Pergamon

Int. Trans. Opl Res. Vol. 5, No. 3, pp. 211–231, 1998

PII: S0969-6016(98)00007-0 0969-6016/98 $19.00 + 0.00

Design of Interconnection of Local Area Networks

VIRGILIO JOSÉ MARTINS FERREIRA FILHO[1] and ROBERTO DIÉGUEZ GALVÃO[2]

[1]School of Engineering/Federal University of Rio de Janeiro, Rio de Janeiro, Brazil and [2]COPPE/Federal University of Rio de Janeiro, Rio de Janeiro, Brazil

This paper addresses the interconnection of local area networks (LANs) using bridges. The interconnection problem is difficult to solve and feasible solutions are usually obtained through heuristics. The description of a simplified approach to solve the problem of interconnecting LANs is followed by the formulation of the problem and the development of a Lagrangean relaxation. A Tabu search metaheuristic for designing minimum cost spanning tree topologies is also presented. Computational results are given for the interconnection of up to ten local area networks. © 1998 Published by Elsevier Science Ltd on behalf of IFORS. All rights reserved

Key words: Topological design, telecommunication networks, local area networks.

1. INTRODUCTION

Local area networks (LANs) are used by many corporations and universities. They may be used for sharing databases, increasing processing power, accessing a specific machine and/or specialised programs. They provide higher reliability and bring the services of a computer closer to the end-user. As the importance of LANs grows, the requirements placed on these networks exceed the capacity of a single LAN. This has caused considerable interest in the interconnection of LANs. LANs interconnection provides a way to extend the LAN environment in physical extent, number of stations, performance and reliability (see Backes, 1988; Bux *et al.*, 1987; Green *et al.*, 1990).

Transparent (spanning tree) bridges (Backes, 1988) are one of the most commonly used devices in the interconnection of LANs. These bridges are based on a logical tree topology. Logical trees may be obtained by implementing a minimal spanning tree algorithm (Perlman, 1984). As the logical topology must be a tree, the physical topology must be carefully designed to avoid excessive delays in the messages that pass through the LANs. While performance analysis and protocol specification for internetworking has received the attention of many researchers, the design of the corresponding topologies has not been explored in a systematic way and the literature in this field is scarce. The papers by Boorstyn and Frank (1977) and Gerla and Kleinrock (1977) offer an early overview of network design optimisation problems and are a good introduction to the theme. Gavish (1991) and Ferreira Filho and Galvão (1994) are surveys of network design problems. The issues of *Annals of Operations Research* (Gavish, 1992a) and *Operations Research* (Gavish, 1995), dedicated to Operations Research in Telecommunications, provide the reader with a broad understanding of the subject. Papers that deal directly with interconnection design are: Gupta and Ross (1991), Liang and Yee (1994), Clarke and Anandalingam (1990), Fetterolf (1990), Fetterolf and Anandalingan (1991, 1992a,b), Ersoy and Panwar (1993), Ferreira Filho (1995).

Corresponding author: Roberto D. Galvão, Prog. Eng. Produção, COPPE/UFRJ, Caxia Postal 68507, 21945-970 Rio de Janeiro, RJ, Brazil

1.1. Topological design of the interconnection of local area networks

The problem of interconnecting local area networks may be defined as follows: **given** the location of end-users and the matrix of external traffic requirements (which specifies message rates and lengths between each pair of origin/destination users), **find** the topology of the interconnection, the capacity of the local area networks and the routes to be used in order to **minimize** the total system cost, **subject** to constraints on the interconnection design (which may include restrictions on end-to-end delay and reliability), **and to** connectivity constraints. There are no exact procedures available for this problem and the proposed methodologies generally use heuristics to obtain feasible solutions and Lagrangean relaxation to obtain the corresponding lower bounds.

The present paper describes a mathematical programming model for the interconnection design that does not rely on direct decomposition of the problem, but uses simplifying assumptions that make the problem easier to solve. One of these assumptions consists of considering that both the number of networks to be interconnected and the station assignments are known. This assumption is justified by the natural grouping of users that exists in organizations (e.g. stations in the same department or division). It is also assumed that the topology of the interconnection is a tree. This is a technological restriction imposed by most bridges. With this assumption the message routes are automatically determined, since in tree topologies there is only one route for each origin/destination pair.

Under the assumptions above the problem may be defined as: **given** a set of LANs (number, location and capacity) to be interconnected and the matrix of external traffic requirements, **find** the topology of the interconnection (location of the bridges), seeking to **minimise** the overall network costs, **subject to** the condition that the topology must be a tree. The overall network costs consist of two major cost components: the investment costs (costs of installing the bridges) and the delay costs imposed on messages carried by the networks.

1.2. Organization of the paper

This paper is organized as follows. A formulation for the problem of interconnecting local area networks is given in Section 2. In Section 3 a Lagrangean relaxation for the simplified formulation of Section 2 is presented. A Tabu search metaheuristic to obtain feasible solutions for the problem is given in Section 4; this is followed, in Section 5, by the presentation of some computational results. Finally, the conclusions are given in Section 6.

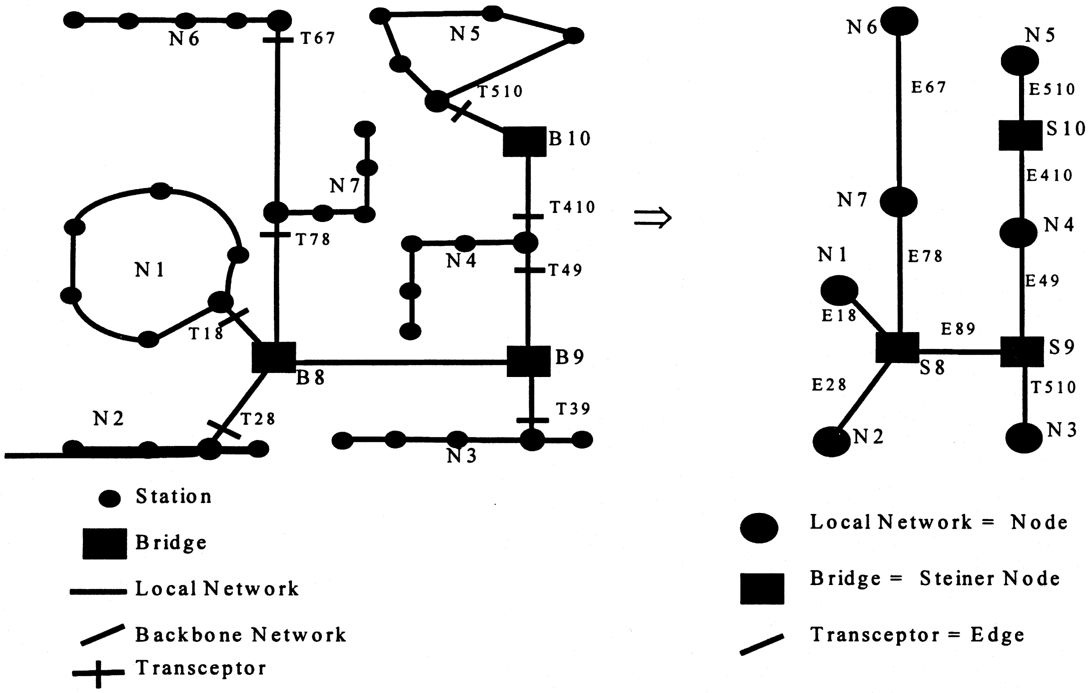
2. PROBLEM FORMULATION

The problem consists of interconnecting networks using bridges, which may be of two basic types: bridges with two ports (transceptors) or bridges with three or more ports (bridges). This problem can be represented by a graph $G = (V,L)$, where the nodes (V) represent the networks and the bridges (subset of networks represented by N, subset of bridges by S), and the edges (L) represent the transceptors (see Fig. 1).

The problem is to choose a subset of L that connects all the nodes in N at minimum cost. This problem is known as the Steiner problem in graphs and its solution is a subtree in $G = (V,L)$ that spans all nodes in N and some nodes in S (called Steiner nodes).

The traffic requirements are represented by a matrix $\lambda = [\lambda_{ij}]_{v \cdot v}$ where λ_{ij} represents the average traffic demand from node V_i to node V_j. Since the traffic demand between bridges and networks, and between bridges and bridges is zero, it follows:

$$\lambda_{ij} = 0 \; if \begin{cases} i \in S \text{ and } j \in N \\ i \in N \text{ and } j \in S \\ i \in S \text{ and } j \in S \end{cases}$$



(a) Physical Representation (b) Mathematical Model

Fig. 1. Problem representation.

Notation

In order to simplify the formulation, the term commodity is used to represent a specific pair of origin-destination communicating networks. A summary of the notation and the mathematical formulation of the problem are given below.

Indices

i, j, k, n, s, t	node indices.
l, b	bridge indices.
r	route index.
p	commodity index.

Sets

Π	set of indices of all origin/destination pairs (commodities) in the network.
S_p	subset of candidate routes that support commodity p, $p \in \Pi$.
$R = \bigcup_{p \in \Pi} S_p$	set of all candidate routes.
N	set of indices of all networks.
S	set of indices of all bridges.
$S_0 = S \cup \{0\}$	where 0 represents an auxiliar bridge (used to model the Steiner tree).
$V = N \cup S$	set of indices of all nodes in the graph.
V_0	$N \cup S_0$.
L_r	set of indices of all links used by route r.
$L = \bigcup_{r \in R} L_r$	set of candidate links that can be used in the interconnection.
$L_0 = \; = L \cup \{(0,i) \mid i \in S \cup \{1\}\}$	where 1 is any node that belongs to N.
$P_k = \{(k,j) \mid (k,j) \in L\} \cup \{(j,k) \mid (j,k) \in L\}$	that is, the set of all links where the node k is an extreme point.

Parameters

λ_p	trafficc generated by commodity p (messages/time unit). If the node N_i is the origin and the node N_j is the destination of commodity p then: $\lambda_p = \lambda_{ij}$.
τ_i	detection and propagation delay in network i, in time units.
$\overline{X_{Ni}}$	average transmission time of a message in network i.

$\overline{X^2_{Ni}}$ mean-square transmission time in network i.
$\overline{X_{Bij}}$ average processing time of a message in transceptor (i,j).
$\overline{X^2_{Bij}}$ mean-square processing time in transceptor (i,j).
$\overline{X_{Si}}$ average processing time of a message in bridge i.
$\overline{X^2_{Si}}$ mean-square processing time in bridge i.
Ca cost of the message delay in the interconnection.
Cb_{ij} transceptor B_{ij} installation cost.
Cs_i bridge S_i installation cost.

Indicator functions

These functions must be provided for a given set of candidate routes, links and networks (they are not decision variables).

$$\delta_{ij}^r = \begin{cases} 1 \text{ if link } (i,j) \in L \text{ appears in route } r, \\ 0 \text{ otherwise.} \end{cases}$$

$$\rho_i^r = \begin{cases} 1 \text{ if network } i \in N \text{ appears in route } r, \\ 0 \text{ otherwise.} \end{cases}$$

$$\xi_i^r = \begin{cases} 1 \text{ if bridge } i \in S \text{ appears in route } r, \\ 0 \text{ otherwise.} \end{cases}$$

$$\Delta_{ij} = \begin{cases} 0 \text{ if } i \in S \text{ and } j \in S \\ 1 \text{ if } \begin{cases} i \in N \text{ and } j \in N \\ i \in N \text{ and } j \in S \\ i \in S \text{ and } j \in N \end{cases} \end{cases}$$

Decision variables

$$Y_{ij} = \begin{cases} 1 \text{ if a link exists between nodes } i \text{ and } j, \\ 0 \text{ otherwise.} \end{cases}$$

$$\upsilon_i = \begin{cases} 1 \text{ if bridge } i \in S \text{ is used,} \\ 0 \text{ otherwise.} \end{cases}$$

$$W_r = \begin{cases} 1 \text{ if route } r \in R \text{ is selected,} \\ 0 \text{ otherwise.} \end{cases}$$

Design variables

$\lambda_{Ni} = \sum_{r \in R} W_r \lambda_r \rho_i^r$ number of messages passing through the network N_i per unit of time;
$\lambda_{Bij} = \left(\sum_{r \in R} W_r \lambda_r \delta_{ij}^r\right)\Delta_{ij}$ number of messages passing through the transceptor B_{ij} per unit of time;
$\lambda_{Si} = \sum_{r \in R} W_r \lambda_r \xi_i^r$ number of messages passing through the bridge S_i per unit of time.

2.1. Mathematical formulation

The objective of the interconnection design is to balance the overall investment in the network (installation of transceptors and bridges), versus the cost of delays imposed on network users.

Higher investments in the network lead to lower queueing delays (and costs). The design attempts to minimize the sum of these two costs. The costs related to the installation of the transceptors and the bridges can be obtained from the following expression:

$$Cinst = \sum_{(i,j)\in L} (Cb_{ij} Y_{ij} \Delta_{ij}) + \sum_{i\in S} (Cs_i \upsilon_i).$$

Costs associated with the delay encountered by messages carried over the network are calculated by $Ca.\lambda.T$, where T is the average end-to-end delay of messages and $Ca.\lambda$ is the unit cost of message delay in the network. By varying the value of Ca we can change the tradeoffs between direct network costs and the delays imposed on network users. Not all the above cost components exist in each design. When some of these components are irrelevant, they can be set equal to zero.

2.2. Delay model

The delay of messages in the interconnection is a combination of the queueing delays due to LAN's, transceptors and bridges, as described in Bertsekas and Gallager (1987). The delay in each one of these devices is a function of the total flow of messages in the device. Because of the complexity of the overall design problem we need an efficient way of approximating the average interconnection delay.

The approximate average end-to-end delay T is obtained by adding the delays incurred by a message in each one of the commodities (origin-destination pair), weighted by the probability of occurrence of this commodity; that is,

$$T = \sum_{p\in\Pi} \frac{\lambda_p}{\lambda} T_p,$$

where $\lambda = \sum_{p\in\Pi} \lambda_p$ and T_p is the sum of the delays in each of the networks, transceptor and bridges in the route that supports the commodity p (μ_p). Thus

$$T_p = \sum_{N_i\in\mu_p} T_{N_i} + \sum_{B_{ij}\in\mu_P} T_{B_{ij}} + \sum_{S_i\in\mu_p} T_{S_i}.$$

This model for the end-to-end delay is similar to that presented by Kleinrock (1964) to packet switched networks and accepts the Independence Assumption (which states that each time a message is received in a network or in a bridge a new packet length is chosen independently from an exponential distribution) as valid. If static routing is assumed, and considering that the topology is a tree, there is a unique path associated with each commodity p. Using the indicator functions defined previously, we can calculate the aggregate flows in each of the devices: networks, transceptors and bridges (λ_N, λ_B, λ_S, respectively). There is a queue associated with each channel. We assume that all queues operate independently and have infinite buffers. As seen above, the arrival rate for each channel is given by λ_N, λ_B or λ_S, according to the type of the device. After finding the delays caused by networks, transceptors and bridges, and some algebraic manipulation, the average end-to-end delay may be expressed as

$$T = \frac{1}{\lambda} \left\{ \sum_{i\in B} T_{N_i} \lambda_{Ni} + \sum_{(i,j)\in L} T_{B_{ij}} \lambda_{Bij} + \sum T_{s_i} \lambda_{Si} \right\}.$$

The average delay in the networks may be calculated by the simplified expression presented by Bertsekas and Gallager (1987) for ETHERNET type networks, under the following assumptions: non-persistent CSMA/CD protocol; Poisson arrivals with rate λ and a finite number of stations. Although the simplified expression does not represent exactly the operational conditions of an ETHERNET network (which uses unslotted 1-persistent CSMA/CD with binary exponential backoff, according to the IEEE 802.3 standard), for small values of the $\tau/\overline{X}$ approximation it is good (Bertsekas and Gallager, 1987), and is adequate for the objectives of the present work, since it will be incorporated into a design strategy where other simplifications are necessary in order to make the process tractable.

The average delay incurred by a message in a transceptor or in a bridge may be obtained from the M/G/1 queueing model. The arrivals process to bridges is given by the departure process from a limited number of networks. In this situation the assumption of Poisson arrivals is somewhat unrealistic, but the intractability of other models justifies this simplification.

Given the above considerations, the total delay may be obtained from the following expression, where the first term is the delay in the networks, the second term the delay in transceptors and the last term the delay in bridges:

$$T = \frac{1}{\lambda}\left\{\sum_{i\in N}\left(d_i + \frac{a_i + b_i\lambda_{Ni}}{2 - c_i\lambda_{Ni}}\right)\lambda_{Ni} + \sum_{(i,j)\in L}\left(e_{ij} + \frac{f_{ij}\lambda_{Bij}}{2 - 2e_{ij}\lambda_{Bij}}\right)\lambda_{Bij} + \sum_{i\in S}\left(g_i + \frac{h_i\lambda_{SI}}{2 - 2g_i\lambda_{Si}}\right)\lambda_{Si}\right\},$$

where the coefficients were used to simplify the expression; they are given by:

$$a_i = 4.62\tau_i,\ b_i = 2\tau_i\overline{X_{Ni}} + \overline{X^2_{Ni}},\ c_i = 2(3.31\tau_i + \overline{X_{Ni}}),\ d_i = \overline{X_{Ni}},$$

$$e_{ij} = \overline{X_{Bij}}, f_{ij} = \overline{X^2_{Bij}},\ g_i = \overline{X_{Si}},\ h_i = \overline{X^2_{Si}}.$$

The coeficients a_i, b_i, c_i and d_i were obtained directly from Bertsekas and Gallager (1987). The other coefficients are the first and second moments of the processing time in the interconnection devices.

The problem may be then formulated as:

Problem P

$$Z = Min\left\{\begin{array}{l} Ca.\left\{\sum_{i\in N}\left(d_i + \frac{a_i + b_i\lambda_{Ni}}{2 - c_i\lambda_{Ni}}\right)\lambda_{Ni} + \sum_{(i,j)\in L}\left(e_{ij} + \frac{f_{ij} + \lambda_{Bij}}{2 - e_{ij}\lambda_{Bij}}\right)\lambda_{Bij} + \sum_{i\in S}\left(g_i + \frac{h_i + \lambda_{Si}}{2 - 2g_i\lambda_{Si}}\right)\lambda_{Si}\right\} \\ + \sum_{(i,j)\in L}(Cb_{ij}Y_{ij}\Delta_{ij}) + \sum_{i\in S}(Cs_i\upsilon_i) \end{array}\right\} \tag{1}$$

subject to:

$$\lambda_{ii} \le \lambda_{Ni} \le \frac{1}{c_i}\ \forall i \in N \tag{2}$$

$$0 \le \lambda_{Si} \le \frac{1}{g_i}\upsilon_i\ \forall i \in S \tag{3}$$

$$(\lambda_{ii} + \lambda_{jj})\Delta_{ij}Y_{ij} \le \lambda_{Bij} \le \frac{1}{e_{ij}}\Delta_{ij}Y_{ij}\ \forall(i,j) \in L \tag{4}$$

$$\lambda_{Ni} \ge \sum_{r\in R} W_r\lambda_r\rho^r_i\ \forall i \in N \tag{5}$$

$$\lambda_{Si} \ge \sum_{r\in R} W_r\lambda_r\xi^r_i\ \forall i \in S \tag{6}$$

$$\lambda_{Bij} \ge (\sum_{r\in R} W_r\lambda_r\delta^r_{ij})\Delta_{ij}\ \forall(i,j) \in L \tag{7}$$

$$\sum_{r \in S_p} W_r = 1 \ \forall p \in \Pi \tag{8}$$

$$\{Y_{ij}\} \text{ forms a spanning tree on } (V_0, L_0) \tag{9}$$

$$Y_{0i} + Y_{st} \leq 1 \ \forall (s,t) \in P_i, i \in S \tag{10}$$

$$\upsilon_i = 1 - Y_{0i} \ \forall i \in S \tag{11}$$

$$Y_{ij} \in \{0,1\} \ \forall (i,j) \in L_0 \tag{12}$$

$$\upsilon_i \in \{0,1\} \ \forall i \in S \tag{13}$$

$$W_r \in \{0,1\} \ \forall r \in R \tag{14}$$

Restrictions (5), (6) and (7) express the number of messages passing through networks, bridges and transceptors, respectively. Limits on the amount of traffic that can be handled by networks, bridges and transceptors are imposed by constraints (2), (3) and (4), respectively. From (8) it follows that exactly one route must be selected for each commodity (this assumes a non-bifurcated flow).

Constraints (9), (10) and (11) ensure that the topology is a Steiner tree (see Beasley, 1989). Constraints (9) ensure that the links chosen form a spanning tree on the augmented graph $G_0 = (V_0, L_0)$. Constraints (10) establish that if link $(0,i)$ $i \in S$ is used, no link that is connected to node i may be used (therefore node i has degree 1). Constraints (11) ensure that if a bridge is connected to the auxiliary node 0 then it is not present in the effective optimal solution. Thus any formulation for this augmented spanning tree problem with additional restrictions may be considered as equivalent to the Steiner tree problem on the original graph.

Finally, constraints (12), (13) and (14) express the binary nature of the decision variables Y, υ and W.

The model presented above has a high degree of complexity. Among the reasons for this complexity are the nonlinear nature of the objective function, the interdependence between the costs of the routes and the corresponding flows and the condition that the underlying network must form a Steiner tree.

3. A LOWER BOUND FROM A LAGRANGEAN RELAXATION

Problem P is a difficult nonlinear combinatorial optimisation problem. In fact, simplified versions of this problem, namely the capacitated minimum spanning tree problem and the Steiner problem in graphs, which handle queueing delays implicitly, are NP-complete (see Garey and Johnson, 1979). In this section we present a Lagrangean relaxation of P that provides a lower bound for the problem through subgradient optimisation.

3.1. A Lagrangean relaxation of P

A Lagrangean relaxation of P is obtained by multiplying constraints (5), (6), (7) and (10) by a vector of Lagrangean multipliers $M = [\mu_i, \sigma_i, \eta_{ij} \ \theta_{ist}]$, $M \geq 0$, and adding them to the objective function.

Lagrangean problem P(M)

$$Z(M) = Min\left\{\begin{array}{l} Ca.\left\{\sum_{i\in N}\left(d_i - \frac{1}{Ca}\mu_i + \frac{a_i + b_i\lambda_{Ni}}{2 - c_i\lambda_{Ni}}\right)\lambda_{Ni}\right\}+ \\ +Ca.\left\{\sum_{i\in S_0}\left(g_i - \frac{1}{Ca}\sigma_i + \frac{h_i\lambda_{Si}}{2 - 2g_i\lambda_{Si}}\right)\lambda_{Si}\right\}+ \\ +Ca.\left\{\sum_{(i,j)\in L_0}\left(e_{ij} - \frac{1}{Ca}\eta_{ij} + \frac{f_{ij}\lambda_{Bij}}{2 - 2e_{ij}\lambda_{Bij}}\right)\lambda_{Bij}\right\} \\ + \sum_{(i,j)\in L_0}(C\theta_{ij}Y_{ij}) + \sum_{i\in S}(Cs_i\upsilon_i - \sum_{(s,i)\in P_i}\theta_{ist})+ \\ +\sum_{r\in R}\left[\sum_{i\in N}\mu_i\rho_i^r + \sum_{i\in S_0}\sigma_i\xi_i + \sum_{(i,j)\in L_0}\eta_{ij}\delta_{ij}^r\Delta_{ij}\right]W_r\lambda_r \end{array}\right\} \tag{15}$$

subject to: (2), (3), (4), (8), (9), (11), (12), (13) and (14), where:

$$C\theta_{ij} = \begin{cases} \sum_{(s,t)\in P_j} q_{jst} & \text{if } i = 0 \text{ and } j \in S \\ 0 & \text{if } i = 0 \text{ and } j \in N \\ q_{iij} + Cb_{ij} & \text{if } (i,j) \in L \text{ and } i \in S \text{ and } j \in N \\ q_{jij} + Cb_{ij} & \text{if } (i,j) \in L \text{ and } i \in N \text{ and } j \in S \\ q_{iij} + Cb_{ij} & \text{if } (i,j) \in L \text{ and } i \in S \text{ and } j \in S \\ Cb_{ij} & \text{if } (i,j) \in L \text{ and } i \in N \text{ and } j \in N \end{cases}$$

It is well known (see Geoffrion, 1974) that for any vector of multipliers $M = [\mu_i,\sigma_i,\eta_{ij}\theta_{ist}]$, $M \geq 0$, that is, $\mu_i \geq 0$ ($\forall_i \in N$), $\sigma_i \geq 0$ ($\forall_i \in S$), $\eta_{ij} \geq 0$ ($\forall (i,j) \in L$) and $\theta_{ist} \geq 0$ ($\forall s,t \in P_i, i \in S$), the value of the objective function of the Lagrangean problem P(M) is a lower bound for problem P, that is, $Z(M) \leq Z$ ($\forall M \geq 0$). As we are interested in obtaining the best possible lower bound it is necessary to find the vector M* which solves the Lagrangean dual problem: $Z_D(M^*) = \text{Max}_{M\geq 0}\{Z(M)\}$. Any computational procedure for obtaining $Z_D(M^*)$ requires that the Lagrangean problem P(M) be efficiently solved.

It is not difficult to see that for any given vector of multipliers M the Lagrangean problem P(M) can be decomposed into four subproblems, $P_1(M)$, $P_2(M)$, $P_3(M)$ and $P_4(M)$, such that:

$$Z(M) = Z_1(M) + Z_2(M) + Z_3(M) + Z_4(M).$$

Subproblem P_1(M)

$$Z_1(M) = Min\left\{Ca.\left\{\sum_{i\in N}\left(d_i - \frac{1}{Ca}\mu_i + \frac{a_i + b_i\lambda_{Ni}}{2 - c_i\lambda_{Ni}}\right)\lambda_{Ni}\right\}\right\} \tag{16}$$

subject to: (2).

Subproblem P_2(M)

$$Z_2(M) = Min\left\{Ca.\left\{\sum_{i\in S_p}\left(g_i - \frac{1}{Ca}\sigma_i + \frac{h_i\lambda_{Si}}{2 - 2g_i\lambda_{Si}}\right)\lambda_{Si}\right\}\right\} \tag{17}$$

subject to: (3).

Subproblem P_3(M)

$$Z_3(M) = Min\left\{Ca.\left\{\sum_{(i,j)\in L_0}\left(e_{ij} - \frac{1}{Ca}\eta_{ij} + \frac{f_{ij}\lambda_{Bij}}{2 - 2e_{ij}\lambda_{Bij}}\right)\lambda_{Bij}\right\} + \sum_{(i,j)\in L_0}(C\theta_{ij}Y_{ij}) + \sum_{i\in S}(Cs_i\upsilon_i - \sum_{(s,t)\in P_i}\theta_{ist})\right\} \tag{18}$$

subject to: (4), (9), (11), (12) and (13).

Subproblem P_4(M)

$$Z_4(M) = Min\left\{\sum_{r\in R}\left[\sum_{i\in N}\mu_i\rho_i^r + \sum_{i\in S_0}\Sigma_i\xi_i^r + \sum_{(i,j)\in L_0}\eta_{ij}\xi_{ij}^r\Delta_{ij}\right]W_r\lambda_r\right\} \quad (19)$$

subject to: (8) and (14).

In the decomposition above the first subproblem minimizes the cost of the delays in the LANs, the second the cost of the delays in the bridges, the third the cost of the delays in the transceptors and the installation costs of bridges and transceptors, the fourth the total distance travelled in the selected routes.

In Section 3.2 we present the solution of the four subproblems for a fixed value of the vector of multipliers $M = [\mu_i, \sigma_i, \eta_{ij}\ \theta_{ist}]$, $M \geq 0$. These solutions are used in the subgradient optimization procedure presented in Section 3.3, with the objective of finding the vector M* providing the best possible lower bound for the problem. That is, we want to solve: $Z_D(M^*) = \text{Max}_{M\geq 0}\{Z(M)\}$.

3.2. Solution of the Lagrangean subproblems

3.2.1. Subproblem P_1^i(m). This subproblem can be decomposed into $|N|$ subproblems, one for each node (LAN). The *i*-th subproblem is given by:

Subproblem P_1^i(M)

$$Z_l^i(M) = Min\left\{Ca.\left(d_i - \frac{1}{Ca}\mu_i + \frac{a_i + b_i\lambda_{Ni}}{2 - c_i\lambda_{Ni}}\right)\lambda_{Ni}\right\} \quad (20)$$

subject to:

$$\lambda_{ii} \leq \lambda_{Ni} \leq \frac{2}{c_i}. \quad (21)$$

This is a convex nonlinear optmization problem. The objective function can be shown to be convex by proving that the Hessian is positive demidefinite in the interval given by (21) (see Ferreira Filho, 1995). λ_{Ni}^* can be calculated in the following way:

$$\frac{\partial Z_l^i(m)}{\partial\lambda_{Ni}} = 0 \Longrightarrow d_i - \frac{1}{Ca}\mu_i + \frac{2a_i + 4b_i\lambda_{Ni}^* - b_ic_i\lambda_{Ni}^{*2}}{(2 - c_i\lambda_{Ni}^*)^2} = 0$$

let

$$\tilde{\lambda}_{Ni} = \frac{2}{c_i}\left(1 - \sqrt{\frac{-2b_i + a_ic_i}{2c_i\left(d_i - \frac{\mu_i}{Ca}\right) - 2b_i}}\right).$$

It is possible to define

$$\lambda_{Ni}^* = \begin{cases} \tilde{\lambda}_{Ni} & \text{for } \mu_i > \left(d_i + \frac{a_i}{2}\right)Ca \text{ and } \tilde{\lambda}_{Ni} > \lambda_{ii}, \\ \lambda_{ii} & \text{otherwise.} \end{cases}$$

Replacing λ_{Ni}^* in the expression of $Z_l^i(M)$ leads to the optimal value $Z_1^i(M) = \tilde{Z}_l^i(M)$. The solution of subproblem $P_1(M)$ is then given by:

$$\tilde{Z}_1(M) = \sum_{i \in N} \tilde{Z}_l^i(M).$$

3.2.2. Subproblem $P_2(M)$. Subproblem $P_2(M)$ is also a nonlinear combinatorial optimization problem. It can be simplified by taking advantage of the following observation. Given a solution $\overline{v_i}$ to subproblem $P_2(M)$, the optimization over the λ_{Si} variables is separable over the bridges. From (3) it follows that $\overline{v_i} = 0 \Longrightarrow \lambda_{Si} = 0$ and $\overline{v_i} = 1 \Longrightarrow \lambda_{Si} = \lambda_{Si}^*$, where λ_{Si}^* is the optimal solution of the following problem:

Problem H_i

$$H_i(M) = Min\left\{Ca.\left(g_i - \frac{1}{Ca}\sigma_i + \frac{h_i \lambda_{Si}}{2 - 2g_i \lambda_{Si}}\right)\lambda_{Si}\right\} \tag{22}$$

subject to: $0 \leq \lambda_{Si} \leq 1/g_i$.

Problem H_i is also a convex nonlinear optimization problem. The solution of this problem can be obtained by making

$$\frac{\partial H_i(M)}{\partial \lambda_{Si}} = 0 \Longrightarrow g_i - \frac{1}{Ca}\sigma_i + \frac{2h_i \lambda_{Si}^* - g_i h_i \lambda_{Si}^{*2}}{2(1 - g_i \lambda_{Si}^*)^2} = 0$$

and is given by:

Let

$$\tilde{\lambda}_{Si} = \frac{1}{g_i}\left(1 - \sqrt{\frac{h_i}{2g_i\left(g_i - \frac{\sigma_i}{Ca}\right) - h_i}}\right).$$

It is possible to define

$$\lambda_{Si}^* = \begin{cases} \tilde{\lambda}_{Si} & \text{for } \sigma_i > g_i Ca, \\ 0 & \text{otherwise.} \end{cases}$$

Replacing λ_{Si}^* in the expression of $H_i(M)$ leads to the optimal value $H_i(M) = \tilde{H}_i(M)$. The solution of subproblem $P_2(M)$ is given by:

$$Z_2(M) = \sum_{i \in S}(\tilde{H}_i(M)).$$

3.2.3. Subproblem $P_3(M)$. Subproblem $P_3(M)$ is also a nonlinear combinatorial optimization problem. It can be simplified by taking advantage of the following observation. Given a solution Y_{ij} to subproblem $P_3(M)$, the optimization over the λ_{Bij} variables is separable over the links (transceptors). From (4) it follows that $\overline{Y_{ij}} = 0 \Longrightarrow \lambda_{Bij} = 0$ and $\overline{Y_{ij}} = 1 \Longrightarrow \lambda_{Bij} = \lambda_{Bij}^*$, where λ_{Bij}^* is the optimal solution of the following problem:

Problem G_{ij}

$$G_{ij}(M) = Min\left\{Ca.\left(e_{ij} - \frac{1}{Ca}\eta_{ij} + \frac{f_{ij}\lambda_{Bij}}{2 - 2e_{ij}\lambda_{Bij}}\right)\lambda_{Bij}\right\} \tag{24}$$

subject to:

$$(\lambda_{ii} + \lambda_{jj})\Delta_{ij} \leq \lambda_{Bij} \leq \frac{1}{e_{ij}}\Delta_{ij}. \tag{25}$$

Once again this is a convex nonlinear optimization problem. Its solution can be obtained by making

$$\frac{\partial G_{ij}(M)}{\partial \lambda_{Bi}} = 0 \Longrightarrow e_{ij} - \frac{1}{Ca}\eta_{ij} + \frac{2f_{ij}\lambda^*_{Bij} - e_{ij}f_{ij}\lambda^{*2}_{Bij}}{2(1 - e_{ij}\lambda^*_{Bij})^2} = 0$$

and is given by:

$$\text{let } \tilde{\lambda}_{Bij} = \frac{1}{e_{ij}}\left(1 - \sqrt{\frac{f_{ij}}{2e_{ij}\left(e_{ij} - \frac{\eta_{ij}}{Ca}\right) - f_{ij}}}\right)\Delta_{ij}.$$

It is possible to define

$$\lambda^*_{Bij} = \begin{cases} \tilde{\lambda}_{Bij} & \text{for } \eta_{ij} > Ca.e_{ij} \text{ and } \tilde{\lambda}_{Bij} > (\lambda_{ii} + \lambda_{jj}), \\ (\lambda_{ii} + \lambda_{jj})\Delta_{ij} & \text{otherwise.} \end{cases}$$

Replacing λ^*_{Bij} in the expression of G_{ij} leads to the optimal value of the objective function $G_{ij}(M) = \tilde{G}_{ij}(M)$.

As $\upsilon_i = 1 - Y_{0i}$, then:

$$Z_3(M) = Min\left\{\sum_{(i,j)\in L_0}(\tilde{G}_{ij} + C\theta_{ij})Y_{ij} + \sum_{i=0}\sum_{j\in S}(-Cs_j Y_{ij}) + \sum_{i\in S}\left(Cs_i - \sum_{(s,i)\in P_i}\theta_{ist}\right)\right\}.$$

Without loss of generality Subproblem $P_3(M)$ may be rewritten as:

$$Z_3(M) = Min\left\{\sum_{(i,j)\in L_0} C_{ij}Y_{ij} + \sum_{i\in S}\left(Cs_i - \sum_{(s,i)\in P_i}\theta_{ist}\right)\right\}. \tag{26}$$

subject to: (9) and (12), where

$$C_{ij} = \begin{cases} \tilde{G}_{ij} + C\theta_{ij} - Cs_j & \text{if } i = 0 \text{ and } j \in S; \\ \tilde{G}_{ij} + C\theta_{ij} & \text{otherwise.} \end{cases}$$

If cost (C_{ij}) is associated with link (i,j), subproblem $P_3(M)$ consists of finding the minimal spanning tree of the corresponding graph.

3.2.4. Subproblem $P_4(M)$. This subproblem may be rewritten as:

$$Z_4(M) = Min\left\{\sum_{p\in\Pi}\sum_{r\in S_p} W_r Q_r\right\} \tag{27}$$

subject to: (8) and (14), where:

$$Q_r = \left[\sum_{i\in N}\mu_i\rho^r_i + \sum_{i\in S}\sigma_i\xi^r_i + \sum_{(i,j)\in L}\eta_{ij}\delta^r_{ij}\Delta_{ij}\right]\lambda_r.$$

This problem can be decomposed into $|\Pi|$ subproblems $P^p_4(M)$, one or each origin-destination pair, the p-th subproblem being:

Subproblem $P^p_4(M)$

$$Z_4^p(M) = Min\left\{\sum_{r\in S_p} W_r Q_r\right\} \tag{28}$$

subject to:

$$\sum_{r\in S_p} W_r = 1 \tag{29}$$

$$W_r \in \{0,1\}\ \forall r \in S_p \tag{30}$$

Subproblem $P_4^p(M)$ consists of choosing a minimal cost route for commodity p. This is equivalent to finding the shortest path from origin to destination for p. To be able to find the shortest routes for all commodities it is necessary to express the coefficient Q_r in terms of costs over the links. The cost of a route in Subproblem $P_4^p(M)$ is proportional to the traffic in the route; its value corresponds to the sum of three components: the summation of $(\eta_{ij})\lambda_r$ over all transceptors, plus the summation of $(\mu_i)\lambda_r$ over all networks, plus the summation of $(\sigma_i)\lambda_r$ over all bridges in this route. Since there are no bifurcated routes, it can be considered that for each link (i,j) there corresponds a cost α_{ij} given by:

$$\alpha_{ij} = \begin{cases} (\eta_{ij} + \mu_i)\lambda_r & \text{if } i \in N \text{ and } j \in N \\ (\eta_{ij} + \mu_i)\lambda_r & \text{if } i \in N \text{ and } j \in S \\ (\eta_{ij} + \sigma_i)\lambda_r & \text{if } i \in S \text{ and } j \in N \\ (\sigma_i)\lambda_r & \text{if } i \in S \text{ and } j \in S \end{cases}$$

That is, the cost of link (i,j) includes the cost of the link (transceptor) plus the cost of the node (network or bridge) at the link's origin. The cost of a route related to a commodity p is obtained by the summation of the costs of all links in the route that supports the commodity, plus a constant, $(\mu_{j^*})\lambda_r$, the cost of the destination node j^*. This constant does not interfere with the choice of the optimal route that supports commodity p. Subproblem $P_4^p(M)$ does not contain any negative cycles because the Lagrangean multipliers η_{ij}, μ_i and σ_i are nonpositive. Let CMP be the cost of the shortest path related to commodity p, obtained using α_{ij} as the link cost. $P_4^p(M)$ and CMP can be then related in the following way:

$$Z_4^p = CM^p + \mu_j.\lambda_p\ \forall p | i \in N, j^* \in N,$$

where: j^* = index of the destination node of the pair p; λ_p = traffic associated with commodity p.

Subproblem $P_4^p(M)$ is equivalent to finding the shortest path between all origin destination pairs in a graph, using α_{ij} as the link costs. The solution of $P_4^p(M)$ is given by:

$$Z_4(M) = \sum_{p\in\Pi} Z_4^p(M).$$

3.3. *Maximization of the lower bound*

Function Z(M) is nondifferentiable over M; therefore classical optimization procedures cannot be used to solve the Lagrangean dual, Max{Z(M)}. A subgradient optimization procedure will be used to estimate Z(M^*).

Subgradient optimisation (see, for example, Held *et al.*, 1974; Geoffrion, 1974; Fisher, 1981) is an iterative procedure in which, in its k-th iteration, the updated vector of Lagrangean multipliers M^k is obtained. Using these values, the Lagrangean subproblems are solved. Based on this solution, the multipliers are updated for the next iteration in the following way:

$$M^{K+1} = Max\{0,(M^K + t_k\nabla(M^K))\},$$

where $\nabla(M^K)$ are subgradient directions given by:

$$\nabla(\mu_i^k) = -\lambda_{Ni}^k + \sum_{r\in R} W_r^k \lambda_r \rho_i^r \ \forall i \in N \tag{31}$$

$$\nabla(\sigma_i^k) = -\lambda_{Si}^k + \sum_{r\in R} W_r^k \lambda_r \xi_i^r \ \forall i \in S_0 \tag{32}$$

$$\nabla(\eta_{ij}^k) = -\lambda_{Bij}^k + \left(\sum_{r\in R} W_r^k \lambda_r \delta_{ij}^r\right)\Delta_{ij} \ \forall (i,j) \in L_0 \tag{33}$$

$$\nabla(\theta_{ist}^k) = -1 + Y_{0i}^k + Y_{st}^k \ \forall (s,t) \in P_i, i \in S, \tag{34}$$

and t_k is the step size. It has been shown (Poljack, 1967) that $Z(M^k)$ converges to $Z(M^*)$ when the sequence t_k satisfies the following conditions: $\text{Lim}_{k\to\infty}(t_k) \longrightarrow 0$ and $\sum_{k\to\infty} t_k \longrightarrow \infty$. Unfortunately these two conditions are difficult to be satisfied simultaneously. In practice t_k is approximated by the following expression:

$$t_k = \frac{\overline{Z} - Z(M^k)}{\|\nabla(M^k)\|^2} S_k,$$

where $\overline{Z}$ is a upper bound on $Z(M^*)$ and S_k is a scalar between 0 and 2. Since $Z(M^*) \leq \overline{Z}$, any feasible solution to the topological design problem can be used as $\overline{Z}$.

4. A TABU SEARCH HEURISTIC FOR THE INTERCONNECTION OF LOCAL AREA NETWORKS

Tabu search was proposed by Glover (1986). Its philosophy is to derive and exploit a collection of intelligent problem solving techniques. Tabu search is concerned with imposing restrictions to guide a search process to access regions of otherwise difficult access. These restrictions operate in several forms, both by direct exclusion of search alternatives classified as "forbidden" or tabu, and also by transference to alternative search regions. A fundamental element underlying tabu search is the use of flexible memory. From the standpoint of tabu search, flexible memory embodies the dual process of creating and exploiting structures for taking advantage of history (hence combining the activities of acquiring and storing information and profiting from stored information). For more details on Tabu search, see Glover *et al.* (1993) and Glover and Laguna (1993).

A key search parameter of tabu search implementation is the *tabu_tenure*. The value of this parameter must be adjusted to prevent the search from cycling (i.e., from indefinitely repeating the same sequence of moves) and to allow enough flexibility (as measured by the percentage of available moves at any given iteration). Given the importance of this parameter several implementations of it have been proposed (see Glover and Laguna, 1993). In this paper we use a random implementation calculated as follows: *tabu_tenure = tenure_min + U*(tenure_max − tenure_min)*; where *tenure_max* and *tenure_min* are given parameters (in the test problems we used values of $2n$ and n, respectively, where n is the number of networks) and U is a random number uniformly distributed between 0 and 1. The *tabu_tenure* value is changed at every *factor*tabu_tenure* iterations, where *factor* ranges between 2 and 3 (we used the value 2).

Tabu search methods are designed to select at each step the best move available given the current search state. Starting from a feasible solution, new solutions are obtained by repeatedly making moves from the current solution to another solution in its neighbourhood. We use the exchange of edges to move from one solution to the next. The exchange of edges is defined in two steps: in the first step an edge is chosen to be included in the solution, forming a cycle; in the second step one of the edges of the cycle is deleted, restoring the tree topology.

Let the edge to be included be identified by (*node, neighbour*). During the search a list is available with the frequencies of the nodes included in the solution. When it is necessary to chose an edge for inclusion, both *node* and *neighbour* must be chosen. The *node* in the list with the lowest frequency is identified. Ties are broken by numerical sequence. The *neighbour* is chosen in descending order of traffic intensity since this measure favours the direct link between networks with high traffic intensity. The edge to be deleted is chosen from the cycle formed. The deleted edge must be such that it leads to a minimum cost tree. In order to broaden the search space, *d neighbours* are evaluated for each *node*. The parameter *l* is used to reduce the computational cost of the algorithm. When the solution value drops *l* times for one *node* the search is interrupted and restarted from a new *node*.

The evaluation of solutions generated by moves is the most computationally expensive module of tabu search. In order to reduce this cost a procedure that calculates the savings of the move was implemented. This procedure restricts the cost evaluation to the elements (networks, bridges, transceptors) of the cycle formed.

In order to check the admissibility of moves a *tabu_list* is maintained. A move is admissible if it is non-tabu or if its tabu status is overridden by an *aspiration criterion* (a condition that allows a forbidden move to be overriden). The simplest (and also the most widely used) aspiration criterion renders a tabu move admissible if its execution leads to a solution that is better than the best found so far.

The data structure used for *tabu_list* was a V·V matrix (where V is the number of nodes, i.e. the number of networks plus bridges). The main diagonal contains the frequency of the nodes already included in visited solutions. The elements of the triangular matrix above the main diagonal show the iteration until which the inclusion of the corresponding edge is tabu. Likewise, the elements below the main diagonal show the iteration until which the exclusion of the corresponding edge is tabu. As the exchange move is a double attribute move, this move may be tabu if one of the attributes is tabu, or if both attributes are tabu simultaneously. We used the second possibility, which is less restrictive than the first.

Once the admissible moves are defined and the best one is chosen, the move is performed. The data structures are then updated. The search continues until some stopping condition is reached. The most used stopping conditions are the maximum number of iterations allowed (we used the value 500) and the maximum number of iterations performed since the last change in the best solution (we used the value 200).

The basic tabu search algorithm we developed is:

4.1. Algorithm—Tabu search

Step 1 Obtain an initial topology (a tree) and compute its cost;
Step 2 Initialize iter_coenter, best_cost, tabu_tenure and tabu_list;
Step 3 Do while stopping conditions are not met:
  A - Increment iter_counter;
  B - Choose the node with the lowest frequency in tabu_list, breaking ties by the numerical sequence of the nodes,
  C - Initialize neighbor_counter and profit_counter;
  D - If neighbor_counter $\leq$ d then:
    1 - Find the node's nearest neighbour;
    2 - Set up the link (node, neighbor);
    3 - Identify the circuit formed;
    4 - Initialize current_min_cost
    5 - Repeat for all edges in the cycle (except the edge (node, neighbor)):
      - Delete the edge;
      - Compute the cost of the resulting tree;
      - If the move is non tabu or if its cost is less than best_cost then:
        - If cost < current_min_cost then:
          - current_min_cost = cost;
          - save this topology as the best_current_solution;
        - If cost < best_cost then:

- best_cost = cost;
- increment profit_counter;
- save this topology as the best_solution;
- If profit_counter > 1 then go to A;
- Restore the edge deleted;
6 - Increment neighbor_counter;
E - Update the stopping conditions;
F - If stopping conditions are not met then:
1 - Recover best_current_solution;
2 - Calculate tabu_tenure;
3 - Update tabu_list;
Step 4 Make final solution ← best_ solution and END.

5. COMPUTATIONAL RESULTS

In order to test the proposed methodology the algorithms described in Section 3 and Section 4 were programmed in Pascal and computational experiments were performed in a 486 DX–50 microcomputer. The test data were chosen so that the influence of the following parameters could be observed: amount of total trafficc (FT), the ratio installation cost over delay costs (Cf/Ca) and the number of networks to be connected (N).

In order to obtain the coefficients for the objective function the following assumptions were made:

- all networks with capacity of 10 Megabits per second (MBPS);
- packet size exponentially distributed with average of 1600 bits;
- bridge processing time exponentially distributed with average of 200 microseconds;
- detection and propagation delay in networks (τ) of 10 microseconds;
- the installation cost of bridges and transceptors are identical and given by Cf.

Each network's own traffic set at 25%, 50% and 75% of the network capacity, corresponding to light, medium and high traffic, respectively. Interconnection of 4, 6, 8, 9 and 10 networks were considered. The ratio Cf/Ca was tested for the values 0, 0.1 and 1.

The traffic matrices were generated from uniform distributions in the following way:

$$\lambda_{ij} = \begin{cases} \bigcup\left[\dfrac{0{,}36*FT*CAP}{N-1} \quad ; \quad \dfrac{0{,}44*FT*CAP}{N-1}\right] & \forall ij | i \neq j \\ \bigcup[0{,}54*FT*CAP \quad ; \quad 0{,}66*FT*CAP] & \forall ij | i = j, \end{cases}$$

where CAP = 10 MBPS, $FT \in [0.25, 0.50, 0.75]$ and $N \in [4, 6, 8, 9, 10]$. This expression generates traffic that is $(60 \pm 6\%)$ internal to the network and $(40 \pm 4)\%$ uniformly distributed among the other $(N-1)$ networks.

A total of 93 problems were solved: three initial test problems, plus 90 problems generated by combination of the parameters Cf/Ca, FT and N. For each combination of parameters two instances are carried out and the values indicated in Tables 1–3 are the mean of the two values. Table 1 shows the gap between the best feasible solution UB obtained by the Tabu search algorithm and the Lagrangean bound LB corresponding to the problems in each group. This gap is expressed as a percentage of the upper bound, i.e. $GAP = (UB - LB)/UB \cdot 100\%$. The average gaps obtained are in the range 0,09–48%. The overall average gap for the 90 problems was 18,29%. In face of the complexity of the problem these results may be considered good.

Table 2 and Table 3 show the average running times (in seconds), for each combination of parameters, for the subgradient algorithm and the tabu search heuristic, respectively. Considering that this is a design and not an operational problem, the computational times (average 58 s and maximum 300 s for obtaining the lower and upper bounds) may be considered relatively small.

Table 1. Average gap (%)

CF/CA	FT	N 4	6	8	9	10	Total
	0,25	0,13	30,89	12,71	16,68	4,04	12,89
0	0,5	0,09	45,61	5,29	3,56	0,62	10,35
	0,75	4,44	42,69	25,37	20,73	15,94	21,83
0 Average		1,55	39,73	14,46	12,21	6,87	14,87
	0,25	32,09	16,59	2,98	14,38	2,41	13,80
0,1	0,5	13,43	47,60	0,41	24,15	0,96	17,93
	0,75	3,15	48,01	28,02	20,86	19,61	23,93
0,1 Average		16,22	37,40	10,47	19,07	7,66	18,25
	0,25	0,70	10,67	28,91	24,92	45,62	22,17
1	0,5	0,57	24,05	32,43	22,61	0,50	16,03
	0,75	23,70	33,09	33,20	26,21	20,83	27,41
1 Average		8,32	22,60	31,51	24,58	22,32	21,87
	0,25 Average	10,98	19,38	14,87	17,59	17,36	16,13
	0,5 Average	4,70	39,08	12,71	16,04	0,69	14,73
	0,75 Average	10,43	41,26	28,87	22,60	18,80	24,39
Mean with respect to N		8,70	33,24	18,81	18,39	12,28	18,29

Table 2. Average running time (secs.) for the lower bound

CF/CA	FT	N 4	6	8	9	10	Total
	0,25	0,05	21,86	30,93	37,16	258,56	77,45
0	0,5	3,11	18,84	29,20	52,18	0,50	23,62
	0,75	10,82	16,98	24,17	37,93	57,09	29,40
0 Average		5,58	19,23	28,10	43,81	105,38	41,69
	0,25	6,29	19,72	39,49	71,61	130,87	56,60
0,1	0,5	9,75	21,04	0,25	51,72	0,50	19,72
	0,75	10,19	17,00	24,25	37,98	57,21	29,32
0,1 Average		8,74	18,89	21,33	57,51	62,86	36,55
	0,25	3,00	19,52	65,70	77,81	127,19	58,64
1	0,5	0,06	22,06	54,38	37,43	0,52	22,89
	0,75	2,39	17,08	24,20	37,99	57,18	27,77
1 Average		1,82	19,55	48,09	51,07	61,63	36,43
	0,25 Average	3,72	20,37	45,37	64,55	172,20	63,31
	0,5 Average	4,30	20,57	27,94	48,32	0,50	22,12
	0,75 Average	7,80	17,02	24,21	37,96	57,16	28,83
Mean with respect to N		5,37	19,24	32,50	51,40	76,62	38,19

Figure 2 shows the behaviour of the gap for each of the problems tested. In this graph the problems are firstly sorted by the number of networks connected, secondly by the parameter FT, and thirdly by the ratio Cf/Ca. Problems 1–18 correspond to 4-network problems; 19–36, to 6-network; 37–54, to 8-network; 55–75, to 9-network; and, finally, 76–93 to 10-network problems. Table 4 presents the complete data for all the problems. The detailed analysis of the

Table 3. Average running time (secs.) for the Tabu search heuristic

CF/CA	FT	N 4	6	8	9	10	Total
	0,25	3,68	12,39	18,62	25,24	36,47	19,28
0	0,5	3,74	12,67	22,91	33,58	35,45	22,75
	0,75	5,03	12,55	18,62	24,99	36,12	19,46
0 Average		4,15	12,54	20,05	28,74	36,01	20,57
	0,25	3,68	5,97	18,57	28,67	35,71	20,21
0,1	0,5	5,60	6,34	23,26	25,03	37,24	20,00
	0,75	5,55	13,24	18,57	24,99	36,17	19,70
0,1 Average		4,94	8,52	20,13	26,64	36,37	19,98
	0,25	3,57	5,96	18,57	28,79	32,95	17,97
1	0,5	3,41	6,34	18,54	24,15	37,55	18,00
	0,75	3,40	10,28	18,54	25,32	33,97	18,30
1 Average		3,46	7,53	18,55	26,08	34,82	18,09
	0,25 Average	3,64	8,11	18,58	27,84	35,04	19,22
	0,5 Average	4,25	8,45	21,57	28,02	36,75	20,32
	0,75 Average	4,66	12,02	18,58	25,10	35,42	19,15
Mean with respect to N		4,18	9,53	19,58	27,16	35,74	19,57

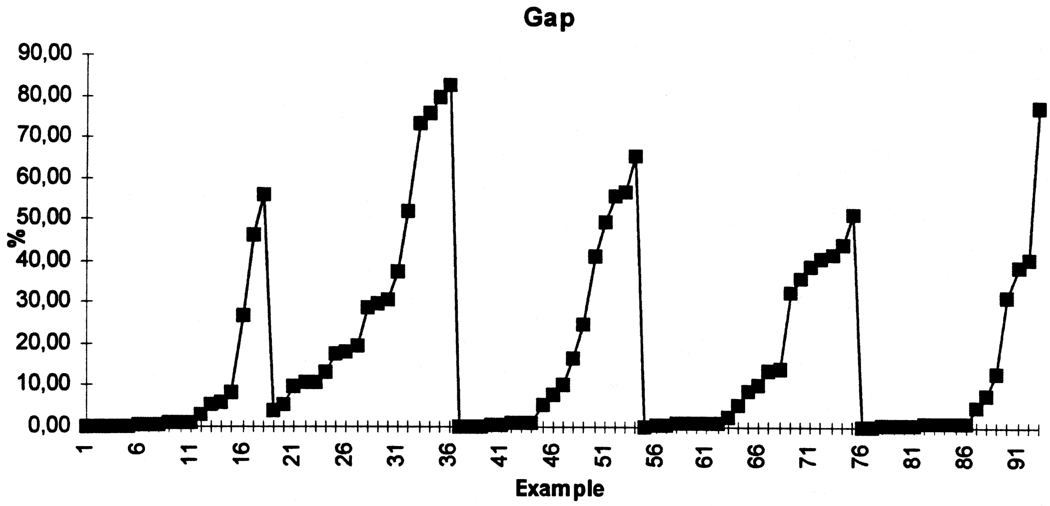


Fig. 2. The quality of solutions.

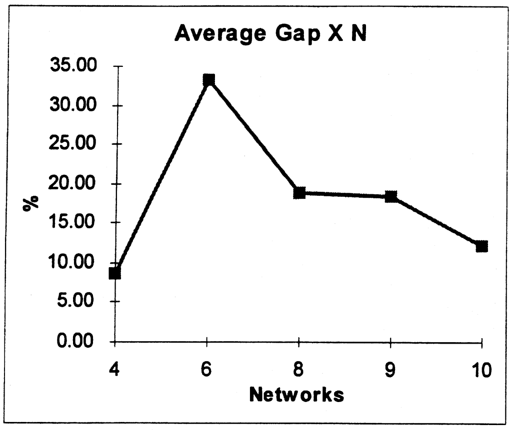
influence of each parameter in the performance of the methodology is provided in Fig. 3. Figure 3a and Fig. 3b show comparative graphs of the behaviour of the average gap and running time versus number of networks. It can be observed that running times increase and quality of solution improves (for six or more networks) when the number of networks increases.

Figure 3c and Fig. 3d show comparative graphs of the behaviour of the average gap and running time versus the ratio Cf/Ca. It can be observed that the running times are practically independent of the ratio. The quality of solution deteriorates when the ratio increases.

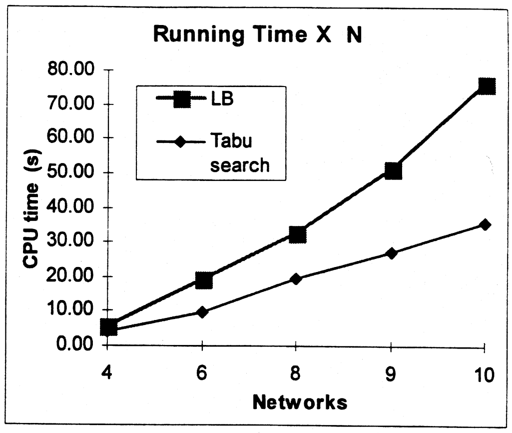
Finally, Fig. 3e and Fig. 3f show average gap and running time versus weighting factor. The smallest running times were obtained for weighting factors of 50%. The value of the gap increases with the amount of traffic.

Table 4. Complete data for the problems tested

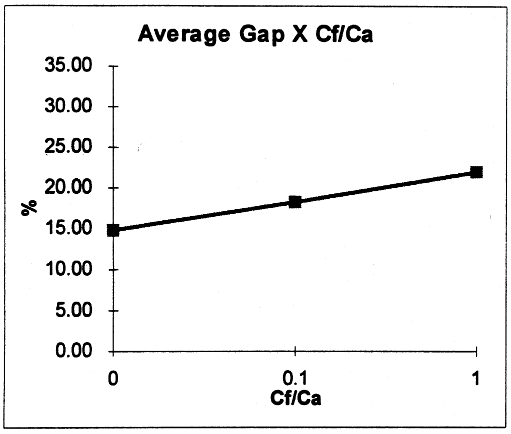
Problem	N	CF/CA	FT	Problem	N	CF/CA	FT	Problem	N	CF/CA	FT
1	4	0	0,25	32	6	0	0,75	63	9	0	0,50
2	4	0	0,25	33	6	0,1	0,75	64	9	0	0,50
3	4	0,1	0,25	34	6	0,1	0,75	65	9	0	0,50
4	4	0,1	0,25	35	6	1	0,75	66	9	0,1	0,50
5	4	1	0,25	36	6	1	0,75	67	9	0,1	0,50
6	4	1	0,25	37	8	0	0,25	68	9	1	0,50
7	4	0	0,50	38	8	0	0,25	69	9	1	0,50
8	4	0	0,50	39	8	0,1	0,25	70	9	0	0,75
9	4	0,1	0,50	40	8	0,1	0,25	71	9	0	0,75
10	4	0,1	0,50	41	8	1	0,25	72	9	0,1	0,75
11	4	1	0,50	42	8	1	0,25	73	9	0,1	0,75
12	4	1	0,50	43	8	0	0,50	74	9	1	0,75
13	4	0	0,75	44	8	0	0,50	75	9	1	0,75
14	4	0	0,75	45	8	0,1	0,50	76	10	0	0,25
15	4	0,1	0,75	46	8	0,1	0,50	77	10	0	0,25
16	4	0,1	0,75	47	8	1	0,50	78	10	0,1	0,25
17	4	1	0,75	48	8	1	0,50	79	10	0,1	0,25
18	4	1	0,75	49	8	0	0,75	80	10	1	0,25
19	6	0	0,25	50	8	0	0,75	81	10	1	0,25
20	6	0	0,25	51	8	0,1	0,75	82	10	0	0,50
21	6	0,1	0,25	52	8	0,1	0,75	83	10	0	0,50
22	6	0,1	0,25	53	8	1	0,75	84	10	0,1	0,50
23	6	1	0,25	54	8	1	0,75	85	10	0,1	0,50
24	6	1	0,25	55	9	0	0,25	86	10	1	0,50
25	6	0	0,50	56	9	0	0,25	87	10	1	0,50
26	6	0	0,50	57	9	0,1	0,25	88	10	0	0,75
27	6	0,1	0,50	58	9	0,1	0,25	89	10	0	0,75
28	6	0,1	0,50	59	9	0,1	0,25	90	10	0,1	0,75
29	6	1	0,50	60	9	0,1	0,25	91	10	0,1	0,75
30	6	1	0,50	61	9	1	0,25	92	10	1	0,75
31	6	0	0,75	62	9	1	0,25	93	10	1	0,75



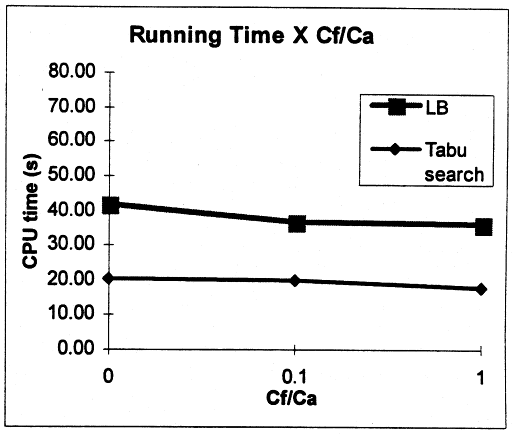
(3a)



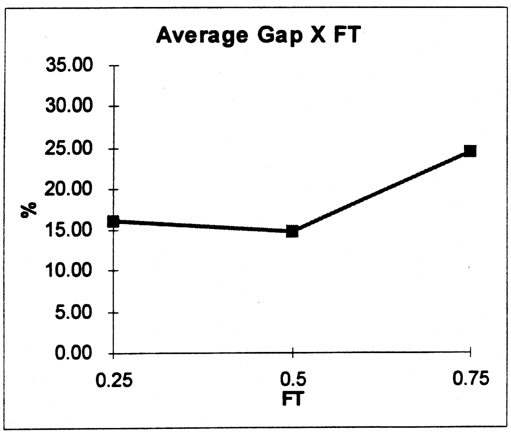
(3b)



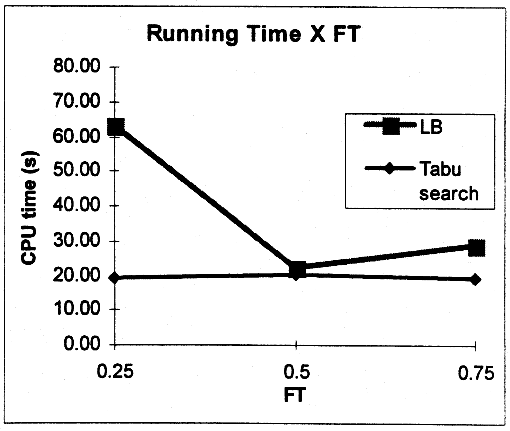
(3c)



(3d)



(3e)



(3f)

Fig. 3. Performance of the methodology.

The results we obtained for the quality of the solutions versus the number of networks and the amount of traffic are coincident with those obtained by Fetterolf and Anandalingam (1992a).

5.1. Comparison with other methodologies

It is not a trivial task to compare the computational efficiency of the algorithms developed in the preceding sections with those available in the literature, since different authors addressed different problems using different computers, operating systems, programming languages and compilers. They also used non-standard data sets. In spite of this, the results reported in this paper, average gaps in the range 0,09–48%, with mean of 18,29%, and running times: average of 58 s and maximum of 300 s, may be considered comparable to those corresponding to other methodologies available in the literature, for problems with similar degrees of complexity.

For the problem of interconnecting networks using transceptors (bridges with 2 ports) there are some results reported:

- Ferreira Filho and Galvão (1995), using Lagrangean relaxation and exchange heuristics report an average gap of 16.44% for the interconnection of up to 21 networks.
- Ersoy and Panwar (1993), using simulated annealing and lower bounds given by the optimal distribution flow between networks, obtained gaps in the range 18,2–61,6%, with running times of up to 181 min. in a PC-AT (124 s on a CONVEX 120 minicomputer), for the interconnection of up to 30 networks.
- Fetterolf and Anandalingan (1992a), using a Lagrangean heuristic, report an average gap between 0 and 25%, with running times of up to 58 min. on a Sun 360 workstation, for interconnecting up to 20 networks.

Computational results related to similar network design problems (under different nomenclatures) are reported by:

- Liang and Yee (1994) consider the problem of determining which gateways to use to interconnect wide area networks (WANs). They used Lagrangean relaxation and a heuristic based on the Frank and Wolfe method. The results reported, for networks of up to 47 nodes, show an average gap of 10% and running times of 1 h on a Sun workstation.
- Kim and Tcha (1992) study hierarchical two-level networks where the backbone network is of the tree type and the local access networks are of the star type. A dual-based lower bounding procedure was incorporated into a branch-and-bound algorithm. Computational experiments were conducted on a variety of problems going up to 50 backbone nodes and 200 demand points. The dual heuristics developed obtained feasible solutions within 17% of the corresponding optimal solutions.
- Fetterolf and Anandalingan (1992b) address the problem of designing LAN-WAN computer networks with transparent bridges. A simulated annealing heuristic was developed for this problem, with some computational results reported. The CPU time for 30 sites was 32.7 min on a Sun 360 workstation.
- Gavish (1992b) addresses the topological design of computer networks. Lower bounds are obtained by Lagrangean relaxation while feasible solutions are obtained by partial enumeration and exclusion of nodes heuristics. Computational tests involving up to 30 backbone nodes and 200 demand points were run on a IBM 3083, requiring from 20 to 40 min of CPU to generate lower bounds and feasible solutions. The gaps between lower bounds and feasible solutions are in the range 5–10%.

6. CONCLUSION

This paper addresses the interconnection of local area networks. This is a difficult problem and feasible solutions are usually obtained only through heuristics. A mathematical model for a simplified problem was presented and a Lagrangean relaxation and a tabu search heuristic developed for its solution. Computational results for a variety of problems with up to 10 interconnected networks are given. The average gap between feasible solutions and the corresponding lower bounds was 18.29%. These values are compatible with results reported in the literature for related problems.

In relation to future research in this area, it is desirable to extend the model we developed to incorporate additional features which better approximate the model to a practical situation. Some of these features are:

- selection of bridge capacities: in the model developed in this paper the bridge capacities are predetermined. A more realistic model should consider the bridge allocation problem, that is, given a discrete set of bridge capacities, determine which capacity to assign to each bridge;
- use of routers and other types of interconnection devices which don't require the topology to be a tree;
- consideration of additional constraints such as maximum hops in a route and reliability;
- modeling of ultra high speed networks and communication devices.

REFERENCES

Backes, F. (1988) Special issue on LANs interconnection. *IEEE Network,* Vol. 2, No. 1.

Beasley, J. E. (1989) An SST-based algorithm for the Steiner problem in graphs. *Networks* **19,** 1–16.

Bertsekas, D. and Gallager, R. (1987) *Data Networks.* New Jersey: Prentice-Hall.

Boorstyn, R. and Frank, H. (1977) Large scale network topological optimization. *IEEE Transactions on Communications* **25,** 29–47.

Bux, W., Grillo, D. and Maxemchuk, N. F. (1987) Special issue on LANs interconnection. *IEEE Journal on Selected Areas in Communications,* Vol. 5, No. 12.

Clarke, L. W. and Anandalingam, G. (1990) Optimal design of survivable LAN-LAN networks. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics,* pp. 293-297.

Ersoy, C. and Panwar, S. (1993) Topological design of interconnected LAN/MAN networks. *IEEE Journal on Selected Areas in Communications* **11,** 1172–1182.

Ferreira Filho, V. J. M. (1995) *Projeto Topologico de Redes de Computadores (Topological Design of Computer Networks).* Ph.D Thesis, Post-Graduate School of Engineering (COPPE), Federal University of Rio de Janeiro.

Ferreira Filho, V. J. M. and Galvão, R. D. (1994) A survey of computer network design problems. *Investigación Operativa* **4,** 183–211.

Ferreira Filho, V. J. M. and Galvão, R. D. (1995) Interconnection of local area networks with an application in the Federal University of Rio de Janeiro. *Pesquisa Operacional* **14,** 37–72.

Fetterolf, P. C. (1990) Design of data networks with spanning tree bridges. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics,* pp. 298–300.

Fetterolf, P. C. and Anandalingan, G. (1991) Optimizing interconnections of local area networks: an approach using simulated annealing. *ORSA Journal on Computing* **3,** 275–287.

Fetterolf, P. C. and Anandalingan, G. (1992a) Optimal design of LAN-WAN internetworks: an approach using simulated annealing. *Annals of Operations Research* **36,** 275–298.

Fetterolf, P. C. and Anandalingan, G. (1992b) A Lagrangean relaxation technique for optimizing interconnection of local area networks. *Operations Research* **40,** 678–688.

Fisher, M. L. (1981) The Lagrangean relaxation method for solving integer programming problems. *Management Science* **27,** 1–18.

Garey, M. R. and Johnson, D. S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness.* New York: Freeman.

Gavish, B. (1991) Topological design of computer communication networks — Local access design methods. *Annals of Operations Research* **33,** 17–71.

Gavish, B. (1992a) Operations research in telecommunications. *Annals of Operations Research* **36,** 1–402.

Gavish, B. (1992b) Topological design of computer communication networks — the overall design problem. *European Journal of Operational Research* **58,** 149–172.

Gavish, B. (1995) Special issue on telecommunications systems: modeling, analysis and design. *Operations Research* **43,** 1–193.

Geoffrion, A. M. (1974) Lagrangean relaxation for integer programming. *Mathematical Programming Study* **2,** 82–114.

Gerla, M. and Kleinrock, L. (1977) On the topological design of distributed computer networks. *IEEE Transactions on Communications* **25,** 48–60.

Glover, F. (1986) Future paths for integer programming and links with artificial intelligence. *Computers and Operations Research* **13,** 533–549.

Glover, F., Laguna, M., Taillard, E. and de Werra, D. (1993) Tabu Search. *Annals of Operations Research* **41,** 1–492.

Glover, F. and Laguna, M. (1993) Tabu Search. In *Modern Heuristic Techniquesfor Combinatorial Problems,* ed. C. R. Reeves, pp. 70-150. Oxford: Blackwell Scientific.

Green, P. E., Naemura, K. and Willianson, R. (1990) Special issue on heterogeneous computer networks interconnection. *IEEE Journal on Selected Areas in Communications,* Vol. 8, No. 1.

Gupta, S. and Ross, K. W. (1991) Performance modeling and optimization of networks of bridged LANs. *Queueing Systems* **9,** 113–132.

Held, M., Wolfe, P. and Crowder, H. (1974) Validation of subgradient optimization. *Mathematical Programming* **6,** 62–88.

Kim, J. and Tcha, D. (1992) Optimal design of a two-level hierarchical network with tree-star configuration. *Computers and Industrial Engineering* **22,** 273–281.

Kleinrock, L. (1964) *Communication Nets: Stochastic Message Flow and Delay.* New York: McGraw-Hill.

Liang, S. C. and Yee, J. R. (1994) Locating internet gateways to minimize nonlinear congestion costs. *IEEE Transactions on Communications* **42,** 2740–2750.

Perlman, R. (1984) An algorithm for distributed computation of a spanning tree. *Proceedings of the 9th Data Communications Symposium,* Montreal.

Poljack, B. T. (1967) A general method for solving extremum problems. *Soviet Mathematics* **8,** 593–597.