JB 97-06

Entitled:

Hierarchical, Layered Modeling and Performance Evaluation of Hybrid Communication Networks

Authors:

J.S. Baras, M.S. Corson, K. Jang, M. Li,
A. Misra, H. Xie

Conference

MILCOM'97
Monterey, CA
November 2-5, 1997

HIERARCHICAL, LAYERED MODELING AND PERFORMANCE EVALUATION OF HYBRID COMMUNICATION NETWORKS *

J. S. Baras, M.S. Corson, K. Jang, M. Li, A. Misra and H. Xie

Center for Satellite and Hybrid Communication Networks
Institute for Systems Research
University of Maryland
College Park, Maryland 20742

ABSTRACT

Commercial and military networks are rapidly becoming hybrid networks; that is interconnected wireless terrestrial, wireline terrestrial and satellite networks. There is a great need for software tools to design, model and also do performance analysis efficiently for such large heterogeneous networks. This is the topic addressed in this paper. The hybrid network modeling and performance evaluation tool is currently under development at the University of Maryland. The tool consists of an algorithmic structure and a software architecture. The algorithm is hierarchical in form, deriving its structure from that of the network which follows the network's routing hierarchy. At each level, a node represents a set of nodes at a lower level. The boundary of a node is characterized by a set of statistical models. The statistics, and their parametric representations for the models of the level 1 nodes are obtained from the OPNET simulations which are run offline. The model defines procedures that convey modeling, simulation and performance data up and down the hierarchy of models. We consider the following performance metrics only: end-to-end delay, throughput, blocking probabilities, cell or packet loss rate. We describe algorithms based on loss network approximations and fast progressive solutions of the resulting fixed point equations.

INTRODUCTION

Modeling of communication networks is a daunting challenge given the complex nature and large scope of today's and tomorrow's communication systems. The approach we pursue is top-down in design, but bottom-up in function-relying on low-level, discrete event simulation of small communication networks to generate statistical parameters for higher-level, hierarchical, analytical models to permit fast approximate performance evaluation of strategic communication networks.

The discrete event simulation package we are using is OPNET. OPNET will be used to accurately model small communication networks. The purpose of simulating networks at the low tactical level is two-fold: (i) to accurately verify the functionality of network protocols and quantify their operation in terms of desired performance metrics and (ii) to provide appropriate statistical parameters to feed into the lowest level of a hierarchical analytic model. In the context of a simulation, each small network becomes a node at the lowest level of the mathematical hierarchy. These nodes are referred to as "level 1" nodes.

The network performance evaluation tool is currently under development at the University of Maryland. The tool consists of an algorithmic structure and a software architecture. Figure 1 illustrates the hierarchical layers of the tool. The algorithm is hierarchical in form, deriving its structure from that of the network which follows the network's routing hierarchy. The mathematical basis for the model is a hierarchical modeling of traffic flows based on Markov modulated Poisson and Bernoulli processes. These models have been known in practice to model accurately a variety of multimedia traffic over a variety of channels and communication media. At each level, a node represents a set of nodes at a lower level. The boundary of a node is characterized by a set of statistical models. The statistics, and their parametric representations for the models of the level 1 nodes are obtained from the OPNET simulations which are run offline .

At each level, topology is represented by logical nodes and logical links. At the lowest (which is the highest

*This work was supported in part by the U.S. Department of the Army, Army Research Laboratory under Cooperative Agreement DAAL01-96-2-0002, Federated Laboratory ATIRP Consortium, and in part by the Center for Satellite and Hybrid Communication Networks, under NASA cooperative agreement NCC3-528.

0-7803-4249-6/97/$10.00 © 1997 IEEE

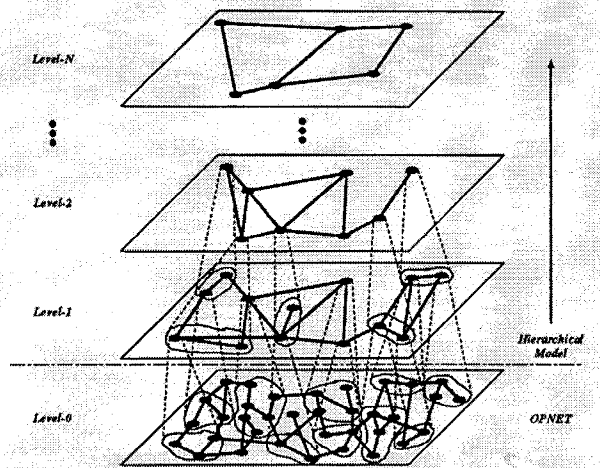

Figure 1: Hierarchical Layers

resolution) level, each node and link represents a real switching system and a physical link. At higher layers, each node may represent either a real system, or a subnetwork, or a group of switching systems. At higher layers links may correspond to either a real physical link or a virtual link.

The models in this approach form a hierarchical, progressive chain (from coarse to fine). We will then develop techniques from multi-level optimization, multigrid numerical methods for progressive computation of performance metrics or trade-off curves. Progressive here means that the computations of the previous layer are used and are improved upon

$$\phi(J_n, K_n) = \phi(J_{n+1}, K_{n+1}) + \Delta_n^{n+1}(d_n, \lambda_{n+1})$$

$n = 1, \cdots, N, 1 =$ fine, $N =$ coarse. Here J and K are two performance metrics. The subindex indicates the layer or accuracy of the computation. The second term in the right hand side indicates the correction term that utilizes the data at the new layer and some aggregate variables propagated from the previous layer. In this paper we describe the algorithms developed todate. We consider the following performance metrics only: end-to-end delay, throughput, blocking probabilities, cell or packet loss rate.

We are developing fast algorithms to compute trade-off curves, and to perform interactive trade-off analysis, among these performance metrics by linking the hierarchical modeling and simulation system to multi-objective optimization packages. In this way we can compute performance metric sensitivities with respect to network parameters using the simulation model; a particular instance of this idea is what is called simulation differentiation. The tool under development is useful for both military and commercial networks. A typical application is depicted in Figure 2. Here we want to evaluate in some detail – using discrete event simulation – a small network (one on the west-coast) while taking into account the effects of the larger network. Since it is not feasible to perform detailed discrete event simulation for the large network it is better to simulate the large network in some aggregate fashion and interject these aggregate results into the discrete event simulation of the small network. Similar examples are abundant in performance evaluation of satellite constellation networks.

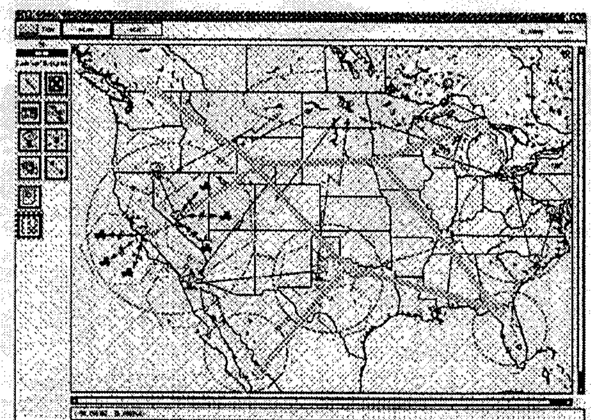

Figure 2:

ALGORITHMS BASED ON FIXED-POINT METHODS

The first analytical method that we have investigated, is that of *fixed point* methods for estimating blocking in multihop, multirate integrated service networks with state dependent admission control and routing. This analytical technique is at the connection level, and applies to synchronous transfer mode (circuit-switched) services, and to asynchronous transfer mode (ATM) services. These methods have been shown to be efficient and accurate in previous work [3-5]. They also can be used for worse-case analysis since they are quite accurate under overload conditions.

In our work, we have concentrated on integrated services networks, *vs* voice where past work focused, and on hierarchical solutions and formulations of the fixed point equations. One of the difficulties of moving these effective methods into multirate methods has been the fact that in multirate networks the computational complexity of these algorithms is infeasible. Here we use some methods first introduced in [3] to circumvent this computational complexity obstacle.

The network model that we assumed is as follows.

We consider general networks of N nodes, indexed $1, 2, \cdots, N$. A link (i,j) between nodes i and j has capacity $C_{i,j}$, counted in bandwidth units termed *trunks*. Links are undirected; (i,j) and (j,i) denote the same link. Calls offered to the network fall into S classes, differentiated by bandwidth and possibly by priority via trunk reservation, as described below. A call of class s has bandwidth b_s, meaning that if the call is admitted to the network then it is allocated b_s trunks on each link of a path from its source to its destination. A link's state is characterized by the number of calls in progress n_s, of each call class.

A call is admissible on a given path if it is admissible on every link of that path. Consider a call of class s offered to node pair (i,j). Link admissibility may be decided with or without control. In the absence of control, the call is admissible on a link if the call's bandwidth b_s is less than or equal to the idle bandwidth $C - \Sigma_t b_t n_t$. Admission controls considered here are of trunk reservation type. Specifically, the call is admissible on link (i,j) if the calls bandwidth b_s is less than or equal to the idle bandwidth $C - \Sigma_t b_t n_t$ plus r, where r is a static parameter, which may depend on s, i, j, and other factors, such as the length of the route associated with the call. Thus, choosing parameter r to be small allows the call type greater access to scarce capacity. If admitted on a path in the network the call simultaneously seizes for its exclusive use b_s trunks on every link of the path. Similarly, the call simultaneously releases b_s trunks on every link of the path when it completes.

We consider routing policies where each node pair (i,j) is assigned an ordered list $P_{i,j}$ of paths between nodes i and j: $P_{i,j}^{(1)}, P_{i,j}^{(2)}, \ldots$. The call is admitted on the first path in the list where it is admissible. If the call is not admissible on any path then it is blocked; that is, rejected and lost.

Traffic is modeled as follows. Calls of class s are assumed to arrive to node pair (i,j) according to a Poisson process with rate $\lambda_{i,j}(s)$. If admitted to the network, the call holds for an exponential period of time with mean $\mu_{i,j}(s)$. Thus, these calls constitute a load of $\rho_{i,j}(s) = \lambda_{i,j}(s)/\mu_{i,j}(s)$.

The key performance metrics are the node pair by node pair, class by class call blocking probabilities,

$$Pr\{blocking\ a\ call\ of\ class\ s\ arriving\ to\ node\ pair\ (i,j)\},$$

or (equivalently) the availability probabilities, defined as corresponding probabilities of admitting the call.

The fixed point method that we use is based on the following two approximations (assumptions):

$(A1)$ (Link Independence): The probability that a cell is admissible on a given path in the network is the product of the probabilities that the call is admissible on each link of that path.

$(A2)$ (Poisson Rates): Arrivals of calls of a given class to a given link are described by a Poisson process. The rate of the process may depend on the state of the link.

It is important to note that although arrival rates to node pairs are given and are assumed to be Poisson, admission control and routing depends on link-states and thus perturbs the Poisson character of the link arrivals. These two assumptions, lead to a system of equations, whose unknowns are, for each link (i,j) and call class s:

(i) arrival rates $\nu_{i,j}(s)$ of calls of class s from any node pair that includes link (i,j) on some route, given that link (i,j) is in a state that admit calls of class s, and

(ii) probabilities $a_{i,j}(s)$ that link (i,j) is in a state that admits calls of class s.

The fixed point formulation, essentially consists of two mappings. One such mapping is for every node pair (i,j) from probabilities $a_{k,l}(s)$ (with k and l dependent on the routing structure) to rates $\nu_{i,j}(s)$. The other mapping is for each link (i,j) from rates $\nu_{i,j}(s)$ to probabilities $a_{i,j}(s)$.

To estimate equilibrium network performance, we solve for the fixed point of the composition of these two mappings, simultaneously fixing the $\nu_{i,j}(s)$ and $a_{i,j}(s)$. The algorithms as in [3] are inspired by the close mathematical similarity of these equations to certain problems in network reliability.

The mapping from admissibility probabilities to arrival rate is now described.

Let $\nu_{i,j}^{(m)}(k,l,s)$ = rate that (k,l) admits class s calls on its m^{th} route, if that route contains link (i,j); otherwise $\nu_{i,j}^{(m)}(k,l,s) = 0$. Thus, link (i,j) receives calls of class s at aggregate rate

$$\nu_{i,j}(s) = \Sigma_m \Sigma_{k,l} \nu_{i,j}^{(m)}(k,l,s).$$

We now proceed to compute $\nu_{i,j}^{(m)}(k,l,s)$, assuming link (i,j) belongs to the m^{th} route, for node pair (k,l).

For $m = 1$

$$\nu_{i,j}^{(1)}(k,l,s) = \lambda_{k,l}(s) \prod_{(u,v)\in P_{k,l}^{(m)},(u,v)\neq(i,j)} a_{u,v}(s)$$

where $\lambda_{k,l}(s)$ denotes the original rate of arrivals for calls of class s for node pair (k,l) and $a_{u,v}(s)$ is the admissibility probability for class s on link (u,v).

For $m > 1$,

$$\nu_{i,j}^{(m)}(k,l,s) = \lambda_{k,l}(s) \cdot R_{i,j}^{(m)}(k,l,s) \cdot \prod_{(u,v)\in P_{k,l}^{(m)},(u,v)\neq(i,j)} a_{u,v}(s)$$

$$R_{i,j}^{(m)}(k,l,s) = Pr\{\text{none of the paths } P_{i,j}^{(1)} \cdots P_{i,j}^{(m-1)} \text{ are admissible} \mid \text{path } P_{i,j}^{(m)} \text{ is admissible}\}.$$

Combining the formulas above completes the mapping from link admissibility probabilities to link arrival rates.

We next describe the mapping from arrival rates to admissibility probabilities when there is admission control in the network. It is well known that admission control destroys the product-form of the link occupancy probabilities $P(n)$, which in turn destroys the efficient and exact computation of those probabilities by a formula such as:

$$a(s) = 1 - \Sigma_{n=C-b_s+1}^{C} P(n)$$

where C is the link capacity, S the number of classes and b_s the bandwidth of each class s call. Instead, we now have to compute the equilibrium distribution of a Markov chain with a large state space; a prohibitive computational task. Instead we employ an approximation first introduced in [6,7], which transforms the problem into one-dimensional one.

The following method appears to be very effective. Letting $\alpha(s)$ denote the average number of calls of type s in progress,

$$\alpha(s) = a(s)\lambda(s)/\mu(s).$$

since these calls enter into service at rate $a(s)\lambda(s)$ and depart at rate $\alpha(s)\mu(s)$. Treat the $\alpha(s)$ as given, and consider the one-dimensional Markov chain, which, for any state n and call classes s, jumps to:

(c) state $n + b_s$ with rate $\lambda(s)I(C - n \geq r + b_s)$, where r is the trunk reservation parameter, and to

(d) state $n - b_s$ with rate $\mu(s)n\frac{\alpha(s)}{\Sigma_t \alpha(t)}I(n \geq b_s)$. In the true system, this jump down depends on the number of calls in progress n_t for each class t, rather than the aggregate state $n = \Sigma_t n_t$, so this is an approximation, that follows from estimating n_s by $n\alpha(s)/\Sigma_t \alpha(t)$.

Then the probability of admitting a call of class s is approximately $a(s) = 1 - \Sigma_{n=C-b_s-r_s+1}^{C} P(n)$. Taking $P(n)$ to be the equilibrium distribution for this one-dimensional chain, we arrive at a small system of fixed point equations. In [6,7] it is shown that this approximation becomes asymptotically exact in overload conditions, for large C and $b_s << C$. More refined approximations are also discussed in [6,7].

In our work todate we have developed hierarchical versions of these algorithms that are based on hierarchical routing in the network being analyzed. We are currently working on versions of these hierarchical algorithms that are progressive. We are developing implementations that work at different layers while exchanging information during the computation. We are also experimenting with parallel implementations of the hierarchical algorithm. Basically we have arrival rates and admissibility probabilities at each level, except that they correspond to aggregate links and nodes. The definition of these quantities is delicate because they should lead to an efficient progressive computation of blocking probabilities. If these algorithms can speed up performance by 10 times over discrete-event simulation in a single layer, we can (for instance) speed up by 1000 in a three layer application.

In the paper we will present results on the accuracy of these computations realistic network.

SOFTWARE ARCHITECTURE

The software implementation of the performance evaluation tool is written in Java. Users will access the tool via any World Wide Web browser that supports Java from any hardware platform. The software architecture is fundamentally client/server-based. Users access the simulation kernel–hosted on a server at the University–via a browser-based, graphical user interface (GUI) which forms the client. Thus, while simulations are constructed within the client, they are executed within a high-speed, multi-processor server.

The software architecture is CORBA-compliant, with all Java objects being specified first in CORBA's Interface Definition Language (IDL). The Object Request Broker of choice is Iona's OrbixWeb. Simulation objects may be made persistent via an object adapter to ObjectStore.

Nodes and channels form the basis of the network model's object structure. Nodes form an object hierarchy of their own, and have a different representation on both the client and server. Server node objects contain all data that must be persistent from run to run, and client node objects are decorated versions of the server objects containing additional GUI functionality. Channel objects also have differing representations on both client and server. Client channel objects primarily capture the spatial, topological and qualitative characteristics of the channel whereas, on the server, channel objects are translated into mathematical quantities imbedded in the model.

Figure 3, depicts the current GUI to the hierarchical performance evaluation tool. We are also building various GUIs to display the performance evaluation results. Figure 4 depicts a GUI that is used to evaluate satellite coverage and placement.

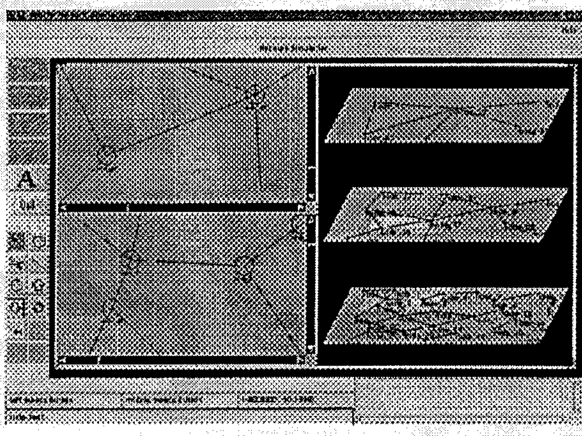

Figure 3: The GUI to the hierarchical performance evaluation tool.

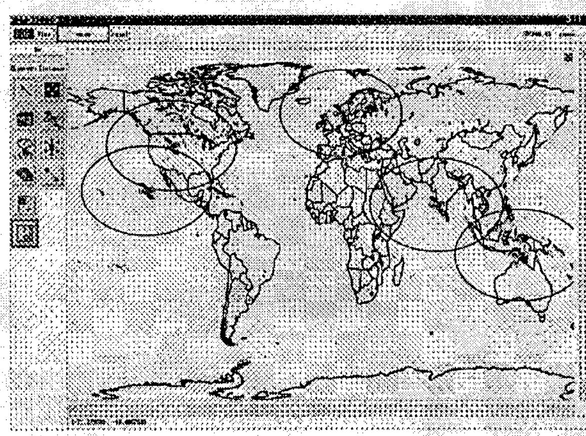

Figure 4: Simulation Tool

REFERENCES

1. J. S. Baras *et al*, "Progress Report on ATIRP 2.1 Task", June 1996, University of Maryland at College Park.

2. F.P. Kelly, "Loss Networks", *The Annals of Applied Probability*, Vol. 1, No. 3, pp. 319-378, Aug. 1991.

3. A.G. Greenberg and R. Srikant, "Computational Techniques for Accurate Performance Evaluation of Multirate, Multihop Communication Networks", preprint, 1996.

4. S.P. Chung and K.W. Ross, "Reduced Load Approximations for Multirate Loss Networks", *IEEE Trans. on Communications*, Vol. 41, No. 8, pp. 1222-1231, Aug. 1993.

5. S.P. Chung, A. Kashper and K.W. Ross, "Computing Approximate Blocking Probabilities for Large Loss Networks with State-Dependent Routing", *IEEE/ACM Trans. on Networking*, Vol. 1, No. 1, pp. 105-115, Feb. 1993.

6. N. Bean, R.J. Gibbens, and S. Zachary, "Asymptotic Analysis of single resource loss systems under heavy traffic, with applications to integrated services network", *Adv. Appl. Prob.*, March 1995.

7. N. Bean, R.J. Gibbens, and S. Zachary, "The Performance of single resource loss systems in multiservice networks", in *Proc. of the 14th International Teletraffic Congress*, 1994.