Traffic Management of a Satellite Communication Network Using Mean Field Annealing

Ambalavanar Arulambalam and Nirwan Ansari
Center for Communications and Signal Processing Research
Department of Electrical and Computer Engineering
New Jersey Institute of Technology, University Heights, Newark, NJ 07102

Abstract—The performance of non-hierarchical circuit switched networks at moderate load conditions is improved when alternate routes are made available. However, alternate routes 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 that reserves a fraction of the capacity of each link for direct routed calls is used. In this paper, 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 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 a neural network-based optimization technique called mean field annealing. The simulation results show that this method of traffic management performs better than pure dynamic routing with a fixed configuration.

I. Introduction

For circuit-switched networks, it is known that non-hierarchical routing performs better than hierarchical static routing [1]. The dynamic non-hierarchical routing (DNHR) [2] and an adaptive routing algorithm known as dynamic control routing (DCR) [3] are two of the most common examples of non-hierarchical networks. It has been shown that allowing alternate routes in non-hierarchical networks results in improved performance at moderate loads, but it suffers severely at overload conditions. Many control mechanisms have been proposed to overcome this stability problem. A study done by Akinpela [4] suggests that reserving some portion of the capacity at high loads avoids the instability. In a study done by Mitra et. al. [5], optimal trunk reservation is found for a fully connected, completely symmetric network.

In this paper a traffic management scheme is proposed to improve the efficiency of a circuit-switched satellite communication network. The proposed scheme incorporates the idea of dynamically adapt-

II. The Traffic Management Scheme

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.

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. The proposed scheme can be explained with the help of the block diagram shown in Figure 1. The scheme is made up of four functional modules: map generator, router, controller and arbitrator.

A. 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, \( c \) and \( \bar{F} \). Vector \( \bar{F} \) denotes the link capacities of the network, and the elements of vector \( \bar{F} \) 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, mean field annealing is used to find an optimal map which will minimize the total block rate of the network.

B. 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.
• If direct link routing and alternate routing fail, the call is blocked.

C. 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.

D. 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.

III. Analytical Model
A queuing 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 queuing 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.

These assumptions help us study the network based only on the proposed scheme, where only the block rate is used as the cost. A detailed discussion of the validity of these assumptions can be found in [3].

Calculation of The Block Rate
Denote the link from node i to node j by (i, j), and denote the O-D pair from node i to node j by (i−j). The average network blocking probability, B, can be obtained by summing all the call blocking probabilities and normalizing the sum by the total arrivals to the network.

\[ B = \frac{1}{A} \sum_{(i,j)} \lambda_{i,j} B_{i,j}, \]  
(1)

where \( B_{i,j} \) is the probability that call (i−j) is blocked from the network and \( \lambda \), the rate of the total input traffic to the network, is given by

\[ \Lambda = \sum_{(i,j)} \lambda_{i,j}. \]  
(2)

Arrival rates \( \lambda_{i,j} \) are known quantities. Under the previously stated assumption of independent link blocking probabilities, \( B_{i,j} \) can be expressed in terms of \( B_{ij} \) (probability that any call is blocked in \( (i,j) \)) and \( B_{ij}^R \) (probability that an alternate route is blocked in \( (i,j) \)). 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:

\[ B_{i,j} = B_{ij} \prod_{m \in M_{i,j}} [1 - (1 - B_{im}^R)(1 - B_{mj}^R)], \]  
(3)

where \( m \) denotes a tandem node used in an alternate route and \( M_{i,j} \) denotes a set of tandem nodes that forms the alternate routes for \( (i,j) \).

The link blocking probabilities \( B_{ij} \) and \( B_{ij}^R \) in Eq. (3) are derived from the birth-death process of an \( M/M/m/m \) queuing model [6]. The detailed derivation can be found in [7]. The network block rate, \( \bar{B} \), calculated above, is used as the cost function in determining the best map in the map generator module.

IV. Map Generation by Mean Field Annealing
As we know, simulated annealing is a powerful optimization technique to solve combinatorial optimization problems [8], but it is computationally intensive, especially for large problems. As an alternative, mean field annealing (MFA) [9]-[12], which provides a good tradeoff between performance and computational complexity, can be used in minimizing the call blocking probability.

In order to solve the problem by mean field annealing, 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.

3578
In all cases, the total network capacity is fixed to a constant value. In order to represent all these cases, an encoding method to denote the neurons is necessary, and is described below.

**Neuron Encoding**

A neuron is denoted by $S_{ijr}$, $S_{ij}$ or $S_{ijr}$ 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" or "0" according to the following rule:

$$S_{ijr} = \begin{cases} 
1 & \text{if } r \text{ circuits in link } (i,j) \text{ handle alternately routed calls}, \\
0 & \text{otherwise}. 
\end{cases} \quad (4)$$

**Associative 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 associative matrix $N$ should be defined [13]. If there are $N_N$ nodes, then there are $N_N$ rows and $N_N$ columns in matrix $N$, as shown below.

$$N = \begin{bmatrix}
n_{11} & n_{12} & \cdots & n_{1N_N} \\
n_{21} & n_{22} & \cdots & n_{2N_N} \\
& \vdots & \ddots & \vdots \\
n_{N_N1} & n_{N_N2} & \cdots & n_{N_NN_N}
\end{bmatrix}, \quad (5)$$

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

$$n_{ij} = \begin{cases} 
1 & \text{if a link exists between nodes } i \text{ and } j, \\
0 & \text{otherwise.} 
\end{cases} \quad (6)$$

**Formulation of Energy Function**

In constrained optimization problems, the energy function has two terms: the cost term and the constraint term. Here, 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. 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 other cases that 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. Combining Eqns. (1) and (3), an expression for the cost term $E_0$ is obtained:

$$E_{ab} = \prod_{m \in M_{i,j}} E_{ab}^m$$

$$E_{ab}^m = 1 - (1 - \sum_r B_{r,m}^R \sum_r B_{r,m}^R S_{mjr})$$

$$E_0 = \frac{1}{N} \sum_i \sum_j \lambda_{i,j} E_{ab} E_{ab} n_{ij}, \quad (7)$$

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

Constraint terms 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:

$$E_1 = \sum_i \sum_j \sum_r S_{ijr} S_{ijr} \quad (8)$$

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":

$$E_2 = (\sum_i \sum_j \sum_r S_{ijr} - N_L)^2 \quad (9)$$

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

$$E = \alpha \times E_0 + \beta \times E_1 + \gamma \times E_2, \quad (10)$$

where $\alpha$, $\beta$, $\gamma$, are the Lagrange parameters.

**Application of Mean Field Theory**

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

$$V_{ijr} = \langle S_{ijr} \rangle. \quad (11)$$

The relaxation follows a Boltzmann distribution [8]:

$$P(V) = \frac{1}{Z} e^{-\beta V}, \quad (12)$$

Now define the local field,

$$h_{ijr} = -\frac{\partial E}{\partial S_{ijr}}. \quad (13)$$
Mean field theory approximation is used to approximate the local field $h_{ijr}$ by its thermal average (mean field):

$$h_{ijr}^{MFT} = <h_{ijr}> = -\frac{\partial E}{\partial V_{ijr}}.$$  

(14)

Thus, an expression for $V_{ijr}$ can be obtained:

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

or

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

(15)

(16)

A detailed description and the derivation of mean field equations can be found in [9]–[11]. In order 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 Eqns. (7)–(9).

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

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

$$E_{m} = 1 - (1 - \sum_r B_{imr} V_{imr})(1 - \sum_r B_{mjr} V_{mjr})$$

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

$$E_1 = \sum_i \sum_j \sum_{r \neq i} \sum_{r' \neq r} V_{ijr} V_{i'jr}.$$  

(17)

(18)

(19)

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 $V_{ijr}$. Thus, $\frac{\partial E}{\partial V_{ijr}}$ is expressed as:

$$\frac{\partial E}{\partial V_{ijr}} = \frac{\partial E_0}{\partial V_{ijr}} + \frac{\partial E_1}{\partial V_{ijr}} + \frac{\partial E_2}{\partial V_{ijr}}.$$  

(20)

Determination of Lagrange Parameters

The selection of Lagrange parameters are critical in the annealing process. An approximate method to find these Lagrange 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 Eq. (20) the parameter $\alpha$ governs the balance between “cost” and “constraint” terms, and the constants $\beta$ and $\gamma$ 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

$$\beta > \alpha, \gamma.$$  

(21)

In order to find a relationship between $\alpha$ and $\gamma$, assume that constraint (1) is satisfied, (i.e., $E_1 = 0$). This leads to the fact that $\frac{\partial E_0}{\partial V_{ijr}} = 0$. In Eq. (20), $\frac{\partial E_0}{\partial V_{ijr}}$ is always positive. But $\frac{\partial E_2}{\partial V_{ijr}}$ 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 $\frac{\partial E_2}{\partial V_{ijr}}$ will be positive; and if the total number of neurons having “1” is less than the number of links, then the term $\frac{\partial E_2}{\partial V_{ijr}}$ 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.” In order to turn “on” a neuron, $\frac{\partial E}{\partial V_{ijr}} < 0$. This leads to the following:

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

(22)

Since there are not enough neurons required to be “on,”

$$0 < \sum_i \sum_j V_{ijr} - N_L.$$  

(23)

Thus, Eq. (22) becomes:

$$\alpha (\frac{\lambda_{i-j}}{\lambda} B_{ijr} E_{ab}) + 2\gamma < 0.$$  

(24)

Furthermore, the maximum values of $B_{ijr}$ and $E_{ab}$ are “1” and the fraction $\frac{\lambda_{i-j}}{\lambda}$ will never be more than “1.” If only one neuron is needed to be turned “on,” then

$$0 < \sum_i \sum_j V_{ijr} - N_L.$$  

(25)

Using these conditions, the relationship between $\alpha$ and $\gamma$ can be written as:

$$\alpha - 2\gamma < 0,$$

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

(26)

When more than one specific reservation parameter is assigned to a link, a neuron must be turned “off.” In order to turn “off” a neuron, $\frac{\partial E}{\partial V_{ijr}} > 0$. Then,

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

(27)
Since there are more neurons which are "on" than required,

$$\sum_i \sum_j \sum_r V_{ijr} - N_L > 0. \quad (28)$$

Thus, Eq. (27) becomes:

$$\alpha \left( \frac{\lambda_i + \lambda_j}{\Lambda} B_{ijr} E_{AB} \right) + 2\gamma > 0. \quad (29)$$

Since both terms in the above equation are positive, as long as \( \alpha \) and \( \gamma \) are positive the condition is satisfied.

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

$$\beta > \gamma > \frac{\alpha}{2}. \quad (30)$$

Cooling Schedule

The cooling schedule includes the initial temperature, the stopping criterion, the time spent at each temperature and the temperature updating rule.

A. Initial Temperature

The initial temperature is found by finding a critical point where the energy decreases significantly. Finding this temperature is important to avoid unnecessary computations.

B. Stopping Criterion

The annealing process is stopped when the following saturation conditions are met.

1. All neuron values are within the range \([0,0.1]\) or within the range \([0.9,1.0]\) without exceptions.
2. When the following criterion is met:

$$\frac{\sum_i \sum_j \sum_r (V_{ijr})^2}{N} > 0.95, \quad (31)$$

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

C. Time Spent at Each Temperature

At each temperature, the mean field equations are iterated until the following criterion is met:

$$\sum_i \sum_j \sum_r | V_{ijr}(t+1) - V_{ijr}(t) | < 0.001 N_{on}, \quad (32)$$

where \( N_{on} \) is the number of non-zero neuron elements.

D. Temperature Updating Rule

The following temperature schedule is adopted:

$$T = a \frac{C}{\ln(j)}, \quad (33)$$

where \( a \) is a constant and \( j \) is the iteration index.

Mean Field Annealing Algorithm

After formulating the energy function, finding a suitable cooling schedule and finding all necessary parameters, the mean field annealing algorithm for minimizing the blocking probability is summarized below.

1. Initialize the neurons with random numbers as follows:

$$V_{ijr} = \text{rand}(0,1) n_{ij}. \quad (34)$$

2. Anneal the network until the network is saturated according to the saturation criterion defined before.

3. At each temperature, iterate the MPT equations given below until the convergence criterion is satisfied.

$$V_{ijr} = \frac{n_{ij}}{2} [1 + \tanh \left( \frac{h_{MPT} - n_{ij}}{2T} \right)]. \quad (35)$$

V. Simulations and Results

The proposed traffic management scheme is simulated and the results from different experiments are presented. A mesh-connected network having 11 nodes and 47 links is considered. The total capacity of the network is fixed while the capacity among the links are varied according to the experiments. All measurements of arrival rate and throughput are measured in number of calls per one time unit. The throughput and arrival rate in the plots shown are normalized to the total capacity \( C \) of the network.

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

In order 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 Figure 2. From the results it is evident that near overload conditions make it necessary to impose some reservations in order 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. The mean field annealing algorithm is used in optimizing the network performance. For different arrival rates, the optimization is done by mean field annealing, and the results from this method are
plotted in Figure 3. The measured throughput with different arrival rates shows significant improvement over the previous simulations where no annealing was done.

Similarly, instead of varying the reservation parameter of each link, the capacity of each link is varied and the network parameters are optimized using mean field annealing. 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 Figure 4.

References

Figure 1: System Model

Figure 2: Network Performance with Different Amount of Reservation Parameters

Figure 3: Network Performance after Annealing: Varying the Reservation Parameter Only

Figure 4: Network Performance after Annealing: Varying the Link Capacities Only