intensity per session times the average number of sessions in a commodity.

At this stage, only a small set of cases have been tested, and no conclusive observations can be made. To illustrate uses of the model two examples are used. For the first case the test data is shown in Figure 1. It consists of 215
nodes and 15 NCP candidate locations. The second case in shown in Figure 2, consists of 215 nodes and 25 NCP candidate locations. The average message length was set to 400 bits with an arrival rate of 4 messages per second. The number of commodities was set to 189. The cost data in the two experiments was identical except for link distance dependent costs.

Figures 3 shows the outcome of the design for the case in which the links connecting end user nodes to the network, have low distance dependent costs. The optimal design for this case is a centralized network, the network consists of a single NCP node with all end user nodes directly connected to the NCP. Figure 4 shows the design generated for the same test data, in which the distance dependent costs have been significantly increased. 7 NCP locations have been selected by this design with 13 lines connecting them.

Careful analysis of the designs produced by the new procedures, indicate a very low likelihood for a human designer to come up with competitive designs using trial and error experiments. This is true even when he has at his disposition advanced interactive graphics calculation and presentation. It turns out that in this types of design problems human judgment and experience can not compete with mathematically based methods.

6. Possible Extensions

In order to simplify the exposition the formulation presented in Section 2 concentrated on a simplifying set of assumptions. It is possible to extend the model so as to incorporate additional types of constraints and consider factors which might be important for decisions which have to be made during the design process. We demonstrate how to incorporate such constraints in the formulation. Additional research will be needed to develop viable solution procedures for the expanded formulations.

a. Selecting Link Capacities: Providers of communication links supply them at prespecified capacities. Let $G_{ij}$ be the index set of link capacities that can be acquired between locations $i$ and $j$. $Y_{ijg}$ is a decision variable which is equal to one if a link type $g$ is selected to connect locations $i$ and $j$. $Q_{ijg}$ is its associated capacity, and $S_{ijg}$ and $U_{ijg}$ its associated costs.

The combined problem is reformulated as:

$$Z = \min \left\{ \sum_{i \in I} C_i z_i + \sum_{(i,j) \in E} \sum_{g \in G_{ij}} (Y_{ijg} S_{ijg} + F_{ijg} Y_{ijg} U_{ijg}) \right\}$$

$$+ D \sum_{(i,j) \in E} F_{ij} / (Q_{ij} - F_{ij}) \right\}$$

subject to (2.2—2.8)

and

$$Y_{ijg} = \sum_{g \in G_{ij}} Y_{ijg} \quad \forall (i,j) \in E$$

$$F_{ij} \leq \sum_{g \in G_{ij}} Q_{ijg} Y_{ijg} \quad \forall (i,j) \in E$$

$$Y_{ijg} = 0 \text{ or } 1 \quad \forall i,j,g$$

The combined problem is clearly more complicated than the one presented in Section 2. However, using methods similar to the ones presented in Gavish and Neuman [22, 23], it is possible to derive lower bounds on the optimal solution and heuristics for generating feasible solutions to this problem.

b. Limits on the amount of traffic handled by an NCP:

Processing and technological restrictions may limit the amount of traffic per unit of time which can be handled by an NCP. Those limits can be expressed by the following constraints:

Limit on the traffic volume entering an NCP:

$$\sum_{j \in I} F_{ij} \leq A_i \quad \forall i \in I$$

Limit on the traffic volume exiting an NCP:

$$\sum_{j \in I} F_{ij} \leq B_i \quad \forall i \in I$$

Limit on the total traffic handled by an NCP:

$$\sum_{j \in I} F_{ij} + \sum_{j \in I} F_{ij} \leq D_i \quad \forall i \in I$$

$A_i$ and $B_i$ do not necessarily add up to $D_i$.

c. Restricting the class of selected topologies: In many instances, designers of computer networks impose additional restrictions on the final topology that can be selected. The restrictions might include: Connectivity restrictions which are imposed in order to increase the network reliability and availability. It is customary to impose a $k$-link connectivity restriction on the selected topology ($k$ is typically 2 or 3), thus ensuring against one or two link failures; Diameter restrictions, thus restricting the number of links in the shortest path between any two backbone nodes to be below a given limit. In other cases, the network might be restricted to be a tree.

Let $\sigma$ be the set of restrictions on the $Y$ variables that can be selected in the design, i.e., we add to the formulation given by (3.1—3.8) the restriction that $Y_{i\sigma}$. The Lagrangean problem can still be derived in a similar manner to the one used to derive it earlier, except that now, Subproblem $P(\alpha)$ has the additional restriction that $Y_{i\sigma}$. This may lead to subproblems that are significantly harder
to solve. For example, if we require the topology to form a tree structure, then the subproblem reduces to the Steiner Tree Subproblem. Similarly, imposing a k-connectivity restriction leads to selecting a subgraph with a given property. All such problems are known to be difficult to solve.

References

2. V. Anujo, Routing and flow control in systems network architecture, IBM Systems Journal 18, 208-214.
12. G. L. Francone, "World of Communications" Le Monde Economique ([1985]).
24. A. M. Geoffrion, Lagrangean relaxation and its uses in integer programming, Mathematical Programming Study 3 ([1974]).
25. M. Gerla, Approximations and bounds for the topological design of distributed comp. network., Proc. 6th ACM Data Communications Symposium, Quebec, Canada ([1975]).
Figure 1 Distribution of end user nodes and NCP candidate locations for case 1

Figure 2 Distribution of end user nodes and NCP candidate locations for case 2

Figure 3 The design generated for case 1

Figure 4 The design generated for case 2