Topological Design of Interconnected LAN/MAN Networks

Cem Ersoy, *Member, IEEE,* and Shivendra S. Panwar, *Member, IEEE*

***Abstract*—This paper describes a methodology for designing interconnected LAN/MAN networks with the objective of minimizing the average network delay. We first consider IEEE 802 standard LAN's interconnected by transparent bridges. These bridges are required to form a spanning tree topology. We propose a simulated annealing-based algorithm for designing minimum delay spanning tree topologies. In order to measure the quality of the solutions, we find a lower bound for the average network delay. We extend the algorithm to design the overall LAN/MAN topology consisting of a MAN or high-speed data service interconnecting several clusters of bridged LAN's. Comparison with the lower bound and several other goodness measures show that the solutions are not very far from the global minimum.**

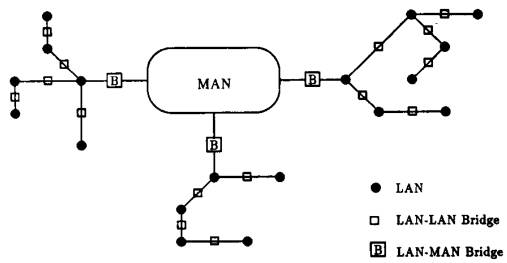


Fig. 1. An example LAN/MAN topology.

I. Introduction

LOCAL area networks (LAN's) have become an indispensable part of the modern working environment. Widespread applications using LAN's and a continuously growing number of users have created a necessity for the interconnection of LAN's. Bridges, routers, and gateways are different interconnection devices used to overcome the geographical and capacity limitations of LAN's. Among these, bridges are the simplest since they operate at the two lowest layers of the Open Systems Interconnection (OSI) model, namely the link and physical layers. Different types of bridges are used for interconnecting various types of LAN's. We consider transparent bridges which are widely used for interconnecting CSMA/CD LAN's and other IEEE 802 standard LAN's [1]. The major advantage of transparent bridges is that they do not require the participation of end users in order to interconnect LAN's. These bridges are self-learning and self-configuring. They learn and store the location of end users by observing source addresses of data packets. They route packets by comparing the destination addresses to the table of learned addresses. In order to operate properly, they require the logical topology of bridges and LAN's to be a spanning tree [2]. Given an arbitrary physical topology, these bridges configure themselves to be part of a deterministic active spanning tree topology. One of the main disadvantages of this is that spanning tree bridges cannot be used to allow all source destination pairs to use the best path that exists between them in the physical topology. However, some degree of flexibility is available since network management can be used to force the shape of this deterministic logical topology to be any spanning tree. The performance of bridged LAN's has been studied in [3]–[5]. The problem of determining which gateways to use to interconnect existing data networks has been discussed in [6]. A simulated annealing technique has been used for designing minimum-cost interconnected CSMA/CD LAN's in [7]. The design of optimally locating bridges and repeaters for minimizing the average delay and fast algorithms for special cases have been studied in [8].

There is a growing interest in the interconnection of geographically separated LAN's or bridged LAN clusters by a backbone metropolitan area network (MAN). Fig. 1 shows an example interconnected LAN/MAN network with several clusters. Each cluster consists of several LAN's interconnected by transparent bridges. The major reason for forming these clusters is the geographical separation of LAN's. There may also be network management or security reasons for keeping several LAN's in a cluster. The overall topology has to be a spanning tree since this is required by the bridges. Depending on the physical distances involved and the capacity required, different backbone network architectures can be used for interconnecting clusters of bridged LAN's. Fiber Distributed Data Interface (FDDI), which is a 100 Mb/s network based on a token ring protocol, has already been used for the interconnection of LAN clusters [9]. There are other promising candidates such as backbone MAN's interconnecting LAN clusters. One of them is the IEEE 802.6 MAN standard: Dual Queue Dual Bus (DQDB), which has typical data rates ranging from 50 to 150 Mb/s [10].

There are also proposals of broadband services such as Switched Multimegabit Data Service (SMDS) for intercon-

Manuscript received June 1992; revised December 1992. This work was supported by the New York State Science and Technology Foundation's Center for Advanced Technology in Telecommunications, Polytechnic University, Brooklyn, NY and by the NSF under Grants NOR-8909719, NCR-9003006, and NCR-9115864. This paper was presented in part at the IEEE INFOCOM'92, Florence, Italy, May 6–8, 1992.

C. Ersoy is now with the Department of Computer Engineering, Boğaziçi University, Istanbul, Turkey.

S. S. Panwar is with the Department of Electrical Engineering, Polytechnic University, Brooklyn, NY.

IEEE Log Number 9211669.

0733–8716/93$03.00 © 1993 IEEE

necting LAN clusters using MAN technology [11]. SMDS is a very high-speed connectionless packet-switched service which is to be offered by the local exchange carriers in the early 1990's. This service is suitable for bridging between IEEE 802 LAN's since it can handle variable size packets of up to 8192 octets. SMDS will provide services with DS1 (1.5 Mb/s) and DS3 (45 Mb/s) rates. The DS3 rate will have access class rates of 4, 10, 16, and 34 Mb/s. The SMDS Interface Protocol is aligned with the IEEE 802.6 MAN standard in order to be suitable for LAN interconnection, since this is expected to be an important application of SMDS.

The problems related to using B-ISDN (Broadband Integrated Services Digital Network) for the interconnection of LAN's have been addressed by Mongiovi *et al.* in [12]. Since the proposed B-ISDN standard, Asynchronous Transfer Mode (ATM) [13], is connection oriented and divides the information into fixed-size cells, it is not directly compatible with the IEEE 802 LAN protocols. ISDN Frame Relay Service has also been suggested for LAN interconnection [14].

In this paper, we address the problem of overall LAN/MAN topology design so that the average network delay is minimized and the maximum end-to-end delay of the overall topology is below a threshold. Topological design of the overall LAN/MAN topology as a whole is very complex and difficult since the total number of LAN's is large and many constraints are involved. In order to reduce the complexity, we define and formulate the problem so that it can be decomposed into smaller problems for each cluster. We show that subproblems for each cluster can be solved independently. After the decomposition, each subproblem involves only the LAN's in that cluster and the backbone MAN. Section II covers the design of individual cluster topologies. In Section II-A, the minimum-delay spanning tree problem for the interconnection of LAN's in a cluster is defined and formulated as a combinatorial optimization problem. There is no efficient exact algorithm for solving this problem. We propose a heuristic algorithm based on simulated annealing for finding solutions to the minimum-delay spanning tree problem in Section II-D. The spanning tree design can then be implemented either by connecting the bridges to the LAN's appropriately or by adjusting network management parameters if the existing physical topology can be pruned to the desired tree. In order to check the quality of solutions found by the simulated annealing algorithm, a lower bound for the problem is found in Section II-C. The algorithm is extended for the design of overall LAN/MAN topologies in Section III. Results of the computational experiments given in Section IV show that by decomposing the problem and using simulated annealing, low-delay LAN/MAN topologies for fairly large networks can be found in reasonable running times. The delays of these topologies are lower than those found by other feasible algorithms. Section V concludes the paper.

II. Design of Minimum Delay Interconnected LAN's

In this section, we will describe the topological design problem for bridged LAN's which belong to the same cluster. The LAN's are located geographically close to each other and interconnected by local transparent bridges. In order to avoid looping of packets, these bridges require the active topology to be a spanning tree [2]. We will describe how individual cluster topologies can be combined to form an overall LAN/MAN topology in Section III.

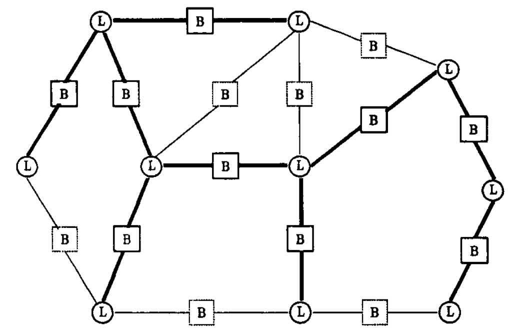


Fig. 2. An example of a bridged LAN network.

A. Definition and Formulation of the Problem

An example physical topology for a group of bridged LAN's is shown in Fig. 2. One of the possible active spanning trees is shown with bold lines. During normal operation, only the bridges on the active tree forward packets. Other bridges remain idle unless activated to form a new tree in case of failures. Since the logical spanning tree topology is independent of the underlying physical topology, real costs do not play an important role in the logical topology design problem. Hence, performance measures can be used as the objective function for finding the optimum logical topology. We address the problem of designing minimum-delay spanning tree topologies for the interconnection of LAN's. In this problem, N LAN's, local traffic on each LAN (t_{ii}), and the traffic requirements between all source and destination LAN pairs (t_{sd}) are given. Our goal is finding the spanning tree topology with the minimum average network delay for the given set of requirements. In other words, we have to decide where to place $(N-1)$ active bridges among all possible LAN pairs such that they form a spanning tree with the minimum delay. In a given network, if the underlying physical topology can be pruned to the minimum-delay spanning tree, bridges can be set up to choose this tree as the active logical topology by adjusting the parameters of the self-configuration algorithm [15].

As in Fig. 2, LAN's are represented as nodes and bridges as edges of a graph. Since many existing bridges have two ports and we are interested in the shape of the minimum-delay topology, this representation is sufficient for our purposes. With slight modification, multiport bridges can be represented as several edges and a central node. In that case, edges of the graph will correspond to bridge ports rather than whole bridges. In this way, both two-port and multiport bridges can be used for implementing minimum-delay spanning tree topologies.

The problem of finding the minimum-delay spanning tree topology is formulated as described:

Problem P:

$$\underset{\{x_{ij}\} \in S}{\text{minimize}}\ T = \frac{1}{\gamma} \cdot \left(\sum_{i=1}^{N} f_i E[T_i] + \sum_{i=1}^{N} \sum_{j=1}^{N} \cdot x_{ij} f_{ij} E[T_{ij}] \right) \quad (1)$$

subject to

$$f_i < \mu_i \qquad \forall LAN\, i \quad (2)$$

$$f_{ij} < x_{ij}\mu_{ij} \qquad \forall Bridge_{ij} \quad (3)$$

where

$$\text{decision variables}\, x_{ij} = \begin{cases} 1, & \text{if there is a bridge between LAN}\, i\, \text{and LAN}\, j \\ 0, & \text{otherwise.} \end{cases}$$

S = set of all possible spanning trees.

The objective function is the average network delay (T) which is a combination of the queueing delays due to LAN's ($E[T_i]$) and bridges ($E[T_{ij}]$) as described in [16]. $E[T_i]$ is a function of the total flow on LAN i (f_i) and $E[T_{ij}]$ is a function of the total flow on the bridge between LAN i and LAN j (f_{ij}). γ is the total input flow to the network. We do not consider propagation delays since LAN's are geographically close to each other. A more detailed description of the delay model will be given in Section II-B. μ_i and μ_{ij} are the LAN and bridge capacities, respectively. Inequalities (2) and (3) are the corresponding LAN and bridge capacity constraints. They are also implied in the objective function, since the delay (T) becomes infinity if the flow values exceed the capacities. In order to keep the problem simple, we do not consider additional constraints. If needed, other constraints such as communication costs or maximum utilization for LAN's and bridges can easily be incorporated into the formulation.

Problem P is a difficult combinatorial optimization problem. In fact, a simplified version of this problem, namely the capacitated spanning tree problem, which handles queueing delay implicitly, is NP-complete [17]. There is no efficient exact algorithm for solving Problem P. The number of possible spanning trees for N LAN's is equal to N^{N-2} [18]. This makes it impractical to determine the minimum-delay topology by exhaustive search except for very small size problems. Our experiments showed that the solution space has many local minima and greedy local search heuristics get stuck in a local minimum. In order to avoid this problem, we based our search heuristic on simulated annealing since this technique works well with problems with many local minima [19]. For small problems, we found the global minimum with complete enumeration. In order to check the quality of the solutions for larger problems, we derived a lower bound for Problem P which is described in Section II-C.

B. Delay Model

We find the average network delay for each topology generated at every iteration of the simulated annealing algorithm. Because of the large number of topologies generated, we need an efficient way of approximating the average network delay; preferably, a closed-form delay model. We consider delays due to LAN's and bridges. The topology is modeled as a network of queues. Different measurement and performance studies have shown that LAN traffic is bursty and consists of batches or trains of packets because of the existing protocols [20], [21]. In order to account for the burstiness of traffic and to have a computationally inexpensive measure, we use $M^X/M/1$ queues with batch Poisson arrivals. Although it is simple, $M^X/M/1$ queueing model may achieve an acceptable fit for the busiest periods of the network [21] and give us the relative performance of different topologies. Other closed-form delay models may also be used in the algorithm. However, more complex delay models will increase the running times.

We are given the mean rate for traffic requirements between each LAN pair (t_{sd}) in terms of batches per second. For a given spanning tree topology, we can find the mean batch flow values of LAN's and bridges λ_i and λ_{ij}, respectively. We know the capacities of LAN's, C_i (bits/second). We assume exponential distribution for packet lengths with mean l and a geometrical distribution for the number of packets in a batch with mean X. The mean service rate for LAN's is equal to the LAN capacity divided by the mean packet length ($\mu_i = C_i/l$ packets/second). The mean service rate for each direction of the bridges is equal to μ_{ij} (packets/second). The expected number of packets ($E[L]$) in each queue is found using the results in [22, pp. 156–160]. Using Little's formula ($E[T] = E[L]/\lambda X$), the expected value of the delay ($E[T]$) for each $M^X/M/1$ queue is given as

$$E[T] = \frac{\rho}{\lambda(1-\rho)} \qquad \text{where}\, \rho = \frac{\lambda X}{\mu}. \quad (4)$$

After finding the delays caused by LAN's and bridges using (4), the average network delay (T) is found by plugging these delay values, total flow values in terms of packets per second ($f = \lambda X$), and the total input flow of the network ($\gamma = X\, \Sigma_{s=1}^{N} \Sigma_{d=1}^{N} t_{sd}$) into the objective function of (1).

C. A Lower Bound on the Average Network Delay

The average network delay is a combination of delays due to LAN's and bridges T_L and T_B, respectively:

$$T = \underbrace{\frac{1}{\gamma}\sum_{i=1}^{N}\lambda_i X E[T_i]}_{T_L} + \underbrace{\frac{1}{\gamma}\sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij}\lambda_{ij} X E[T_{ij}]}_{T_B}. \quad (5)$$

As shown in (5), the sum of the lower bounds on the delay due to bridges and LAN's will give us the lower bound on T. In order to find a bound on the delay due to bridges (T_B), we first find a lower bound on the total flow on all bridges (λ_B^T). Then, we find the optimum distribution of this total flow among all bridges such that T_B is minimized. Similarly, we find a lower bound on the total flow on all LAN's (λ_L^T). The optimum distribution of this flow among all LAN's such that T_L is minimized will give us the lower bound on T_L. The lower bound on T obtained by summing the lower bounds on T_L and T_B enables us to check the quality of the solutions found by simulated annealing and other techniques.

1) A Lower Bound on the Total Flow on Bridges: The total flow on all bridges (λ_B^T) is equal to the sum of all bridge flows. The spanning tree topology with the minimum λ_B^T gives us a lower bound.

Definition 1: Optimum requirement spanning tree problem [23]: In this problem, the cost of communication for a spanning tree is equal to the summation of all the products of a traffic requirement (t_{sd}) times the number of hops on the path corresponding to that requirement (h_{sd}) (i.e., $\Sigma_s \Sigma_d t_{sd} \cdot h_{sd}$). The spanning tree which minimizes this cost of communication is called the *optimum requirement spanning tree* and can be found with an $O(N^4)$ algorithm [23].

Theorem 1: The *optimum requirement spanning tree* defined by Hu in [23] has the minimum λ_B^T among all spanning trees.

Proof: The flow on the bridge between LAN i and LAN j ($Bridge_{ij}$) is equal to the sum of all requirements (t_{sd}) which have to pass through $Bridge_{ij}$ in order to reach their destination

$$\lambda_{ij} = \sum_s \sum_d t_{sd} \cdot y_{sd}^{ij} \tag{6}$$

where y_{sd}^{ij} is equal to one if $Bridge_{ij}$ is on $Path_{sd}$, zero otherwise. Therefore,

$$\lambda_B^T = \sum_{\forall Bridge_{ij}} \lambda_{ij} \tag{7}$$

$$= \sum_{\forall Bridge_{ij}} \sum_s \sum_d t_{sd} \cdot y_{sd}^{ij} \tag{8}$$

$$= \sum_s \sum_d t_{sd} \sum_{\forall Bridge_{ij}} y_{sd}^{ij} \tag{9}$$

and since for a given (s, d) pair the number of nonzero y_{sd}^{ij}'s is equal to the number of bridges on $Path_{sd}$, the number of times t_{sd} appears in the summation is equal to the number of hops in $Path_{sd}$. Thus, the total flow on bridges can be written as

$$\lambda_B^T = \sum_s \sum_d t_{sd} \cdot h_{sd} \tag{10}$$

which is also equal to the cost of communication for the *optimum requirement spanning tree* problem in [23]. Therefore, the cost of the solution to this problem gives the lower bound $\lambda_{B,min}^T$. □

2) A Lower Bound on the Total Flow on LAN's: Similarly, the total flow on all LAN's (λ_L^T) is the summation of flow on all LAN's. The flow on LAN $i(\lambda_i)$ can be found as

$$\lambda_i = t_{ii} + \sum_s \sum_d t_{sd} \cdot w_{sd}^i \tag{11}$$

where w_{sd}^i is equal to 1 if LAN i is on $Path_{sd}$, zero otherwise. Therefore,

$$\lambda_l^T = \sum_i \lambda_i \tag{12}$$

$$= \sum_i \left(t_{ii} + \sum_s \sum_d t_{sd} \cdot w_{sd}^i \right) \tag{13}$$

$$= \sum_s \sum_d t_{sd} \cdot (1 + h_{sd}) \tag{14}$$

$$= \lambda_B^T + \sum_s \sum_d t_{sd}. \tag{15}$$

Since requirement t_{sd} passes through all the LAN's on $Path_{sd}$ including LAN s and LAN d, the number of times it will appear in the summation is equal to one more than the number of hops on $Path_{sd}$. The local traffic t_{ii} affects only LAN i. All local traffic requirements appear in the summation once. The minimum value for the total flow on LAN's ($\lambda_{L,min}^T$) is equal to the summation of $\lambda_{B,min}^T$ and all t_{sd}'s, since only the terms involving λ_B^T can be minimized.

3) Optimum Distribution of Total Flows: After obtaining the lower bounds on the total flow on LAN's and bridges, the problem of finding the lower bounds on T_L and T_B reduces to finding the optimum distribution of these total flows so that T_L and T_B are minimized. In order to make the lower bounds tighter, we also consider some inequality constraints.

Any traffic requirement which has LAN i as its source or destination has to pass through LAN i. Therefore, the flow on LAN i has to be greater than or equal to the *mandatory* flow on that LAN (λ_i^M):

$$\lambda_i \geq \lambda_i^M = \sum_{k=1}^{N} (t_{ik} + t_{ki}) - t_{ii}, \qquad \forall i. \tag{16}$$

In other words, the flow on LAN i consists of the summation of its mandatory flow (λ_i^M) and the transit flow on LAN i (λ_i^{TR}):

$$\lambda_i = \lambda_i^M + \lambda_i^{TR}, \qquad \text{where}$$
$$\lambda_i^{TR} \geq 0 \qquad \forall i. \tag{17}$$

Finding a lower bound for T_L is equivalent to finding the optimum distribution of $\lambda_{L,min}^T$ which minimizes T_L subject to the inequality constraints in (16). This problem is similar to the capacity assignment problem described in [16], but here the capacities and the total flow are known and individual LAN flows are to be found. The optimum distribution of T_L is a standard optimization problem and, as described in Appendix A, it can be solved using Lagrangean techniques. In Appendix A, we also show by using the Kuhn–Tucker conditions [24] that if all LAN capacities are equal, the water-filling type distribution shown in Fig. 3 minimizes T_L. In this distribution,

$$\lambda_i^{TR} = (\lambda_L' - \lambda_i^M)^+ \qquad \forall i \tag{18}$$

where λ_L' is chosen so that

$$\sum_{i=1}^{N} (\lambda_L' - \lambda_i^M)^+ = \lambda_{L,min}^T - \sum_{i=1}^{N} \lambda_i^M \tag{19}$$

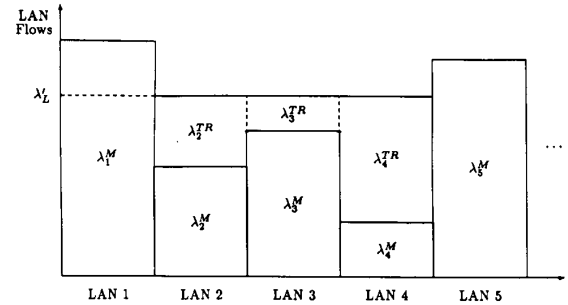


Fig. 3. Distribution of LAN flows.

and $(\xi)^+$ denotes the positive part of ξ, i.e.,

$$(\xi)^+ = \begin{cases} \xi & \text{if } \xi \geq 0 \\ 0 & \text{if } \xi < 0 \end{cases}. \tag{20}$$

Similarly, as described in Appendix A, a lower bound for the delay due to bridges (T_B) can be obtained by optimally distributing $\lambda_{B,min}^T$ on $2(N-1)$ bridge ports so that T_B is minimized. We also show that if all bridges have equal capacity, a uniform distribution of $\lambda_{B,min}^T$ among all bridge ports gives a lower bound for T_B. This lower bound can be improved by considering additional inequality constraints similar to the ones described for LAN's. This time the mandatory traffic concept will be replaced by the $2(N-1)$ largest traffic requirements. Since each of these traffic requirements has to pass through at least one bridge, they impose inequality constraints on bridge flows. In order to minimize T_B, each of these requirements is assigned to a separate bridge port. As in the case of LAN's, it can be verified using the Kuhn–Tucker conditions that a water-filling type distribution similar to the one in Fig. 3 is optimum and gives a lower bound for T_B. However, unlike the LAN case, if the bridge port capacities are not equal, these inequality constraints cannot be used for improving the lower bound on T_B. After finding the lower bound for T_L and T_B, the sum of these terms gives the lower bound for the average network delay T.

D. Simulated Annealing Algorithm

Simulated annealing is a local neighborhood search heuristic technique [25]. Two basic disadvantages of ordinary local search algorithms are that they may get stuck in local minima because they accept only cost improving solutions and that the quality of the final result heavily depends on the initial solution. In contrast, simulated annealing algorithms occasionally accept deteriorations in cost in a controlled manner besides accepting improvements in cost. This property enables them to escape from local minima while keeping the favorable features of local search algorithms, i.e., simplicity and general applicability.

Fig. 4 shows the flowchart of the annealing algorithm used for finding minimum-delay topologies. In general, simulated annealing algorithms are defined by a neighborhood structure and a cooling schedule. The *neighborhood structure* for our problem is defined as follows: Any two spanning tree topologies which have all the branches except one common are called neighbor trees. Given a spanning tree, we can create a neighbor tree by removing a branch, resulting in two separate subtrees. Adding another branch which will connect the two subtrees, but will not create a loop. During the search for a minimum-delay topology, we create a sequence of neighbor trees as described in the flowchart of Fig. 4. We check the feasibility of each topology by comparing the flows of LAN's and bridge ports with their capacities. If the topology is feasible, we find the average network delay using (1). During the local neighborhood search for a minimum-delay spanning tree topology, if the transition from the current topology to a new topology is a delay-decreasing one, it is always accepted; if the transition is a delay-increasing one, the new topology is accepted as the current solution with a certain probability, p, which is given by

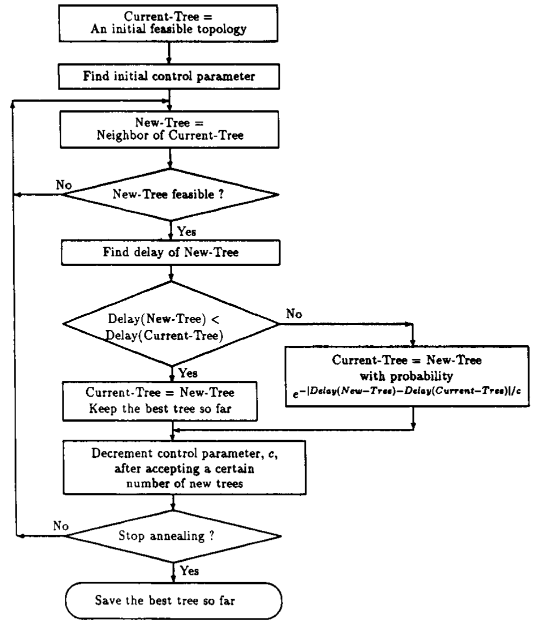


Fig. 4. The flowchart of the simulated annealing algorithm.

$$p = \exp\left(\frac{-[Delay(\text{New Tree}) - Delay(\text{Current Tree})]}{c}\right)$$
$$Delay(\text{New Tree}) > Delay(\text{Current Tree}). \tag{21}$$

Here, c is the control parameter that regulates the probability of accepting a delay-increasing transition. At the beginning, a large value for the control parameter is chosen, resulting in the acceptance of most of the transitions. During the search, we slowly reduce the control parameter towards zero according to the cooling schedule. Lower control parameter values make the acceptance of cost-increasing transitions less probable. At any point during the run, we always keep the best spanning tree generated up to that point.

It has been shown in [26] that the simulated annealing algorithm finds the global optimum. Unfortunately, this implementation requires an infinite number of transitions. A finite-time simulated annealing algorithm for finding high-quality solutions can be implemented with a suitable cooling schedule.

A *cooling schedule* consists of choosing an initial value c_o for the control parameter c the method for decrementing the control parameter, a finite number of transitions at each value of the control parameter, and the stopping criterion. Different cooling schedules have been proposed in [19], [25], and [26]. We experimented with different cooling schedules and chose one similar to the one described in [19], because of its simplicity and efficiency. As already mentioned, the initial value of the control parameter c_o is chosen so that almost all new topologies are accepted at the beginning. The function used for decrementing the control parameter is given by

$$c_{k+1} = \alpha \cdot c_k \qquad k = 0, 1, 2, \cdots \\ 0.75 \leq \alpha \leq 0.99. \tag{22}$$

The control parameter is decreased after acceptance of a fixed number of new topologies. However, since transitions are accepted with decreasing probability, the number of topologies examined at each c_k will increase as c_k goes to zero. In order to avoid extremely long iterations at small values of c_k, the total number of topologies examined at each c_k are bounded by a fixed maximum value. This value is comparable to the size of the neighborhood. Simulated annealing is terminated if the value of the delay does not change after decrementing the control parameter a fixed number of times. This fixed number is chosen such that the algorithm has a sufficiently large probability of visiting at least a major part of the neighborhood of a given solution. In order to guarantee that we are not missing any good solutions in the neighborhood of the annealing solution, we terminate the algorithm by comparing the annealing solution with all of its neighbor topologies.

III. Design of Interconnected LAN/MAN Networks

In this section, we will extend the simulated annealing algorithm to the design of minimum-delay LAN/MAN networks. There are many issues to be resolved in the area of LAN interconnection over MAN's, such as packet delimiting and encapsulation, different protocols, and out-of-sequence packets due to multiple routes [11]. We will not consider these problems here, since we are only interested in the overall minimum delay topology of the LAN's interconnected over a MAN. Fig. 1 shows an example interconnected LAN/MAN network with several clusters. We know the traffic requirements between all LAN pairs and the clustering information (i.e., which LAN's are in each cluster). We want to find a minimum-delay LAN/MAN topology. The overall topology has to be a spanning tree because of the transparent bridges. We assume that we have only one MAN. This MAN can be FDDI, IEEE 802.6 DQDB, or instead of the MAN we can have a connectionless packet switched data service (e.g., SMDS) as a backbone network. We assume that the backbone MAN has a large capacity and the intercluster traffic will not have significant impact on the performance of the MAN. The backbone is modeled as a central node. The bridges between LAN clusters and the MAN are specialized remote bridges. An essential characteristic of remote bridging is that no station may connect directly to the interconnecting medium, in this case the MAN. At least two remote bridges must appear in

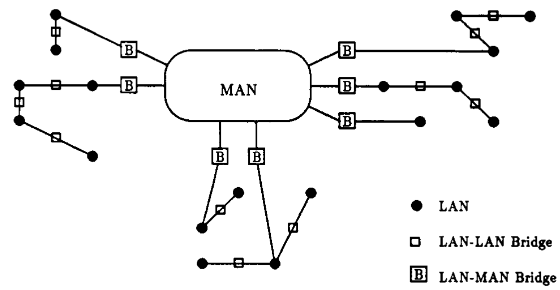


Fig. 5. An example topology with more than one LAN/MAN bridge per cluster.

all communication paths between stations [11]. LAN/MAN bridges usually have higher capacities than ordinary bridges in order to support the cluster traffic.

One approach to find a minimum delay overall topology is solving the problem as if it were a LAN/LAN problem with many nodes and adding an additional constraint, forcing LAN's in different clusters to be interconnected through the MAN. Increasing the number of LAN's increases the running time of the simulated annealing algorithm. We will instead use the approach of decomposing the problem into smaller problems for each cluster and, hence, reduce the running time. In the first phase, we assume that only one LAN in each cluster is directly connected to the MAN. In other words, there is only one LAN/MAN bridge per cluster. This is a reasonable assumption because LAN/MAN bridges are currently five times more expensive than ordinary bridges. Furthermore, in the case of using more than one LAN/MAN bridge, some of the intracluster traffic has to pass through the MAN as shown in Fig. 5 because of the spanning tree requirement. This may not be desirable for security reasons.

The problem is suitable for decomposition because the intracluster traffic does not affect other clusters. Therefore, the topology of each cluster can be determined by considering only the intracluster traffic and the traffic between the MAN and that cluster. At the beginning, we are given an overall traffic requirement matrix, $[t_{sd}]_{N \times N}$, for the whole network. We can calculate smaller traffic requirement matrices $[t_{sd}^k]_{(N_k+1)\times(N_k+1)}$, where N_k is the number of LAN's in cluster k. For example, if the original traffic requirements for the topology in Fig. 1 are given by

$$[t_{s,d}] = \left[\begin{array}{l|l|l} t_{1,1}\cdots t_{1,6} & t_{1,7}\cdots t_{1,11} & t_{1,12}\cdots t_{1,18} \\ \vdots \quad \ddots & \vdots \quad \ddots & \vdots \quad \ddots \\ t_{6,1} & & \\ \hline t_{7,1}\cdots & t_{7,7}\cdots & t_{7,12}\cdots \\ \vdots \quad \ddots & \vdots \quad \ddots & \vdots \quad \ddots \\ t_{11,1} & & \\ \hline t_{12,1}\cdots & t_{12,7}\cdots & t_{12,12}\cdots \\ \vdots \quad \ddots & \vdots \quad \ddots & \vdots \quad \ddots \\ t_{18,1} & & \end{array}\right]$$

we can calculate the traffic matrix $[t^1_{s,d}]$ for cluster 1 as follows:

$$[t^1_{s,d}] = \left[\begin{array}{ccc|c} t^1_{1,1} & \cdots & t^1_{1,6} & t^1_{1,\mathrm{MAN}} \\ \vdots & \ddots & & \vdots \\ t^1_{6,1} & & & t^1_{6,\mathrm{MAN}} \\ \hline t^1_{\mathrm{MAN},1} & \cdots & t^1_{\mathrm{MAN},6} & 0 \end{array}\right]$$

where

$$t^1_{i,\mathrm{MAN}} = \sum_{l=7}^{18} t_{i,l}, \forall i$$
$$t^1_{\mathrm{MAN},i} = \sum_{l=7}^{18} t_{l,i}, \forall i. \tag{23}$$

After finding the new traffic requirement matrices for each cluster, we can solve each subproblem independently. At the beginning, we allow only one LAN/MAN bridge per cluster. We use the simulated annealing algorithm as in the LAN/LAN problem with the additional constraint that each cluster will be connected to the MAN node with only one bridge. After finding the overall topology, we calculate the *maximum cluster access delay* T^k_{CA} for each cluster. T^k_{CA} is the maximum delay between any LAN in cluster k and the MAN in either direction. The maximum end-to-end delay in the overall network will always be bounded by the summation of the two largest T^k_{CA} and the delay due to the MAN (T_{MAN}). We assumed that the intercluster traffic will not have a significant impact on T_{MAN} because of the large capacity of the MAN. For example, in the case of SMDS as a backbone MAN, it is not very likely that one group of users will change the performance of the service because the network management system will try to maintain a guaranteed level of performance. T_{MAN} is, therefore, assumed to have a fixed value for our problem. The *LAN access delay* between LAN i and the MAN (T^i_{LA}) is equal to the larger of the total delay in either direction between LAN i and the MAN ($T_{i,\mathrm{MAN}}$ and $T_{\mathrm{MAN},i}$). The delay from LAN i to the MAN is equal to the summation of the delays of all LAN's (p) and bridges (r, u) lying on the path from LAN i to the MAN, $Path(i, \mathrm{MAN})$, and is given

$$T_{i,\mathrm{MAN}} = \sum_{p \in Path(i,\mathrm{MAN})} E[T_p] + \sum_{(r,u) \in Path(i,\mathrm{MAN})} E[T_{ru}] \quad \forall i \tag{24}$$

and $T_{\mathrm{MAN},i}$ is equal to the summation of the delays in the other direction. Therefore, the *LAN access delay* and the *cluster access delay* are given by

$$T^i_{LA} = \max(T_{i,\mathrm{MAN}}, T_{\mathrm{MAN},i}) \qquad \forall i, \tag{25}$$
$$T^k_{CA} = \max_{i \in cluster\ k} T^i_{LA} \qquad \forall k \tag{26}$$

If T^k_{CA} exceeds a threshold for a cluster, we proceed to the second phase: We allow that cluster to have two LAN/MAN bridges. Because of the spanning tree requirement, that cluster will have two subtrees as shown in Fig. 5. Since this is likely to reduce the depth of the tree, the new T^k_{CA} will be lower. As we have explained before, we can change the topology of one cluster without affecting the other clusters. We find the minimum-delay topology for that cluster as in the first phase but, this time, the maximum number of LAN/MAN bridges is constrained to two. If T^k_{CA} drops below the threshold, we stop. Otherwise, we allow the cluster to have one more LAN/MAN bridge. We continue increasing the number of LAN/MAN bridges until T^k_{CA} drops below the threshold. We repeat this procedure for all clusters. As a result, some of the clusters might have one LAN/MAN bridge, and some of them might have more depending on their traffic requirements.

Another approach could be using additional cost constraints to determine the number of LAN/MAN bridges, but this would increase the complexity of the problem and currently high LAN/MAN bridge prices (approximately five times more expensive than LAN/LAN bridges) would force us to use as few bridges as possible per cluster. In many practical-size problems, one LAN/MAN bridge per cluster is sufficient and the overall minimum-delay topology is found in one phase.

A variation of the problem occurs in the case of using SMDS for interconnecting the clusters of LAN's, since SMDS will provide communications at DS1 (1.5 Mb/s) or DS3 (45 Mb/s) rates. During the design process, we have to decide which rate of service and, in the case of DS3, which access class (4, 10, 16, 34 Mb/s) will be used for each cluster. A possible approach can be described as follows: The total traffic between each cluster and the MAN is known. At first, we choose the lowest and cheapest rate among all SMDS access classes which is sufficient to carry the cluster-MAN traffic for each cluster. We find the minimum-delay cluster topologies and calculate T^k_{CA}. If it is acceptable, we stop; otherwise, if T^k_{CA} is large due to the delay on LAN/SMDS bridge, we choose the next higher SMDS access class for clusters with unacceptable T^k_{CA}. We can decide whether the SMDS connection is the bottleneck or not by using a threshold on the utilization of the SMDS line. We continue increasing the SMDS rates for each cluster until we are satisfied with the delay due to the LAN/SMDS bridge and T^k_{CA}. If the bottleneck is not the SMDS connection, increasing the SMDS access rates does not reduce T^k_{CA}. In that case, we can increase the number of LAN/SMDS bridges per cluster as explained before.

The algorithm for finding an overall LAN/MAN topology can be summarized as follows.

- Find individual traffic matrices for each cluster from the given overall traffic matrix.
- The number of LAN/MAN bridges is initially equal to one for all clusters.
- For all clusters,

1. Find individual minimum delay topology for the cluster.
2. Find *Cluster Access Delay.*
3. If *Cluster Access Delay* exceeds its threshold, add one more LAN/MAN bridge to the cluster and go to 1; otherwise save cluster topology and repeat 1, 2, 3 for the next cluster.

IV. Results and Discussion

A. Methodology for the Experiments

We performed experiments on networks with 6, 7, 10, 15, 20, and 30 LAN's. For 6 and 7 LAN problems, we enumerated

TABLE I
RESULTS FOR THE MINIMUM DELAY INTERCONNECTED LAN PROBLEM

Number of LAN'S	Annealing		$T_{Lower\ Bound}$	Local Search		10,000 Samples		CPU Time PC-AT/CONVEX
	$T_{\min}$	$T_{\max}$		$T_{\min}$	$T_{\max}$	$T_{\min}$	T_{mean}	
6	6.406*	6.406*	5.419	6.406*	7.214	—	—	8 s/< 1 s
7	7.524*	7.746	5.721	7.524*	9.549	—	—	18 s/< 1 s
10	8.323	8.524	6.199	8.639	11.027	9.103	11.445	52 s/< 1 s
15	10.470	10.696	7.168	11.117	14.836	12.419	15.993	6.2 min/4.3 s
20	13.910	14.370	8.908	14.832	19.235	16.655	21.937	25.8 min/18 s
30	17.233	18.112	10.662	19.374	24.583	22.166	26.263	181 min/124 s

* Global minimum in complete enumeration.

all possible spanning tree topologies and found the global minimum. This enabled us to compare the solution of the simulated annealing directly with the global minimum. For larger problems, we used several different goodness measures. In order to estimate the range of the simulated annealing results, we ran the algorithm with 10 different random seeds on the same problem. The difference between the best and the worst solution in 10 runs gave us the range for the annealing solutions. We compared the solutions with the lower bound described in Section II-C. The gap between the lower bound and the annealing solutions is an upper bound for the deviation from the global minimum. We also used the statistical goodness measure described in [7]. In this measure, 10,000 random feasible topologies were generated. A histogram corresponding to the delay values of these topologies was created. This was then compared with the simulated annealing solutions.

The simulated annealing algorithm is a local search heuristic. In order to see its advantages over ordinary local search, we implemented a conventional local search heuristic and ran it several times with different random initial topologies. We compared the best topology found by the local search algorithm with the simulated annealing results. In order to have a fair comparison, we adjusted the number of times the local search was run so that the combined running time of multiple runs of local search was approximately equal to that of the annealing algorithm.

The test problems had various traffic requirement patterns as described.

- Three of the traffic matrices consist of uniformly distributed random traffic requirements with different average values. These average values correspond to light, medium, and heavy loads. In order to consider the unbalanced inter-LAN traffic patterns, some rows and columns of the traffic matrices have larger average values than others. These rows or columns might correspond to LAN's connected to file servers or host computers.
- One of the traffic matrices is such that the traffic between any two LAN's decreases linearly with the "distance" between them. For example, the traffic requirements between LAN 1 and LAN 2 are larger than those between LAN 1 and LAN 5. All traffic requirements have deterministic values according to the "distance" measure.
- The last traffic matrix is uniform. All traffic requirements have the same deterministic value.

Other important parameters of the example problems are as follows: The mean packet length, l, is equal to 192 bytes. This value was measured on interconnected LAN's by Leland and Wilson [21]. The average number of packets in a batch, X, is 8. The capacity of LAN's is 10 Mb/s, which is the standard for CSMA/CD LAN's. The capacity of LAN/LAN bridges is 6,000 packets/second and the capacity of LAN/MAN bridges are 10,000 packets/second for 192-byte packets. Given these parameters, all the delay values presented in the following sections are in milliseconds. The threshold for T_{CA}^{k} in the LAN/MAN problem is 20 ms.

B. Results for the Interconnected LAN Problem

Table I summarizes the results of the annealing algorithm and the comparisons with the lower bound for the interconnected LAN problem in the case of heavy-load random traffic requirements. As an example for the heavy load, the requirement matrix for the 15 LAN problem consists of uniformly distributed random requirements with an average of 3 batches/second. Two rows, corresponding to LAN's with file servers, have higher requirements with an average 10 batches/second. Corresponding minimum delay topologies found by the algorithm are such that maximum utilization on LAN's and bridges reach 80% and 40%, respectively. The minimum and maximum delay values for the simulated annealing algorithm are found by running the algorithm with 10 different random initial topologies. Related columns for the best and worst performances of the algorithm show that the range of simulated annealing solutions is small. This small range of results confirms that the quality of the final solution is not dependent on the initial topology. For small problems, the algorithm found the global minimum most of the time. It found slightly higher delay topologies the rest of the time. There is a gap between the values of the simulated annealing solutions and the lower bound. For different size problems, this gap was in the range of 18.2–61.6% of the lower bound. The gap is a function of the quality of the solutions and the tightness of the lower bound. The lower bound is not very tight. For example, it can be observed from Table I that the gap values between the lower bound and the global minimum for 6 and 7 LAN problems are 18.2% and 31.5% of the lower bound, respectively. Therefore, we conjecture that the annealing solutions for larger problems are not very far from the global minimum.

In fact, Table I shows the worst overall performance of the annealing algorithm because the gap between the lower bound and the annealing solutions becomes smaller for the relatively lighter load cases. The gap values were between 4.6% and

22.5% of the lower bound for light load cases. A complete table of results for the lighter load cases is given in [27]. One of the reasons for the gap between the solutions and the lower bound being large is using the *minimum requirement spanning tree* [23] for finding the bound on the total flow on LANs and bridges. This tree is found without considering the capacity constraints for LAN's and bridges and is not feasible most of the time. Thus, actual flow on a feasible tree is larger than the lower bound for the total flow. The difference between the actual total flow and the lower bound is smaller for light load cases, leading to a smaller gap.

The simulated annealing algorithm outperformed multiple runs of the greedy local search algorithm in all cases studied. The contrast between the range of simulated annealing solutions and that of the greedy local search shows that the simulated annealing algorithm does not get stuck in a local minimum. Related columns of Table I show that the simulated annealing algorithm finds better topologies than the best of 10,000 randomly generated topologies. Furthermore, the majority of the random topologies have significantly higher delays than the simulated annealing solutions. The last column in Table I shows the average CPU times for the simulated annealing algorithm on a PC/AT and CONVEX 120 minicomputer. For smaller problems, the algorithm finds high-quality solutions very quickly even on a personal computer. For larger problems, running times are still reasonable, given that this is an off-line design problem.

C. Results for the Interconnected LAN/MAN Problem

We have experimented with two LAN/MAN interconnection problems. The first problem had a small number of LAN's in each cluster as shown in Fig. 1, so that the global minimum could be found with complete enumeration. The second problem had more clusters and more LAN's in each cluster. The top portion of Table II summarizes the results for the first LAN/MAN problem. The simulated annealing algorithm found the global optimum topology for each of the three clusters in this problem. Since the overall topology is a combination of these cluster topologies, the solution is also globally optimal for the first problem. The short overall running time for the first problem shows the advantage of decomposing the problem. If we had approached the problem as an 18 LAN problem without decomposing it, the running time would have been around 20 minutes instead of 67 seconds on a PC/AT. Only one LAN/MAN bridge was sufficient for each cluster in order to keep the *cluster access delays* below the threshold (i.e., 20 ms).

The lower portion of Table II summarizes the results for the second problem, which has more and larger clusters. Although we do not know the overall global optimum for this problem, we can still check for the global optimum for small clusters. Goodness measures described before show that we have good topologies for larger clusters. Therefore, we conjecture that the simulated annealing algorithm finds high-quality solutions for the LAN/MAN problem. Running times for the algorithm are short because of the decomposition of the problem.

TABLE II
RESULTS FOR TWO LAN/MAN PROBLEMS

	Number of LAN's	$T_{\min}$ Annealing	$T_{\max}$ Annealing	Cluster Access Delay	CPU Time PC-AT/CONVEX
Cluster 1	6	8.421*	8.480	14.643	22 s/< 1 s
Cluster 2	5	7.156*	7.352	14.417	9 s/< 1 s
Cluster 3	7	8.572*	8.674	15.553	36 s/< 1 s
Overall	18	9.483*	9.667	—	67 s/1 s
Cluster 1	6	8.375*	8.437	13.948	23 s/< 1 s
Cluster 2	12	10.929	11.258	17.663	229 s/3.6 s
Cluster 3	7	8.430*	8.768	14.177	34 s/< 1 s
Cluster 4	10	9.555	9.780	16.402	86 s/1.3 s
Overall	35	13.069	13.349	—	372 s/6 s

* Global minimum in complete enumeration

V. CONCLUSION

We have described a method based on simulated annealing for finding minimum-delay spanning tree topologies for interconnected LAN/MAN networks. We have decomposed the LAN/MAN topology design problem into smaller, independent problems for each cluster. We have derived a lower bound for the minimum-delay spanning tree problem using the solution of the optimum requirement spanning tree problem described in [23]. Results of the computational experiments with small problems have shown that the simulated annealing algorithm finds the optimum topology much faster than complete enumeration. For larger problems, comparisons with the lower bound have indicated that the simulated annealing solutions are not very far from the global optimum. Furthermore, comparisons with multiple runs of greedy local search have shown that the quality of the solutions found by the simulated annealing algorithm depends at most weakly on the initial topology and are better than those of ordinary local search algorithms.

Increasing demand for the interconnection of LAN's and better LAN/MAN topologies is continuing. New backbone network architectures are proposed, and the local exchange carriers are offering connectionless data transfer services for interconnecting LAN's. Although they are more complex and slower than bridges, routers are becoming popular. Routers are more flexible and do not require the topology to be a spanning tree. Design of interconnected LAN/MAN topologies using routers is another problem which can be approached in a similar way. Since the topology is no longer required to be a spanning tree, different objective functions such as minimizing the dollar cost and maximizing the throughput can be used. Other constraints can easily be incorporated into the formulation of the problem, such as constraints on the maximum number of hops between any two LAN's, maximum utilization for LAN's, or reliability requirements. Because of its simplicity and general applicability, simulated annealing can be used for finding good feasible solutions for many variations of the interconnected LAN/MAN topology design problem.

APPENDIX A
OPTIMUM DISTRIBUTION OF THE TOTAL FLOW ON LAN'S

A lower bound for the total flow on all LAN's ($\lambda_{L,\min}$) and a set of inequality constraints due to *mandatory* flows on each LAN were presented in Section II-C. The delay due to LAN's is given by

$$T_L = \frac{1}{\gamma}\sum_{i=1}^{N} X\lambda_i E[T_i] = \frac{X}{\gamma}\sum_{i=1}^{N}\frac{X\lambda_i}{\mu_i - X\lambda_i} \tag{27}$$

and

$$\lambda_{L,\min}^T = \sum_{i=1}^{N}\lambda_i. \tag{28}$$

A lower bound for T_L can be obtained by finding the optimum distribution of $\lambda_{L,\min}^T$ on all LAN's subject to the inequality constraints in (16) so that T_L is minimized. This is a standard optimization problem which can be solved using Lagrangean techniques and the following Kuhn–Tucker conditions [24]:

$$\frac{\partial}{\partial\lambda_i}\left(\sum_{i=1}^{N}\frac{X\lambda_i}{\mu_i - X\lambda_i} - \alpha\left(\sum_{i=1}^{N}\lambda_i - \lambda_{L,\min}^T\right) - \sum_{i=1}^{N}\beta_i(\lambda_i^M - \lambda_i)\right) = 0 \qquad \forall i \tag{29}$$

$$\beta_i(\lambda_i^M - \lambda_i) = 0 \qquad \forall i \tag{30}$$

$$\beta_i \geq 0 \qquad \forall i \tag{31}$$

where α and β_i's are Lagrangean coefficients. The optimum distribution for the general case can be found by solving the set of equations specified by the Kuhn–Tucker conditions. If all LAN's have equal capacity (i.e., $\mu_i = \mu_L, \forall i$), it can be verified using the Kuhn–Tucker conditions that the water-filling type distribution of $\lambda_{L,\min}^T$ described in Section II-C and shown in Fig. 3 minimizes T_L. In this distribution, if $\lambda_i \neq \lambda_i^M$, then the corresponding $\beta_i = 0$ due to (30). Consequently, (29) becomes

$$\frac{X\mu_L}{(\mu_L - X\lambda_i)^2} - \alpha = 0 \qquad \forall i \text{ s.t. } \lambda_i \geq \lambda_i^M. \tag{32}$$

Therefore,

$$\lambda_i = \lambda_L' \qquad \forall i \text{ s.t.} \lambda_i \geq \lambda_i^M. \tag{33}$$

For LAN's with $\lambda_i = \lambda_i^M$, (29) becomes

$$\frac{X\mu_L}{(\mu_L - X\lambda_i)^2} - \alpha + \beta_i = 0 \qquad \forall i \text{ s.t.} \lambda_i = \lambda_i^M. \tag{34}$$

Therefore,

$$\beta_i = \alpha - \frac{X\mu_L}{(\mu_L - X\lambda_i)^2} = \frac{X\mu_L}{(\mu_L - X\lambda_L')^2} - \frac{X\mu_L}{(\mu_L - X\lambda_i)^2} \geq 0 \quad \forall i \text{ s.t.} \lambda_i = \lambda_i^M \tag{35}$$

since $\lambda_i > \lambda_L'$. As a result, all β_i's satisfy (31) which shows that we can find β_i's satisfying the Kuhn–Tucker conditions. Hence, the distribution shown in Fig. 3 minimizes T_L.

Appendix B
Optimum Distribution of the Total Flow on Bridges

A lower bound for the total flow on bridges ($\lambda_{B,\min}^T$) was presented in Section II-C. A lower bound for T_B can be obtained by optimally distributing $\lambda_{B,\min}^T$ on all bridge ports such that T_B is minimized. This is a standard optimization problem, which can be solved using a Lagrange multiplier. The delay due to bridges is given by

$$T_B = \frac{1}{\gamma}\sum_{\forall Port_{ij}} X\lambda_{ij}E[T_{ij}] = \frac{X}{\gamma}\sum_{\forall Port_{ij}}\frac{X\lambda_{ij}}{\mu_{ij} - X\lambda_{ij}} \tag{36}$$

and

$$\lambda_{B,\min}^T = \sum_{\forall Port_{ij}}\lambda_{ij}. \tag{37}$$

In order to minimize T_B subject to (37), we form the Lagrangean $\mathcal{L}$

$$\mathcal{L} = \sum_{\forall Port_{ij}}\frac{X\lambda_{ij}}{\mu_{ij} - X\lambda_{ij}} - \alpha\left(\sum_{\forall Port_{ij}}\lambda_{ij} - \lambda_{B,\min}^T\right) \tag{38}$$

where α is the Lagrange multiplier. Stationary points of $\mathcal{L}$ are found by checking where the first derivatives with respect to individual λ_{ij}'s are equal to zero:

$$\frac{\partial\mathcal{L}}{\partial\lambda_{ij}} = \frac{X\mu_{ij}}{(\mu_{ij} - X\lambda_{ij})^2} - \alpha = 0 \qquad \forall Port_{ij}. \tag{39}$$

μ_{ij}'s are known; therefore, we can find all λ_{ij}'s in terms of α using (39). We also know that the summation of all $\lambda_{ij}'s$ is equal to $\lambda_{B,\min}^T$ from which we can find α and λ_{ij}'s. In order to verify that these λ_{ij}'s result in a minimum for T_B, we check if the second partial derivative of $\mathcal{L}$ is positive

$$\frac{\partial^2\mathcal{L}}{\partial\lambda_{ij}^2} = 2X(\mu_{ij} - X\lambda_{ij}) \qquad \forall Port_{ij}. \tag{40}$$

For normal operation, $X\lambda_{ij} < \mu_{ij}$ for all bridge ports; otherwise, the delay will be infinite. This makes the quantity in (40) positive, thus ensuring a minimum. Therefore, λ_{ij}'s found using this distribution give a lower bound on T_L.

If all bridges have equal capacity (i.e., $\mu_{ij} = \mu_B, \forall Port_{ij}$), the uniform distribution of $\lambda_{B,\min}^T$ among all bridge ports gives a lower bound for T_B. For this case, the following bridge port flows give a lower bound for T_B:

$$\lambda_{ij} = \lambda_B = \frac{1}{2(N-1)}\lambda_{B,\min}^T \qquad \forall Port_{ij}. \tag{41}$$

This lower bound for T_B can be improved by considering additional inequality constraints. We pick the $2(N-1)$ largest inter-LAN traffic requirements. Since each of these requirements has to be carried by at least one bridge, they impose inequality constraints on bridge flows. In order to minimize

T_B, each of these requirements is assigned on a separate bridge port. As in the case of LAN's, finding a lower bound on T_B is equivalent to finding the optimum distribution of $\lambda_{B,\min}^T$ on all bridge ports subject to $2(N-1)$ inequality constraints due to the $2(N-1)$ largest requirements. Again, a water-filling type distribution similar to the one shown in Fig. 3 is optimum and gives an improved lower bound on T_B.

If the bridge port capacities are not equal, the $2(N-1)$ largest requirements cannot be used for improving the lower bound on T_B since they cannot be assigned on separate bridge ports. In that case, the lower bound found using the Lagrangean in (38) is used for T_B. As shown in (5), the sum of the lower bounds on T_L and T_B gives the lower bound for the average network delay T.

REFERENCES

[1] "IEEE Standard 802.1, part D: MAC Bridges," in IEEE Project 802 Local and Metropolitan Area Network Standards.

[2] F. Backes, "Transparent bridges for interconnection of IEEE 802 LAN's," *IEEE Netw.*, pp. 5–9, Jan. 1988.

[3] C. K. Kwok and B. Mukherjee, "On transparent bridging of CSMA/CD networks," in *Proc. IEEE GLOBECOM*, 1989, pp. 185–190.

[4] L. Merakos and H. Xie, "Interconnection of CSMA/CD LAN's via an N-port bridge," in *Proc. IEEE INFOCOM*, 1989, pp. 28–37.

[5] C. Ersoy, S. S. Panwar, R. Dalias, and D. Segal, "Transient phenomena in bridged local area networks," in *Proc. IEEE GLOBECOM*, 1990.

[6] S. C. Liang and J. R. Yee, "A gateway allocation algorithm for interconnecting existing data networks," in *Proc. IEEE INFOCOM*, 1989, pp. 468–473.

[7] P. C. Fetterolf and G. Anandalingam, "Optimal design of LAN-WAN internetworks: An approach using simulated annealing," in *Proc. ORSA Telecommun. Conf.*, 1990.

[8] S. Gupta and K. W. Ross, "Performance modelling and optimization of interconnected ethernets," in *Proc. IEEE INFOCOM*, 1991, pp. 1353–1359.

[9] F. E. Ross, "An overview of FDDI: The fiber distributed data interface," *IEEE J. Select. Areas Commun.*, pp. 1043–1051, Sept. 1989.

[10] J. F. Mollenauer, "Standards for metropolitan area networks," *IEEE Commun. Mag.*, pp. 15–19, Apr. 1988.

[11] G. H. Clapp, "LAN interconnection across SMDS," *IEEE Netw. Mag.*, pp. 25–32, Sept. 1991.

[12] L. Mongiovi, M. Farrell, and V. Trecordi, "A proposal for interconnecting FDDI networks through B-ISDN," in *Proc. IEEE INFOCOM*, pp. 1160–1167, 1991.

[13] S. E. Minzer, "Broadband ISDN and asynchronous transfer mode (ATM)," *IEEE Commun. Mag.*, pp. 17–24, Sept. 1989.

[14] P. Marsden, "Interworking IEEE 802/FDDI LAN's via the ISDN frame relay bearer service," *Proc. IEEE*, pp. 223–229, Feb. 1991.

[15] R. Perlman, "An algorithm for distributed computation of a spanning tree in an extended LAN," *Comput. Commun. Rev.*, pp. 44–53, Sept. 1985.

[16] D. Bertsekas and R. Gallager, *Data Networks, 2nd Ed.* Englewood Cliffs, NJ: Prentice-Hall, 1992.

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

[18] W. Nijenhuis, *Combinatorial Algorithms.* New York: Academic, 1975.

[19] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines.* New York: Wiley, 1989.

[20] R. Gusella, "A measurement study of diskless workstation traffic on an Ethernet," *IEEE Trans. Commun.*, pp. 1557–1568, Sept. 1990.

[21] W. E. Leland and D. V. Wilson, "High time-resolution measurement and analysis of LAN traffic: Implications for LAN interconnection," in *Proc. IEEE INFOCOM*, 1991, pp. 1360–1366.

[22] D. Gross and C. M. Harris, *Fundementals of Queueing Theory, 2nd Ed.* New York: Wiley, 1985.

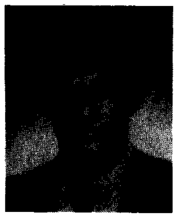
[23] T. C. Hu, "Optimum communication spanning trees," *SIAM J. Comp.*, vol. 3, pp. 188–195, 1974.

[24] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research, 4th Ed.* New York: Holden-Day, 1986.

[25] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Sci.*, pp. 671–680, May 1983.

[26] B. Hajek, "Cooling schedules for optimal annealing," *Math. Operat. Res.*, vol. 13, pp. 311–329, 1988.

[27] C. Ersoy, "Topological design of interconnected local and metropolitan area networks," Ph.D. dissertation, Polytech. Univ., Brooklyn, NY, June 1992.



Cem Ersoy (S'87-M'91-S'91–M'92) was born in Ankara, Turkey, on September 28, 1963. He received the B.S. and M.S. degrees in electrical engineering from Bogazici University, Istanbul, Turkey, in 1984 and 1986, respectively, and the Ph.D. degree in electrical engineering from the Polytechnic University, Brooklyn, NY, in 1992.

From 1984 to 1986, he worked as a Research and Development Engineer at Northern Electric Telecommunications (NETAS), Istanbul, Turkey. At the Polytechnic University, he worked as a Teaching Assistant from 1987 to 1989 and as a Research Assistant from 1989 to 1992. He is currently an Assistant Professor in the Computer Engineering Department of Boğaziçi University, Istanbul, Turkey. His research interests include computer networks and optimization.

Shivendra S. Panwar (S'82–M'85) was born in Delhi, India, on December 15, 1959. He received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, in 1981, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Massachusetts, Amherst, in 1983 and 1986, respectively.

From 1981 to 1985, he was a Research Assistant at the University of Massachusetts. He joined the Department of Electrical Engineering at the Polytechnic University, Brooklyn, where he is currently an Associate Professor. He spent the summer of 1987 as a Visiting Scientist at the IBM T. J. Watson Research Center, Yorktown Heights, NY, and has been a Consultant to AT&T Bell Laboratories, Holmdel, NJ. His research interests include the performance analysis and design of high-speed networks. He is currently the Secretary of the Technical Affairs Council of the IEEE Communications Society and Co-Chairman of the Technical Program Committee of the Second IEEE Network Management and Control Workshop.

Dr. Panwar is a member of Tau Beta Pi and Sigma Xi.