JB 94-16

Entitled:

"Object Oriented Hybrid Network Simulation"

Authors:

(with G. Atallah)

Conference:

Technology 2004
Washington, D.C.
November 8-10, 1994

Object Oriented Hybrid Network Simulation[1]

John S. Baras[2], Ahsun H. Murad, Kap Do Jang, George C. Atallah[3], Ramesh Karne and S. Joseph Campanella
Center for Satellite and Hybrid Communication Networks
Institute for Systems Research
University of Maryland
College Park, MD 20742

ABSTRACT

The simulation of complex contemporary hybrid communication networks is a daunting task and cannot be handled by the array of conventional network simulation tools available today. This paper discusses the key issues involved in the visualization and simulation of complex hybrid networks and presents the design of an object-oriented hybrid network simulator being developed at the Center for Satellite and Hybrid Communication Networks (a NASA center for the commercial development of space), University of Maryland at College Park. This software incorporates a complex system simulation paradigm, advanced visualization techniques, a flexible interface and an object-oriented database to allow the simulator to be used not only as a network simulation tool, but also as a network design tool, as a network performance evaluation tool, and as part of an intelligent network management software.

INTRODUCTION

As the complexity and diversity of networks have grown, simulation has proved to be an important tool in their design, analysis, testing and performance estimation. A typical hybrid communication scenario is shown in Fig. 1. Hybrid networks involve a variety of network elements—both terrestrial and satellite with their associated protocols, and the services they provide, like commercial video and radio transmissions, voice, data and image transmission services. Because of their complex nature, design and evaluation of hybrid networks is a particularly complicated and challenging task.

A number of tools are available for the simulation of communication networks. Some of these are general-purpose simulation packages like BONeS DESIGNER, GPSS/H, MODSIM II, SIMSCRIPT II.5, SES/*workbench*, SIMAN/Cinema V and SLAMSYSTEM; others like OPNET Modeler provide a communication network simulation language; and yet others are specialized communication network simulators like BONeS PlanNet, COMNET III, NETWORK II.5 and L•NET II.5. For the simulation of communication networks, a dedicated communication network simulator offers advantages over a general purpose simulator, namely, speed of network modeling, a library of predefined network components and perhaps, automatic computation/display and analysis of the key communication-related performance statistics. A general-purpose simulator, on the other hand, has the flexibility to model almost any discrete-event system, though, as the generality of the simulator increases, typically, so does the effort required.

The simulation of a complex hybrid communication network offers a unique challenge to simulation tools. Because systems to be simulated are complex and diverse, constructing a system model using a general-purpose simulator would be too time-consuming. On the other hand, most of the current network-simulation tools are not designed to visualize or simulate such complex networks. The deficiencies in simulating hybrid communication networks with conventional communication network tools are (i) difficulty in modeling the system in sufficient detail and accuracy, (ii) difficulty in capturing the entire communication scenario, (iii) slow-speed of execution, (iv) inadequate network performance monitoring/evaluation support both during and at the end of a simulation run.

A number of issues are involved in the simulation of hybrid networks. While being of interest in the simulation of conventional networks, they gain critical status for the simulation of hybrid networks because of the unique demands

[1] Work supported in part by NASA grant NAGW-2777S for the establishment of a NASA Center for the Commercial Development of Space, and by grant NSFD CDR 8803012 through the Engineering Research Center Program.

[2] Martin Marietta Chair in Systems Engineering, Rm. 2249 A. V. Williams Building, University of Maryland, College Park, MD 20742. Ph: (301) 405-6606, Fax: (301) 314-9218, email: baras@isr.umd.edu.

[3] Rm. 3117 A.V. Williams Building, University of Maryland, College Park, MD 20742. Ph: (301) 405-7411, Fax: (301) 314-8586, email: atallah@isr.umd.edu.

placed by the complexity and diversity of these networks. They in turn place stringent demands on the simulation tools and motivate the development of new tools tailored to the simulation of these complex systems.

A Complex System Simulation Paradigm — To get an idea of the complexity, let us consider the simulation of Calling's proposed Teledesic satellite network [1]for the provision of global telecommunication services. The proposed system configuration consists of 840–900 satellites in a 21 low-earth orbital plane configuration, with a possibility of ATM support. The capacity of this system is in excess of 2,000,000 simultaneous full duplex 16kbps channels. With the small data-segment size of ATM packets, the simulation of this system operating even at 1% capacity would require the processing of over 10 million packets per second of simulated time. At 100% capacity, a processing rate of over a billion packets a second would be required.

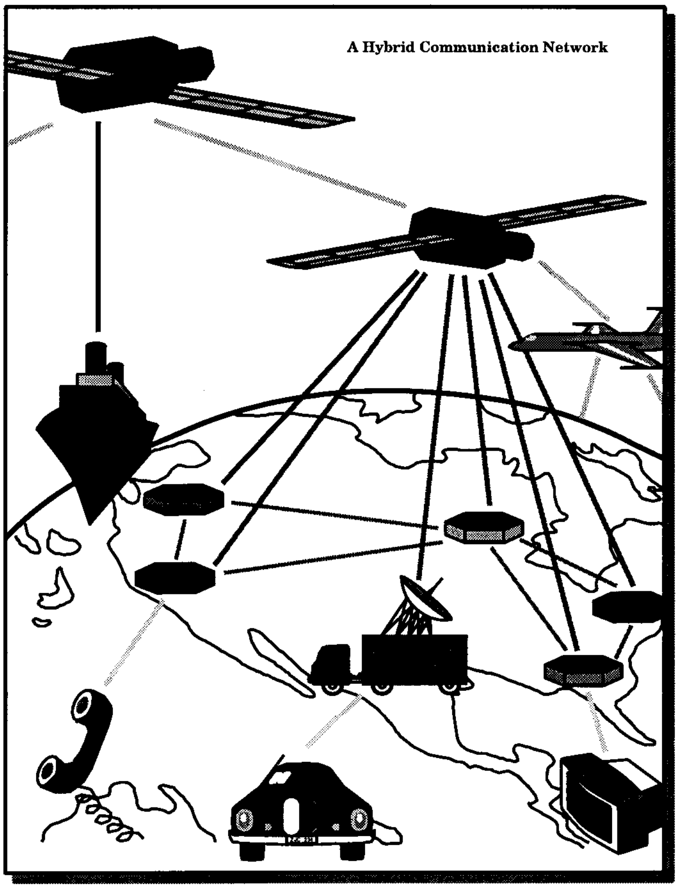


Fig. 1. A Hybrid Communication Scenario.

Assuming the simulation of the life of a packet (generation, end-to-end transfer and consumption) to generate on the order of 100 events (conservative, considering that a typical packet path would involve several satellites), an event based simulator trying to simulate this system would have to process on the order of a billion (1% utilization) to a 100 billion events (100% utilization) for every second of time simulated. With the current technology, even the world most powerful computers would creep through the event-based simulation of such a system—and we haven't even considered the simulation of the terrestrial network that would interact with the Teledesic satellite network, or the work required to simulate the dynamic nature of the satellite/terrestrial network.

Conventional event-driven simulation has been found to be quite efficient for the simulation of conventional networks. However, as the complexity of the system being simulated grows, the vast number of events being generated and required to be individually simulated presents a bottle-neck on the speed of simulation. The back-of-the-envelop calculations we presented for the Teledesic network simulation scenario demonstrate the immensity of the problem. Clearly, a more effective simulation paradigm is required for the simulation of hybrid networks, one that takes advantage of the inherent structure within the sub-systems that make up the complex network. In the hybrid simulation tool, we adapt a multi-tier model based approach– the network system is modeled in terms of a few large functional blocks, which may be recursively defined in terms of smaller functional blocks. The more detailed the model, the more accurate the simulation, and the higher the simulation complexity. This scheme allows a few small, critical blocks to be simulated more accurately, while allowing the other simulation blocks to be replaced by equivalent aggregate parameters, thereby reducing simulation complexity.

Flexibility and Adaptability — A hybrid network, by its very definition, is a diverse system that brings together in a single system, different means of communication, each with its own array of network components, protocols and control strategies. It is not possible for the creator of a simulation tool to provide a model for every possible component that may be required to be simulated. Even if it were, advances in technology and changes in networking ideologies would result in new building blocks that would be required to be simulated, rendering such an effort useless. It is therefore, very important that a hybrid network simulation tool be designed from the bottom-up to allow easy creation of user-defined components and their efficient integration into the simulation software.

Communication Scenario Visualization and the GUI — Conventional communication networks consist of nodes and links (and specializations of these). While these are the same things that make up the communication aspects of a hybrid network, effective representation and visualization of hybrid networks also involves much more. Hybrid satellite/terrestrial networks often span the globe and interact with the world more directly than conventional networks. For example, correct location of satellite earth-stations and mobile base stations requires knowledge of geographical data, so transformable digital geographical databases need to be incorporated. Accurate prediction of traffic patterns for global networks may also require population modeling in terms of density, economic status etc., requiring the incorporation of other databases. Long-term network performance may also require the incorporation of global climatic data. (For example, rain fade in areas of high rainfall may degrade quality of service to unacceptable levels.)

With the ability to incorporate all this multi-faceted information in a simulation comes the difficult problem of visualization and display of this information to the user and providing him with controls to easily manipulate it as desired. The complexity of the network itself may present problems, and require that visualization of the network be done at a higher level with selective presentation of the most relevant data at the top level.

This points towards the development of a sophisticated graphical user interface (GUI)—one that appears clean and simple, and yet utilizes all possible means (like color, size and shape coding) to convey multidimensional information. New network visualization paradigms are also required to display complex network structures.

Data Management, Databases and Network Management — During a simulation run, vast quantities of data may be generated. Typically, this data would be simply dumped into a file, and accessed later for post-simulation analysis. A more efficient approach is to structure the data on the fly by storing it in a database. This allows simulation data to be manipulated and utilized much more easily during the post-simulation phase. More importantly, it allows access to this data while the simulation is being performed, allowing dynamic computation and display of network performance statistics. The database also serves as an interface point for intelligent network management tools that could use the simulated network as a model to predict the long-term behavior of a real network, and use the performance data generated by the simulator to formulate long-term network management policies.

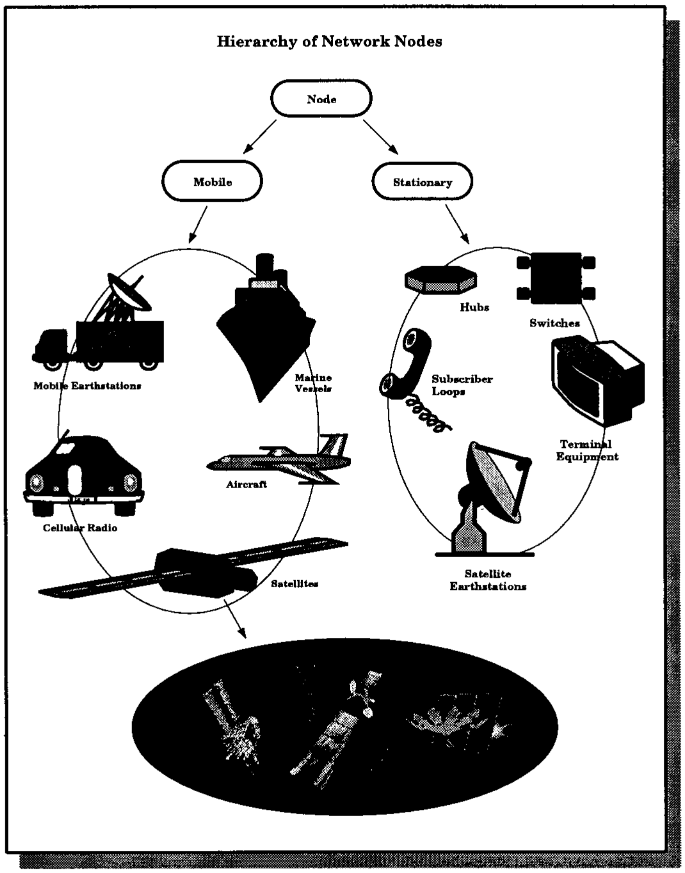


Fig. 2. Object-Oriented Classification of Network Nodes.

Object-Oriented Programming — Object-oriented programming is an advanced approach to structured programming. It is ideally suited to hybrid network simulation for all the reasons that makes it ideal for most complex software efforts: a clean software structure, software and effort reuse, flexibility and adaptability, ease of documentation, and with the availability of modern object-oriented programming languages, efficiency. For example, Fig. 2 illustrates how object-oriented programming may be used to define network nodes; a top-level generic node captures those features common to all nodes, and more specialized node structures are constructed through inheritance and specialization from more general class structures.

THE HYBRID NETWORK SIMULATOR

This section describes the design and implementation of an object-oriented hybrid simulation network being developed here at the Center for Satellite and Hybrid Communication Networks, University of Maryland at College Park. Keeping in view the motivation behind the development of this software, the proposed system is geared towards the simulation of complex hybrid network simulation systems. The basic structure of the software is shown in Fig. 3. The software is divided into three main parts: the interface, the simulation kernel, and the expansion module library. An object-oriented database serves as a medium for storage and exchange of information between the different components.

The Interface

Three categories of users are expected to make use of the hybrid network simulator. The first of these is the human user who wishes to simulate a communications system. To do this, the user sits in front of a terminal, invokes the software, and then interacts with the **Graphical User Interface** (GUI) to create, simulate and evaluate the performance of a simulated network. The second of these is a programmer, who also sits at a terminal, but this time, wishes to create and incorporate a new module into the software, or perhaps to customize an existing one. The programmer (who is expected to have a deep understanding of the internal organization of the simulator) is allowed to directly access many of the internal structures of the software through a special **Application Programmer Interface** (API). The third kind of user that the software caters to is a software application, (e.g., a network management application) that is using the simulator to simulate a selected network, and is using the network-performance data generated by the simulator for some other purpose (e.g., network management). A lean and efficient **Software Interface** (SI) capability is provided for such a user. In short, the GUI is visualization oriented, the API programmer oriented, and the SI efficiency oriented.

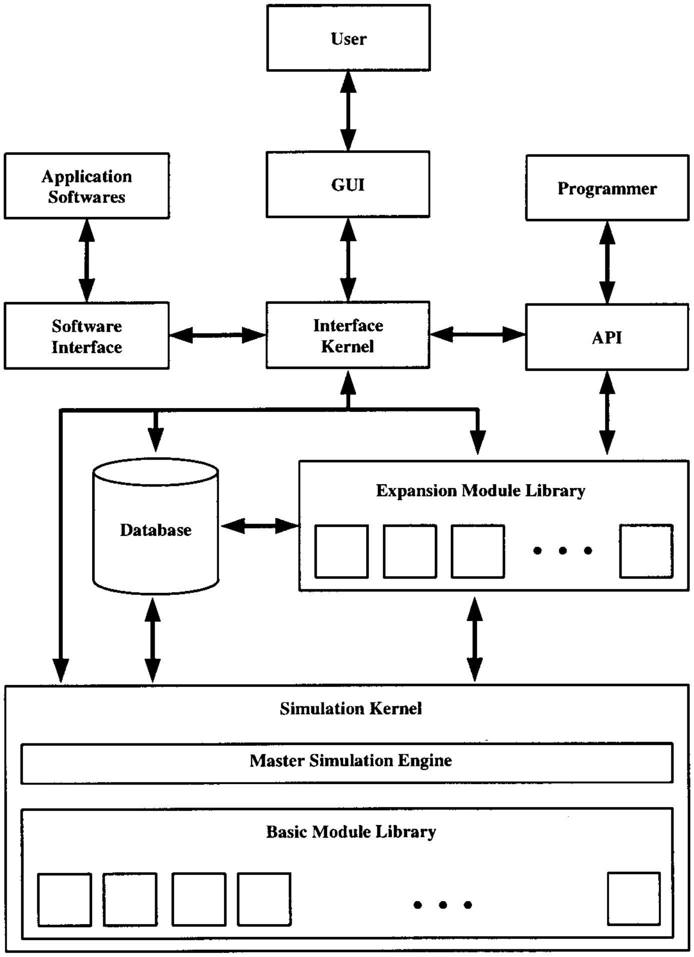


Fig. 3. Block-Diagram of the Hybrid Network Simulator.

Flexibility was the guiding criterion in the design of the interfaces. The software is designed to be easily extendible to many different kinds of graphics platforms, windowing systems, and to be able to interact with a variety of software applications. To do this, the interface is divided into two stages. A platform/user independent **Interface Kernel** provides a canonical set of interface functions that provide complete basic control of the software. The GUI, the API and the software interface are all blocks that use this set of functions. This interface kernel acts as a buffer between the interface and the internal program; changes in the interface do not propagate to the rest of the simulation software. For example, support for a new windowing system can be provided by building a new GUI that still uses the same canonical function set. (Since the API requires greater access to the simulation software, direct interaction also exists between the API module and the simulation software. The API is however expected to be much more simple to adapt to different platforms, as it is basically a text-based programming interface.)

The Simulation Kernel

The heart of the simulation software is the simulation kernel which consists of the **simulation engine** and the **basic module library**. The simulation engine is an integral part of the control mechanism that implements the complex system simulation paradigm which is described below.

The Complex System Simulation Paradigm

Simulation of complex networks requires a rethinking of the technique of simulation. When systems required to be simulated were simple, a time-based simulation paradigm worked well. With more complex systems, an event-based simulation produces significant reduction in complexity, especially if the system is not very active, temporally, i.e., only a few of all the possible events occur at a time. However, as noted earlier, when system complexity increases, even event-based simulation does not work well. The vast number of events being generated, and required to be individually processed creates a bottleneck which slows down the simulation to unacceptable levels.

For simulation of complex systems, a model based approach is often employed. That is to say, the simulation does not try to emulate what would happen in a real system on an event-by-event basis. Rather, a mathematical model of the system is created that tries to approximate the behavior of the entire system. (For example, a queue might be mathematically modeled as a delay element—the delay modeled as a stochastic process—rather than through the enqueue, dequeue and job service operations on an event-by-event basis.) This is however possible only for very well structured networks, and even then, involves a tradeoff between complexity and accuracy of simulation. We must realize, however, that for simulation of a complex system within an acceptable time frame, such a tradeoff between accuracy and speed has to be made. In light of this, a model based simulation is a promising compromise.

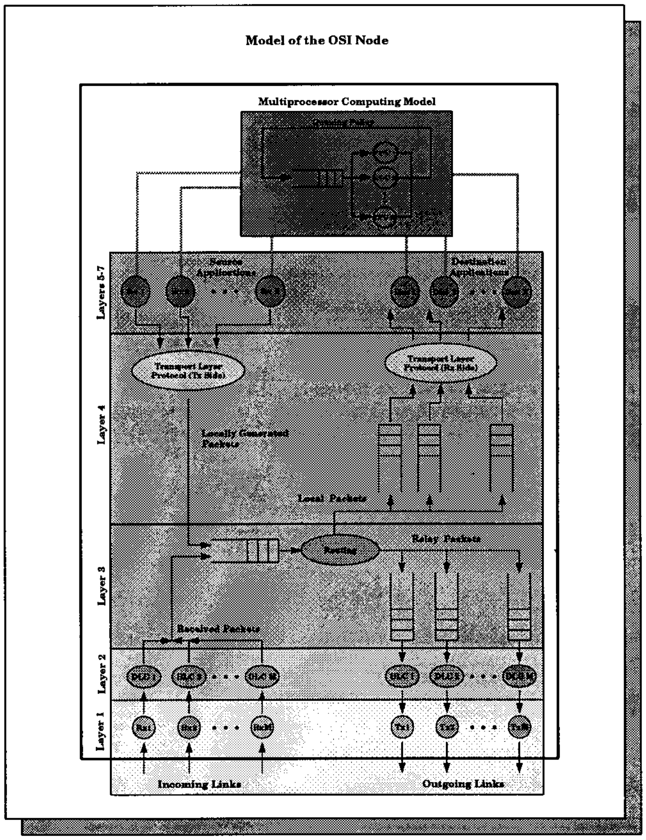


Fig. 4. Structure of the generic OSI Network Node.

Further, it may not be necessary to model the entire network to the same level of accuracy. Non-critical network components may be modeled only coarsely, e.g., a geostationary satellite transponder based up-down link pair may be modeled simply as a broadcast link with a statistical delay and error injection model (rather than as two links and a satellite node). On the other hand, a critical component may be modeled to a much higher level of detail and accuracy, of course, at the cost of an increase in complexity. Often, this modeling may be done in terms of sub-components, which are in turn modeled similarly.

The Basic Modules Library

Also present within the kernel is a **basic modules library**. This is a collection of generic building blocks like queues, switches, transmitters, receivers, timers, e.t.c., and generic nodes and links that can be used to build more complex network components. The modules within the basic module library are however not expected to directly be used as network components. For example, a generic OSI node with a structure as in Fig. 4 may be used as a template to create specialized node structures. The OSI node template is a detailed layer-by-layer model of an OSI structured node with capabilities for multiple incoming and outgoing links with separate transmitters/receivers and queues. Flexibility is provided to implement various Data Link Control (DLC – Layer-2) protocols, routing algorithms (Layer-3), end-to-end transport layer (Layer-4) protocols and other higher level protocols. Also provided is a model of a node processing resource with provisions for modeling multiple CPUs and I/O resources and different job-queuing strategies like First-Come-First-Served, Round-Robin, Shortest-Job-First, e.t.c., and prioritized versions of these.

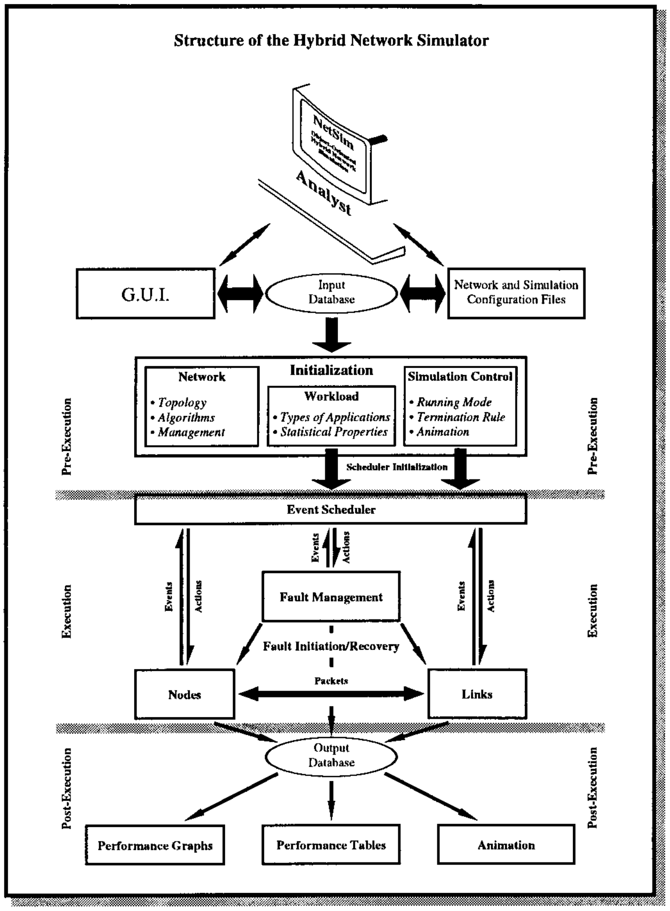


Fig. 5. Temporal View of the Simulation.

Conceptually, the course of a simulation proceeds in three temporal phases, (a) the pre-execution phase, (b) the execution phase, and (c) the post-execution phase. (See Fig. 5.) During the pre-execution phase, the user interacts with the simulation software (either through the GUI, or through files) to create a network scenario to be simulated. During the execution phase, the network is simulated: source applications generate packets of data, which are transported across the network (through links and nodes) and delivered to sink applications. As the simulation proceeds, various network performance statistics are generated and stored in an object-oriented database. As part of the post-execution phase, this database is accessed, and the desired network performance parameters are computed and presented to the user (in the form of performance graphs and tables). As a visual verification of the correctness of the simulation, the user is able to view an animation of the simulation, with the key transactions (like transfer of packets) being displayed dynamically on the GUI.

The Expansion Module Library

The basic module library within the simulation kernel consists of a set of basic components that may be used to build a network component. For example, a specific kind of node could be constructed starting from the generic node and adding transmitters, receivers, queues e.t.c., as required. Unlike the modules within the basic module library, the expansion module library consists of a collection of modules that can directly be used to construct networks. The expansion modules also provide additional functionality. For example, a digital geographical database module would be used to provide map functions; a population module to provide a model of the global population, e.t.c..

Another major difference between the basic module library and the expansion modules library is that while the basic module library is fixed, modules in the expansion module library can be added, deleted, or even changed. At the time of dispatch, the simulation is preconfigured with a collection of modules in the expansion module library. Depending on the needs of users, an application programmer may customize the expansion module library through the API. Users who then wish to simulate a particular communication network will find the exact building blocks that they need in the expansion module library.

The Database

The inclusion of an efficient object-oriented database as an integral component of the simulation software is of vital significance. Vast quantities of information are required to be stored and manipulated as part of the simulation. Online documentation, help messages, test network scenario storage and performance data generated during the course of a simulation run all need to be stored and accessed with speed. The programming effort is greatly reduced through the use of an object-oriented database that specializes in the task of data handling. Equally important is the role of the database as an interface between the different components of the simulation. As a test scenario is created, the information is organized and stored in the database. When simulation is initiated, this information is retrieved to construct a network and simulate it. As simulation proceeds, performance data is again stored in the database. A network-management software accessing the simulator through the software interface would retrieve this information and use it as input to come up with network management strategies, and perhaps dynamically change the network configuration.

SOME TYPICAL APPLICATIONS

In this section, we describe two typical applications of the object-oriented distributed hybrid network simulation software. The first application tries to highlight some of the network visualization capabilities, which have been covered only sketchily in previous sections. The second application is an illustration of how the software may be used as a tool for network management.

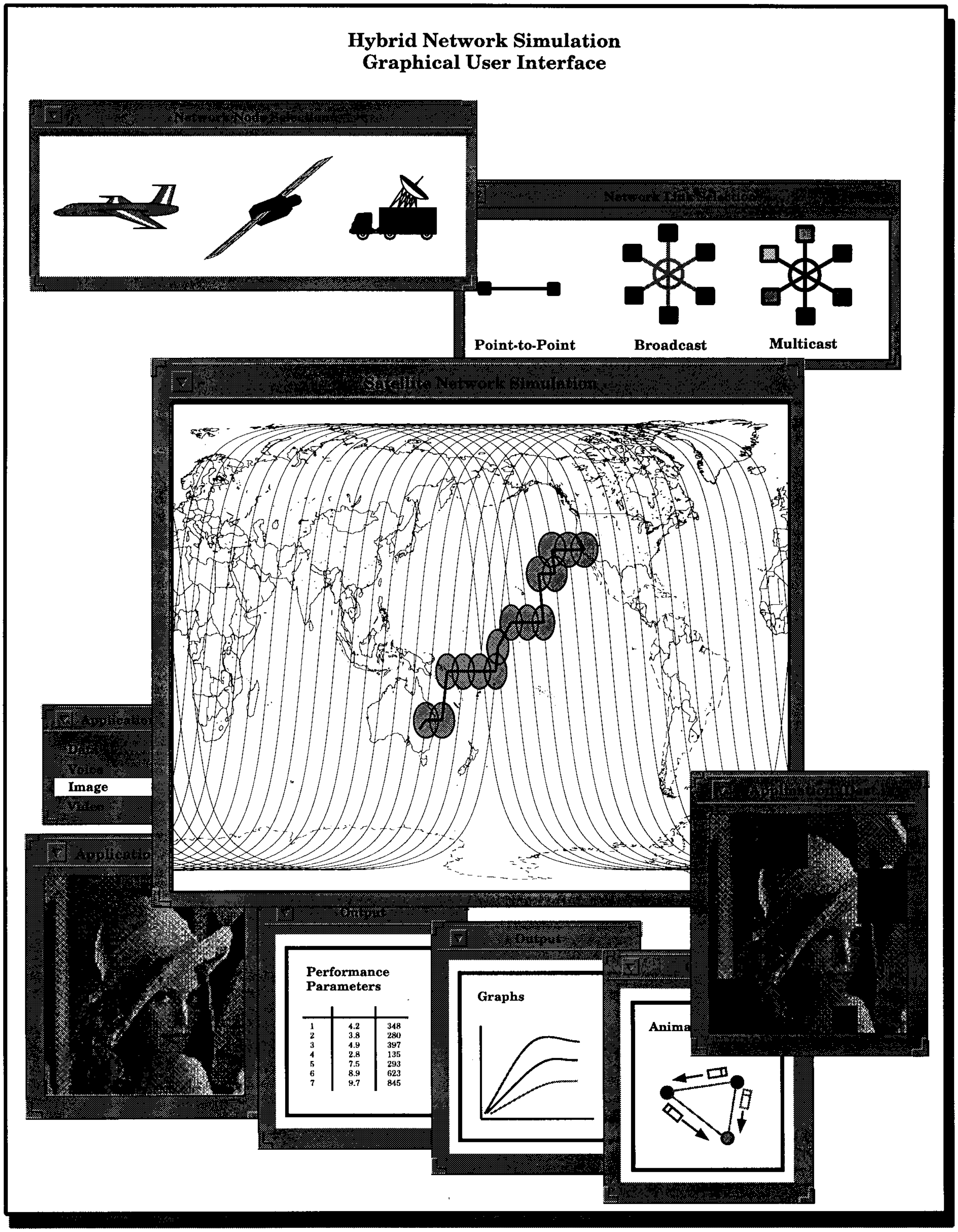
Satellite Constellation Network Simulation

With proposals to provide global communication coverage (telephone/cellular) through a constellation of low-earth orbit (LEO) and medium-earth orbit (MEO) satellites arranged in multiple orbits around the globe, e.g., Iridium, Teledesic and Globalstar, the simulation of satellite networks has suddenly gained prominence. Because of the dynamic nature of satellite networks, both visualization and simulation of such networks presents unique problems. Recognizing their importance, the hybrid network simulator provides a special interface to handle the simulation and visualization of satellites and satellite constellations and their relationship with the earth. The specialized satellite constellation GUI provided by the software incorporates (i) definition of multiple orbits and the positioning of multiple satellites within each orbit; (ii) definition of satellite and earth-station communications parameters; (iii) visualization of the satellite constellation – logically (showing connectivity) and spatially against the globe and on a flat map; (iv) selecting the visual context with respect to a satellite, a point on the globe, or some inertial frame of reference; (v) visualization of end-to-end connection paths, and associated path metrics; (vi) definition and visualization of cell patterns and antenna beam patterns associated with a satellite as it moves around the earth; and (vii) incorporation and visualization of climatic activity and population behavior models.

Fig. 6 shows how some of this information is presented on the GUI. Multiple visualization models are required to effectively visualize the network; for every piece of information, there is a model of visualization that represents it best. In Fig. 6, only the flat-map visualization is shown. The figure shows a communication path between Australia and California over the proposed Teledesic network, and illustrates how the communication path crosses the constellation seam. The path itself is depicted as a bold line, and the shaded ovals represent each satellite's computed footprint (assuming a minimum antenna elevation angle of 40°) projected on the flat map. Other information like beam patterns, connectivity and relative satellite positions can also be effectively visualized on the flat map.

Other visualizations of the satellite constellation network, i.e., a globe-based 3D visualization and a logical view showing communication path and satellite performance related statistics are also available (not shown) to more effectively visualize the network.

Additionally, Fig. 6 also illustrates symbolically, how a network may be assembled using nodes and links and also shows how performance-related data may be viewed dynamically. Finally, the figure also shows an actual distributed image application running on the simulated network (using the software interface feature).



Performance Parameters

1	4.2	348
2	3.8	280
3	4.9	397
4	2.8	135
5	7.5	293
6	8.9	623
7	9.7	845

Fig. 6. Visualization of a Satellite Constellation and related parameters.

Intelligent Network Management

The flexible interface and the incorporation of a database prepare the hybrid network simulation software as a tool for network management in a closed loop with a network management software. (See Fig. 7.) In a typical network, network monitor functions gather data on network performance, and store it in distributed databases. Some of this data is used for short-term network management and fault resolution directly, using dedicated network management functions. Aditionally, as part of the intelligent management of the network, this data is used to drive the simulation (which runs in parallel with the actual network and mimics its behavior). The long-term behavior of the simulated network now acts as a predictor of the long-term behavior of the network, and may be used to better manage the long-tem network performance. The effect of particular network policies can be safely judged by testing them on the simulated network, before they are applied to the running network. For example, in current work, we are developing fast versions of the simulation to provide online feedback advise for hybrid network configuration and fault management.

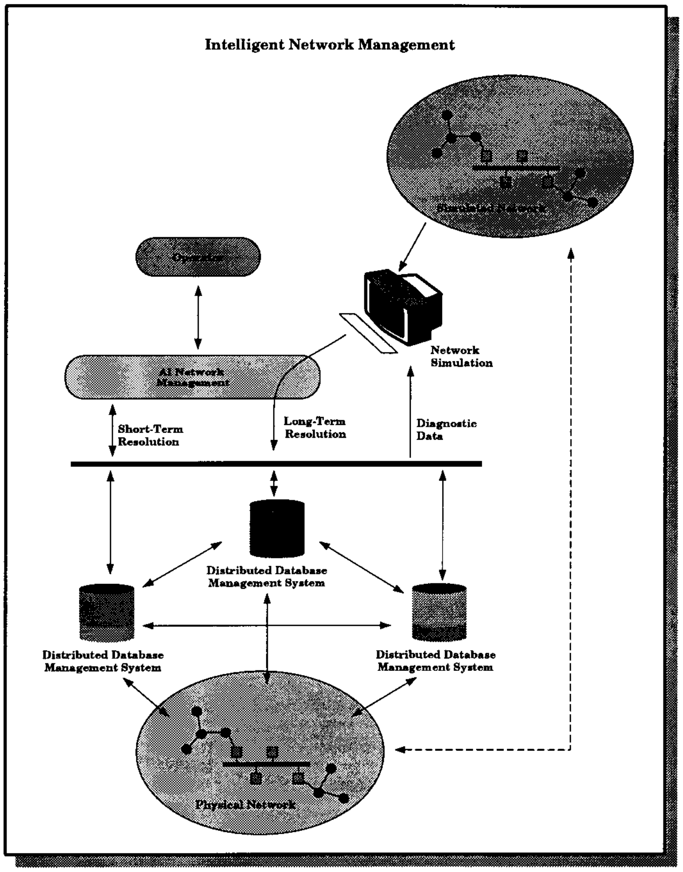

Intelligent Network Management
Simulated Network
Operator
Network Simulation
AI Network Management
Short-Term Resolution
Long-Term Resolution
Diagnostic Data
Distributed Database Management System
Distributed Database Management System
Distributed Database Management System
Physical Network

Fig. 7. Hybrid Network Simulation Software as a Network Management Tool

SUMMARY

This paper identified and discussed key issues involves in the simulation of complex hybrid communication networks, and presented the design features of an object-oriented hybrid communication software being developed at the University of Maryland, Institute for Systems Research. It was shown that conventional simulation paradigms: time-based simulation and event-based simulation, are unsuitable for complex system simulation; rather a model based simulation approach is adapted. It is also shown how the flexible interface design and incorporation of a database enable the software to be used as part of an intelligent network management tool.

BIBLIOGRAPHY

[1] Application of Teledesic Corporation for a Low Earth Orbit Satellite System in the Fixed Satellite Service, FCC File No. _DSS-P/L-94, Teledesic Corp., 5220 Carillon Point, Kirkland, Washington 98033, March 21, 1994.

[2] BONeS DESIGNER User's Guide, Version 2.5, Comdisco System Inc., FosterCity, CA, 1992

[3] GPSS/H User's Manual, Third Edition, Wolverine Software Corp., Annandale, VA, 1989.

[4] MODSIM II Reference Manual, CACI Products Co., La Jolla, CA, 1993.

[5] SIMSCRIPT II.5 Reference Handbook, CACI Products Co., La Jolla, CA, 1993.

[6] SES/workbench User's Manual, Release 2.1, Scientific and Engineering Software, Inc., Austin, TX, 1992.

[7] SIMAN V Reference Guide, Systems Modeling Corp., Sewickly, PA, 1993.

[8] SLAMSYSTEM User's Guide, Version 4.0, Pritsker Corp., Indiannapolis, IN, 1992.

[9] BONeS PlanNet User's Manual, Version 1.0, Comdisco System Inc., FosterCity, CA, 1992

[10] COMNET III User's Manual, Release 1.0, CACI Products Co., La Jolla, CA, 1993.

[11] Network II.5 User's Manual, Release 9.0, CACI Products Co., La Jolla, CA, 1993.

[12] L•NET II.5 User's Manual, CACI Products Co., La Jolla, CA, 1993.

[13] Simulation Software for Communications Networks: The State of the Art, Averill M. Law and Michael G. McComas, IEEE Communications Magazine, March 1994.

[14] Object Oriented Hybrid Network Simulation, A Functional Specification, Working Document, Institute for Systems Research, University of Maryland, College Park, MD.

AUTHOR BIOGRAPHIES

John S. Baras is a Professor in the Department of Electrical Engineering and the Institute for Systems Research, University of Maryland at College Park. He holds the Martin Marietta Chair in Systems Engineering and is a Fellow of the IEEE. He received his Ph.D. in Applied Mathematics from Harvard University in 1973. He has received several research awards for his work including the 1980 outstanding paper award from the IEEE Control Systems Society. He was the Founding Director of the Institute for Systems Research and the Founding Co-Director of the Center for Satellite and Hybrid Communication Networks. His current interests include intelligent control systems, control and systems theory, computer-aided design of control and communication systems, real-time signal processing and understanding, and computer/communication network management and simulation.

Ahsun H. Murad is a Faculty Research Assistant and doctoral student in the Institute for Systems Research/Electrical Engineering Department, University of Maryland at College Park. He graduated with a degree of Bachelor of Technology from the Indian Institute of Technology, Delhi, India (IITD) in 1989, and the degree of Master of Science from The University of Maryland, College Park in 1992, both in Electrical Engineering. He was a recipient of the Bambawala Award for the best project in Electrical Engineering, IITD for his design and hardware implementation of a high-speed HF modem in 1989. In addition to his research in error-control coding for magnetic data recording and distributed error-control schemes, his current research interests include the design and development of tools for hybrid communication network simulation and management.

Kap Do Jang is a Faculty Research Assistant in the Institute for Systems Research, University of Maryland at College Park. He graduated with a Bachelor of Arts in Economics from Sogang University, Seoul, Korea in 1987 and a Bachelor of Science in Computer Science from the University of Maryland at College Park in 1992. His current interests include object oriented design and programming, graphical user interfaces, and object-oriented database design.

George C. Atallah is a Senior Member of Technical Staff in the Institute for Systems Research, University of Maryland at College Park and a former Westinghouse and IBM research and development scientist. He received his Ph.D. in Systems Science and Applied Mathematics from the State University of New York at Binghamton. His current interests include advanced computer applications and architectures, neural networks, complex systems modeling and analysis, hybrid communication network simulation and management, nonlinear systems dynamics and control, systems reliability, maintainability and availability analysis, and systems/concurrent engineering methodology and management.

Ramesh Karne is a Senior Member of Technical Staff in the Institute for Systems Research, University of Maryland at College Park since October, 1993. Before joining the Institute, he was at IBM corporation for over 15 years and was involved with large computer system design, expert system applications and software development. He received the M.S. degree from the University of Wisconsin in 1978, and a Ph.D. in Computer Science from George Mason University in 1992. His research interests include object-oriented technology, artificial intelligence applications, distributed operating systems, communication systems, parallel and distributed simulation and application oriented architectures.

S. Joseph Campanella is an Adjunct Professor, Electrical Engineering, and Senior Research Scientist in the Institute for Systems Research at the University of Maryland. He retired from COMSAT Laboratories after 26 years of research and development in all aspects of commercial satellite communications. He had held the positions of Executive Director of the Communication Laboratory, V.P. and Assistant Director of COMSAT Laboratories and V.P. and Chief Scientist of COMSAT Laboratories. His current research interests include all aspects of satellite communications, hybrid network architectures and new satellites with on-board processing capabilities.