Technical Research Report

Hybrid (Satellite and Terrestrial) Communication Networks: Object Oriented Generic Tools for Simulation and Management

by J.S. Baras and T. Charuhas

CSHCN TR 93-5/ISR TR 93-93

Center for Satellite & Hybrid Communication Networks

A NASA Center for the Commercial Development of Space

University of Maryland Institute for Systems Research

IEEE Communications Society REPRINT

HYBRID (SATELLITE AND TERRESTRIAL) COMMUNICATION NETWORKS: OBJECT ORIENTED GENERIC TOOLS FOR SIMULATION AND MANAGEMENT

John S. Baras
Thomas Charuhas

Reprinted from IEEE Communications Society
"1992 IEEE Military Communications Conference"
October 11–14, 1992

IEEE Communications Society
345 East 47th Street
New York, NY 10017

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

Hybrid (Satellite and Terrestrial) Communication Networks: Object Oriented Generic Tools for Simulation and Management *

JOHN S. BARAS and THOMAS CHARUHAS

Electrical Engineering Department, Systems Research Center
and Center for Satellite and Hybrid Communication Networks
University of Maryland College Park, MD 20742

ABSTRACT We describe a generic object oriented software environment for the simulation and management of hybrid communication networks (satellite and terrestrial) under development at the University of Maryland, as part of the NASA CCDS program. We refer to this environment as a "systems engineering workstation for hybrid networks". It will have the capability to model such networks in a totally object oriented environment and link to network performance evaluation modules. Satellites are treated both as electronic/mechanical systems and as communication nodes. Great flexibility is allowed for defining the various components of the satellite, including its type (i.e. LEO or GEO), on board processing capabilities, bandwidth, multiple access scheme, linkage to terrestrial networks. The satellite link is also treated as an object allowing great flexibility in modeling weather effects, multipath interference and coverage. We show the system's flexibility in studying combined adaptive routing and access for voice-data large hybrid networks.

I. INTRODUCTION

Simulation environments have often been plagued with a fundamental contradiction. Either the environment/tool is so generic that results can only be obtained after a long painstaking effort at software development, or it is so tailored to a specific task, that use of the tool for purposes other than its specific tasks is virtually impossible. Designing tools that are generic enough to handle the diverse aspects of a hybrid communication network, while incorporating enough pre-fabricated functionality is our prime objective. By relying on object oriented technology, we are giving the user the flexibility to create a simulation based on generic objects or highly tailored ones that provide specific functionality.

*Work supported by NASA under the CCDS program and the NSF under the ERC program

II. SOFTWARE BASE

Our hybrid communication network testbed, dubbed SatSim, is an extension to another project being developed at the University of Maryland Systems Research Center, SimNet [5]. SimNet is an object oriented class library written in Jade's Sim++ environment. Sim++ in turn, is a set of classes and utilities based on C++. The software hierarchy of the system is shown in figure 1 and described below.

A. What is C++ ?

C++ is an object oriented language based on C which is generally considered to be a standard

0-7803-0585-X/92 $3.00 © 1992 IEEE

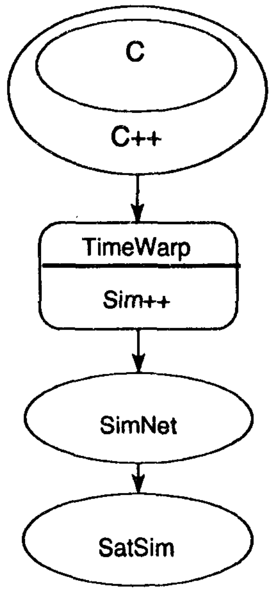

Figure 1: Software hierarchy of SatSim

and is optimized for speed and memory conservation rather than true object-orientedness. Speed is of extreme importance in simulations.

B. What is Sim++ ?

Sim++ is a set of classes written in C++ that allow you to create objects called Sim Entities and schedule events between them. Sim++ also provides classes for creating random variables with several standard probability distributions. Sim++ is built upon TimeWarp, an executive that allows your simulations to be run on either a single processor for sequential execution or on multiple processors for parallel execution. Those multiprocessors could be located on a Transputer board or in several workstations connected via a LAN.

C. What is SimNet ?

SimNet is an extensive set of classes based on Sim++, specifically designed for simulating communication networks. Base class objects such as Nodes, Channels, Transmitters, Receivers, Simulators and OSI Layers are all provided and have general functionality [5]. In order to create one's own simulation, one needs to subclass the existing objects and tailor them to perform the specific duties of a custom simulation. The SimNet system also comes with a parser that will read a file that defines all the objects that will be used in a simulation and create these objects at initialization time. Having a standard input file that describes the system is useful if one wants to automate the process of designing the system to be simulated by adding a graphical front end, as will be described later.

D. What is SatSim ?

SatSim is an extension to SimNet. Every useful class library such as SimNet can be extended by creating new objects by subclassing the previous objects to create a more useful and less generic set of objects. With SatSim we have created a set of new objects that are specifically tailored to simulating hybrid networks that can include satellites, ground stations and mobiles. Motion models for the satellites and mobile users are available as well as channels that model communications between them. These channels can be quite complex, taking into account, for example, different weather conditions and interferences. Custom OSI layer objects are being created, which will allow the user to interchange different channel access, routing and management algorithms in a plug-and-play fashion, while still allowing the flexibility to create custom protocols at any layer of the OSI hierarchy.

III. SATSIM OBJECTS

A. The object-oriented paradigm in discrete events simulations

We cannot emphasize enough how invaluable the object-oriented paradigm is to discrete events simulations. "Object-oriented programming enables a program to be written with a focus on the description of the problem rather than algorithms for solving the problem." [2] SatSim uses this paradigm to its fullest.

B. Hybrid network channels

SimNet contains two base classes for channels: point-to-point and broadcast channels. SatSim

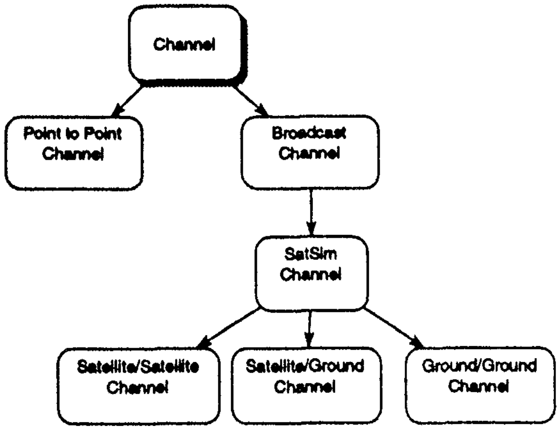

Figure 2: Channel Class Hierarchy

channels are based on this broadcast channel. They are divided into satellite/ground, inter-satellite and ground/ground channels. Each channel, as defined in SimNet contains a physical medium object. This object is used to model the physical medium through which a signal propagates. Noise, delay, loss and interference are some of the characteristics this object models. SatSim uses these objects to simulate the effects of various weather conditions. By dynamically picking the correct physical medium object for the current conditions and attaching it to the channel, varying conditions such as changing weather and cloud cover on the ground can be modeled. Highly unusual physical characteristics can be modeled by subclassing existing physical medium objects to create custom ones. Highly unusual channels (such as meteor burst channels) can also be created in this way.

C. Handling a highly dynamic topology

Satellites and mobile users are generally highly mobile. The main difference between simulating a hybrid network and a standard terrestrial network is that in hybrid networks, the topology varies from one instance to the next. Rather than allowing for an occasional link or node failure as is the case in most terrestrial networks, one must be able to handle links that are constantly being created and destroyed as satellites and ground units pass in and out of range of each other. In order to decide whether a ground station or mobile user is within the footprint of a satellite antenna beam, each such object must be queried for its location. If the location falls within the calculated footprint for the satellite at that instant, channels and links can be created. The number of ground objects are theoretically unbounded. Multiplied by the number of Satellites attempting communications, this number can be very large. This can be quite taxing and will cause a topology update event to be very time consuming. [5]

In order to keep simulation time reasonable, it is important to find an interval between which topology update events will occur, that is small enough to accurately model this highly dynamic system, yet is large enough so as not to cause unnecessary delay in the simulation. Since this is a global event, information about the changing topology must be propagated to all processors on which the simulation is running. Thus one must factor in the latency of the network as well as the speed of the processor when handling such a global event. Distributed simulation will be discussed below.

D. Motion Models

If a node is not stationary, it has a motion model associated with it. SatSim provides motion models for satellites based on orbital parameters and bounded random walk models for mobile users. Custom, fixed trajectory models to simulate vehicles traveling cross-country, for example, are easily implemented.

E. OSI layer models

The lower OSI layer protocols will be implemented in SatSim. Default implementations of layer 2 will exist to model FDMA,TDMA and CDMA. Basic routing schemes for the network layer will be provided, but we are expecting this layer in particular to be often customized in order to simulate complex systems that handle handovers and complex routing schemes.

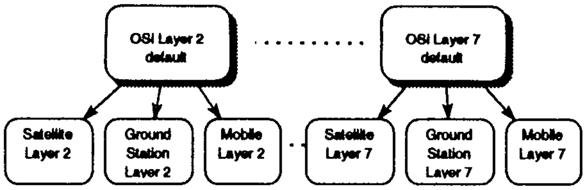

Figure 3: OSI Layer Class Hierarchy

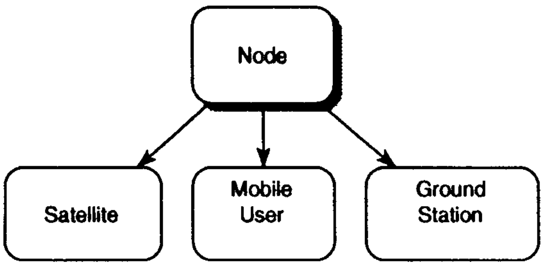

Figure 4: Node Class Hierarchy

F. Nodes

The basic nodes provided are ground stations, which are immobile, satellites and mobile users. The mobile users include mobile terminals located on land and sea-based vehicles as well as individuals carrying mobile phones. The complexity of these nodes can vary tremendously. The on-board processing capabilities of these nodes are implemented in the simulation by attaching OSI layers of different complexities to the node.

G. The Communication Simulator

SimNet contains an object called the Communication Simulator (CS) which handles event processing and distributing. SatSim also has its own Communication Simulator called the SatSim CS which handles the specific tasks of a hybrid network simulator. Updating topology, handling events and keeping track of all the entities are just some of its tasks.

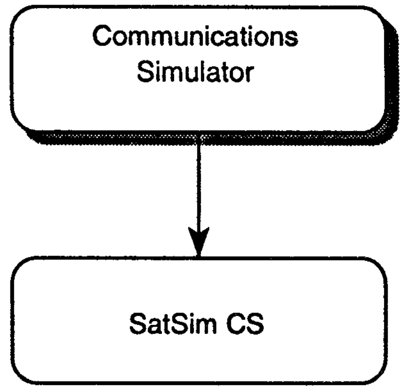

Figure 5: ComSim Class Hierarchy

IV. SATSIM I/O

A. General I/O

SatSim is more than a set of class libraries. Through the NetMonitor, which is being developed, one can graphically observe the simulation as it progresses. Also, the specification of the network, which consists of all the objects in the simulation as well as their properties, are specified by a single file which can be hand-built or automatically generated by a graphical interface we are developing. The key to this interface will be it's generic handling of the objects that comprise the network. When one creates a specific type of satellite or link it is imperative that the graphical interface be able to handle the new properties associated with the new type of object. This would be quite difficult in a non object oriented environment. The graphical interface will be an application running under NeXTstep. The actual simulation will be run on several workstations in a heterogeneous environment over TimeWarp.

The input file is parsed by a YACC-based parser. In this way, the network need not be hard-coded in Sim++ but rather completely described by this file. The file contains all the parameters belonging to the entities in the simulation. Upon startup, the file is read and parsed, the entities created and the initialization parameters are sent as part of an event at time zero.

Since one standard file can be used to describe

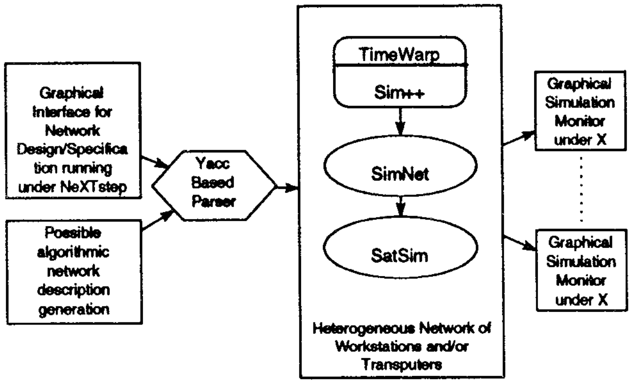

Figure 6: SatSim I/O

the network, it is feasible to algorithmically create multiple network descriptions in order to analyze or evaluate existing or theoretical networks for network management purposes such as real-time evaluation of different routing protocols or topologies.

B. The Graphical Interface

Once the objects and their behavior have been designed and implemented, one must decide what the network will look like. One must know which objects will be part of the simulation, how they will be interconnected and what their parameters are. SatSim includes a graphical front-end that allows the user to add any objects to the"world". The front end is intelligent enough to query the user for all the needed parameters and information associated with that object. For example, a satellite has a set of orbital parameters associated with it that will describe its orbit. In fact, our interface allows the user to animate the orbits on a globe before the simulation is run so the user can be sure his/her satellite is doing what it is supposed to. Ground stations and mobile users can be placed and all their communication characteristics can be defined. It is important to remember that SatSim is supposed to be modified to suit the users particular needs. New objects have new parameters and functionality. The interface is open-ended enough to handle new objects based on the ancestor objects in SimNet and SatSim.

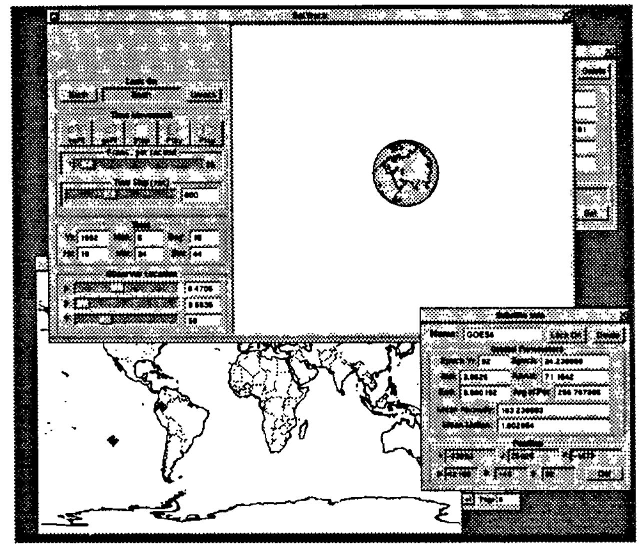

Figure 7: Graphical Interface

V. NETWORK MANAGEMENT

A. Future Networks

Future communication networks will have fundamental differences from past and current networks. Typical characteristics of telephone networks are: homogeneity, modularity (transmissions are separable and evolve independently), hierarchical structure, universal access, contention occurs only at session initiation, there is unified design. The networks of the future have the following characteristics: heterogeneity (1 to 109 bps), no clear modular or hierarchical structure, contention occurs during sessions, some users require guaranteed rates other require guaranteed delays, access is not universal, functions are poorly separated, real-time and other timeliness constraints prevail. In addition constraints on design and operation are imposed by various standards committees and are not always compatible with best operating policies.

These characteristics complicate network operations and management to an unprecedented level. Further complexity results due to the prevalent multi-vendor environment and the increasing importance of network security. On the other hand the continuous advances in microelectronics and the advent of Integrated Services Digital Networks (ISDN), including separate signalling channels/networks, provide the means for increasing and distributing intelligence in the network. The end results are networks that provide extremely sophisticated operations and management.

The key elements are: (1) a transmission/switching system, (2) a separate signaling network, (3) a database, and (4) a support system for the database. Our research agenda addresses several problems that arise in this environment. The area with the greatest challenge and potentially the greatest payoff involves the interaction between the signaling system, the database and the support system.

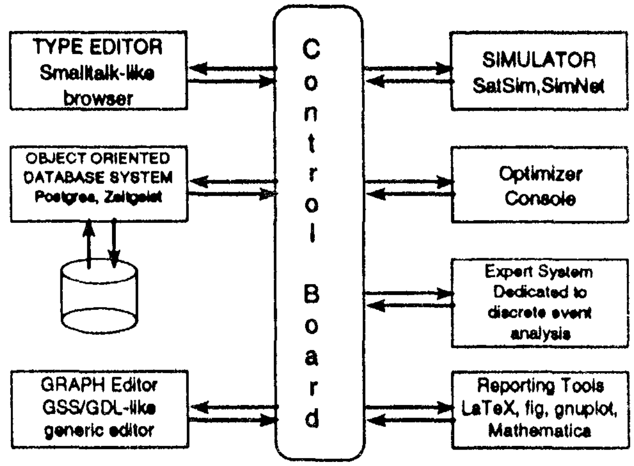

Figure 8: Network Design and Management Workstation

B. Network Design Operations and Management Workstation

Our group is working on several aspects of such a software environment. The basic architecture of the system is illustrated in figure 8.

The environment will be totally object oriented (OO) and the user will interact with the system through an intelligent interface. We have already completed the design and implementation of the type editor [1] and we have initiated the design of the OODB and related OOP distributed simulation tools. Our group has developed several graphic interfaces for network simulation and management in earlier projects.

In an Integrated Network Management system we can identify four main components: (1) a communication network, (2) a simulation package used to explore "what if" questions, (3) an optimization package that provides optimal routings, etc., and (4) a database system that stores performance data, faults, and other useful information. The interesting research problems here stem from the fact that the simulation, the optimization, and the system environments must be tightly coupled with the DBMS, overcoming the need for manual offloading of data into the database. Not only the DBMS serves as a storage and retrieval agent, but it is also used as an active agent between the simulation, the optimization, and the actual running system.

In any system environment, we can identify two different views of the data handled by the DBMS, the *simulation view* and the *system view.* The DBMS should be designed to support both views. Due to the size of the input and output vectors, and the number of input-output pairs generated in the course of a typical network simulation, the data management in such an environment is a time consuming and error prone process. A DBMS can provide means for storing the data and utilities for efficient retrieval.

Similar comments hold for the optimization environment. These latter are nothing but specialized expert systems that solve optimization problems. Hence, in this case, one would like to be able to have an "active" database system that allows, in addition to data storage and retrieval, higher functions such as inferencing and reasoning.

C. Databases and Optimization

The research of our group in this area has several levels of objectives. The first set of objectives is to analyze the requirements of several applications environments and from this analysis to produce a set of database system requirements. Secondly, to derive a set of guidelines/rules for logical and physical database design in this environment. The third objective is to propose enhancements to both the relational and object oriented data models.

In [4] a database system is described, which effectively supports the management of large networks of the future. MANDATE uses special characteristics of network management data and transactions, together with recent advances in database technology, to effectively derive its functionality.

D. Algorithmic design and specification

Certain network management applications call for rapid creation and simulation of a network. For example, different topologies of the same network can be created and simulated in order to find an optimum topology. Theoretically, an on-line system could evaluate a real network using simulation and quickly decide on changes in the event of failures. By having a single file following a simple yet strict syntax, the network description can be created many times per second, rather than input by a human operator; a task that could take days. These tasks can be taken over by a computer. Varying topologies can also be analyzed in this way. Algorithmically designed topologies can be tested many times in order to verify the algorithm.

Dynamic, traffic responsive routing is a key problem in integrated services digital networks. Network heterogeneity complicates the problem. For large networks fast distributed algorithms are needed. we have formulated a model-predictive approach to the problem of demand responsive, dynamic routing for modern and future integrated services networks [3]. We have initiated a study within the simulation environment described here. We have also initiated a study of hybrid cellular-satellite experiments simulations in the environment.

References

[1] M. Scott Corson,"A Distributed, Object-Oriented Communication Network Simulation Testbed", preprint, 1992.

[2] David L. Eldridge, John D. McGregor and Marguerite K. Summers, "Applying the object-oriented paradigm to discrete event simulations using the C++ language", Simulation Feb 1990.

[3] J.S. Baras and C. Muller, "A Type Editor for Network Simulation", Systems Research Center, University of Maryland, Technical Report 1991.

[4] J.R. Haritsa, M.O. Ball, N. Roussopoulos and A. Datta ,"MANDATE: MANaging Networks using DAtabase TEchnology", preprint, Systems Research Center, University of Maryland 1992.

[5] J.S. Baras and S.W. Chen, "Dynamic Routing in Large Hybrid Multi-media Networks: A Model Predictive Control Approach", to appear.