Traffic Management of a Satellite Communication Network Using Stochastic Optimization

Nirwan Ansari, *Senior Member, IEEE,* Ambalavanar Arulambalam, *Student Member, IEEE,* and Santhalingam Balasekar, *Student Member, IEEE*

***Abstract*—The performance of nonhierachical circuit switched networks at moderate load conditions is improved when alternate routes are made available. Alternate routes, however, introduce instability under heavy and overloaded conditions, and under these load conditions network performance is found to deteriorate. To alleviate this problem, a control mechanism is used where, a fraction of the capacity of each link is reserved for direct routed calls. In this work, a traffic management scheme is developed to enhance the performance of a mesh-connected, circuit-switched satellite communication network. The network load is measured and the network is continually adapted by reconfiguring the map to suit the current traffic conditions. The routing is performed dynamically. The reconfiguration of the network is done by properly allocating the capacity of each link and placing an optimal reservation on each link. The optimization is done by using two neural network-based optimization techniques: simulated annealing and mean field annealing. A comparative study is done between these two techniques. The results from the simulation study show that this method of traffic management performs better than the pure dynamic routing with a fixed configuration.**

I. Introduction

SATELLITE communications has gone through a period of tremendous growth, contributing to the information explosion that has changed the world in the last quarter century. Satellites, from their geostationary orbit position, 22 300 miles over the equator, view over one-third of the earth and can instantly connect any pair of points within their coverage [1]. This property, together with their record of high reliability, makes them the most desirable multiple access diversity communication medium. Many studies on circuit-switched telecommunication networks are useful here since a wide spectrum of applications in satellite networks are circuit-switched networks.

For circuit-switched networks, it is known that nonhierarchical routing performs better than the hierarchical static routing [2]. The dynamic nonhierarchical routing (DNHR) method of AT&T [3] and an adaptive routing algorithm known as dynamic control routing (DCR) that is being considered by Telecomm Canada [4] are two of the most common examples of nonhierarchical networks. Several schemes have been proposed and studied by many in the field of networking [5]–[8]. From these studies it is well known that nonhierarchical dynamic routing performs better than the hierarchical static routing. It is also shown that allowing alternate routes in nonhierarchical networks results in an improved performance at moderate loads but suffers severely at overload conditions. Many control mechanisms have been proposed to overcome this stability problem. A study done by Akinpela [8] suggests reservation of some portion of the capacity at high loads avoids the instability. In a study done by Mitra *et al.* [9], optimal trunk reservation is found for a fully connected, completely symmetric network.

In this work a new traffic management scheme is proposed to improve the efficiency of a circuit-switched satellite communication network of the geostationary orbital type. Contributions of this paper include the introduction of the new traffic management model and the application of simulated mean field annealing (MFA), neural network-based technique [10], [11], to carry out the proposed management scheme. Further applications of neural networks in telecommunications can be found in [12]. Analytical results are developed to aid simulations. This work is an extension of the work done by Balasekar and Ansari [13]. The proposed scheme incorporates the idea of dynamically adapting the networks as well as dynamically routing each arrival. The scheme allows the network to change according to the traffic conditions, and thus, improves the grade of service. The system is modeled such that it continuously organizes itself to minimize the cost for varying traffic conditions.

II. The Traffic Management Scheme

A. Network Model

A satellite communication network consists of a number of geostationary satellites, each satellite covering a number of ground stations. This kind of satellite network can be modeled as a mesh-connected topology, where each node represents either a satellite or an earth station. The connections between the nodes denote the links between stations. The links may have any number of circuits, but the total capacity of the network is fixed. Traffic is generated by purely random (e.g., Poisson) sources characterized by two parameters: the average rate of message or call generation and the average length of message or call duration. The satellite communication system is modeled as an appropriate server system which provides a transmission service to the generated traffic or "customer."

The objective here is to design a network such that the overall block rate of the network is minimized, and, thus, the throughput is maximized. Since traffic conditions (i.e.,

Manuscript received June 5, 1994; revised January 29, 1995 and April 20, 1995. This work was supported in part by the NASA Lewis Research Center, Cleveland, OH, Contract NAG3-1244.

The authors are with the Department of Electrical and Computer Engineering, Center for Communications and Signal Processing Research, New Jersey Institute of Technology, Newark, NJ 07102 USA.

Publisher Item Identifier S 1045-9227(96)02872-X.

1045–9227/96$05.00 © 1996 IEEE

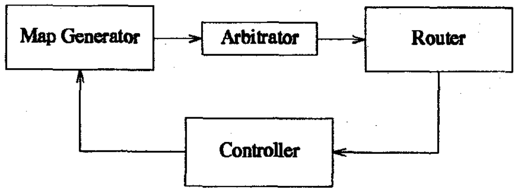


Fig. 1. System model.

arrival rates) change from time to time, a scheme which will dynamically route each call is proposed. The proposed scheme can be explained with the help of the block diagram shown in Fig. 1. The scheme is made up of four functional modules: map generator, router, controller, and arbitrator. The function of each of the modules is described in detail in the subsequent sections.

B. Map Generator

The function of the map generator is to generate a map of the best configuration for current traffic conditions. Maps differ from each other by two parameters, namely, $\vec{c}$ and $\vec{r}$. Vector $\vec{c}$ denotes the link capacities of the network, and the elements of vector $\vec{r}$ denote the number of circuits that can be used by alternately routed calls. Therefore, $c - r$ circuits in a particular link are reserved for direct calls only. The parameter, $c - r$, is referred to as the reservation parameter. Average arrival rates for each origin-destination (O-D) pair, the total capacity of the network, and the current status of the network are given as inputs to this module. Based on this information, an optimization technique is used to find an optimal map which will minimize the total block rate of the network. Two kinds of optimization techniques, simulated annealing and MFA, are employed. Map generation by simulated annealing and by MFA is discussed in Sections IV and V, respectively.

C. Router

The router performs the routing dynamically for every call arriving at the network as follows:

- If the direct link has an idle circuit, an arriving call is routed on the direct link.
- If the direct link has no idle circuits, a randomly selected alternate route is tried. An alternately routed call will be blocked if either one or both links corresponding to that particular O-D pair is in the reserved state (i.e., at least r circuits in the link are busy).
- If direct link routing and alternate routing fail, the call is blocked and then lost from the network.

The routing module performs a simple routing function without much computation and, thereby, reduces the processing delay of each call. There are other schemes which try to dynamically compute the alternate route that minimizes the blocking rate. For instance, the least busy alternative (LBA) model developed by Mitra *et. al.* [9], computes a least busy alternate route. In this work the proper choice of a map eliminates the need for computing the least busy alternate route. Avoiding these computations increases the efficiency of the map generation process.

D. Controller

The controller's job is to keep track of the network's status and performance. The controller decides whether a new map is necessary based on the current network status, which is updated at regular intervals. The arrival rate of calls to each O-D pair and the load balance of the network are two of the most important parameters of which the controller keeps track. The load imbalance is measured using the following technique.

Measurement of Load Imbalance: Let Δ denote the ratio of the network's total flow to the total capacity, where the total flow is the sum of calls accepted by all links. Let δ_{ij} denote the ratio of flow (f_{ij}) to the capacity (c_{ij}) of the link (i, j). At each update a measure of the network's load imbalance, denoted by d, is computed from these ratios as follows:

$$\begin{aligned} \Delta &= \frac{F}{C}, \\ \delta_{ij} &= \frac{f_{ij}}{c_{ij}}, \\ d &= \frac{1}{C} \sum_{(i,j)} (\Delta - \delta_{ij})^2 \end{aligned} \tag{1}$$

where C and F are the total capacity and the total flow of the network, respectively. Therefore, the parameter d indicates the amount by which the network's current load balance deviates from that of the fully balanced network. This measure of disparity in the network load is taken as an indication of potential premature saturation of the network. Let d_t be defined as a threshold value for this imbalance with respect to the network load balance. In the simulations done here, the threshold value is defined as

$$d_t = 0.1 \times \Delta. \tag{2}$$

When d is larger than d_t, the network is considered to be in the inefficient state and the controller module calls the map generator module to come up with a better map for the current traffic condition. When d_t is small and close to zero, even a small deviation of network load balance from the ideal condition is not tolerated. In this situation, the map generator is called upon very frequently. But too frequent changes in the configuration may not be cost effective when considering other costs associated with changing the configuration of a satellite network. To avoid this, parameter d is updated after a fixed number of network updates. In the simulations here, network status is updated at the end of every time unit and the parameter d is computed after 10 updates. Up to 100 time units are used in measuring the traffic pattern.

E. Arbitrator

The function of the arbitrator is to decide whether changing the map will be beneficial to network performance, and thus, it is used as a cost saving measure. The routing of calls must be uninterrupted and the optimization has to be done in real time. Hence, there may be some instances where the map configured from the most recent network experience may not actually reflect the optimal performance for present traffic conditions if the traffic pattern changes too quickly. Therefore,

Fig. 2. Representing the reserved state.

the arbitrator's function is to change the map that is being generated if the change will result in better performance. The above situation can, in fact, be eliminated to a certain extent by properly choosing the duration that the network is monitored for measuring the network traffic condition, and by properly choosing the tolerance level.

III. Analytical Model

A queueing model is employed here to analyze the network. Since the average block rate is used as the cost, an expression for the average block rate must be developed. Before carrying out the analysis, the following assumptions are made.

- New calls and overflow calls to any link form a Poisson process and are independent.
- Holding times of calls are exponentially distributed.
- Each link is represented by an $M/M/m/m$ queueing model, where m is the number of circuits in that link.
- The average holding time of calls is assumed to be one time unit.
- Link blocking probabilities are independent.
- Processing and propagation delays are negligible.

The last assumption helps us study the network based only on the proposed scheme, where only the block rate is used as the cost. Furthermore, this is close to a real situation in circuit-switched networks due to long average holding time. Even though calls arriving at an alternate route is clearly not Poisson, this is found to be a reasonable assumption, especially when each link derives traffic from many end users. Similarly, the link blocking independence assumption is found to be reasonable, and this greatly reduces the complexity of the analysis. A detailed discussion of the validity of these assumptions can be found in [4].

A. Notations

1) (i, j) Link from node i to node j.
2) $(i-j)$ The O-D pair from node i to node j.
3) λ_{i-j} External (new) arrival rate of $(i-j)$.
4) Λ The total rate of input traffic to the network.
5) γ_{ij} Total arrival rate to (i, j).
6) f_{ij} Flow in (i, j).
7) c_{ij} Capacity, in number of circuits, of (i, j).
8) r_{ij} Number of circuits in (i, j) that allow alternately routed calls.
9) m A tandem node used in an alternate route.
10) M_{i-j} Set of tandem nodes that forms the alternate routes for $(i-j)$.
11) R^k_{i-j} An alternate route for $(i-j)$, with node k as the tandem node.
12) B_{i-j} Probability that call $(i-j)$ is blocked from the network.
13) B_{ij} Probability that any call is blocked in (i, j).
14) B^R_{ij} Probability that an alternately routed call is blocked in (i, j).
15) $\overline{B}$ Average network blocking probability.

For notational convenience, subscripts are dropped when referring to a particular link.

B. Calculation of The Block Rate

Denote the average network blocking probability as $\overline{B}$, which can be obtained by summing all the call blocking probabilities and normalizing the sum by the total arrivals to the network

$$\overline{B} = \frac{1}{\Lambda} \sum_{(i-j)} \lambda_{i-j} B_{i-j} \tag{3}$$

where Λ, the rate of the total input traffic to the network is given by

$$\Lambda = \sum_{(i-j)} \lambda_{i-j}. \tag{4}$$

Arrival rates λ_{i-j} are known quantities. Under the previously stated assumption of independent link blocking probabilities, B_{i-j} can be expressed in terms of B_{ij} and B^R_{ij}. A call $(i-j)$ is blocked from the network only when the direct link and the possible alternate routes are busy. The alternate routes are busy when either or both of the links constituting that route are busy. Hence, the call blocking probability B_{i-j} is

$$B_{i-j} = B_{ij} \prod_{m \in M_{i-j}} [1 - (1 - B^R_{im})(1 - B^R_{mj})]. \tag{5}$$

C. Queueing Analysis

The link blocking probabilities B_{ij} and B^R_{ij} in (5) are derived from the birth-death process of an $M/M/m/m$ queueing model [15]. Let λ and v denote the arrival rates of new and overflow calls, respectively. In this model a state is defined as the number of circuits occupied in a given link (see Fig. 2). In the given link, $c-r$ circuits are reserved for direct calls, and r circuits can be used by either direct calls or alternately routed calls. If a new call arrives to the link or a call is serviced by the link, the state of a link changes from one to another. If the transition rate from state i_1 to state i_2 is denoted by $p(i_1, i_2)$, then

$$p(i, i+1) = \begin{cases} \lambda & (c > i \geq r) \\ \lambda + v & (r > i \geq 0) \end{cases} \tag{6}$$

$$p(i, i-1) = i\mu \quad (c \geq i > 0) \tag{7}$$

where μ is the service rate ($\mu = 1$), $\gamma \doteq \lambda + v$ and $\alpha \doteq \lambda/\gamma$.

Let p_i denote the steady state probability of having i circuits occupied in a given link. The following steady-state probabilities are obtained from the balance equations for an $M/M/m/m$ queueing model. Detailed derivation is found in [15]

$$p_i = \begin{cases} \dfrac{\gamma^i}{i!} p_0 & (0 \leq i \leq r) \\ \dfrac{\gamma^i (1-\alpha)^{i-r}}{i!} p_0 & (r < i \leq c) \end{cases} \tag{8}$$

where p_0 is found using the flow conservation equation, $\sum_{i=0}^{c} p_i = 1$, and is given below

$$p_0 = \left[\sum_{i=0}^{r} \frac{\gamma^i}{i!} + \sum_{i=r+1}^{c} \frac{\gamma^i (1-\alpha)^{i-r}}{i!} \right]^{-1}. \tag{9}$$

Substituting (9) into (8), p_i can be solved in terms of γ, α, and c. The link blocking probabilities B and B^R are then related to these parameters via (10) and (11) below

$$B = p_c \tag{10}$$

$$B^R = \sum_{i=r+1}^{c} p_i. \tag{11}$$

To solve for the link blocking probabilities B and B^R, one needs to relate the unknown link arrivals γ and the parameter α to the known node-to-node arrivals λ and the reservation parameter r of the links in the network. This is done by finding the flow f_{ij} on each link with two different methods, as described in the following section.

D. Calculation of Flow

The flow on each link can be calculated using two different methods. Knowing the link blocking probabilities, the flow is found by first using γ, and then λ and the overflow rate from other O-D pairs.

Method 1: For a given map (i.e., with c and r for each link specified) the flows f_{ij} of all links (i, j) can be computed from

$$f_{ij} = \gamma_{ij}(1 - B_{ij}). \tag{12}$$

Method 2: Alternatively, flows in each link can also be given in terms of the node-to-node traffic λ and the link blocking probabilities. Suppose, for link (i, j), the rate of accepted direct calls is denoted by u_{ij}, and the rate of accepted alternately routed calls is denoted by y_{ij}, then the flow on link (i, j) is

$$f_{ij} = u_{ij} + y_{ij} \tag{13}$$

where

$$u_{ij} = \lambda_{i-j}(1 - B_{ij}). \tag{14}$$

With $f_{i'-j'}^{k}$ denoting the rate of accepted calls in link (i', j') on alternate route $R_{i'-j'}^{k}$, y_{ij} in (13) is given by

$$y_{ij} = \left[\sum_{i':(i,j) \in R_{i'-j}^{i}} f_{i'-j}^{i} + \sum_{j':(i,j) \in R_{i-j'}^{j}} f_{i-j'}^{j} \right]. \tag{15}$$

Suppose $Q_{i'-j'}^{k}$ is used to denote the probability that call $(i' - j')$ is not blocked on the alternate route $R_{i'-j'}^{k}$ while it is blocked on all other alternate routes available for this O-D pair, then

$$f_{i'-j'}^{k} = \lambda_{i'-j'} B_{i'j'} Q_{i'-j'}^{k} \tag{16}$$

where $\lambda_{i'-j'} B_{i'j'}$ is the rate of call overflow $(i' - j')$ from the direct link (i', j'). $Q_{i'-j'}^{k}$ can be given in terms of the link blocking probabilities and can be written as

$$Q_{i'-j'}^{k} = \left[\prod_{\substack{m \neq k \\ m \in M_{i'-j'}}} B(R_{i'-j'}^{m}) - \prod_{m \in M_{i'-j'}} B(R_{i'-j'}^{m}) \right] \tag{17}$$

where $B(R_{i'-j'}^{m})$ is the blocking probability of the alternate route $R_{i'-j'}^{m}$ and this can be expressed in terms of the link blocking probabilities for alternately routed calls of the links that form this alternate route, as given below

$$B(R_{i'-j'}^{m}) = [1 - (1 - B_{i'm}^{R})(1 - B_{mj'}^{R})]. \tag{18}$$

Using the initial estimate of the quantities α, link arrivals γ are obtained from external arrival rates λ. Then using (6)–(11), the link blocking probabilities are computed. These probabilities are used to compute the flow by using (13)–(18). For these flows, the new link arrivals can be computed from (12) and then a new set of α for each link are calculated as follows:

$$\alpha_{ij} = \frac{\lambda_{ij}}{\gamma_{ij}}. \tag{19}$$

The above steps are repeated until the flows found in successive iterations converge. Finally, from the resulting blocking probabilities, the desired quantity $\overline{B}$ is obtained from (3)–(5). This $\overline{B}$, the network block rate, is used as the cost function in determining the best map in the map generator module, as explained in the next two sections.

IV. MAP GENERATION USING SIMULATED ANNEALING

To improve the performance of the network, a map which will minimize the total block rate of the network must be chosen. As explained in Section III, the block rate $\overline{B}$ depends on the capacity per link and on the number of circuits that can be used by alternately routed calls in a link. These two independent variables, c and r, make the solution space very large. Choosing the best map from this solution space is computationally time consuming. Therefore, a powerful optimization technique should be applied to find an optimal map. In this paper, two neural network-based optimization techniques, simulated annealing and MFA, are used. This section describes the application of simulated annealing in the map generation process. The next section describes how MFA can be applied to speed up the map generation task.

A. Introduction to Simulated Annealing

First proposed by Kirkpatrick [16], simulated annealing is a stochastic optimization technique used to solve complex problems. Since then it has been applied in many areas. Generally, a combinatorial optimization problem consists of a set S of configurations or solutions and a cost function C, which determines the cost $C(s)$ for each configuration. An iterative improvement scheme known as local search could be performed to find the minimum cost. During a local search process, an initial solution s_i is given and then a new solution s_n is proposed at random. If the new cost $C(s_n)$ is less than the current cost $C(s_c)$, which is the same as $C(s_i)$, then the new configuration is accepted, and the new solution s_n becomes the current solution s_c. If the new cost is larger than the current cost then a new solution is proposed, again at random. This procedure continues until the minimum solution is found. Unfortunately, a local search may get stuck at the local minima and may escape the global minima. To alleviate the problem of getting trapped at local minima, simulated annealing occasionally allows "uphill moves" to solutions of higher cost according to the so-called metropolis criterion [14].

The simulated annealing algorithm is based on the analogy between the simulation of the annealing of solids and the problem of solving large combinatorial optimization problems. For this reason the algorithm became known as "simulated annealing." The simulated annealing process consists of first "melting" the system being optimized at an effective high temperature, then lowering the temperature gradually until the system "freezes" and no further changes occur [16]. At each temperature, the system is allowed to reach thermal equilibrium, characterized by a probability of being in a state s, $P(s)$. This probability is proportional to the Boltzman probability factor and can be written as

$$P(s) \propto e^{-E(s)/kT} \tag{20}$$

where $E(s)$ is the energy of the system at state s, k is the Boltzman constant, and T denotes the temperature at which thermal equilibrium is maintained. Now, apply a small perturbation to the current state s_c of the system and obtain a new state s_n. Denote p, the ratio between the probabilities of finding the system in two states s_c and s_n, as

$$\begin{aligned} p &= \frac{P(s_n)}{P(s_c)} \\ &= e^{-\{[E(s_n)-E(s_c)]/kT\}}. \end{aligned} \tag{21}$$

If the difference in energy, ΔE, between the current state and the slightly perturbed one is negative (i.e., the new state has lower energy than the current state), then the process is continued with the new state. If $\Delta E > 0$, then the probability of acceptance of the perturbed state is given by (21). This acceptance rule for new states is referred to as the metropolis criterion. Following this criterion, the system eventually evolves into a state of thermal equilibrium. Detailed discussions on the convergence of simulated annealing type algorithms and other similar algorithms can be found in [17]–[19].

B. Application of Simulated Annealing in Map Generation

In this section an optimal map generation process using simulated annealing is described. Before applying the algorithm, a cost function, the cooling schedule and a generation mechanism (or, equivalently, a neighborhood structure) must be defined.

1) Cost Function: In this work the total block rate of the network is chosen as the cost. An expression for finding the total block rate $\overline{B}$ is given in (3). Since the block rate depends on the capacity of the link and the reservation parameter of each link, the energy function can be written as follows:

$$E(s) = \overline{B}(\vec{c}, \vec{r}) \tag{22}$$

where

$$\vec{c} = \{c_{ij} : (i, j) \in \text{all links}\}$$

and

$$\vec{r} = \{r_{ij} : (i, j) \in \text{all links}\}.$$

2) Cooling Schedule: Once the cost function is determined the following parameters should be specified:

1) the initial value of temperature T_0;
2) the final value of temperature T_f (stopping criterion);
3) the number of transitions at each temperature; and
4) a rule for changing the current value of the control parameter T_j into the next one, T_{j+1}.

These particular parameters are referred to as the cooling schedule [17].

a) Initial Temperature: The initial value of temperature T_0 is determined in such a way that virtually all the transitions are accepted. If a value is too high then all the transitions will be accepted most of the time. If the value is too low then the solution may not be optimal. To avoid these problems, a temperature must be chosen so that it is in the critical point where the number of accepted maps is about to decrease significantly. This particular temperature is referred to as the critical temperature.

b) Stopping Criterion: A stopping criterion is usually based on the argument that execution of the algorithm can be terminated if the improvement in cost (that which would be expected in the case of continuing execution of the algorithm) is small [16]. Therefore, if the maps generated consecutively do not vary significantly in cost, then the algorithm is terminated and the final map is considered the "best" map.

c) Number of Transitions at Each Temperature: The generation of new states is stopped and the next temperature level is tried when either the number of acceptances reaches a certain level (L_{accept}) or after generating a specific number (L_{trials}) of new states. Due to the variations in the experiments, different values of L_{accept} and L_{trials} are tried in the simulations.

d) Temperature Updating Rule: The difference between two temperatures and the value of the temperature relative to the difference in the possible range of costs of two states in the solution space are two parameters that need to be given close attention in determining the rule for decreasing the temperature. Since the size of the network affects ΔE, the control parameter is related to the total capacity C of the network. The rule of decreasing the temperature can be expressed as follows:

$$T = a\frac{C}{\ln(j)} \tag{23}$$

where a is a constant and j is the iteration index which is incremented linearly to produce the desired temperature schedule.

3) Neighborhood Structure: A neighborhood structure is necessary when implementing the simulated annealing algorithm. Different neighborhoods are defined for each of the cases considered during the simulations.

Case 1) Varying Reservation Parameter of the Links: If a particular link has c circuits, the number of circuits that can be reserved varies from zero to c. Therefore, when finding a neighbor, a random number in $[0, c]$ is chosen, and this number is assigned as the reservation parameter of that link.

Case 2) Varying Link Capacities Only: Here, one link is chosen at random, and one circuit is deducted from that link and assigned to another link which will benefit by this exchange. To be practical and to obtain reasonable results, a lower bound capacity and an upper bound capacity are assigned to each link. These conditions will avoid the state where some of the links have almost no circuits and some circuits have a very large number of circuits. In addition, they will avoid unnecessary computations.

Case 3) Varying Both Link Capacities and Reservation: The number of combinations in this case is very large. To be practical and to avoid unnecessary calculations, a similar control measure described for Case 2) is used. The same approach is taken here to allocate the circuits but, in addition, a reservation parameter between zero and the link capacity is randomly assigned.

Once the cost function, the cooling schedule, and the neighborhood structure are defined above, the map generation problem can be solved by simulated annealing. Simulation results are given in Section VI.

V. MAP GENERATION USING MFA

Simulated annealing is a powerful optimization technique, but it is computationally intensive, especially for large problems. As an alternative, MFA [21]–[24], which provides a good trade-off between performance and computational complexity, can be used in minimizing the call blocking probability. In MFA, two operations used in simulated annealing are still needed: a thermostatic operation which schedules the decrease of temperature, and a random relaxation process which searches for the equilibrium solution at each temperature. The relaxation process, however, has been replaced by a search for the mean value of the solution. Equilibrium can be reached faster by using the mean [24] and, thus, MFA speeds up the computational process.

To solve the problem by MFA, the problem should be mapped into a neural network and an energy function should be formulated. Since each map differs from the others in terms of the link capacities and the reservation parameter of each link, the energy function should be able to incorporate all possible combinations. Three different cases of map generation are considered:

1) The reservation parameter of each link is varied while keeping the capacity of each link constant.
2) The capacity of each link is varied while keeping the reservation parameter constant.
3) Both capacity and the reservation parameter are varied for all links.

In all cases, the total network capacity is fixed to a constant value. To represent all these cases, an encoding method to denote the neurons is necessary, and is described below.

A. Neuron Encoding

A neuron is denoted by S_{ijr}, S_{ijc}, or S_{ijcr} depending on the case considered in the analysis. For example, for the case where only the link reservation parameters vary, the neuron is denoted by S_{ijr}. S_{ijr} takes on "1" when the link (i, j) has r circuits which are allowed to handle alternately routed calls and takes on "0" otherwise. This can be defined mathematically as follows:

$$S_{ijr} = \begin{cases} 1 & \text{if } r \text{ circuits in link } (i, j) \text{ handle} \\ & \text{alternately routed calls} \\ 0 & \text{otherwise} \end{cases} . \tag{24}$$

B. Associativity Matrix N

Since the nodes in the network may not be fully connected, some of the neurons are always fixed to zero. Most of the time, the number of neurons which are "off" is very large. Therefore, the computations for those clamped neurons can be avoided, which speeds up the computational process. To implement this clamping technique into the neural network, an associativity matrix $\mathbf{N}$ should be defined [25]. If there are N_N nodes, then there are N_N rows and N_N columns in matrix $\mathbf{N}$, as shown below

$$\mathbf{N} = \begin{bmatrix} n_{11} & n_{12} & \cdots & n_{1N_N} \\ n_{21} & n_{22} & \cdots & n_{2N_N} \\ \vdots & \vdots & \vdots & \vdots \\ n_{N_N1} & n_{N_N2} & \cdots & n_{N_NN_N} \end{bmatrix} \tag{25}$$

where n_{ij} takes on either "0" or "1" according to

$$n_{ij} = \begin{cases} 1 & \text{if there is a link between} \\ & \text{node } i \text{ and node } j \\ 0 & \text{otherwise} \end{cases} . \tag{26}$$

C. Formulation of Energy Function

In this work, the energy function will have the form of the Hopfield energy function [26]. In constrained optimization problems, the energy function has two terms: the cost term and the constraint term. In the problem considered in this paper, the cost term is the total block rate of the network and the constraint term is the penalty imposed to the cost for violating the constraints. Therefore, the energy function can be written as follows:

$$E = \alpha \times \text{"cost"} + \beta \times \text{"constraint1"} + \gamma \times \text{"constraint2"} + \cdots \tag{27}$$

where α, β, and γ are the relative weights.

For illustrative purposes, the energy function for the case where only the reservation parameter is varied is derived below. With additional or different constraints, energy functions for the other cases which can be similarly derived are omitted.

In this case only the reservation parameter is changed. The capacity of each link is fixed at a constant value. The "cost" term in (27) is the total block rate of the network. Combining (3) and (5), an expression for the cost term E_0 is obtained

$$\begin{aligned} E_{DB} &= \sum_r B_{ijr} S_{ijr} \\ E_{AB} &= \prod_{m \in M_{i-j}} E_{AB}^m \\ E_{AB}^m &= 1 - \left(1 - \sum_r B_{imr}^R S_{imr}\right)\left(1 - \sum_r B_{mjr}^R S_{mjr}\right) \\ E_0 &= \frac{1}{\Lambda} \sum_i \sum_j \lambda_{i-j} E_{DB} E_{AB} n_{ij} \end{aligned} \tag{28}$$

where E_{DB} is the energy term corresponding to the direct blocking, E_{AB} the energy term corresponding to the alternate blocking, B_{ijr} the probability that a direct call is blocked in the link (i, j) with r circuits available for alternately routed calls, and B_{ijr}^R the probability that an alternately routed call is blocked in the link (i, j) with r circuits available for alternately routed calls.

Constraint terms of the energy function are defined as follows:

1) Each link is restricted to have only one particular reservation parameter. If more than one reservation parameter is assigned to a link then a penalty term is imposed

$$E_1 = \sum_i \sum_j \sum_r \sum_{r' \neq r} S_{ijr} S_{ijr'}. \tag{29}$$

2) The total number of neurons that are "on" must be equal to the number of links, N_L, in the network.

Thus, this constraint avoids the situation where all neurons are turned "off"

$$E_2 = \left(\sum_i \sum_j \sum_r S_{ijr} - N_L\right)^2. \tag{30}$$

The total energy is the sum of the cost and the constraints, and can be written as

$$E = \alpha \times E_0 + \beta \times E_1 + \gamma \times E_2 \tag{31}$$

where α, β, and γ, are the relative weights.

D. Application of Mean Field Theory

Once the representation scheme of neurons and the energy function are developed, mean field theory can be applied in the map generation process. The relaxation in both simulated annealing and MFA follows a Boltzman distribution [16], which is given by

$$P(S) = \frac{1}{Z} e^{-E(S)/T} \tag{32}$$

where S is any one of the possible configurations specified by the corresponding neuron set, $E(S)$ is the energy of the corresponding configuration, T is the parameter called temperature, and Z is the partition function given by

$$Z = \sum_{(S)} e^{-E(S)/T}. \tag{33}$$

In mean field theory, instead of concerning the neuron variables directly, the means of neurons are considered

$$V_{ijr} = \langle S_{ijr} \rangle. \tag{34}$$

Let the value of neuron S_{ijr} be "1" when the neuron S_{ijr} is on, and let the value of neuron S_{ijr} be "0" when the neuron S_{ijr} is "off." Now (34) becomes

$$\begin{aligned} V_{ijr} &= (1).P(S_{ijr} = 1) + (0).P(S_{ijr} = 0) \\ V_{ijr} &= P(S_{ijr} = 1) \end{aligned} \tag{35}$$

where $P(S_{ijr} = 1)$ and $P(S_{ijr} = 0)$ are the probabilities for $S_{ijr} = 1$ and $S_{ijr} = 0$, respectively, and V is the mean configuration corresponding to S. Thus, in terms of the mean field variable, (32) becomes

$$P(V) = \frac{1}{Z} e^{-E(V)/T}. \tag{36}$$

Now define the local field

$$h_{ijr} = -\frac{\partial E}{\partial S_{ijr}} \tag{37}$$

and

$$S_{ijr} = \begin{cases} 1 & \text{if } h_{ijr} > 0 \\ 0 & \text{if } h_{ijr} < 0 \end{cases}. \tag{38}$$

The probability of a particular neuron S_{ijr} to be "on" and "off" is given by the following two equations:

$$P(S_{ijr} = 1) = \frac{e^{h_{ijr}/T}}{e^{-h_{ijr}/T} + e^{h_{ijr}/T}} \tag{39}$$

$$P(S_{ijr} = 0) = \frac{e^{-h_{ijr}/T}}{e^{-h_{ijr}/T} + e^{h_{ijr}/T}}. \tag{40}$$

Mean field theory approximation is used to approximate the local field h_{ijr} by its thermal average (mean field). Therefore, (39) and (40) become

$$P(S_{ijr} = 1) = \frac{e^{h_{ijr}^{MFT}/T}}{e^{-h_{ijr}^{MFT}/T} + e^{h_{ijr}^{MFT}/T}} \tag{41}$$

$$P(S_{ijr} = 0) = \frac{e^{-h_{ijr}^{MFT}/T}}{e^{-h_{ijr}^{MFT}/T} + e^{h_{ijr}^{MFT}/T}} \tag{42}$$

where

$$\begin{aligned} h_{ijr}^{MFT} &= \langle h_{ijr} \rangle \\ &= \left\langle -\frac{\partial E}{\partial V_{ijr}} \right\rangle. \end{aligned} \tag{43}$$

Now, combining (35), and (41)–(43), an expression for V_{ijr} is obtained as follows:

$$V_{ijr} = \frac{e^{h_{ijr}^{MFT}/T}}{e^{-h_{ijr}^{MFT}/T} + e^{h_{ijr}^{MFT}/T}}, \tag{44}$$

or

$$V_{ijr} = \frac{1}{2}\left[1 + \tanh\left(\frac{h_{ijr}^{MFT}}{2T}\right)\right]. \tag{45}$$

A detailed description and the derivation of mean field equations can be found in [21]–[23]. To apply the mean field equations, thermal average, h_{ijr}^{MFT}, should be evaluated. This is done by first replacing the neuron variable S_{ijr} by the mean of neuron, V_{ijr}, in (28)–(30), as follows:

$$E_{DB} = \sum_r B_{ijr} V_{ijr}$$

$$E_{AB} = \prod_{m \in M_{i-j}} E_{AB}^m$$

$$E_{AB}^m = 1 - \left(1 - \sum_r B_{imr}^R V_{imr}\right)\left(1 - \sum_r B_{mjr}^R V_{mjr}\right)$$

$$E_0 = \frac{1}{\Lambda}\sum_i \sum_j \lambda_{i-j} E_{DB} E_{AB} n_{ij} \tag{46}$$

$$E_1 = \sum_i \sum_j \sum_r \sum_{r' \neq r} V_{ijr} V_{ijr'} \tag{47}$$

$$E_2 = \left(\sum_i \sum_j \sum_r V_{ijr} - N_L\right)^2. \tag{48}$$

Now the thermal average of the local field, h_{ijr}^{MFT}, can be evaluated by taking the partial derivative of the above equations with respect to V_{ijr}, as follows:

$$h_{ijr}^{MFT} = \langle h_{ijr} \rangle = \left\langle -\frac{\partial E}{\partial V_{ijr}} \right\rangle. \tag{49}$$

Thus $\partial E/\partial V_{ijr}$ is expressed as

$$\frac{\partial E}{\partial V_{ijr}} = \alpha \frac{\partial E_0}{\partial V_{ijr}} + \beta \frac{\partial E_1}{\partial V_{ijr}} + \gamma \frac{\partial E_2}{\partial V_{ijr}}. \tag{50}$$

E. Evaluation of Relative Weights

The selection of Lagrange parameters are critical in the annealing process. These parameters must be selected carefully to guarantee convergence. If the parameters are selected poorly, the process may not approach a global minimum, or divergence may occur. To avoid these problems, an approximate method to find these parameters is proposed for the case where only the reservation parameter is varied. A similar approach can be taken to determine the parameters for the other cases.

In (50) the parameter α governs the balance between "cost" and "constraint" terms, and the constants β and γ determine the importance of the constraints. Since only one reservation parameter value should be assigned to a link, energy term E_1 should be weighed heavier than the others. Thus, we have

$$\beta > \alpha, \gamma. \tag{51}$$

To find a relationship between α and γ, assume that constraint (1) is satisfied, (i.e., $E_1 = 0$). This leads to the fact that $\partial E_1/\partial V_{ijr} = 0$. In (50), $\partial E_0/\partial V_{ijr}$ is always positive. But $\partial E_2/\partial V_{ijr}$ may be positive or negative depending on whether the total number of neurons which are "on" is equal to the total number of links in the network. If the total number of neurons having "1" is more than the number of links, then the term $\partial E_2/\partial V_{ijr}$ will be positive; and if the total number of neurons having "1" is less than the number of links, then the term $\partial E_2/\partial V_{ijr}$ will be negative.

Each link must have a specific reservation parameter between zero and the capacity of the link. When all the neurons corresponding to the link are "off" (i.e., no specific reservation parameter is assigned to that link) a neuron must be forced to turn "on." To turn "on" a neuron, $\partial E/\partial V_{ijr} < 0$. This leads to the following:

$$\alpha \frac{\partial E_0}{\partial V_{ijr}} + \gamma \frac{\partial E_3}{\partial V_{ijr}} < 0 \tag{52}$$

$$\alpha\left(\frac{\lambda_{i-j}}{\Lambda} B_{ijr} E_{AB}\right) + 2\gamma\left(\sum_i \sum_j \sum_r V_{ijr} - N_L\right) < 0. \tag{53}$$

Since there are not enough neurons required to be "on"

$$\left(\sum_i \sum_j \sum_r V_{ijr} - N_L\right) < 0. \tag{54}$$

Furthermore, the maximum values of B_{ijr} and E_{AB} are "1" and the fraction λ_{i-j}/Λ will never be more than "1." If only one neuron is needed to be turned "on," then $(\sum_i \sum_j \sum_r V_{ijr} - N_L) = -1$. Using these conditions, the relationship between α and γ can be written as

$$\alpha - 2\gamma < 0. \tag{55}$$

Therefore

$$\gamma > \frac{\alpha}{2}. \tag{56}$$

When more than one specific reservation parameter is assigned to a link, a neuron must be turned "off." To turn "off" a neuron, $\partial E/\partial V_{ijr} > 0$. Then

$$\alpha \frac{\partial E_0}{\partial V_{ijr}} + \gamma \frac{\partial E_3}{\partial V_{ijr}} > 0 \tag{57}$$

$$\alpha\left(\frac{\lambda_{i-j}}{\Lambda} B_{ijr} E_{AB}\right) + 2\gamma\left(\sum_i \sum_j \sum_r V_{ijr} - N_L\right) > 0. \tag{58}$$

Since there are more neurons which are "on" than required

$$\left(\sum_i \sum_j \sum_r V_{ijr} - N_L\right) > 0. \tag{59}$$

Applying similar arguments as before (58) becomes

$$\alpha\left(\frac{\lambda_{i-j}}{\Lambda} B_{ijr} E_{AB}\right) + 2\gamma > 0. \tag{60}$$

Since both terms in the above equation are positive, as long as α and γ are positive the condition is satisfied.

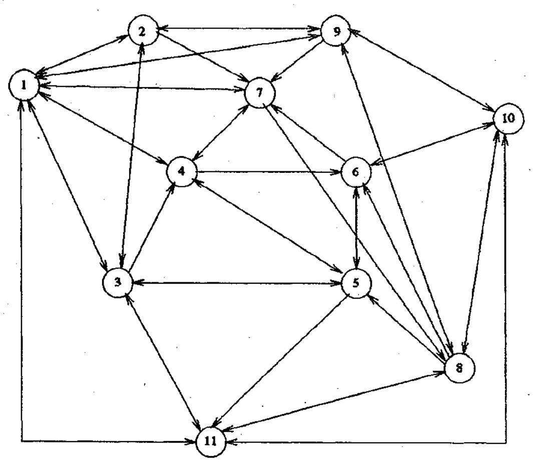


Fig. 3. Network used in simulation.

Node j

Node i	1	2	3	4	5	6	7	8	9	10	11
1	0	1	1	1	0	0	1	0	1	0	1
2	1	0	1	0	0	0	1	0	1	0	0
3	1	1	0	1	1	0	0	0	0	0	1
4	1	0	0	0	1	1	1	0	0	0	0
5	0	0	1	1	0	1	0	0	0	0	1
6	0	0	0	0	1	0	1	0	0	1	0
7	1	0	0	1	0	0	0	1	0	0	0
8	0	0	0	0	1	1	0	0	1	1	1
9	1	1	0	0	0	0	1	1	0	1	0
10	0	0	0	0	0	1	0	1	1	0	1
11	1	0	1	0	0	0	0	1	0	1	0

Fig. 4. Connections between nodes of the network.

As a rule of thumb, by incorporating all these relationships, the following rule is obtained:

$$\beta > \gamma > \frac{\alpha}{2}. \tag{61}$$

F. Cooling Schedule

Similar to simulated annealing, a cooling schedule must be specified. The cooling schedule includes the initial temperature, the stopping criterion, the time spent at each temperature and the temperature updating rule.

1) Initial Temperature: The initial temperature is found by finding a critical point where the energy decreases significantly. Finding this temperature is important to obtain the best map and to avoid unnecessary computations.

2) Stopping Criterion: The annealing process is stopped when the following saturation conditions are met.

A) All neuron values are within the range [0.0, 0.1] or within the range [0.9, 1.0] without any exceptions.

B) When the following criterion is met for Case 1):

$$\frac{\sum_i \sum_j \sum_r (V_{ijr})^2}{N} > 0.95 \tag{62}$$

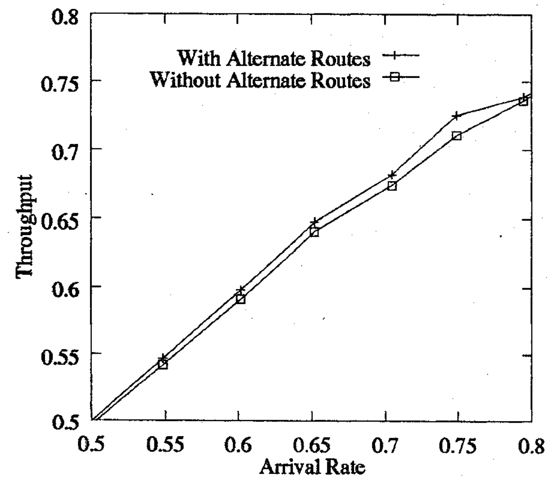


Fig. 5. Routing of calls with and without alternate routes in moderate load.

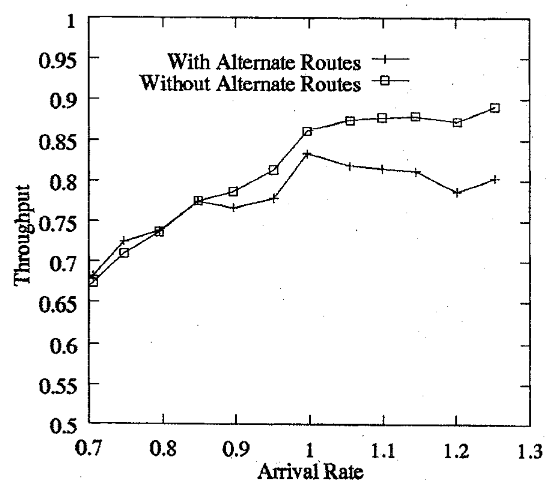


Fig. 6. Routing of calls with and without alternate routes in overload conditions.

where N is the number of neurons that have values within the range [0.9, 1.0].

3) Time Spent at Each Temperature: At each temperature the mean field equations are iterated until the following convergence criterion is met for Case 1):

$$\sum_i \sum_j \sum_r |V_{ijr}(t+1) - V_{ijr}(t)| < 0.001 N_{on} \tag{63}$$

where N_{on} is the number of nonzero neuron elements.

4) Temperature Updating Rule: The following temperature schedule, which is the same as the one used in simulated annealing, is adopted:

$$T = a \frac{C}{\ln(j)} \tag{64}$$

where a is a constant and j is the iteration index.

G. MFA Algorithm

After formulating the energy function, finding a suitable cooling schedule and finding all necessary parameters, the

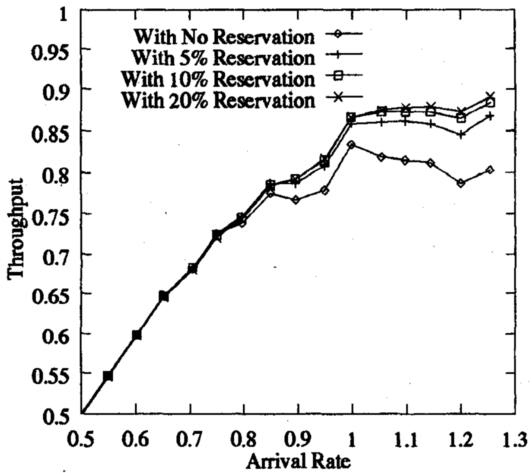


Fig. 7. Network performance with different amount of reservation parameters.

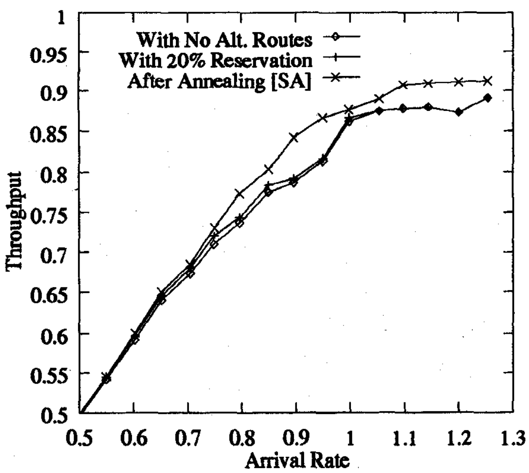


Fig. 8. Network performance after simulated annealing: Varying the reservation parameter only.

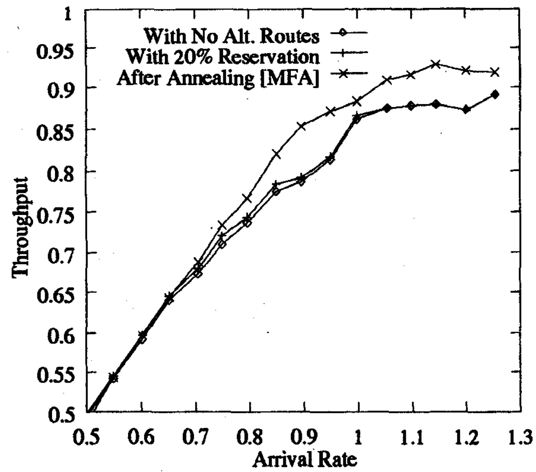


Fig. 9. Network performance after MFA: Varying the reservation parameter only.

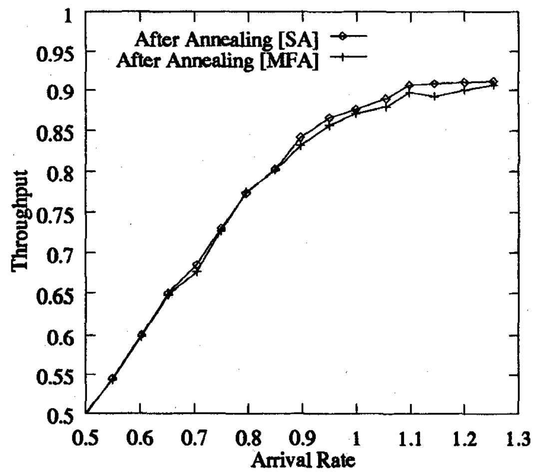


Fig. 10. Comparison of simulated annealing and MFA: Varying the reservation parameter only.

MFA algorithm for minimizing the blocking probability is summarized below.

1) Initialize the neurons with random numbers as follows:

$$V_{ijr} = \text{rand}\,[0,\ 1]n_{ij}. \tag{65}$$

2) Anneal the network until the network is saturated according to the saturation criterion defined before.
3) At each temperature, iterate the MFT equations given below until the convergence criterion is satisfied for Case 1)

$$V_{ijr} = \frac{n_{ij}}{2}\left[1 + \tanh \frac{h_{ijr}^{MFT}}{2T} n_{ij}\right]. \tag{66}$$

VI. Simulations and Discussion of Results

The proposed traffic management scheme is simulated and the results from different experiments are presented in this section. A reasonably sized network (see Fig. 3) having the following properties:

- The network is mesh-connected.
- The network has 11 nodes and 47 links.
- The link capacities may vary from link to link.
- The total capacity of the network is fixed.
- Each O-D pair has a direct link, and alternate routes varying from zero to four possible paths.

The following elements of the simulation are predefined:

- the associativity matrix of the network is shown in Fig. 4, i.e., the connection between nodes of the network (a "1" indicates a link from i to j);
- the O-D pairs and the possible alternate routes for each O-D pair; and
- the arrival rates to each O-D pair.

Various types of experiments are performed on the network to study the proposed scheme. Some of the experiments are listed below.

- network performance with and without alternate routes;
- network performance with a reservation scheme;
- performance of the network after varying only the reservation parameter using simulated annealing and MFA;

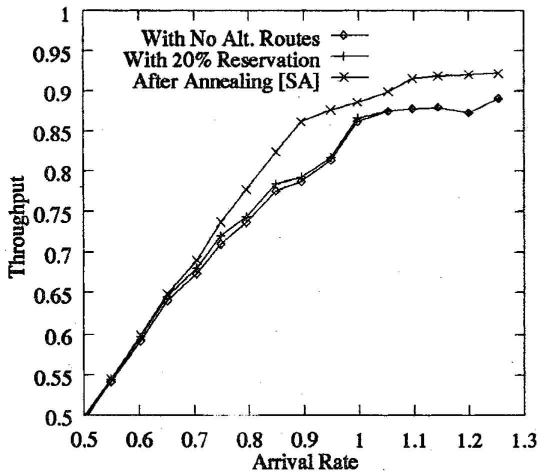


Fig. 11. Network performance after simulated annealing: Varying the link capacity only.

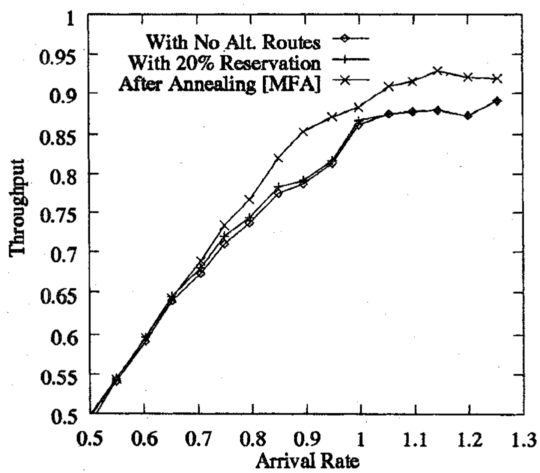


Fig. 12. Network performance after MFA: Varying the link capacity only.

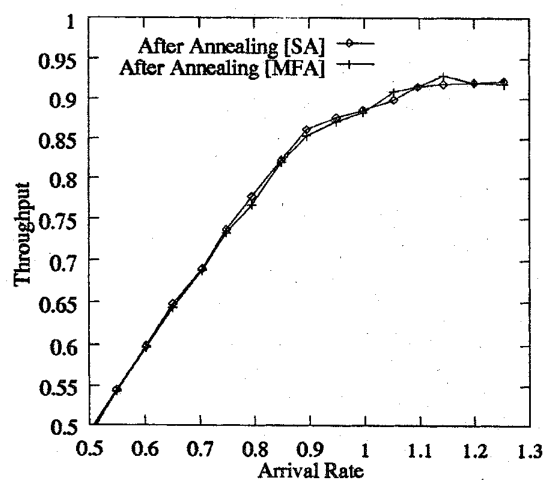


Fig. 13. Network performance after MFA: Varying the link only.

TABLE I
COMUTATION TIMES OF ANNEALING METHODS

Varying Parameter	SA (Time units)	MFA (Time units)
Reservation	264.5	12.3
Capacity	412.4	16.7

TABLE II
PARAMETER VALUES USED IN SIMULATED ANNEALING

Parameter	Varying Reservation	Varying Capacity
a	0.002	0.004
$i_{initial}$	25	35
increment value of i	10	10

- performance of the network after varying only the link capacities using simulated annealing and MFA.

The throughput and arrival rate in the plots shown are normalized to the total capacity C of the network. All measurements of arrival rate and throughput are measured in number of calls per one time unit.

To show the effects of allowing alternate routes, the network is simulated with and without alternate routes under moderate and heavy loads. The results from these simulations are shown in Figs. 5 and 6. Through these simulations, it is verified that at moderate load the network performs better by having alternate routes. At overload conditions, however, the network performance deteriorates and becomes unstable.

To alleviate instability and to improve the performance of the network with alternate routes under heavy and overloaded conditions, the reservation scheme, where a fraction of circuits in the links are reserved only for direct calls, is applied to the network. The network is simulated with some portion of the capacity of each link reserved for direct calls. This is done with a network which has 20 circuits per link and the reservation parameters used are 5, 10, and 20%. The results from this simulation are plotted in Fig. 7. From the results it is evident that near overload conditions make it necessary to impose some reservations to overcome the instability that alternate routing causes.

Unlike the previous simulation, where all the links were assigned the same reservation parameter, in the next simulation the reservation parameter of each link is allowed to vary while keeping the capacity of each link fixed. Both the simulated annealing and MFA methods are tried in optimizing the network performance. When optimization by simulated annealing is performed, initially, 20% of the link capacities are reserved for direct calls. Annealing is done for different arrival rates, and the results from the simulation are plotted in Fig. 8. For different arrival rates, the optimization is also done by MFA. The results from this method are plotted in Fig. 9. To show the advantage of using MFA (Fig. 10), computation time measurements are shown in Table I. The parameters used in the simulations are given in Tables II and III. The measured throughput with different arrival rates shows significant improvement over the previous simulations where no annealing was done. Even though simulated annealing

TABLE III
PARAMETER VALUES USED IN MEAN FIELD ANNEALING

Parameter	Varying Reservation	Varying Capacity
a	0.002	0.004
$i_{initial}$	25	35
increment value of i	10	10
α	0.5	0.5
β	2.0	2.0
γ	0.3	2.0
κ	–	0.4

produces better results than MFA, the computing time required by the MFA is significantly lower than the time required by the simulated annealing.

Similarly, instead of varying the reservation parameter of each link, the capacity of each link is varied and the network parameters are optimized using the two annealing techniques. The constraint here is that the total capacity of the network is fixed. The network is simulated with 20% of the link capacity reserved for direct calls only. The results obtained from these simulations are presented in Figs. 11–13. Again, to show the computational efficiency of MFA, computation time measurements are tabulated in Table I. The parameters used in the simulations are given in Tables II and III. (An additional parameter k is needed for the later case.

VII. CONCLUSIONS

From the simulation results it can be verified that having alternate routes improves the performance of a satellite network at moderate load conditions. When having alternate routes, however, the network becomes unstable as the offered load is increased to a heavy load region, and after a critical point the performance deteriorates. To overcome this undesirable effect, a control scheme is implemented where some portion of the link capacity is reserved for routing direct calls only. The results obtained here show that this is an effective control mechanism in avoiding instability in overloaded traffic conditions. Furthermore, this implementation of the reservation scheme improves the throughput.

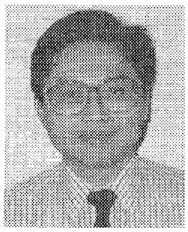
Largely due to the ease of changing the link capacities in a satellite communication network (in contrast to terrestrial networks), the network was able to adapt to the current traffic pattern while routing each call dynamically. The use of simulated annealing and MFA "fine tunes" the network configuration and, thus, improves the performance. The problem with the simulated annealing algorithm is that the annealing process is very time consuming. MFA reduces processing time significantly and produces a configuration which improves performance.

One of the important features of the proposed traffic management scheme is the flexibility with which the factors that affect the network performance can be included. This is done effectively by properly defining the cost function for both the simulated annealing and the MFA algorithms. An added feature of the scheme for improving the throughput of the network is that only external arrival rates are used, and this is easily measured from the network.

The selection of appropriate parameters for simulated annealing and MFA is essential. If the selection is poor the algorithms may diverge and fail to produce an optimal solution. In this paper an approximate method to find the parameters for MFA is analyzed.

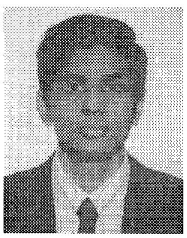
REFERENCES

[1] S. J. Campanella, "Small low cost earth stations, a major trend," *IEEE Spectrum,* vol. 24, pp. 42–43, Jan. 1987.
[2] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis.* Reading, MA: Addison-Wesley, 1987.
[3] G. R. Ash, "Design and control of networks with dynamic nonhierarchical routing," *IEEE Commun. Mag.,* vol. 28, no. 10, pp. 34–40, 1990.
[4] A. Gerard and M. Bell, "Blocking evaluation for networks with residual capacity adaptive routing," *IEEE Trans. Commun.,* vol. 37, pp. 1372–1380, 1989.
[5] A. Gersht and A. Shulman, "Optimal routing in circuit switched communication networks," *IEEE Trans. Commun.,* vol. 37, pp. 1203–1211, 1989.
[6] J. S. Kaufman, "Blocking in shared resource environment," *IEEE Trans. Commun.,* vol. 29, pp. 1474–1481, 1981.
[7] J. M. Aein and O. S. Kosonych, "Satellite capacity allocation," *Proc. IEEE,* vol. 65, no. 3, pp. 332–342, 1977.
[8] J. M. Akinpela, "Overload performance of engineering networks with nonhierarchical and hierarchical routing," *AT&T Bell Labs Tech. J.,* vol. 63, no. 7, pp. 1261–1281, 1984.
[9] D. Mitra, R. J. Gibbens, and B. D. Huang, "State-dependent routing on symmetric loss networks with trunk reservations I," AT&T Bell Lab., Murray Hill, NJ, Tech. Rep., Nov. 1990
[10] S. Haykin, *Neural Networks—A Comprehensive Foundation.* New York: Macmillan, 1994.
[11] S. Y. Kung, *Digital Neural Networks.* Englewood Cliffs, NJ: Prentice-Hall, 1993.
[12] B. Yuhas and N. Ansari, *Neural Networks in TeleComm.* Norwell, MA: Kluwer, 1994.
[13] S. Balasekar and N. Ansari, "Adaptive map configuration and dynamic routing to optimize the performance of a satellite communication network," in *Proc. '93 IEEE GLOBECOM,* Houston, TX, Nov. 29–Dec. 2, 1993, pp. 986–990.
[14] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *J. Chemical Physics,* no. 21, pp. 1087–1091, 1953.
[15] L. Kleinrock, *Queueing Systems, vol. 1.* New York: Wiley, 1975.
[16] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Sci.,* vol. 220, pp. 671–680, 1983.
[17] S. Gemen and D. Gemen, "Stochastic relaxation, Gibbs distributions and Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 6, pp. 721–741, 1984.
[18] B. Gidas, "Nonstationary Markov chains and convergence of the annealing algorithm," *J. Statist. Physics,* no. 39, pp. 73–131, 1985.
[19] S. B. Gelfand, "Analysis of simulated annealing type algorithms," Ph.D. dissertation, Massachusetts Inst. Technol., 1987.
[20] P. J. M. Van Laarhoven and E. H. L. Arts, *Simulated Annealing: Theory and Applications.* Dordrecht, Holland: Reidel, 1987.
[21] C. Peterson, "A mean field theory learning algorithm for neural networks," *Complex Syst.,* vol. 1, pp. 995–1019, 1987.
[22] ——, "Neural networks and NP-complete optimization problem; a performance study on the graph bisection problem," *Complex Syst.,* vol. 2, pp. 59–89, 1988.
[23] C. Peterson and E. Hartman, "Explorations of the mean field theory learning algorithm," *Neural Networks,* vol. 2, pp. 475–494, 1989.
[24] G. Bilbro and R. Mann *et al.,* "Optimization by mean field annealing," *Advances in Neural Information Processing System 1,* D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 91–98.
[25] N. Ansari, E. S. H. Hou, and Y. Yu, "A new method to optimize the satellite broadcasting schedules using the mean field annealing of a Hopfield neural network," *IEEE Trans. Neural Networks,* vol. 6, pp. 470–483, Mar. 1995.
[26] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.,* vol. 52, pp. 141–152, 1985.



Nirwan Ansari (S'83–M'88–SM'94) received the B.S.E.E. degree (summa cum laude) from the New Jersey Institute of Technology (NJIT), Newark, in 1982, the M.S.E.E. degree from the University of Michigan, Ann Arbor, in 1983, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 1988.

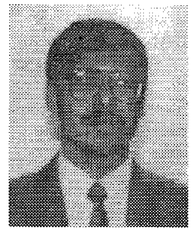
In 1988, he joined the Electrical and Computer Engineering Science Department at NJIT, where he is an Associate Professor and the Assistant Chair for Graduate Studies. His current research interests include neural computing, pattern recognition, data fusion, nonlinear signal processing, ATM networks, and adaptive CDMA detection. He serves as a frequent Referee, a Session Chair, a Session Organizer, and a Technical Representative for various major journals, conferences, and federal agencies. He coedited a book, *Neural Networks in Telecommunications*, (Boston: Kluwer, 1994).



Ambalavanar Arulambalam (S'90) received the B.S. and M.S. degrees in electrical engineering from the New Jersey Institute of Technology, Newark, in 1992 and 1993, respectively. He is currently working toward the Ph.D. degree at the same university.

His research interests include congestion control in ATM local and wide area networks, congestion and flow control, network management and performance evaluation of ATM networks, and the application of neural networks in telecommunications.

Mr. Arulambalam is a member of Tau Beta Pi and Eta Kappa Nu honor societies.



Santhalingam Balasekar (S'87) received the B.S. and M.S. degrees in electrical engineering from Rutgers University, Piscataway, NJ, in 1990, and the New Jersey Institute of Technology (NJIT), Newark, in 1992. He is currently working toward the Ph.D. degree at NJIT.

His research interests include data communication networks and ATM networks and ATM networks.

Mr. Balasekar is a member of Phi Theta Kappa honor society.