AND - ATM Network Design System

Charles A. Brooks, Shuyue Wei, Hong Li, and Cyrus J. Walker
Department of Electrical Engineering and Computer Science
University of Illinois at Chicago
cbrooks@eecs.uic.edu

Abstract

In this paper Augmented Infinitesimal Perturbation Analysis (APA) is used for gradient estimation in a hybrid simulation/analytic system for computer communications network capacity planning called ATM Network Design (AND) System. Usage of APA provides an alternative to traditional network planning methods that assume independence of packet service times. A steepest descent algorithm determines locally optimal minimum average network delays based on these APA estimates. The algorithm employs an Armijo line search. In addition a linear, separable capacity constraint is assumed. It is assumed that input regulators (e.g., leaky bucket regulators) are used at the source of each virtual circuit. All virtual circuit external arrivals are modeled as independent Poisson processes. Regenerative simulation is used.

1 Introduction

Capacity planning for packet-based communication networks is a well-developed research area. For expediency we will not attempt to cover all prior and current aspects of this research. Before explaining the contribution of the work presented in this paper, we will discuss the goals of capacity planning and review a sampling of pertinent literature in this area.

Methods for capacity planning have been of interest to managers of point-to-point communication networks (such as the ARPANET) since their inceptions in the late 1960s and early 1970s. The massive proliferation of computers over the past two decades, and the recent advent of the Information Age have led to a very rapid increase in the number of users on these networks. As a result of this and the ever increasing traffic demands of newly introduced applications (e.g., real-time multimedia applications) the traffic load presented to computer communication networks is increasing at a even greater pace than that of the population of users. Although advanced fiber-optic technology has provided an increase in network capacity by several orders of magnitude, the corresponding increase in traffic offerred to these networks continues to make capacity planning an active research area.

Essentially capacity planning is the determination of link capacities for point-to-point networks. This determination is needed in all stages of network design, development, implementation, usage, and management. That is, it is of importance in the initial planning and design of a network as well as in the continued ongoing maintenance of a network.

Many factors contribute to the delay a packet encounters while traversing a network. Depending on the physical topology of the network, the reception of a packet at its destination may be delayed due to processing times, transmission delays, propagation delays, and/or queueing delays. The latter is usually the result of resource contention, which is characteristic of the resource sharing frequently used in computer networks.

In the performance analysis of computer networks, an assumption frequently used is that packet transmission times are independent. This assumption (aka the "independence assumption") along with other weaker assumptions admit a closed-form analytical expression for the average delay. These assumptions, which are presented in Section 2, allow the application of queueing-theoretic product-form results for delay analysis [15].

Product-form results were applied in early fundamental research on network planning by Fratta, Gerla, Kleinrock and others (cf. [8]). These results also appear in later work on network planning based on combinatorial optimization (e.g., see [7]).

As shown by Brooks, Xiao, Varaiya, and others [17, 3], Perturbation Analysis [10] provides better accuracy in delay prediction than methods using product-form results. The work presented here is an extension of earlier applications of Discrete Event System [5] methodologies to communication

0-8186-7758-9/97 $10.00 © 1997 IEEE

networks by Cassandras (cf. [13]).

In subsequent sections of this paper a number of topics are discussed. Section 2 defines the Capacity Assignment problem and presents Kleinrock's solution as well as the AND solution. Section 3 describes AND. In section 4 results are presented for a simple 16-node ATM network. Section 5 describes the algorithm used. Conclusions and suggestions for future work are given in Section 6.

2 The Capacity Assignment Problem

The Capacity Assignment (CA) problem was first formulated in the 1970s [12]. The problem is defined below:

CA subproblem

Given: Flows, network topology, and

source-destination traffic

Minimize: Average end-to-end packet delay

With respect to: Capacities

Under constraint: Total link cost $\leq D$

2.1 Kleinrock's Solution

To solve the CA problem, Kleinrock [12] formulated the average delay in the network as

$$T = \frac{1}{\gamma}\sum_{i=1}^{M}\frac{\lambda_i}{\mu C_i - \lambda_i} \tag{1}$$

where, $T =$ the average delay (in seconds/packet), $\gamma =$ the total external traffic entering the network (in packets/sec), $M =$ the total number of links in the network, $\mu =$ the inverse of the average packet length (in packets/bit), $\lambda_i =$ the packet flow through link i (in packets/sec), and $C_i =$ the capacity of link i (in bits/sec).

In deriving this formula for the average delay, all external traffic sources are modeled as independent Poisson processes and the "independence assumption" [11] is made: each time a packet is received at a node within the network, a new packet length is independently generated from an exponential distribution with rate μ. Henceforth, we shall refer to Equation 1 as the product-form solution.

The total cost, D, is separable, i.e.,$D = \sum_{i=1}^{M} d_i(C_i)$, where $d_i(\cdot)$ is a function only of capacity C_i, independent of the traffic on the link.

To solve the CA subproblem Lagrangean relaxation is used. The optimal capacities are found for a variety of cost functions. For a linear cost function, $d_i(C_i) = d_i C_i$, the optimal solution to the CA subproblem is

$$C_i = \frac{\lambda_i}{\mu} + \left(\frac{D_e}{d_i}\right)\frac{\sqrt{\lambda_i d_i}}{\sum_{j=1}^{M}\sqrt{\lambda_j d_j}}, i = 1,\ldots,M, \tag{2}$$

where

$$D_e \doteq D - \sum_{i=1}^{M}\frac{\lambda_i d_i}{\mu}, \tag{3}$$

is the "excess dollars." Note that the excess dollars are distributed in proportion to the square root of the cost-weighted traffic, $\lambda_i d_i$, over all channels. This is called the "square root channel capacity assignment (SRCCA)."

Note that Kleinrock assumes that there is a continuum of available link capacities. Based on this assumption (in addition to the independence assumption) the average delay is convex with respect to each link capacity. This indicates that the solution is globally optimal. Additional analysis is required to map this solution to to the limited number of available link capacities. This mapping may not preserve global optimality.

2.2 AND's Simulation-based Solution

Although usage of the independence assumption provides a simple closed-form solution for the average delay, the accuracy of this assumption is questionable [3]. To illustrate the inaccuracy of product-form results, Kleinrock's SRCCA will be used as a basis for comparison with results provided by a simulation-based software system called AND, which is an abbreviation for ATM Network Design system. AND solves the CA problem by generating gradient estimates via Monte-Carlo simulation-based APA. These estimates are then used in an iterative, gradient-based algorithm using steepest descent with projection and an Armijo line search. These methods are discussed in detail in Sections 4 and 5 of this paper.

2.3 AND Execution

AND can be abstracted into several parts (see Figure 1). At the center is the graphical user interface (GUI) by which the user creates the network topology. This topology may be stored as a layout file, which may be saved and retrieved at a later time. The GUI also creates header files which are compiled with libraries to create the object code for the analysis. This procedure is used to eliminate the overhead introduced by dynamic memory allocation.

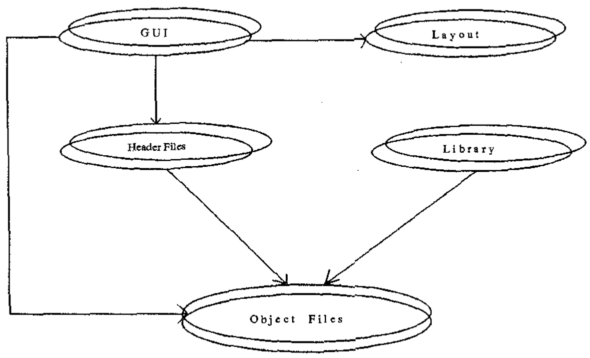

Figure 1. AND Execution

3 Graphical User Interface

The graphical user interface is a menu-driven, interactive program. The menu system allows the user to set up the network and simulation attributes, to run the simulation, to update and view the network, to manage the results of the analysis, and to present them efficiently. The menu structure is divided into several groups according to functionality. The menu system is designed according to convention, frequency of use, order of use, and categorical or functional groups. The hierarchy of the menu system is shown in Figure 2.

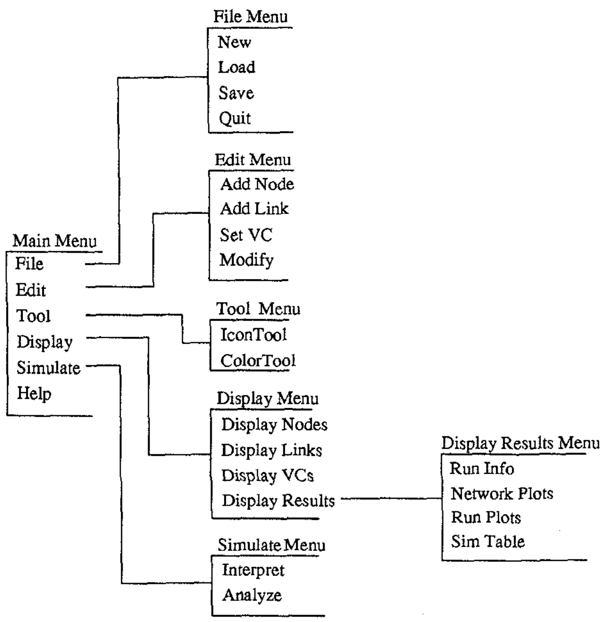

Figure 2. The hierarchy of the menu

The Main Menu presents six choices: the File Menu, the Edit Menu, the Tool Menu, the Display Menu, and the Simulate Menu. The File Menu is for storing, and retrieving network topologies. The Edit Menu allows the user to interactively create a network topology consisting of nodes, links, and input-regulated virtual circuits.

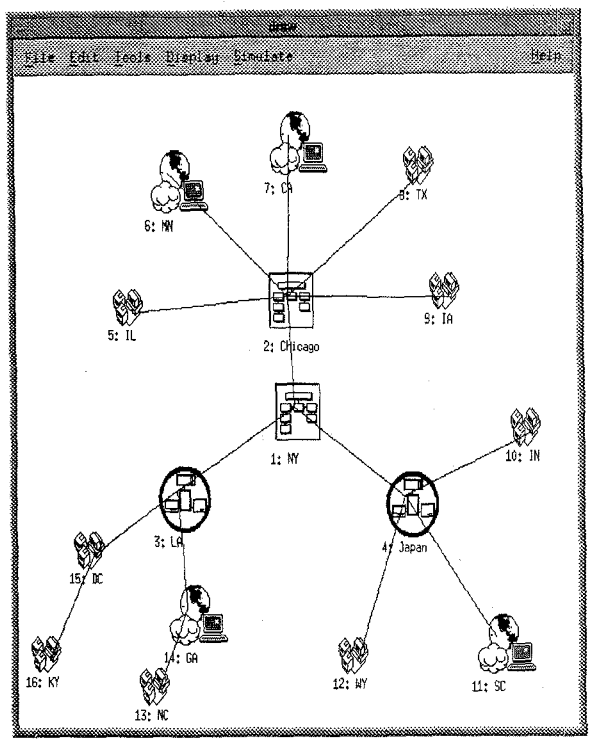

Figure 3. Graphical User Interface

The Tool Menu allows user-customization of the network representation. (See Figure 3 for an example.) Network topology information and results of the analysis can be displayed via the Display Menu. The Simulate Menu interprets the network topology and invokes the analysis. The Help Menu provides general guidelines to usage of AND.

4 Results

The model we consider is a 16-node ATM network interconnected by 30 (unidirectional) links (see Figure 3). The traffic consists of 62 input-regulated virtual circuits with arrival rates ranging from 1.0 to 25.0 cells per millisecond. For each input regulator σ is chosen to be 1 and ρ is chosen to be 4 times the virtual circuit arrival rate. It is assumed that all links have equal weights (cost coefficients). All simulations were run for 30 regenerative cycles[1] and up to 100 simulation iterations.

Table 1 displays the average delay versus cost for this network model. Both the predicted average delay and the initial simulation average delay are listed.

The "formula" average delay is the average delay deter-

Cost (K$)	Formula (msec)	Initial Simulation (msec)	Percent Difference
2.4	904.3	333.6	171%
2.5	519.5	195.2	166%
2.6	364.4	134.6	171%
2.7	280.6	116.0	142%
2.8	228.2	98.9	131%
3.0	166.1	77.5	114%
3.25	123.9	62.2	99%
3.5	98.9	52.6	88%
4.0	70.4	41.2	71%
5.0	44.6	29.6	51%
6.0	32.7	23.4	40%
7.5	23.3	18.1	29%
9.0	18.1	14.7	23%
10.0	15.8	13.1	21%
11.0	14.0	11.8	19%
12.0	12.5	10.8	16%
13.0	11.4	9.9	15%
15.0	9.6	8.5	13%

Table 1. Accuracy of Independence Assumption

mined by using Equation 1 with the capacities determined by Kleinrock's SRCCA (which makes the independence assumption). The "initial simulation" average delay is the result of simulating the network without making the independence assumption, and using the same capacities as those in the "formula" curve. Table 1 effectively shows that making the "independence assumption" leads to approximate performance results. The "accuracy" of the approximation is dependent on several factors including network topology and source traffic. Table 1 shows that in this instance the approximation predicted the average delay within a factor of 2.

Cost (K$)	Initial Simulation (msec)	AND Soln. (msec)	Percent Difference
2.4	333.6	255.1	30.8%
2.5	195.2	168.2	16.1%
2.6	134.6	128.9	4.4%
2.7	116.0	107.4	8.0%
2.8	98.9	93.5	5.8%
3.0	77.5	74.8	3.6%
3.25	62.2	61.0	2.0%
3.5	52.6	51.6	1.9%

Table 2. AND Results

Table 2 exhibits the gain in performance derived by application of the sample-path-based optimization algorithm. Usage of APA and gradient-based optimization resulted in a significant reduction in average delay in some cases.

5 Algorithms

AND uses Augmented Infinitesimal Perturbation Analysis (APA) and Gradient Descent methods. These methods will be examined in this section. To understand APA, IPA, its predecessor, is first introduced. The unbiased and strong consistent gradient estimates [2] determined by APA are used in a gradient descent algorithm to determine a locally optimal solution to the Capacity Assignment problem.

5.1 Infinitesimal Perturbation Analysis (IPA)

The queuing delay faced by a cell at the output link of an ATM switch is modeled as that in a G/G/1 queue. The IPA algorithm used is presented in detail in [14]. These results are extended to simultaneously estimate the gradient estimates for all links in a network from a single Monte-Carlo simulation.

To estimate the gradient of the average delay with respect to the inverse of the link capacity, accumulators measure sample gradients at each link. Note that M sample gradients are computed simultaneously, where M is the number of links in the network. The following procedure is used:

1. Let i = the link at which the capacity is perturbed, and j = the link at which the packet has arrived. All link accumulators are initialized to zero. I.e., $acc_{ij} = 0.0, \forall i, j$.

2. At the beginning of each busy cycle, the accumulator of link j is set to zero if the first arrival in the busy period is an external arrival (i.e., $acc_{ij} = 0.0, \forall i$); or the accumulator of link j is set to the accumulator value of the upstream link, k, from which the packet arrived (i.e., $acc_{ij} = acc_{ik}, \forall i$).

3. For each packet arrival (in the same busy period) at link j, the accumulator measuring sample derivatives with respect to perturbations of link capacity j is incremented by an amount equal to the packet size, which is derivative of the packet service time with respect to the inverse of the link capacity. Without loss of generality it is assumed that packet sizes (as well as link capacities) are in units of one cell (and cells-per-sec). Therefore the accumulator is incremented by one. I.e., $acc_{jj} = acc_{jj} + 1$.

4. These packet sample derivative values are collected as cells depart the network. The empirical average of the sample derivatives is used to determine the gradient estimate for that link.

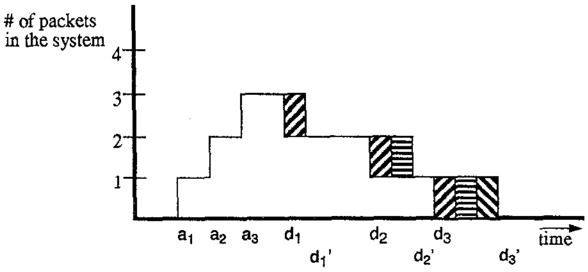

Figure 4. Capacity Infinitesimal Perturbation Analysis

Figure 4 illustrates the concept behind IPA. To determine the sample gradients, the inverse of a link's capacity is increased by a small "perturbation." In the Figure, $\{a_i\}_{i=1}^{3}$ correspond to the arrivals of three cells in a busy period of the perturbed link, and $\{d_i\}_{i=1}^{3}$ correspond to the respective nominal departure times. Perturbing the link results in increases in the service time of each cell. This also delays the initiation of service for cells not served at the beginning of the busy period.

As a result the departure times of each cell is increased. $\{d_i'\}_{i=1}^{3}$ correspond to these perturbed departure times. These perturbations also affect the arrival times at subsequent links. This propagation of perturbations is reflected in the second part of the third step of the IPA algorithm. Note that IPA determines sample derivatives of the end-to-end cell delay with respect to θ, and does not actually introduce real perturbations in the link capacities.

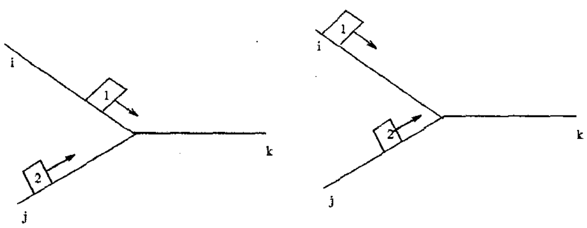

Figure 5. Nominal and Perturbed Event Orders

The major drawback of IPA is that there are discontinuities in the delay with respect to nominal and perturbed link capacities (as the perturbations decrease in size) [10]. This is illustrated in Figure 5. In this Figure link i and link j merge into link k. In the nominal sample path the inverse of the capacity of link i is equal to θ, and cell one link i precedes the arrival of cell two from link j at link k.

In the perturbed sample path the inverse of the capacity of link i is perturbed (increased) by $\Delta\theta$. This causes a delay in the arrival at link k of the cell one, which in turn results in a change in the order of cell arrivals at link k. This in turn will cause a jump discontinuity in the end-to-end delay of each cell with respect to $\Delta\theta$. (See Figure 6.)

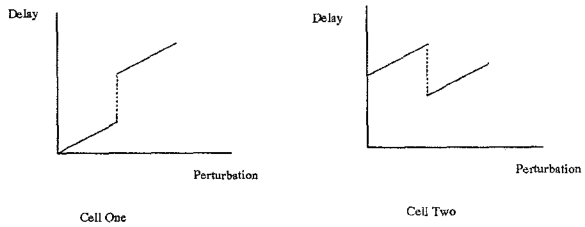

Figure 6. Cell Delay versus Perturbation

If both cells depart the network at the destination of link k, the two jump discontinuities will cancel one another and no discontinuity will appear in the average delay with respect to $\Delta\theta$. However if both cells do not depart immediately, their "perturbations" may propagate further along in the network. The effect of these perturbations may not be $o(\Delta\theta)$, and therefore will not disappear as $\Delta\theta$ tends to zero. This will produce a discontinuity in the average delay with respect to θ. If these continuities are dense over the interval of valid capacities, the average delay is not differentiable (cf. [4]). The effect of these discontinuities will be accounted for in the APA gradient estimate, which will also be referred to as the bias estimate.

5.2 Augmented Infinitesimal Perturbation Analysis (APA)

As mentioned previously event order changes can result in biased IPA gradient estimates [9]. To estimate this bias APA [6] is used. For simplicity assume that we are only interested in the gradient estimate at a single link. However note that APA, like IPA, can be applied in a parallel fashion to simultaneously determine bias estimates at all links.

Let the interval $\Theta = [a, b]$ represent a continuum of inverses of available link capacities, and let $\theta \in \Theta$ represent the inverse of the link capacity at which the gradient is being calculated. Let $T = T(\theta)$ represent the average end-to-end cell delay (as a function of θ) over an infinite time horizon, and let $T_n(\theta)$ represent the average end-to-end cell delay after n "events," where an event corresponds to either an external cell arrival, or an (internal or external) cell departure.

For a fixed sequence of events, $\omega \in A^n$, T_n can be represented as a function of θ, and $\{\psi_i\}_{i=1}^{n}$, where ψ_i corresponds to a sample of a uniform random variable. That is, $T_n = T_n(\theta, \psi_1, \ldots, \psi_n)$. It is assumed that clock values in the model are generated by inversion.

Continuing along these lines the expectation of T_n is

$$\mathbf{E}T_n(\theta, \psi_1, \ldots, \psi_n)$$

$$= \sum_{\omega} \int_{\psi_1^{min}}^{\psi_1^{max}} \cdots \int_{\psi_n^{min}}^{\psi_n^{max}} T_n(\theta, \psi_1, \ldots, \psi_n) d\psi_1 \ldots d\psi_n,$$

where it is assumed that the limits of integration are determined by enforcing deterministic similarity. That is the values of ψ for which events do not change order form an interval from ψ^{min} to ψ^{max}.

Following this course, Gaivoronski et al [6] determine this bias estimate:

$$\sum_{i=1}^{n} \frac{T_n(\psi_i^{max}) \frac{d}{d\theta} \psi_i^{max} - T_n(\psi_i^{min}) \frac{d}{d\theta} \psi_i^{min}}{\psi_i^{max} - \psi_i^{min}}.$$

5.3 Gradient Descent Algorithm

A steepest descent algorithm is used to perform a nonlinear minimization of the average delay with respect to link capacities with a linear constraint on the total cost of all of the links in the network. The algorithm uses an Armijo inexact line search algorithm [1] (as is demonstrated in [16]), and the gradient projection method.

6 Conclusions and Future Work

AND System has been shown to be an effective method for solving the capacity assignment problem in ATM networks. To our knowledge, AND is the first large-scale usage of APA. This is contrary to the common belief that APA is only practical for small problems. AND is another demonstration of the effectiveness of the Armijo line search, as well as being a demonstration of regenerative simulation.

We conclude that the difference in average delay determined by product-form and simulation results can be significant depending on the total network link cost. To be exact this difference is also dependent on other factors such as the network topology and the traffic specification. Elaboration of these dependencies is the subject of future research.

References

[1] L. Armijo. Minimization of functions having Lipschitz continuous first-partial derivatives. *Pacific J. Math.*, 16(1):1–3, 1966.

[2] C. A. Brooks. Using augmented infinitesimal perturbation analysis to determine asymptotically unbiased gradient estimates for capacity planning in ATM intree networks. Technical Memorandum UIC-EECS-96-3, University of Illinois at Chicago, Department of Electrical Engineering and Computer Science, May 1996. Submitted to the Journal of Discrete Event Dynamic Systems.

[3] C. A. Brooks and P. Varaiya. Using perturbation analysis to solve the capacity and flow assignment problem for general and ATM networks. In *Proc. of 3rd International Conference on Telecommunication Systems Modeling and Analysis*, pages 19–32, March 1995.

[4] X.-R. Cao, W.-B. Gong, and Y. Wardi. Ill-conditioned performance functions of queueing systems. *IEEE Trans. Automatic Control*, 40:1074–1079, 1995.

[5] C. G. Cassandras. *Discrete Event Systems: Modeling and Performance Analysis.* Aksen Associates Incorporated Publishers, 1993.

[6] A. A. Gaivoronski, L. Y. Shi, and R. S. Sreenivas. Augmented infinitesimal perturbation analysis: An alternate explanation. *Discrete Event Dynamic Sys.: Th. and Appl. (J. Discrete Event Dynamic Systems)*, 2:121–138, 1992.

[7] B. Gavish and I. Neuman. A system for routing and capacity assignment in computer communication networks. *IEEE Trans. Comm.*, 37(4):360–366, April 1989.

[8] M. Gerla and L. Kleinrock. On the topological design of distributed networks. *IEEE Trans. Comm.*, 25(1):48–60, January 1977.

[9] P. Heidelberger, X.-R. Cao, M. A. Zazanis, and R. Suri. Convergence properties of infinitesimal perturbation analysis estimates. *Mgmt. Sci.*, 34(11):1281–1302, November 1988.

[10] Y.-C. Ho and X.-R. Cao. *Perturbation Analysis of Discrete Event Dynamic Systems.* Kluwer Academic Publishers, 1991.

[11] L. Kleinrock. *Communication Nets: Stochastic Message Flow and Delay.* McGraw-Hill Book Company, 1964.

[12] L. Kleinrock. *Queueing Systems: Volume 2.* Wiley, 1976.

[13] J. Pan and C. G. Cassandras. Infinitesimal perturbation analysis of a queueing system with bursty traffic. *Discrete Event Dynamic Sys.: Th. and Appl. (J. Discrete Event Dynamic Systems)*, 4:325–358, 1994.

[14] R. Suri. Perturbation analysis: The state of the art and research issues explained via the GI/G/1 queue. *Proc. IEEE*, 77(1):114–137, January 1989.

[15] J. Walrand. *An Introduction to Queueing Networks.* Prentice-Hall, 1988.

[16] Y. Wardi and K. Lee. Application of descent algorithms with Armijo stepsizes to simulation-based optimization of queueing networks. In *Proc. of the 30th Conference on Decision and Control*, pages 110–115, 1991.

[17] N. Xiao, F. F. Wu, and S.-M. Lun. Dynamic bandwidth allocation using perturbation analysis. In *IEEE INFOCOM*, pages 383–389, June 1994.