CAPACITY AND FLOW ASSIGNMENTS

IN LARGE COMPUTER NETWORKS

Bezalel Gavish
University of Rochester
Rochester, New York 14627

and

Irina Neuman
New York University
New York, New York 10006

January 1986

Center for Research on Information Systems
Computer Applications and Information Systems Area
Graduate School of Business Administration
New York University

Working Paper Series

CRIS #111

GBA #86-2

To appear in **Infocom 1986**, Miami, Florida.

Capacity and Flow Assignment in Large Computer Networks

Bezalel Gavish
University of Rochester
Rochester, NY, 14627

Irina Neuman
New York University
New York, NY, 10006

Abstract

This paper presents a model and the corresponding solution method for the problem of jointly selecting a set of primary routes and assigning capacities to the links in a computer communication network. The network topology and the traffic characteristics are known; a set of candidate routes for each communicating pair of nodes, and a set of candidate capacities for each link are also given. The goal is to obtain the least costly feasible design, where the costs include both capacity and queuing components.

The resulting combinatorial optimization problem is solved using Lagrangean relaxation and subgradient optimization techniques. The method was tested on several topologies, and in all cases good feasible solutions, as well as tight lower bounds were obtained.

1. Introduction

As a result of the important advantages they offer, both the number and the range of applications supported by communication based computer systems have significantly increased. A variety of computer networks, such as SNA[13], BNA[15] and DECNET[6] architectures, TELENET[20], TYMNET[21], TRANSPAC[5], AIS/NET-1000[1] and DATAPAC[4] are currently available. This paper deals with the following problem faced by the network designer whenever a new network is set up or when an existing network is to be expanded: how to simultaneosly select the link capacities and the routes to be used by the communicating nodes in the network, such as to insure an acceptable performance level at a minimum cost. The topology of the network and estimates of the external traffic requirements are given. 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 [11] suggest that, at steady state, there is no significant difference between the delays induced in the network by good static and adaptive routing strategies.

Static routing policies are generally implemented by providing each pair of communicating nodes in the network with an ordered set of routes, out of which the first available route is chosen whenever a session is initiated (see [3]). In this paper we concentrate on the choice of the primary route, i.e the most "recommended" among the routes in the candidate set.

Much of the existing literature deals with the two imbedded subproblems independently. This is often inappropiate, since the close interplay between the capacity value of a link, and the delay incurred by a given flow on that link, makes it difficult to claim that a truly good solution has been found for either of these subproblems when considered separately.

The literature focussing on the capacity and flow assignment (CFA) problem is very limited. In [18], the authors incorporate the heuristic methods for capacity assignment developed in [17], 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, 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 [12] four heuristic methods for solving the CFA problem based on their flow deviation algorithm [7]. A weakness common to all existing attempts to solve the CFA problem is that no means, either theoretical or empirical, are provided in order to evaluate the quality of the heuristic solution they generate. This may seriously hamper their usefulness for real life applications.

The remainder of the paper is organized as follows: in section 2 the CFA problem is formulated as a nonlinear integer programming problem. Section 3 presents the Lagrangean relaxation of the problem, and section 4 shows how a subgradient optimization procedure can be used to improve on the quality of the lower bound provided by the relaxation. In section 5, we show how good feasible solutions to the problem can be generated during the computation of the lower bound. Detailed results of computational tests are presented in section 6. We conclude by discussing some related open problems and suggesting further research.

2. Problem Formulation

The queuing phenomena are captured by modeling each link as a server whose service rate is determined by its capacity, and by viewing messages on the link as customers competing for its service. The resulting model is that of a network of queues. We assume unlimited buffering space and no processing delay at the nodes, so that the delays incurred by messages in the network are solely due to the limited bandwidth of the links. For ease of exposition, propagation delays, which are negligible for terrestrial links, are ignored. We make the common assumptions of Poisson external arrivals and exponentially distributed message lengths. We also use the independence assumption, first introduced by Kleinrock in [16].

Since the model deals in a unified way with both the flow and the capacity assignment issues, the following two distinct types of costs are considered:

- capacity costs, comprised of a fixed setup cost (including a base monthly charge and a term proportional to the distance between the two nodes), and a variable cost, which is a function of the traffic on the line;

- queuing costs, associated with the delay incurred by messages in the network.

The following notation will be used throughout the paper:

$L=$ the total number of links in the network.

$I_l=$ the index set of line types available for link l, $l \in L$.

Q_{lk} = the capacity of line type $k, k \in I_l$.

S_{lk} = the fixed cost of line type $k, k \in I_l$.

C_{lk} = the variable cost of line type $k, k \in I_l$ per unit of traffic on link l.

D = unit cost of delay.

R = the set of candidate routes. It may be obtained through various route generation procedures or may be provided by the users.

Π = the set of communicating origin-destination pairs in the network.

S_p = the set of candidate routes for $p, p \in \Pi$. We assume that $S_p \cap S_q = \phi$ for $p \neq q$.

λ_r = the message arrival rate of the unique origin-destination pair associated with route $r, r \in R$. We define $\lambda_p = \lambda_r, \forall r \in S_p$.

γ = the total external arrival rate.

δ_{rl} = an indicator function, taking the value one if link l is used in route r, and zero otherwise.

$1/\mu$ = the average message length.

x_r = a decision variable, which is one if route r is chosen to carry the flow of its associated origin-destination pair, and zero otherwise.

y_{lk} = a decision variable, which is one if line type k is assigned to link l, and zero otherwise.

In terms of the x_r and y_{lk} variables defined above the CFA problem is:

Problem P1

$$Z_{P1} = min \left\{ \sum_{l \in L} \frac{D \sum_{r \in R} \lambda_r \delta_{rl} x_r / \mu}{\sum_{k \in I_l} Q_{lk} y_{lk} - \sum_{r \in R} \lambda_r \delta_{rl} x_r / \mu} + \right.$$

$$\left. \sum_{\substack{l \in L \\ k \in I_l}} S_{lk} y_{lk} + \sum_{\substack{l \in L \\ k \in I_l}} C_{lk} y_{lk} \sum_{r \in R} \lambda_r \delta_{rl} x_r / \mu \right\} \tag{1}$$

subject to:

$$\sum_{r \in R} \lambda_r \delta_{rl} x_r / \mu \leq \sum_{k \in I_l} Q_{lk} y_{lk} \qquad \forall l \in L \tag{2}$$

$$\sum_{k \in I_l} y_{lk} = 1 \qquad \forall l \in L \tag{3}$$

$$\sum_{r \in S_p} x_r = 1 \qquad \forall p \in \Pi \tag{4}$$

$$x_r = 0,1 \qquad \forall r \in R \tag{5}$$

$$y_{lk} = 0,1 \qquad \forall k \in I_l, l \in L \tag{6}$$

The constraints in (2) ensure that the flow on each link is feasible, i.e that it does not exceed the capacity value assigned to the link. Constraints (3) and (4) guarantee that only one line type is chosen for each link, and only one route for each origin-destination pair, respectively.

Problem P1 is a nonlinear combinatorial optimization problem. For fixed values of the y_{lk} variables, (2) is equivalent to the constraint set of the multiconstrained knapsack problem, a classical optimization problem known to be in the NP-complete class. Also, the nonlinearity of the objective function and the very large number of constraints and variables corresponding to the size of today's communication networks, significantly increase the complexity of the problem.

The problem is reformulated by introducing a new set of decision variables. A similar reformulation, that better highlights the underlying structure of the problem, was introduced earlier in [9].

Define f_l to be the utilization of link l, i.e that proportion of its capacity used by the actual message flow. f_l can be expressed as:

$$f_l = \frac{\sum_{r \in R} \lambda_r \delta_{rl} x_r / \mu}{\sum_{k \in I_l} Q_{lk} y_{lk}} \qquad \forall l \in L \tag{7}$$

In terms of the new set of decision variables, the CFA problem becomes:

$$Z_{P2} = min \left\{ \sum_{l \in L} \frac{D f_l}{1 - f_l} + \sum_{\substack{l \in L \\ k \in I_l}} S_{lk} y_{lk} \right.$$

$$\left. + \sum_{\substack{l \in L \\ k \in I_l}} C_{lk} f_l y_{lk} (\sum_{i \in I_l} Q_{li} y_{li}) \right\}$$

subject to:

$$\sum_{r \in R} \lambda_r \delta_{rl} x_r / \mu \leq f_l \sum_{k \in I_l} Q_{lk} y_{lk} \qquad \forall l \in L \tag{9}$$

$$0 \leq f_l \leq 1 \qquad \forall l \in L \tag{10}$$

$$\sum_{r \in S_p} x_r = 1 \qquad \forall p \in \Pi \tag{11}$$

$$\sum_{k \in I_l} y_{lk} = 1 \qquad \forall l \in L \tag{12}$$

$$x_r = 0,1 \qquad \forall r \in R \tag{13}$$

$$y_{lk} = 0,1 \qquad \forall k \in I_l, l \in L \tag{14}$$

Since the objective function is nondecreasing increasing in the f_l directions, (7) is rewritten as an inequality in (9). The quadratic term in the last sum of (8) can be linearized as a result of the following observation:

Lemma 1 The following relation holds over the feasible region defined by (9)-(14):

$$\sum_{\substack{l \in L \\ k \in I_l}} C_{lk} f_l y_{lk} (\sum_{i \in I_l} Q_{li} y_{li}) = \sum_{\substack{l \in L \\ k \in I_l}} C_{lk} Q_{lk} f_l y_{lk}$$

Proof The constraints in (12) imply that y_{lk} may be one for a single k value in each I_l set, i.e:

$$y_{lk} y_{li} = \begin{cases} y_{lk} & \text{if } k = i \quad \forall k, i \in I_l \\ 0 & \text{otherwise} \end{cases}$$

for all y_{lk}'s that satisfy (12).

As a result of the above lemma, problem P2 can be rewritten as:

$$Z_{P2} = min \left\{ \sum_{l \in L} \frac{Df_l}{1-f_l} + \sum_{\substack{l \in L \\ k \in I_l}} S_{lk} y_{lk} + \sum_{\substack{l \in L \\ k \in I_l}} C_{lk} Q_{lk} f_l y_{lk} \right\}$$

subject to: (9)-(14)

3. Lagrangean Relaxation

The Lagrangean relaxation of the problem is formed by multiplying the constraints in (9) by a vector of non-positive Lagrange multipliers $\alpha_l, l \in L$, and adding them to the objective function. The resulting problem is:

Problem $\mathbf{P}(\alpha)$

$$L(\alpha) = min \left\{ \sum_{l \in L} \frac{Df_l}{1-f_l} + \sum_{\substack{l \in L \\ k \in I_l}} S_{lk} y_{lk} + \sum_{\substack{l \in L \\ k \in I_l}} C_{lk} Q_{lk} f_l y_{lk} + \sum_{l \in L} \alpha_l (f_l \sum_{k \in I_l} Q_{lk} y_{lk} - \sum_{r \in R} \frac{\lambda_r \delta_{rl} x_r}{\mu}) \right\}$$

subject to: (10)-(14)

It is known from optimization theory [10] that for any vector of multipliers, $L(\alpha)$ is a lower bound on the value of the objective function of the original problem. The best Lagrangean bound is given by the vector α^* that corresponds to: $L(\alpha^*) = max_{\alpha \le 0}\{L(\alpha)\}$.

Two important issues when Lagrangean methods are used for difficult combinatorial problems are the ease of solving the relaxed problem, and efficiently obtaining the vector α^* (or a good approximation of it). In the following we show how in the present case the relaxed problem can be readily solved, while the second issue is dealt with in the next section.

For any α, the objective function of the Lagrangean can be rewritten as: $L(\alpha) = L_1(\alpha) + L_2(\alpha)$, where:

$$L_1(\alpha) = min \left\{ \sum_{l \in L} \frac{Df_l}{1-f_l} + \sum_{k \in I_l} S_{lk} y_{lk} + \sum_{k \in I_l} Q_{lk} f_l y_{lk} (C_{lk} + \alpha_l) \right\}$$

$$L_2(\alpha) = min \left\{ \sum_{r \in R} x_r (\sum_{l \in L} -\alpha_l \lambda_r \delta_{rl} / \mu) \right\}$$

i.e it can be decomposed into a component depending only on the link decision variables f_l and y_{lk}, and a second component depending on the routing variables x_r. Since the set of coupling constraints is no longer present in the relaxed problem, $P(\alpha)$ can be decomposed into:

Subproblem $P_1(\alpha)$

$$L_1(\alpha) = min \left\{ \sum_{l \in L} \frac{Df_l}{1-f_l} + \sum_{k \in I_l} S_{lk} y_{lk} + \sum_{k \in I_l} Q_{lk} f_l y_{lk} (C_{lk} + \alpha_l) \right\}$$

$$0 \le f_l \le 1 \qquad \forall l \in L$$

$$\sum_{k \in I_l} y_{lk} = 1 \qquad \forall l \in L$$

$$y_{lk} = 0,1 \qquad \forall k \in I_l, l \in L$$

and **Subproblem $P_2(\alpha)$**

$$L_2(\alpha) = min \left\{ \sum_{r \in R} x_r (\sum_{l \in L} -\alpha_l \lambda_r \delta_{rl} / \mu) \right\}$$

$$\sum_{r \in S_p} x_r = 1 \qquad \forall p \in \Pi$$

$$x_r = 0,1 \qquad \forall r \in R$$

Subproblem $P_1(\alpha)$ may be further separated into $|L|$ subproblems, one for each link in the network, where the subproblem associated with the lth link is:

$$L_1^l(\alpha) = min \left\{ \frac{Df_l}{1-f_l} + \sum_{k \in I_l} S_{lk} y_{lk} + \sum_{k \in I_l} Q_{lk} f_l y_{lk} (C_{lk} + \alpha_l) \right\}$$

$$0 \le f_l \le 1$$

(16) $$\sum_{k \in I_l} y_{lk} = 1$$

(17) $$y_{lk} = 0,1 \qquad \forall k \in I_l$$

To solve the above subproblem, we take advantage of the fact that the set of candidate capacities for each link will generally be of small cardinality, and exhaustively search the I_l set. Thus, for any given values of the y_{lk} variables that satisfy the constraints in (16) and (17), the subproblem becomes: **Subproblem $P_1^l(\alpha, k)$**

$$L_1^l(\alpha, k) = min \left\{ Df_l/(1-f_l) + Q_{lk}(C_{lk} + \alpha_l) f_l \right\} + S_{lk}$$

subject to:

$$0 \le f_l \le 1$$

where the k index corresponds to the y_{lk} variable chosen to be one.

The solution to the subproblem is:

$$f_l(k) = \begin{cases} 1 - \sqrt{\frac{-D}{(C_{lk}+\alpha_l)Q_{lk}}} & \text{if } \frac{-D}{(C_{lk}+\alpha_l)Q_{lk}} < 1 \text{ and } \alpha_l < -C_{lk} \\ 0 & \text{otherwise} \end{cases}$$

$L_1^l(\alpha)$ is given by: $L_1^l(\alpha) = min_{k \in I_l} \{ L_1^l(\alpha, k) \}$ and $L_1(\alpha)$ by: $L_1(\alpha) = \sum_{l \in L} L_1^l(\alpha)$

Similarly, subproblem $P_2(\alpha)$ can be decomposed into $|\Pi|$ subproblems, one for each origin-destination pair , where the pth subproblem is: **Subproblem $P_2^p(\alpha)$**

$$L_2^p(\alpha) = min \left\{ \sum_{r \in S_p} a_r x_r \right\}$$

$$\sum_{r \in S_p} x_r = 1$$

$$x_r = 0,1 \qquad \forall r \in S_p$$

where $a_r = \sum_{l \in L} -\alpha_l \lambda_r \delta_{rl} / \mu$

$P_2^p(\alpha)$ is solved by setting $x_r = 1$ for that index $b \in S_p$ that satisfies: $a_b = min_{r \in S_p} a_r$. This gives the value of $L_2^p(\alpha)$, and therefore of $L_2(\alpha)$.

4. The Subgradient Optimization Procedure

This section presents the methods used in order to obtain that value of $L(\alpha)$ that is as close as possible to Z_{P2}, the original objective function value, i.e provides the tightest lower bound on the value of the objective function.

A subgradient optimization procedure is used in order to estimate α^*. This iterative method was found to be effective in producing good lower bounds in a variety of combinatorial optimization problems[2,8,9,14].

Let $x_r(\alpha^i)$, $y_{lk}(\alpha^i)$ and $f_l(\alpha^i)$ be the optimal solution to the Lagrangean problem for a fixed vector α^i. The subgradient directions are given by:

$$\gamma_l(\alpha^i) = f_l(\alpha^i) \sum_{k \in I_l} Q_{lk} y_{lk}(\alpha^i) - \sum_{r \in R} \frac{\lambda_r \delta_{rl} x_r(\alpha^i)}{\mu} \quad \forall l \in L$$

The vector of multipliers corresponding to the $(i+1)$-st subgradient iteration can be computed as:

$$\alpha_l^{i+1} = \alpha_l^i + t_i \gamma_l(\alpha^i)$$

Poljack has shown in [19] that the convergence of $L(\alpha)$ to $L(\alpha^*)$ is guaranteed whenever the sequence of t_i's converges to zero and $\sum_{i=0}^{\infty} t_i = \infty$. Since such a sequence cannot be numerically generated, most existing applications of the subgradient method use the following heuristic rule for computing the t_i's:

$$t_i = s_i \frac{\overline{Z}_{P2} - L(\alpha^i)}{\|\gamma(\alpha^i)\|^2}$$

where $\overline{Z}_{P2}$ is an overestimate of the value of the objective function and s_i a scalar whose value is halved whenever no improvement in the value of the Lagrangean function is observed in a predetermined number of iterations.

The following steps comprise the subgradient optimization procedure:

1. Initialization:

a. using a heuristic, get an initial value for $\overline{Z}_{P2}$ (or set $\overline{Z}_{P2} = \infty$);

b. select an arbitrary initial value α^0 for the multipliers;

c. set $\alpha^* = \alpha^0$, improvement counter IMP = 0, iteration counter ITR = 0, current best value of the Lagrangean function $L(\alpha^*) = 0$, and stepsize $s_i = s$ (a value between 0 and 2).

2. Solving the Lagrangean problem:

a. Set IMP =IMP +1;

b. Solve problem $P(\alpha^i)$ using the current multipliers α^i, and obtain the values for $L(\alpha^i)$, $x_r(\alpha^i)$, $y_{lk}(\alpha^i)$ and $f_l(\alpha^i)$.

3. Testing and updating the parameters:

a. If $L(\alpha^i)$ is greater than the current $L(\alpha^*)$, then set $L(\alpha^*) = L(\alpha^i)$, $\alpha^* = \alpha^i$, and IMP = -1 ;

b. If $x_r(\alpha^i)$, $y_{lk}(\alpha^i)$ and $f_l(\alpha^i)$ are feasible for problem P2, compute the corresponding value of Z_{P2}, and if it is less than $\overline{Z}_{P2}$, set $\overline{Z}_{P2} = Z_{P2}$;

c. If the value of IMP has reached a prespecified limit, set $s_i = s_i/2$, $\alpha^i = \alpha^*$, IMP = 0, and go to step 2;

d. check for termination conditions. The algorithm stops whenever the total number of iterations exceeds a prespecified limit, the stepsize s_i becomes exceedingly small, or when the values of the overestimate and of the Lagrangean are acceptably close, i.e the algorithm has converged within a given tolerance limit.

4. Updating the multipliers:

The multipliers to be used in the next iteration are computed as: $\alpha_l^{i+1} = min\{0, \alpha_l^i + t_i \gamma_l(\alpha^i)\}$.

5. Set ITR=ITR + 1. Go to step 2.

A price often to be paid for the ease with which the relaxed problem can be solved is that, even after applying the subgradient procedure,the resulting lower bound is still of poor quality. This is explained in our case by the fact that the relaxed constraints express the very connection between the two sets of decision variables. The lower bound is tightened by generating additional constraints (i.e constraints that would be redundant in the original problem, but that may prove to be binding in the relaxed problem) and thus reducing the feasible region over which the Lagrangean problem is defined. The main idea behind the redundant constraint generation is to try to make some of the structure of the set of candidate routes "known" to the link related subproblems, i.e an attempt to recapture some of the meaning lost through relaxation.

Define $A_l = \{p : \delta_{rl} = 1 \quad \forall r \in S_p\}$, i.e the set of origin-destination pairs whose primary route must use link l, and $B_l = \{p : \delta_{rl} = 1 \quad \text{for some } r \in S_p\}$ i.e the set of origin-destination pairs that might use link l as part of their primary path. As a result, the following tighter formulation of subproblem $P_1^l(\alpha, k)$ is obtained:

$$L_1^l(\alpha, k) = min \left\{ \frac{Df_l}{1 - f_l} + Q_{lk}(C_{lk} + \alpha_l) f_l \right\} + S_{lk}$$

$$0 \le L_{lk} \le f_l \le U_{lk} \le 1$$

$$L_{lk} = \sum_{p \in A_l} \sum_{r \in S_p} \lambda_r / \mu Q_{lk}$$

$$U_{lk} = \sum_{p \in B_l} \sum_{r \in S_p} \lambda_r / \mu Q_{lk}$$

The solution to the subproblem is now:

$$f_l(k) = \begin{cases} \overline{f}_l(k) & \text{if } -D/(C_{lk} + \alpha_l)Q_{lk} < 1, \text{ and} \\ & \quad \alpha_l < -C_{lk}, \text{ and} \\ & \quad L_{lk} \le \overline{f}_l(k) \le U_{lk} \\ L_{lk} & \text{if } -D/(C_{lk} + \alpha_l)Q_{lk} \ge 1, \text{ or} \\ & \quad \alpha_l \ge -C_{lk}, \text{ or} \\ & \quad \overline{f}_l(k) < L_{lk} \\ U_{lk} & \text{if } \overline{f}_l(k) > U_{lk} \end{cases}$$

where: $\overline{f}_l(k) = 1 - \sqrt{-D/(C_{lk} + \alpha_l)Q_{lk}}$

The reformulated Lagrangean problem produced significantly tighter lower bounds. Finally, the following observation was also used in an attempt to further improve the quality of the bound: in any feasible solution, the value of the flow on any link l must be expressible as a sum of the message rates of some of the origin-destination pairs that might use link l as part of their primary route. As a result, for each k, f_l is defined only over a discrete set of values in the interval $[L_{lk}, U_{lk}]$. This additional restriction only marginally improved the quality of the bound.

5. Heuristic Procedures

It is important to obtain good upper bounds, not only because they represent a benchmark against which, in the absence of the optimal solution, the quality of the lower bound provided by the Lagrangean can be measured, but foremost because they represent feasible solutions to the original problem. If the gap between the two bounds is reasonably small, the solution corresponding to the upper bound can safely be used instead of the optimal one.

The algorithm presented earlier can be extended so that, using the solutions to the Lagrangean problem obtained during the subgradient procedure as a starting point and with some additional computational effort, a sequence of feasible solutions is generated. The following ideas were incorporated as part of the heuristic procedure:

1. Each time that a new solution to the Lagrangean problem is generated, it is checked for feasibility in terms of the relaxed set of constraints. If it is feasible and of lower cost, it replaces $\overline{Z}_{P2}$, the current value of the overestimate.

2. In order to increase the chances of identifying feasible solutions, the following observation was used: whenever subproblem $P_2^{p}(\alpha)$ is solved, it is often the case that more than one route have the same reduced cost a_r (where "same" is thought to mean within an $\epsilon << 1$ away from the minimum). A list of such routes is kept for each origin-destination pair. Several candidate solutions can then be generated by randomly selecting a route from each list, and checking the resulting assignment for feasibility in terms of the capacity assignment provided by the Lagrangean. This randomization procedure significantly increases the power of the algorithm to identify feasible solutions.

3. Taking advantage of the small cardinality of the sets, guarantees the generation of several feasible solutions at each subgradient iteration. Since the objective function is decomposable over the links, it is easy to obtain the best feasible capacity assignment for any given flow on a link, i.e to compute $\sum_{l \in L} min_{k \in I_l} A(f_l^j, Q_{lk})$ for $j = 1 \dots J$,where $A(f_l^j, Q_{lk})$ is the cost of link l as a function of its capacity and of the flow assigned to it, and J is the number of candidate solutions generated by the randomization procedure outlined above.

4. It is possible to follow the capacity improvement step with a route improvement procedure which is based on a modified version of the model. The solution to the following problem is the flow assignment that corresponds to the lowest queuing and variable costs for a given capacity assignment:

$$Z_F = min \left\{ \sum_{l \in L} \frac{Df_l}{1 - f_l} + C_{lj} Q_{lj} f_l \right\}$$

subject to:

$$\text{(18)} \qquad \sum_{r \in R} \lambda_r \delta_{rl} z_r / \mu \le f_l Q_{lj} \qquad \forall l \in L$$

$$L_{lj} \le f_l \le U_{lj} \qquad \forall l \in L$$

$$\sum_{r \in S_p} z_r = 1 \qquad \forall p \in \Pi$$

$$z_r = 0, 1 \qquad \forall r \in R$$

where $j \in I_l$ corresponds to the line type assigned to link l.

The constraints in (18) are relaxed and a subgradient procedure is applied to the resulting Lagrangean problem. The tests have shown that the algorithm converges very fast, a solution tolerance of under 1% being generally obtained in less than 40 iterations. Thus, a nearly optimal flow assignment is obtained without a major computational effort.

5. The above procedure can significantly change the flow pattern on the links, and as a result some other capacity assignment may become preferable in terms of the overall cost. We therefore iterate between the capacity and route improvement algorithms until no further change in the overall cost can be achieved.

The search for a local optimum is automatically triggered whenever a feasible solution with an objective function value less than $\hat{Z}$ is generated by the randomization procedure, where $\hat{Z}$ is the best feasible solution obtained so far without attempting any further improvement. Furthermore, every N iterations, the user is given the option to initiate a search using the best solution generated at the current iteration as a starting point.

6. Computational Results

The model and the algorithm presented in this paper are currently implemented in a system that allows for an easy and flexible definition of the topologies to be used and of the model parameters. At the end of each major iteration (defined as a given number of subgradient iterations, to be specified by the user), control is returned to the user. At this point, the procedure may either be stopped, if a satisfactory solution was reached, or continued. At the begining of each major iteration, the user may change the values of some of the parameters that control the procedure, like J, the number of candidate solutions to be generated by the randomization procedure, N, the number of subgradient iterations after which a search for a local optimum may be initiated, or the stopping conditions for the subgradient optimization procedure. A comprehensive output corresponding to the best feasible solution generated so far is produced, and it can be viewed by the user at the end of each major iteration. In addition to the current value of the Lagrangean, the overestimate and its corresponding average message delay, the output also gives a detailed description of the capacity assignment, specifying for each link in the network, the line type currently assigned to it, its message rate and utilization, its associated fixed, variable and queuing costs, and the percentage of the overall cost attributable to it, thus presenting the user with a full picture of the current solution that can be used as a basis for gaining further insights into the characteristics of the problem under consideration.

In order to obtain a feeling for its perfomance and behaviour under different conditions, the system was tested on several topologies and for different parameter values. Some of the results of these experiments are presented here. The runs were performed on a VAX 11/780 machine running under VMS.

Four different topologies (fig. 1-4) were used in the experiments. In all cases, each node was allowed to communicate with each other node in the network (i.e a host is assumed to be located at each node), resulting in n(n-1)/2 origin-destination pairs, where n is the number of nodes in the network. Also, it was assumed that two sessions were active at each node, each of them generating a traffic of one message per second on an average, resulting in an average traffic of four messages per second for both directions.

The set of candidate routes was obtained by the combined effect of two route generation algorithms. The first is based on a capacitated minimum cost flow algorithm. When specialized so that all arcs have a maximum capacity and cost of one, the algorithm generates a set of edge disjoint paths between any two communicating nodes in the network. For the second method, the "cost" of each arc is again assumed to be one, and a set of candidate routes for each origin-destination pair is generated by using a modified shortest path algorithm, that proceeds as follows:

For each communicating pair of nodes (i,j):

1. Determine the shortest path from i to j, and store it as a candidate route. Obviously, this will be the minimum number of hops route. Set K = number of hops in this route.

2. Set the cost of the Kth arc in the route to ∞.

3. Recompute the shortest path between i and j. If the cost of the path is finite, store it as another candidate route.

4. Reset the cost of the Kth link to 1. Set K=K-1. If K=0, stop. Else go to step 2.

The above procedure will generate up to K + 1 different routes.

Since the two algorithms may generate some identical routes, duplicate routes are eliminated.

The experiments were conducted with two main purposes in mind: first, to test the performance of the algorithms, and second, to examine the impact of various parameters on the solution generated, and thus to get a feeling for the appropriateness of the model to be used as a flexible design tool. The capacity and delay costs used as a base case are presented in Table 1. For simplicity of exposition and without loss in generality, the same set of candidate capacities was considered for all links. The values for the capacity costs are the same as the ones used in [17] and [18]. The cost of delay is an estimate based on the value to the user of the time spent while awaiting for an answer from the system. Notice also the economies of scale exhibited by the structure of the variable capacity costs.

In the following tables, in addition to the values of the best Lagrangean, best feasible solution,its breakdown into major cost components, and the ratio of the upper bound to the lower bound, we also show the value of the average delay per message (measured in milliseconds) corresponding to the best feasible solution, which can be viewed as a measure of the response time in the network.

Table 2 shows the results for different mean message lengths, measured in bits. A change in the average message length corresponds to a change in the amount of total traffic the network is expected to support. In most cases, since the capacity cost components are always dominant in the overall cost, an increase in the total load results in higher average message delays. Notice though that in the case of the OCT network, the average delay went down as a result of increasing the message length from 400 to 500 bits. As a result, the corresponding increase in the fixed capacity cost is even more significant now (32%, as opposed to roughly 18% in all other cases).

Table 3 examines the solutions obtained for different costs of delay. As expected, when the cost of a unit of delay increases, the expected delay in the network goes down, but at the expense of an increase in the line and in the traffic flow costs. Whenever the cost of a unit of delay is difficult to predict, the designer may easily generate several solutions corresponding to different values of this parameter. The resulting curve, that corresponds to the tradeoff between response time and link costs, can then be used by the decision maker as a basis for selecting the prefered alternative.

The impact of variations (50% and 150% of the base costs) in the fixed and variable costs are examined in Tables 4 and 5, respectively. It can be observed from Table 4 that as the fixed capacity costs increase, their dominance in the total cost becomes even more marked. As a result, links are assigned lower capacity values, and the average message delay goes up correspondingly. On the other hand, due to the economies of scale incorporated in the structure of the variable costs (see table 1), higher capacity values tend to become relatively more attractive as the weight of the variable cost in the overall cost increases. This effect can be observed in Table 5, though it is less significant.

Finaly, Table 6 gives a more detailed picture of the way in which the capacity assignment corresponding to the best feasible solution is affected by changes in the model parameters. The entries in the table give the capacity value (expressed in Kbps) asigned to the line. The one factor that, predictably enough, seems to have the most significant impact is the load applied to the network, represented here by changes in the average message length. Notice though that, as the network gets closer to saturation, further increases in the load have less of an impact on the capacity assignment. For instance, when the average message length increases from 100 to 200 bits, 15 links in the ARPA network are assigned a higher capacity, while, as a result of a similar change from 500 to 600 bits, only 5 links are affected. A similar observation can also be made with respect to the other parameters, namely that, due to the heavy weight in the total cost of the fixed capacity cost, as their values move in a direction that tends to increase the capacity assigned to a link, the impact of the change is felt less in terms of the capacity assignment and more in terms of the queuing cost incurred, and therefore of the expected delay in the network. For instance, a change from 400 to 1000 in the value of the cost of delay results in five links being assigned a higher capacity, while when the cost of delay increases from 2000 to 3000, a more significant change, the capacity value of only two links goes up, while link 18 is even assigned a lower capacity.

The effect of the economies of scale in the structure of the variable costs mentioned before, can also be observed in Table 6 in terms of the capacity assignment. The impact is not very significant though, and it becomes apparent only for major changes in the values of these costs.

It is important to keep in mind that the solutions generated by the model are based on often rough estimates of the external traffic requirements. It is then highly desirable to have a robust solution , i.e a solution whose cost when used under real traffic conditions does not significantly differ from its estimated cost. The next set of experiments tested the sensitivity of the solution to this parameter.

Define:

$\Lambda_e =$ the matrix of estimates of traffic requirements

$\Lambda_a =$ the matrix of actual traffic requirements

$A_e =$ capacity and routing obtained based on λ_e

$A_a =$ capacity and routing obtained based on λ_a

The following measures are then of interest: $C(\Lambda_e, A_e)$ is an estimate of the solution cost, i.e the cost of the solution as determined by the algorithm during the design stage, $C(\Lambda_a, A_e)$, is the actual cost of this solution when implemented, i.e its cost under real traffic conditions, and $C(\Lambda_a, A_a)$, the cost of the solution that would have been generated, had the actual traffic conditions been known. An important ratio that can be used as a measure of the robustness of the solutions generated by the algorithm, is $C(\Lambda_a, A_e)/C(\Lambda_a, A_a)$. Notice that this ratio will not always be greater than one, since in both cases we deal only with heuristic solutions.

In testing, great uncertainty in estimating the external traffic requirements was allowed for, by randomly generating errors within intervals ranging from ± 10% to ± 50% The results showed that the ratio is very close to one, meaning that there will be no significant difference between the actual cost of the solution generated by the algorithm, and the cost of the solution that could have been obtained had the real values of the external arrival rates been known. The solutions generated are not very sensitive to variations in the external arrival traffic, definitely an encouraging fact.

7. Conclusions

A model and solution methods for the problem of capacity and primary route assignment in computer communication networks were presented. What we see as the main value of this approach is that the model as well as the optimization procedure deal simultaneously with both aspects of the problem, thus driving the solution towards a global optimum. From the computational experience, it can be concluded that the procedure is both efficient and effective in identifying robust solutions that are satisfactorily close to the lower bound.

The present model can be generalized to deal with different classes of customers, characterized by different priorities, message lengths, and/or delay requirements. Work is currently in progress on modeling and developing the solution techniques for the case when the delay phenomena are represented as a network of nonpreemptive head of the line priority queues.

References

1. S. Abraham, H. Bodner, C. Harrington and R. White, *Advanced information systems (AIS™)/NET 1000 service; Technical overview*, Proc. 2nd Joint Conf. IEEE Computer and Commun. Soc.,San Diego, CA (1983), 87–90.
2. P. Afentakis and B. Gavish, *Computationaly efficient optimal solutions to the lot-sizing problem in multistage assembly problems*, Management Sci **30** (1984), 222–239.
3. V. Ahuja, *Routing and flow control in systems network architecture*, IBM Syst. J **18** (1979), 298–314.
4. P.M. Cashin, *Datapac network protocols*, Proc. 3rd Int. Conf. Comput. Commun, Toronto, Ont., Canada (1976), 150–155.
5. R. Despres and G. Pichon, *The TRANSPAC network status report and perspectives*, Proc. Online Conf. Data Networks Develop. Use, London, England (1980), 209–232.
6. *Digital Network Architecture, General description*, Digital Equipment Corporation, Maynard, MA (1978).
7. L. Fratta, M. Gerla and L. Kleinrock, *The flow deviation algorithm: an approach to store-and-forward computer communication network design*, Networks **3** (1973), p. 97–133.
8. B. Gavish, *Topological design of centralized computer networks: Formulations and algorithms*, Networks **12** (1982), 355–377.
9. B. Gavish and S.L. Hantler, *An algorithm for the optimal route selection in SNA networks*, IEEE Trans. Commun. COM–31 (1983), 1154–1161.
10. A.M. Geoffrion, *Lagrangean relaxation and its uses in integer programming*, Mathematical Programming Study **2** (1974), 82–114.

11. M. Gerla, *Deterministic and adaptive routing policies in packet-switched computer networks*, ACM- IEEE 3-rd Data Comm. Symp., Tampa FL (1973).
12. M. Gerla and L. Kleinrock, *On the topological design of distributed computer networks*, IEEE Trans. Commun. **COM-25** (1977), 48–60.
13. J. Gray and T. McNeil, *SNA multiple-system networking*, IBM Sys. J. **18** (1979), 263–297.
14. M. Held and R. Karp, *The travelling salesman problem and minimum spanning trees, Part II*, Math. Program. **1** (1971), 6–25.
15. S. Johari, *Burroughs network architecture: an overview*, Proc. 2nd Joint Conf. IEEE Computer and Commun. Soc., San Diego, CA (1983), 503–511.
16. L. Kleinrock, "Communication nets: stochastic message flow and delay", McGraw-Hill, 1964.
17. K. Maruyama, K. Fratta and D. Tang, *Heuristic design algorithm for computer communication networks with different classes of customers*, IBM J.Res.Devel. (1977), 360–369.
18. K. Maruyama and D. Tang, *Discrete link capacity assignment in communication networks*, 3rd ICC Toronto (1976), 92-97.
19. B.T. Poljack, *A general method for solving extremum problems*, Sov.Math.Doklady **8** (1967), 593–597.
20. "TELENET Communications Corporation Packet Switching Network", New York: Auerbach, 1978.
21. L. Tlymes, *Routing and flow control in TYMNET*, IEEE Trans. Commun **COM-29** (1981), 392–398.

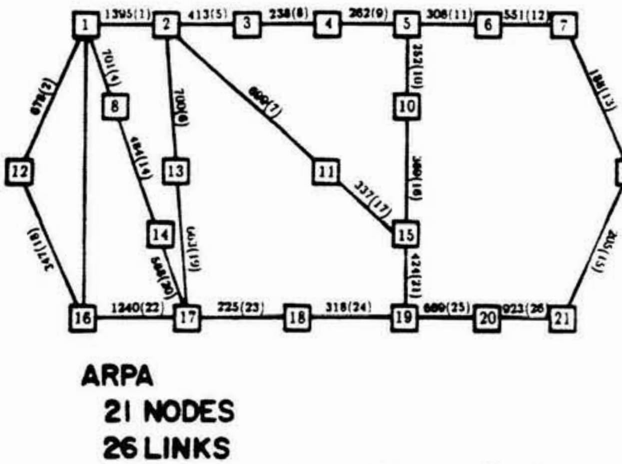


Figure 1: Topology and distances for the ARPA network

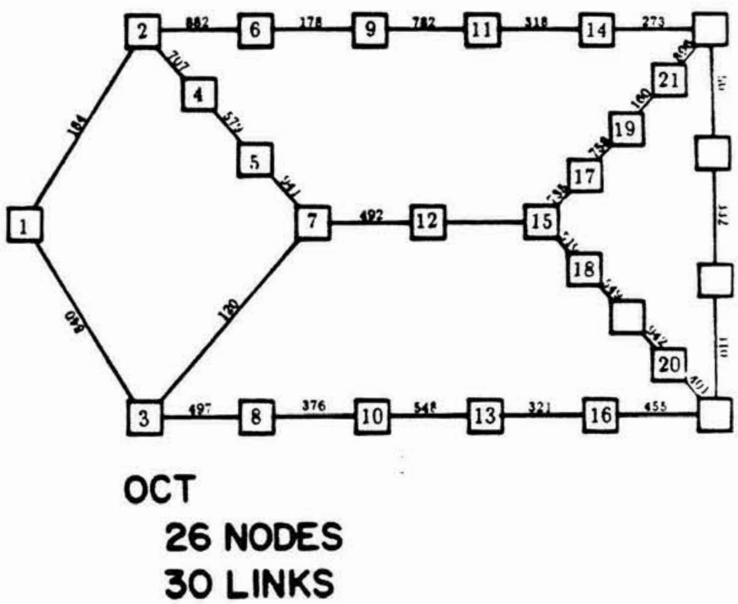


Figure 2: Topology and distances for the OCT network

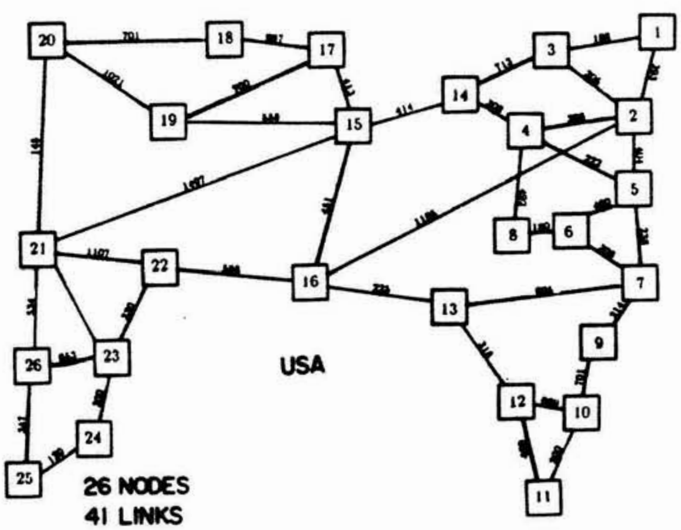
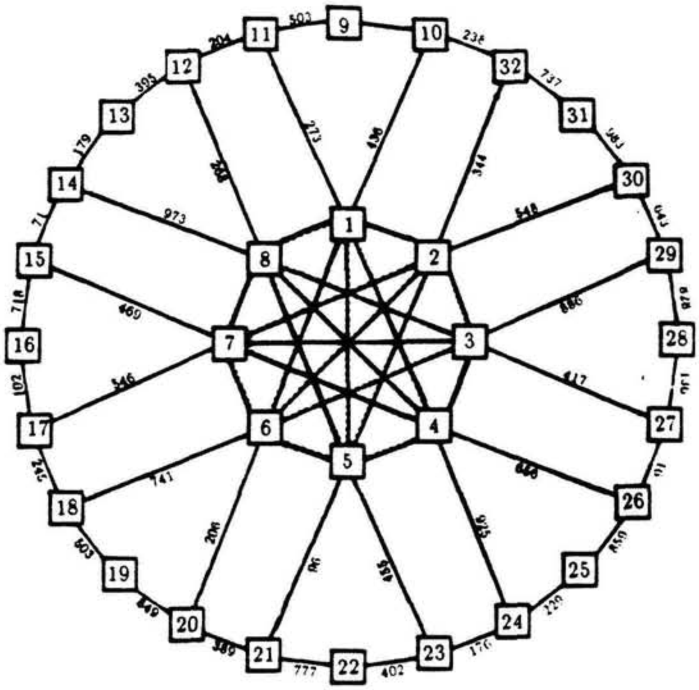


Figure 3: Topology and distance for the USA network



RING
32 NODES
60 LINKS

	2	3	4	5	6	7	8
1	563	X	57	273	326	X	505
2		623	X	453	898	447	X
3			846	X	452	720	426
4				442	X	262	473
5						660	X
7							620

Figure 4: Topology and distances for the RING network

Network ID	Message length	Lower bound	Upper bound	Queuing cost	Fixed cost	Variable cost	Upper/Lower	Average message delay
ARPA	100	107838	132111	25586	90684	15840	1.225	7.6
ARPA	200	165543	186457	37505	121149	27802	1.126	11.2
ARPA	300	224499	245798	61014	145280	39504	1.095	18.2
ARPA	400	288567	311079	82854	177358	50867	1.078	24.7
ARPA	500	355536	377538	98620	210958	67960	1.061	29.6
ARPA	600	428101	446692	111768	263663	71261	1.043	33.3
OCT	300	399496	421346	93196	256673	71477	1.055	17.9
OCT	400	524964	560794	165875	303460	91459	1.068	31.9
OCT	500	652882	698284	145212	445740	107332	1.069	27.9
USA	300	342003	366102	87428	219264	59410	1.070	16.8
USA	400	435906	463281	96663	294975	71643	1.063	18.4
USA	500	534203	560667	114226	357093	89348	1.049	22.0
RING	300	453291	487288	106233	300166	80890	1.075	13.4
RING	400	571368	595285	138505	352472	104307	1.042	17.5
RING	500	686015	714269	161495	427962	124812	1.041	20.4

TABLE 2: Summary of computational results for different message lengths

CAPACITY [bps]	SETUP COST [dollars/month]	DISTANCE COST [dollars/month/mile]	VARIABLE COST [dollars/month/bps]
4800	650	.4	.36
9600	750	.5	.252
19200	850	2.1	.126
50000	850	4.2	.03
108000	2400	4.2	.024
230000	1300	21	.02
460000	1300	60	.017

Delay Cost = 2000 [dollars/month/message]

Average Message Length = 400 [bits]

TABLE 1: Capacity set and base costs used in computational experiments

Network ID	Delay cost	Lower bound	Upper bound	Queuing cost	Fixed cost	Variable cost	Upper/Lower	Average message delay
ARPA	1	170337	196582	236	142113	54234	1.154	140.2
ARPA	100	183236	207904	8606	145646	53651	1.134	51.2
ARPA	400	212030	231667	25676	152902	52909	1.092	38.5
ARPA	1000	246569	265822	47187	166047	52589	1.078	28.1
ARPA	2000	287428	311079	82854	177358	50867	1.082	24.7
ARPA	3000	313269	343919	89679	204730	49510	1.097	17.8
OCT	1	313863	361092	162	267606	93324	1.150	62.2
OCT	100	338797	386999	14162	277311	95520	1.142	54.5
OCT	1000	453570	474905	93978	288133	92794	1.047	36.1
OCT	2000	524964	560794	165875	303460	91459	1.068	31.9
OCT	3000	578460	620706	184092	298386	88227	1.073	17.2
USA	1	247415	299175	418	215361	833395	1.162	160.9
USA	100	283124	320249	16328	223691	80230	1.131	62.8
USA	1000	374321	404534	61726	266984	75824	1.080	23.7
USA	2000	435906	463281	96663	294975	71643	1.063	18.4
USA	3000	485875	510445	127416	319670	72358	1.050	16.3
RING	1	311688	385053	435	272311	112307	1.160	109.6
RING	100	356498	407808	17501	278205	112102	1.143	44.1
RING	1000	494690	518119	83122	327899	107098	1.047	20.9
RING	2000	571368	595285	138505	352472	104307	1.042	17.5
RING	3000	629623	664235	182058	379326	102851	1.064	16.3

TABLE 3: Summary of computational experiments for different delay costs

Network ID	Multiplier of fixed cost	Lower bound	Upper bound	Queuing cost	Fixed cost	Variable cost	Upper/Lower	Average message delay
ARPA	.5	198527	211307	57775	104329	49203	1.064	17.2
ARPA	1.	287428	311079	82854	177358	50867	1.082	24.7
ARPA	1.5	366563	390073	92450	254513	51110	1.085	27.5
ARPA	3.	584014	645497	104802	487497	53197	1.120	31.2
OCT	.5	354963	377116	77864	213179	86074	1.062	15.0
OCT	1.	524964	560794	165875	303460	91459	1.068	31.9
OCT	1.5	675660	710960	170773	448169	92038	1.052	32.8
USA	.5	299310	312465	79225	161429	71811	1.043	15.2
USA	1.	435906	463281	96663	294975	72643	1.063	18.4
USA	1.5	557588	598242	112852	409995	75395	1.072	21.7
RING	.5	389009	410976	103446	207018	100512	1.055	13.0
RING	1	571368	595285	138505	352472	104307	1.042	17.5
RING	1.5	728896	765569	159691	498921	106957	1.050	20.1

TABLE 4: Summary of computational experiments for different fixed capacity costs

Network ID	Multiplier of variable cost	Lower bound	Upper bound	Queuing cost	Fixed cost	Variable cost	Upper/Lower	Average message delay
ARPA	0	238517	260165	82800	177358		1.090	24.6
ARPA	.5	263763	285643	82851	177358	25434	1.082	24.7
ARPA	1.	288567	311079	82854	177358	50867	1.078	24.7
ARPA	1.5	312435	336694	81207	179417	76070	1.077	24.2
ARPA	3.	384011	412758	81622	179417	151718	1.074	24.3
OCT	0	438502	473164	125188	347976		1.086	24.1
OCT	.5	480119	515348	135990	334030	45328	1.073	26.2
OCT	1	524964	560794	165875	303460	91459	1.068	31.9
OCT	1.5	569980	608917	112363	362701	133853	1.070	21.6
USA	0	361728	388792	115282	273510		1.074	22.2
USA	.5	398991	425307	108421	279321	37565	1.066	20.8
USA	1.	435906	463281	96663	294975	72643	1.063	18.4
USA	1.5	472374	500171	94640	296230	109301	1.058	18.2
RING	0	472373	491079	143814	347266		1.039	18.1
RING	.5	519770	543165	141980	348846	52659	1.045	17.8
RING	1.	571368	595285	138505	352472	104307	1.042	17.5
RING	1.5	622047	647842	136323	353346	156173	1.0413	17.4

TABLE 5: Summary of computational experiments for different variable capacity costs

Link ID	Base case	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9
1	108	230	230	108	108	230	108	108	108	230
2	108	108	108	48	50	108	50	50	50	108
3	108	108	108	108	48	108	108	50	19.2	108
4	108	108	108	50	50	108	108	50	108	108
5	230	230	230	108	180	230	230	230	230	230
6	108	108	108	108	50	108	108	108	108	108
7	108	230	230	108	50	108	108	108	108	108
8	230	230	230	108	108	230	230	230	230	230
9	230	230	230	108	108	230	230	230	230	230
10	230	230	230	108	50	230	230	108	230	230
11	230	230	230	108	108	230	230	230	230	230
12	230	230	230	108	50	230	108	108	108	230
13	108	230	230	108	50	230	108	108	108	230
14	108	108	108	50	50	108	108	50	108	108
15	108	230	108	108	50	108	108	108	108	108
16	230	230	230	108	50	230	230	108	230	230
17	108	230	230	108	50	230	108	108	108	230
18	108	108	108	108	50	50	50	50	50	50
19	108	108	108	108	50	108	108	108	108	108
20	108	230	108	108	50	108	108	108	108	108
21	230	230	230	108	108	230	230	230	230	230
22	108	108	108	108	50	108	108	108	108	108
23	230	460	230	230	108	230	230	230	230	230
24	230	460	230	230	108	230	230	230	230	230
25	230	230	230	108	108	230	230	230	230	230
26	230	108	108	108	50	108	108	108	108	108

Case 1: Average message length = 600 [bits]
Case 2: Average message length = 500 [bits]
Case 3: Average message length = 200 [bits]
Case 4: Average message length = 100 [bits]
Case 5: Cost of delay = 3000 [dollars/month/message]
Case 6: Cost of delay = 1000 [dollars/month/message]
Case 7: Cost of delay = 400 [dollars/month/message]
Case 8: Fixed capacity costs = 3 * base case
Case 9. Variable capacity costs = 3 * base case

TABLE 6: Impact of parameter changes on the capacity assignment (ARPA network)