SOLVING BLOCKING PROBLEMS IN HYBRID NETWORKS

Arik Kashper

AT&T Bell Laboratories, Holmdel, U.S.A.

Abstract

The recent deployment of a new sophisticated approach to call routing, known as Dynamic Nonhierarchical Routing (DNHR), drastically changed the traditional hierarchical structure of AT&T's toll network and created the need for new traffic network planning methodologies. In this paper, we focus on major aspects of trunk servicing methodology for the new hybrid network environment. In particular, we describe a two-stage optimization procedure that solves trunk group and point-to-point blocking problems in hierarchical and DNHR parts of the network, respectively, in an economical manner.

1. Introduction

1.1 New Network Structure

With the deployment of the Dynamic Nonhierarchical Routing (DNHR) network, the architecture of the AT&T Communications toll network has changed from its traditional multilevel hierarchy to a hybrid network environment that contains DNHR switching nodes replacing upper levels of the hierarchy and hierarchical switching nodes at the lower levels[1]. The work by Ash, et al, [2] demonstrated that DNHR represents a more flexible and efficient routing scheme that takes full advantage of the expanding Common Channel Interoffice Signaling (CCIS) network and the increased intelligence of the 4ESS electronic switching systems. The initial DNHR cutover, in July 1984, included all 10 regional centers, four sectional centers, and two primary centers. By 1987, all 93 of the existing 4ESS switches in the AT&T Communications network will be converted to DNHR switching nodes[1].

At the 10th International Teletraffic Congress held in Montreal in 1983, papers by Ash, et al, [2], Field, [6], Haenschke, et al, [7], and David, et al, [5], described different aspects of the DNHR feasibility, operations, and economic benefits. It was shown that the DNHR method - two-link dynamic routing with crankback - leads to economically attractive, high performance intercity or metropolitan networks and necessitates the development of new more centralized and automated operations support systems.

1.2 Trunk Network Administration

Trunk network administration in hierarchical and DNHR networks is composed of two major functions: trunk forecasting and trunk servicing. Trunk forecasting at AT&T Communications network is a biannual process that projects future demands and converts them into economical network designs for several future years. These network designs specify the multiyear schedule of trunk augments/disconnects and corresponding routing changes.

Because of the probabilistic nature of the demand projection, the forecasted network may not provide the desired level of blocking performance for actual traffic loads. Accordingly, a servicing system complements the forecasting process to restore service degradation due to forecast errors [8].

Although trunk forecasting and servicing functions in the hierarchical and DNHR portions of the network are identical, the definitions of acceptable service and the methods to achieve it are quite different. The hierarchical network is forecasted and serviced to guarantee a certain average blocking on final trunk groups, while the DNHR network is forecasted and serviced to achieve a certain level of point-to-point blocking between DNHR switches.

1.3 Overview

For the past several years there has been a large effort to improve the methodology of the trunk network administration process. Most of the papers dealt with the trunk forecasting techniques and assumed that all the switching nodes in the network are either hierarchical or DNHR [3,4,9,15]. In contrast, in this paper, we concentrate on several aspects of trunk servicing methodology for the new, hybrid network environment.

More specifically, we will outline a two-stage optimization procedure that combines trunk servicing measures in both parts of the network to restore a desired level of blocking on final trunk groups in the hierarchy and a desired point-to-point blocking level in the DNHR network. We will start by examining the impact of servicing actions in one part of the network on traffic loads offered to the other part. Then, we will formulate a new trunk servicing proposal focusing on several improvements in the hierarchical servicing methodology and the use of trunk reservation in servicing the DNHR part of the network.

2. Trunk Servicing and Network Interactions

Servicing systems collect and analyze network measurements on a regular basis to detect blocking problems and, when they exist, develop corrective actions to achieve the desired blocking level in a timely and cost-effective manner. In the hierarchical part of the network, servicing actions are limited to trunk group augmentations [15]. In the DNHR part of the network, however, we can take advantage of the new capabilities of the 4ESS switch and attempt to solve point-to-point blocking problems by changing routing and introducing trunk reservation for problem parcels.

As is illustrated in Figure 1, because the DNHR point-to-point offered loads include overflows from the subtending hierarchy, trunk augmentations in the hierarchy may affect the offered loads and, consequently, the blocking performance of the DNHR part of the network. Similarly, the DNHR servicing measures will change the DNHR carried loads and, therefore, will affect loads offered to the subtending hierarchical switching offices.
In the i-th day of a study period that normally contains 4 business weeks, corresponding change in overflow to the DNHR network. During the second stage of our servicing procedure, we will identify blocking problems in the hierarchical part of the network, but, in most cases, will not result in a significant increase in the carried load between the two DNHR switches. Consequently, to avoid overtrunking, our servicing procedure will solve the final group blocking problem. Note that the current servicing action may affect the cost of planned trunk provisioning activities in the future.

3.2 Cluster Optimization

When a statistically significant service problem on the final group has been detected, the group's high blocking can be decreased by adding capacity to the final group and/or reducing the overflow by augmenting subtending high usage groups. In [12], a heuristic servicing algorithm was proposed that achieves the desired blocking level by finding undersized high-usage trunk groups, augmenting them, and, if the problem still exists, resizing the final group.

The algorithm in [12] was developed prior to the introduction of new minimum-cost multiyear trunk forecasting process, which we described at the Tenth ITC [9]. Consequently, in this section we will develop a mathematical model that reflects the new trunk implementation realities and present an algorithm that minimizes the immediate and future costs of solving the final group blocking problem. Note that the current servicing action may affect the cost of planned trunk provisioning activities in the future.

3.2.1 Notation

First, we will focus on all the high-usage groups in the network cluster. Note that a network cluster consists of all high-usage groups which originate at a common switching office and overflow to a common final.

To formulate our optimization model for augmenting high-usage groups we need to introduce the following notation:

- \( T_j(k) \) - number of trunks in service on the subtending group \( j \) at forecast period \( k \)
- \( u_j(k) \) - number of planned trunk augmentations/disconnects on the subtending group \( j \) at forecast period \( k \)
- \( d_j(k) \) - the demand in trunks on the subtending group \( j \) at forecast period \( k \) as specified by the trunk forecasting process [9]
- \( \beta_j \) - additional number of trunks on the subtending group \( j \) that compensates for one final trunk
- \( \delta_j \) - maximum reduction in the final trunk requirement that can be obtained by augmenting trunk group \( j \)
- \( z(x) \) - deficit in final trunks that we will cover by servicing the subtending high-usage groups
- \( z_j \) - portion of the deficit (number of final trunks) covered by servicing subtending high-usage group \( j \)
- \( M \) - total number of subtending high-usage

Our analysis revealed, however, that in the range of engineering interest, the impact of the DNHR servicing on the hierarchical part of the network is not significant and may be ignored. Intuitively, this is explained by the fact that a small increase in the carried load between the two DNHR switches generally translates into an increase in the load offered to all subtending final groups and, therefore, does not cause a severe service degradation for a particular final group.

Analogously, servicing final groups in the hierarchy may increase point-to-point loads offered to the DNHR part of the network, but, in most cases, will not result in a significant blocking problem for an individual DNHR point-to-point pair. However, if we augment the high-usage group A-C, for example, we will reduce the offered load to the point-to-point pair B-C only. Thus, by augmenting high usage group A-C we may be also solving the DNHR B-C blocking problem.

Consequently, to avoid overtrunking, our servicing procedure will identify blocking problems in the hierarchical part of the network, first, solve them, and then calculate the corresponding change in overflow to the DNHR network. During the second stage of our servicing procedure, we will reevaluate and solve remaining point-to-point blocking problems in the DNHR part of the network.

3. Hierarchical Trunk Servicing

3.1 Background

In the hierarchical network environment, the demand servicing process is based on trunk group measurements collected by the trunk servicing system. To monitor network blocking performance, the statistic \( B \) is computed for the busy hour, where \( B \) represents the fraction of calls blocked in the i-th day of a study period that normally contains 4 consecutive business weeks.

Because of the statistical nature of the demand and relatively small size of the blocking sample, the statistic \( B \) computed on

**FIGURE 1 HYBRID NETWORK**

NOTATION

- DNHR SWITCHING OFFICES
- HIERARCHICAL SWITCHING OFFICES

Our analysis revealed, however, that in the range of engineering interest, the impact of the DNHR servicing on the hierarchical part of the network is not significant and may be ignored. Intuitively, this is explained by the fact that a small increase in the carried load between the two DNHR switches generally translates into an increase in the load offered to all subtending final groups and, therefore, does not cause a severe service degradation for a particular final group.

Analogously, servicing final groups in the hierarchy may increase point-to-point loads offered to the DNHR part of the network, but, in most cases, will not result in a significant blocking problem for an individual DNHR point-to-point pair. However, if we augment the high-usage group A-C, for example, we will reduce the offered load to the point-to-point pair B-C only. Thus, by augmenting high usage group A-C we may be also solving the DNHR B-C blocking problem.

Consequently, to avoid overtrunking, our servicing procedure will identify blocking problems in the hierarchical part of the network, first, solve them, and then calculate the corresponding change in overflow to the DNHR network. During the second stage of our servicing procedure, we will reevaluate and solve remaining point-to-point blocking problems in the DNHR part of the network.

3. Hierarchical Trunk Servicing

3.1 Background

In the hierarchical network environment, the demand servicing process is based on trunk group measurements collected by the trunk servicing system. To monitor network blocking performance, the statistic \( B \) is computed for the busy hour, where \( B \) represents the fraction of calls blocked in the i-th day of a study period that normally contains 4 consecutive business weeks.

Because of the statistical nature of the demand and relatively small size of the blocking sample, the statistic \( B \) computed on
groups in the network cluster

\[ N \] - number of years in the planning horizon

\[ \rho \] - discount factor that measures the worth of the next year's dollars in terms of present dollars.

\[ c^2_t(u(k)) \] - capital cost defined on Figure 2. On Figure 2, \( c^2_t \) and \( b^2_t \) represent the per-trunk capital cost and salvage value at forecast period \( k \), respectively.

\[ c^3_t(u(k)) \] - labor cost defined on Figure 2. On Figure 2, \( a^3_t \) and \( b^3_t \) represent the labor costs of connecting and disconnecting one trunk at forecast period \( k \).

\[ c^4_t(T(k)+u(k)) \] - maintenance cost defined on Figure 2. On Figure 2, \( a^4_t \) represents the per-trunk maintenance cost.

Note that \( T(0) \) represents the number of trunks currently in service, and \( u(0) \) represents the servicing action to be determined.

![Figure 2: Trunk Provisioning Costs](image)

3.2.2 Mathematical Model

Our goal is to find a servicing policy for the network cluster that will minimize the cost of all trunk rearrangements over the planning horizon

\[
\min J(x) = F(x) + L(x),
\]

where \( x \) is the number of trunks added to the final. \( F(x) \) and \( L(x) \) are the present worth of all trunk provisioning costs over the planning horizon for the final group and for all subtending high-usage groups respectively.

We would like to note that the method for computing \( F(x) \) is provided in [10]. Thus, we can devote our attention to calculating \( L(x) \). In particular, we need to find how to cover the remaining deficit in final trunks, \( z(x) \), most economically by augmenting the subtending high-usage groups

\[
\min L(x) = \min \sum_{j=1}^{M} L_j(z_j)
\]

We need to minimize \( L(x) \) with respect to all present and future rearrangements \( u_j(k) \) subject to the following constraints:

1. The total trunk requirement must be sufficient to solve the final blocking problem

\[
u_j(0) \geq \beta_j z_j, \quad \text{and}\]

\[
\sum_{j=1}^{M} z_j = z(x).
\]

2. The number of trunks in service on the high-usage group \( j \) at forecast period \( k \) must be greater or equal to the originally forecasted trunk requirement

\[
T_j(k) + u_j(k) \geq d_j(k), \quad \text{for} \quad k=1,2,..,N,
\]

where \( T_j(k) \) is defined by the recursion

\[
T_j(k+1) = T_j(k) + u_j(k).
\]

3. The unknown variables \( z_j \) must satisfy feasibility constraints

\[
0 \leq z_j \leq \delta_j.
\]

3.2.3 Servicing Solution

The solution to the optimization problem (1)-(4) exploits the multiyear forecasting algorithm for high-usage groups which we derived in [10]. Indeed, when \( z_j \) are fixed, the objective function (1) can be decomposed as follows:

\[
\min L(x) = \sum_{j=1}^{M} \min L_j(z_j),
\]

where \( L_j(z_j) \) represents the present worth of trunk provisioning costs at the subtending group \( j \). Thus, we arrive at a single high-usage group capacity expansion problem: minimize \( L_j(z_j) \) subject to constraints

\[
T_j(k) + u_j(k) \geq d_j(k), \quad \text{for} \quad k=0,1,..,N,
\]

where \( d_j(0) = T_j(0) + \beta_j z_j \).

Therefore, the original trunk servicing problem described by (1)-(4) can be reformulated as follows: find

\[
\min \sum_{j=1}^{M} L_j(z_j)
\]

while solving the final blocking problem

\[
\sum_{j=1}^{M} z_j = z(x),
\]

where \( z_j \) satisfy feasibility constraints, and \( L_j(z_j) = \min L_j(z_j) \) is computed by a simple, efficient algorithm which is based on the following result.

Under quite general assumptions on the cost parameters, we have shown [10] that the optimal policy, \( u_j^*(k) \), has the form
4.1 Background

4.1.1 DNHR Overview

As we have shown in Figure 1, calls overflowing the hierarchical part of the network will enter the DNHR part at a unique entry point and will leave the DNHR part at a unique exit point. Within the DNHR network a call is routed according to a specified sequence of paths. A path may contain either one or two links. If the second link in a two-link connection is busy, the via switch sends a "crankback" CCIS message to the DNHR entry node and then the next path in the sequence is tried. The dynamic nature of DNHR is accomplished by changing the preplanned sequence of paths up to ten times during the day. The ability to adjust the routing allows us to benefit from the noncoincidence of traffic loads and leads to a more efficient network design [3].

4.1.2 Servicing Function

To monitor point-to-point blocking in the DNHR environment, 4ESS switching offices collect point-to-point traffic data. The point-to-point blocking performance is analyzed weekly by the DNHR servicing system to determine which parcels are not receiving adequate service. Similarly to the hierarchical environment, the need for demand servicing is identified if the service violation is statistically significant. If servicing is needed, the DNHR servicing system suggests a combination of trunk augments and routing changes to restore the desired level of point-to-point blocking. In [4], the initial DNHR servicing algorithm was proposed. The algorithm is based on "flow optimization," "engineering," and "blocking correction" routines [3] and results in a servicing solution that includes both trunk augmentations and routing changes.

It is important to note that in practice trunk augmentations on the demand servicing basis are often expensive and can be implemented only with a minimum of 2 weeks delay. Moreover, there is always a reluctance to add trunks at the end of the busy season. The delay in servicing may result in significant revenue loss as well as customer aggravation. In the DNHR environment, however, we can improve network blocking performance in a timely manner by avoiding trunk augmentations to the extent possible.

There are two major ways to achieve this goal: change traffic routings to maximize the use of existing capacities and introduce trunk reservation to provide better service for the high blocking point-to-point pairs. The problem of minimizing the maximum point-to-point blocking in the network by routing changes only was considered by Nassar, [11]. She showed that the initial servicing algorithm [4] does not lead to a routing-only solution of blocking problems and developed a new, heuristic procedure that adjusts the engineered network routing for the difference in forecasted and actual loads. Our task in the section 4.2 will be the investigation of the use of trunk reservation in DNHR servicing.

4.2 The Use of Trunk Reservation

4.2.1 Trunk Reservation Background

Consider a trunk group that handles high-priority and low-priority traffic streams. A high-priority call will be cleared if on arrival there are no idle trunks in the group, whereas a low-priority call will be cleared if on arrival there are less than R+1 idle trunks in the group. This service discipline is called Trunk Reservation (TR) and R is called the reservation level.

Trunk reservation is known as an effective network management technique that preserves the performance of either the hierarchical or DNHR networks under overload conditions [7,14]. We will show that the TR method can be also used to solve some of the DNHR point-to-point blocking problems.

4.2.2 Parcel Blocking with TR

To apply the TR method in servicing, we need to know how to compute the blockings for high and low-priority traffic streams when TR is employed. Under the reasonable assumption that these streams can be modeled by the Interrupted Poisson Processes, the blockings were derived analytically by Songhurst [14]. To simplify the computations, we present less accurate but simpler approximation for the blockings that exploits the decomposition principle introduced in [13].

We assume that the high-priority traffic stream is described by the pair \((a_i, z_i)\), where \(a_i\) is the offered load and \(z_i\) is the peakedness factor \((z_i > 1)\); the low-priority traffic stream is random (Poisson) with intensity \(a_1\) (see Figure 3). Applying the decomposition principle, we split the peaked traffic stream
into a Poisson component $a_2 z_1$ and a zero-variance component $(a_2 z_1 - a_2, 0)$. This is equivalent to the substitution

\[
(a_1, v_1) \sim (v_1, v_1) - (v_1 - a_1, 0)
\]

where $v_1 = a_1 z_1$.

\[
(a_1, z_1) - \text{HIGH PRIORITY PEAKED TRAFFIC} \quad \approx \quad (a_1, z_1) - \text{LOW PRIORITY POISSON TRAFFIC} \quad a_1 z_1 - \text{REMAINING HIGH PRIORITY POISSON TRAFFIC OFFERED TO AN "EQUIVALENT" TRUNK GROUP}
\]

**FIGURE 3 DECOMPOSITION PRINCIPLE**

Dedicating $a_1(z_1-1)$ trunks to carry the zero-variance stream component, we arrive at the congestion model of Figure 3. Now, we have two random streams with parameters $a_2$ and $a_1$ offered to a trunk group of $S = N + a_1(z_1-1)$ trunks. In this case, the probability of blocking for the high and low-priority streams is given by

\[
b_1 = \frac{(a_2 z_1 + a_2)^{s-R}}{S!} P_s, \quad \text{and} \quad b_2 = P_s \left( a_2 z_1 + a_2 \right)^{s-R} \sum_{j=0}^{s-R} \frac{(a_2 z_1)^j}{j!},
\]

where $P_s$ is defined by

\[
P_0 = \left( \frac{\left( \sum_{j=0}^{s-R} \frac{(a_2 z_1-a_1)^j}{j!} \right) + (a_2 z_1+a_2)^{s-R} \sum_{j=0}^{s-R} \frac{(a_2 z_1)^j}{j!}}{j!} \right)^{-1}.
\]

Our numerical experience shows that for small values of $R$ these formulas provide a satisfactory approximation for parcel blockings. Thus, we can now estimate point-to-point blocking changes when TR is used on one of the trunk groups. Note that the DNHR point-to-point blocking probability is computed by the product of individual path blockings.

**4.2.3 Solving Blocking Problems**

Now we will outline an heuristic procedure that attempts to minimize the maximum point-to-point blocking in the DNHR network by introducing TR for problem pairs. To preserve high utilization of all DNHR trunk groups we will consider only three trunk reservation levels $R = 1, 2, 3$. These levels proved to be sufficient to significantly change the blocking for problem parcels.

To specify how to use TR in servicing, we use the following iterative procedure:

(i) Assign (upgrade, if necessary) the level of acceptable point-to-point blocking.

(ii) Compute point-to-point blockings for all node pairs and identify the most severe servicing problem.

(iii) Identify trunk groups on which TR should be applied. Determine the trunk group and the reservation level $R$ to obtain the maximum reduction in the point-to-point blocking for the chosen node pair subject to the constraint: point-to-point blocking probabilities for all other pairs should remain less than the level of acceptable blocking defined in (i).

In practical situations, usually, there are many point-to-point pairs with virtually zero blocking probability and few high blocking pairs. According to our numerical experience, in these cases, the TR method can be effectively utilized to solve the DNHR point-to-point blocking problems. In contrast to network management applications, our iterative procedure frequently results in introducing TR to protect overflow rather than direct traffic. On the last choice via-path, for example, where the contribution of the problem parcel is small relative to the other traffic, the application of TR tends to be feasible and beneficial.

**4.2.4 DNHR Servicing Overview**

To summarize our discussion, in this section we formulate the major steps in the DNHR servicing procedure.

**Identify DNHR servicing problems.**

We use switch data to detect point-to-point service violations in the DNHR part of the network. We adjust the violations for the effect of the hierarchical servicing. If the resulting blocking problems are statistically significant, the corresponding point-to-point pairs become candidates for servicing.

**Relieve blocking problems by routing changes**

To reduce the number of blocking problems, we use the procedure described in [11] to change the traffic routing. To minimize the cost of servicing and to avoid the delay, trunk augmentations are not allowed at this step of the servicing procedure.

**Improve network blocking performance by trunk reservation.**

For a problem pair, we determine on which trunk groups TR control should be imposed. These trunk groups are selected so as to reduce the blocking for the problem pair with no significant service degradation for other traffic.

**Solve blocking problems by trunk augmentation**

Finally, as the last resort, trunk group additions [34] can be used to bring the network blocking performance to the desired level.

**REFERENCES**


