Learning Telephone Network Trunk Reservation Congestion Control using Neural Networks

R. M. Goodman and B. E. Ambrose

California Institute of Technology
Pasadena, CA91125, USA
`rogo@micro.caltech.edu`

Abstract

The paper reports on the use of a neural network to learn the trunk reservation problem, which is a form of congestion control. The telephone network is symmetric. The error for backpropagation purposes is the network blocking, after simulating light, medium and heavy traffic and appropriately weighing the results. The results indicate that we can be very confident that the neural network can learn the task and that it performs better than schemes with fixed trunk reservations. The results indicate a 13% decrease in blocking for this assumed traffic pattern.

1 Introduction

Congestion control is an important topic in the design of telephone networks and will be even more so in design of ATM networks. In this paper, neural networks are used to decide the level of trunk reservation to apply in congestion situations. The communication networks investigated are symmetric fully connected telephone networks.

2 Trunk Reservation

A simple example of a small network is given in Figure 1. This shows a simple five node network, with nodes labelled A to E. Each node has a set

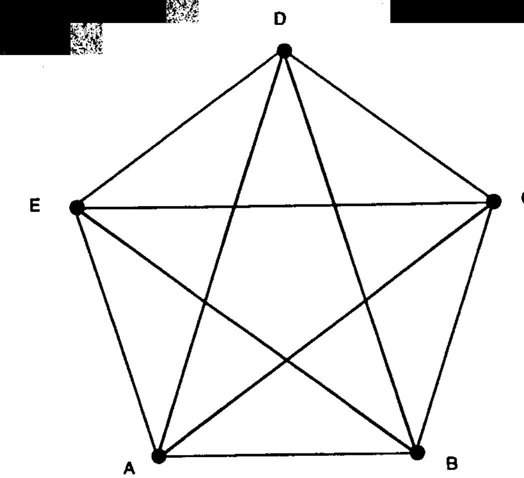

Figure 1: Example of a Symmetric Five Node Network

of links between them. This small network can be used to demonstrate the concepts underlying trunk reservation.

Calls are usually setup using either one or two hops. For example, a call from A to E could be setup using either the trunks from A to E, or else A to B and then B to E. In lightly loaded traffic conditions, using two hops allows more opportunities to complete telephone calls. However in heavily loaded traffic conditions, the two trunk groups used by a two hop call could instead be used to set up two one hop calls and reduce the network blocking.

One hop traffic is called *direct routed* traffic. Two hop traffic is called *alternate routed* traffic.

Trunk reservation is a policy whereby in each trunk group, alternate routed traffic is blocked if there is less than R circuits free on the trunk group. R is known as the trunk reservation parameter. This has the effect of favoring the direct routed traffic at the expense of alternate routed traffic, increasing the network efficiency in time of high load. The parameter R can be very small and still have a substantial effect as is illustrated in Figure 2. A fixed point model [1, 2] was used to generate this plot.

Akinpelu in [2] shows that trunk reservation prevents networks having two stable states at high loads. Kelly in [3] references a result by Hunt and Laws that a policy that chooses the least busy alternative for routing

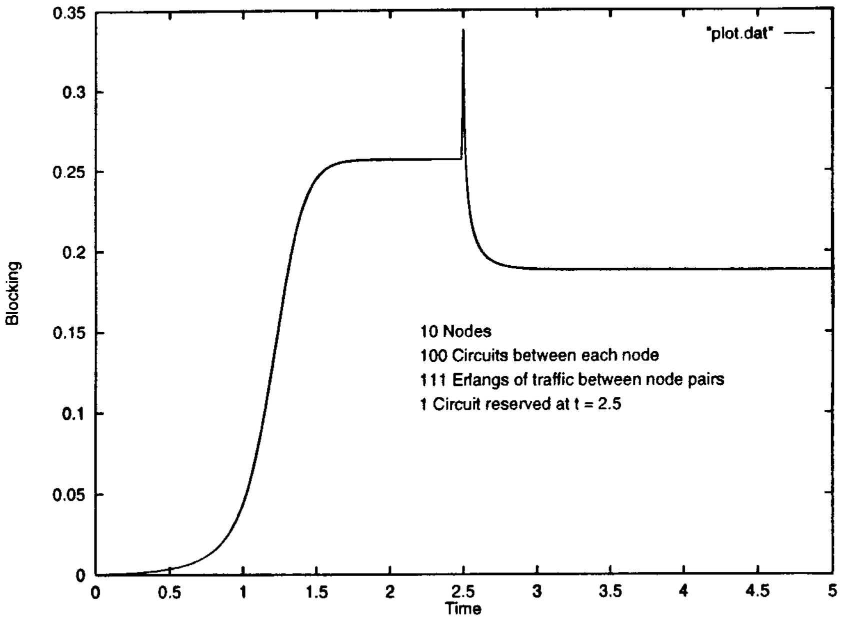

Figure 2: Effect of turning on trunk reservation at $t = 2.5$

and implements trunk reservation is an asymptotically optimal policy in minimizing blocked traffic, as the number of network nodes increases.

Closed form solutions for the correct value of the trunk reservation parameter, R, to use as a function of network load, traffic mix on the trunk group and number of circuits on the trunk group are not known [1]. In this paper, a neural network is used to decide on the value of the trunk reservation parameter as a function of switch load and trunk group traffic mix.

The remainder of the paper talks about the constraints for this problem, training the neural network, the architecture of the neural network, the traffic mix, results and conclusions.

3 Constraints

In a large network, it is unrealistic to expect each node to have a global view of the network. Therefore it was decided to require the inputs to the neural network to be local information, in other words, statistics about the route or statistics about the switch to which the route is attached.

The main benefit of using a neural network on this problem is the ability of network to adjust its weights to reflect changes in the nature of the traffic in the communications network. An unsupervised learning approach was taken. In this case, the neural network is not provided with training examples taken from analytic studies of trunk reservation. Instead simulation is used to provide the training set. The neural network is required to adjust its weights to minimize the overall network blocking in the simulated network.

4 Neural Network

The neural network had two inputs and two hidden units. A linear output unit was used to aid in function fitting. Since good results were obtained with two hidden units, the number of hidden units was not varied.

The inputs were the switch loading and the traffic mix. The *switch loading* was defined to be the number of occupied trunks attached to the switch divided by the total trunks attached to that switch. The *traffic mix* was defined to be the number of direct routed calls on the trunk group divided by the total number of calls on the trunk group. This second parameter was used experimentally. The results indicate it does not have a strong influence on the value of the trunk reservation parameter chosen and could be left out. See Figure 3.

The neural network will output fractional values for trunk reservation. These were interpreted as follows. If the trunk reservation parameter from the neural network was 0.3, then 0.3 of the time a trunk reservation parameter of 1 was used, and the remainder of the time a trunk reservation parameter of 0 was used.

Each simulation cycle represented a 3000 minute or 50 hour traffic simulation. This generated about 18000 test cases and 18000 training cases. Each training case or test case fell into one of the following classes:

- a direct routed call was blocked because the trunk reservation parameter was not high enough. The training data contained a new higher value of trunk reservation.
- an alternate routed call was blocked indicating that the trunk reservation parameter was too high. The training data contained a new lower value of trunk reservation.
- to keep the neural network output steady in regions of the input space that rarely experienced blocking, 1% of the simulated call arrivals were used as training data, keeping the value of the trunk reservation parameter unaltered.

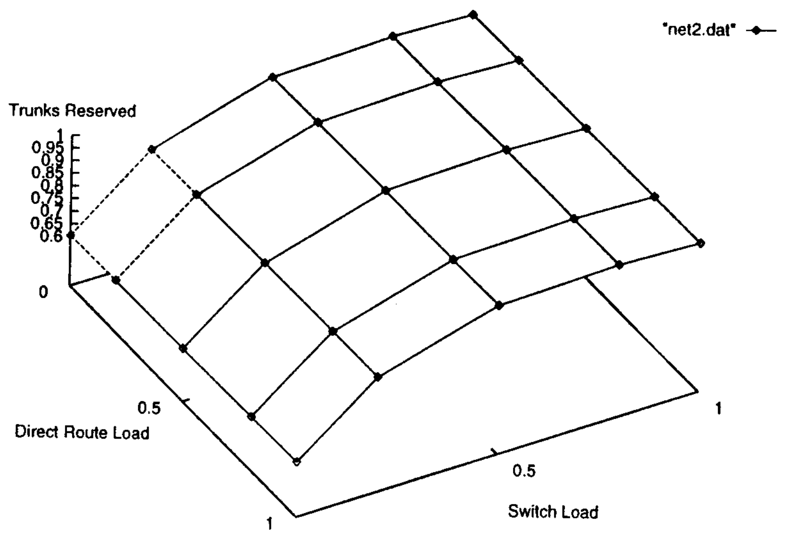

Figure 3: Neural Network Output

Quickprop [4] was then run for 200 iterations to tune the neural network weights. About 10 such simulation cycles were necessary to get good results. See Figure 4.

5 Traffic Mix

For the traffic simulation, a mixture of light (4.4 Erlangs), medium (6.7 Erlangs) and heavy (8.9 Erlangs) traffic was used. The simulated network had 10 nodes and 10 circuits between each node. Because light traffic occurs far more often in real life than medium or heavy, a weight of 0.8 was given to the light, 0.15 to the medium and 0.05 to the heavy traffic for the purpose of computing blocking.

The light, medium or heavy traffic level was indirectly input to the neural network through the switch load input variable. As the traffic level increases, the expected switch load will increase also.

It is important to note that online training in a "live" network could be substituted for the traffic simulations. Once enough new training cases were available, the neural network weights could be tuned. In this way the neural network could continually adjust itself to the traffic mixes found in

	Blocking	±
Neural Network (train)	0.0129164	±0.0000348
Fixed Res of 0 (train)	0.0148592	±0.0000568
Fixed Res of 3 (train)	0.0158059	±0.0000341
Neural Network (test)	0.0129246	±0.0000368
Fixed Res of 0 (test)	0.0148622	±0.0000477
Fixed Res of 3 (test)	0.0157703	±0.0000417

Table 1: Blocking for Neural Network and Fixed Reservation Parameters

the real network.

6 Results

Figure 4 shows how the training and test error decrease with the number of simulation iterations. For comparison, fixed reservation schemes with 3 trunks and 0 trunks reserved are also shown.

The results are given in tabular form in Table 1. The results indicate a 13% decrease in blocking compared to not using any trunk reservation. An even greater decrease is found compared to using a value of 3 for the trunk reservation parameter.

7 Conclusions

In this paper, neural networks have been applied to telephone network congestion relief and done better than the conventional technique of a fixed reservation scheme. In addition, it has been shown that:

- The neural network can be applied to a difficult unsupervised learning problem.
- In the test network, a large decrease in average blocking was seen.
- By substituting live traffic for the simulations in this paper, the neural network can learn in real-time.

It should be possible to extend the results in this paper to networks containing trunk groups of different sizes. This would require an extra input to the neural network containing this information, and longer simulation runs to make sure all parts of the input space were well represented.

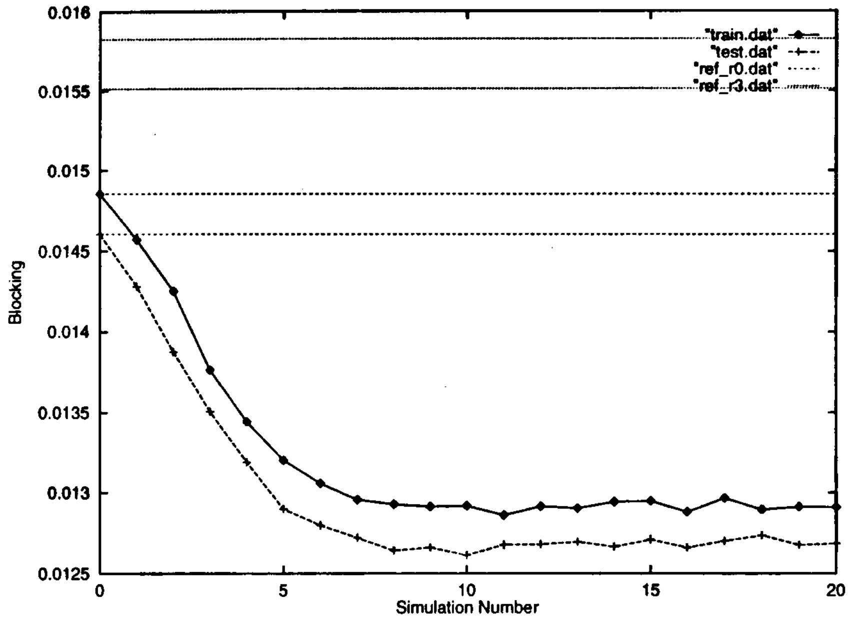

Figure 4: Learning the Trunk Reservation Problem

References

[1] D. Mitra and J. B. Seery, "Comparative evaluations of randomized and dynamic routing strategies for circuit-switched networks," *IEEE Transactions on Communications*, vol. 39, pp. 102–116, Jan. 1991.

[2] J. M. Akinpelu, "The overload performance of engineered networks with nonhierarchical and hierarchical routing," *Bell System Technical Journal*, pp. 1261–1280, Sept. 1984.

[3] F. P. Kelly, "Bounds on the performance of dynamic routing schemes for highly connected networks," *Mathematics of Operations Research*, vol. 19, pp. 1–21, Feb. 1994.

[4] S. E. Fahlman, "Faster-learning variations on back-propagation: An empirical study," in *1988 Connectionist Models Summer School*, Morgan Kaufmann, 1988.