Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

CALL-BY-CALL SIMULATOR FOR TELEPHONE NETWORKS*

Harold M. Heggestad

MIT Lincoln Laboratory
Lexington, MA 02173-9108

Thomas Lam

Defense Information Systems Agency
Reston, VA 22090-5500

James W. Forgie

MIT Lincoln Laboratory
Lexington, MA 02173-9108

Constance K. McElwain

MIT Lincoln Laboratory
Lexington, MA 02173-9108

ABSTRACT

A Call-by-Call SIMulator (CCSIM) of a switched communication network has been developed under the sponsorship of the Defense Information Systems Agency. It has been used for a variety of applications involving engineering and network management studies for the Defense Switched Network (DSN), undergoing a series of evolutionary extensions in the process. It is now a uniquely powerful tool for telephone network simulation applications. The purposes of this paper are to describe the goals, functionality, design and implementation of CCSIM; to summarize the applications for which it has proven a valuable tool to date; and to describe its current state of development, particularly in the context of understanding how to exploit CCSIM in future applications.

1 INTRODUCTION

In the early 1980s the Defense Communications Agency (now redesignated the Defense Information Systems Agency, or DISA) was planning the transition from the aging AUTOVON military telephone system to the modern DSN. The computer control and common-channel signalling (CCS) systems in the new DSN switches, typified by the Northern Telecom,Inc.(NTI) DMS-100/200 series, offered opportunities for great performance improvement as well as challenges in designing the DSN to achieve these new goals. One area of particular interest was selection of call routing algorithms for implementation in the DSN switches, and DISA tasked Lincoln Laboratory with evaluating and recommending alternatives that would best satisfy military requirements in the DSN environment. The literature described many types of algorithms, but most were aimed at the commercial world; it was difficult to determine which were applicable, and how well, in the presence of special military functions such as multi-level precedence and preemption. Analytic evaluation of network routing algorithms is notoriously difficult, and out of this situation grew the project of creating the initial architecture of the CCSIM described below. Each interesting candidate routing algorithm was implemented in CCSIM, and comprehensive statistics were recorded for later analysis and comparison as each candidate was subjected to an identical series of simulation runs.

In the late 1980s the emphasis of DISA's tasking for Lincoln shifted to network management (NM) for the emerging DSN. AUTOVON NM had been rather straightforward, since the control options for the old AUTOVON switches were very limited; thus the existing cadre of AUTOVON network managers were not prepared by their training and experience to handle the greatly increased range of indicators and controls available to them in the DSN. The body of NM knowledge in the commercial world, where modern computer-controlled switches like the NTI DMS-100s had already been in use for several years, was largely inapplicable because the goals and constraints were very different from those of the military world. While commercial NM seeks to maximize revenue in a richly-connected network treating all customers as peers, in the DSN the goals are to assure service for critical military users in a cost-conscious, very sparsely-connected network, while providing the lower-precedence users with the best service they can get with the remaining resources of the network. Thus there was a need to create knowledge about NM for a new and much more complex DSN which did not yet exist. This situation led to revisiting CCSIM and implementing a number of modifications and extensions that transformed CCSIM into a testbed for NM experiments and knowledge generation. As the resulting body of DSN NM knowledge accumulated, it became clear that two additional applications were now feasible: a DSN NM

*This work was sponsored by the Defense Information Systems Agency.

operator trainer based upon CCSIM, and a DSN NM Expert System capturing the NM knowledge base in software. Both of these new applications were subsequently implemented; the Trainer is installed at the DISA Center for Engineering, and the NM Expert System is deployed as an operational tool in the European and Pacific Theater DSN Operations Centers.

The original CCSIM was written in RATFOR (RATional FORtran), a language that features C-like constructs for programming convenience but (after a pre-processing operation) produces conventional Fortran code. This was done to permit porting of the system to a variety of computers, and in particular to an older IBM mainframe in use at DISA at the time. Some of the routines added in more recent CCSIM modifications were coded in the C language, primarily to take advantage of existing software modules for handling UNIX socket communications. The graphics interfaces of CCSIM are written in XView.

CCSIM's domain of applicability is small to medium-sized networks, where the upper bounds are determined by available real memory and running time constraints for the platform in use. Typical array sizes on contemporary Sun workstations have been (e.g.) 50 switching nodes, 2,000 trunk groups, 10,000 simultaneously active calls, 18,000 generated calls in a run. On contemporary Suns with DSN theater-size networks and traffic levels, CCSIM typically runs faster than real time.

2 SUMMARY OF FEATURES

The features of the current CCSIM are briefly described as follows, and are elaborated in the succeeding sections:

Network Description:
- User-supplied files
- Switches, trunks, connectivity

Traffic Matrices:
- User-supplied files
- Average busy-hour call intentions:
  - Each source-destination pair
  - Five precedence levels
  - Two call types (voice and data)

Routing Options:
- Kinds of Tables:
  - Engineered
  - Forward
  - Modified Forward
- Adaptive (tables can be changed while running)
- Crankback

Preemption Options:
- Ruthless or Friendly Search
- Blind-Out or Blind-Back Preemption

Signalling Options:
- In-Band, CCS, or mixture

Traffic Generation:
- Dynamically variable
- Voice calls, data calls, or mixture
- Selectable ratios of precedences
- Variable busy-destination percentages

Call Processing:
- Models every en route switch
- Accounts for signalling and propagation delays

Switch Reports:
- Modeled after NTI DMS-100/200
- Turned on/off on per-switch basis

Statistics Generated:
- Six types for each of five precedences

Simulator Commands:
- From Graphics Interface and/or Command File
- Damage/restore switches and trunks
- Apply/remove NM controls

3 SYSTEM OVERVIEW

The purpose of this section is to provide a general overview of CCSIM functionality and operation.

CCSIM is a call-by-call simulator which, for a theater-wide telephone network, generates every source-to-destination call attempt; all route selection processing in all the switches; all blocking, preemption, call failure, retry and termination events; and two classes of statistics reports, namely comprehensive summaries of everything that happened during a run, and periodic switch reports that replicate the actual snapshots of data available to a network manager in the real network. CCSIM's clock runs in 0.1-second "ticks". Traffic overloads or damage to switches and trunks can be imposed by the operator at any time during a run, and the ensuing behavior of CCSIM will reflect real-world performance of the network under such conditions. NTI DMS-100 switch control commands are implemented in CCSIM, such that the operator can apply controls to remedy traffic problems or damage conditions, and can then watch CCSIM respond to the controls. Entire simulation runs can be replicated precisely, except that different control strategies can be applied in order to develop and evaluate NM techniques.

At the conclusion of a run, the comprehensive statistics reports can be scrutinized to determine exactly what happened and how well the control strategy performed. The limited "soda straw" views of the network state provided by the switch reports can be studied and compared in the light of the comprehensive statistics, so as to learn how to recognize problem statistics, so as to learn how to recognize problem conditions in the real network when only the switch reports are available. The relation between desired and actual network behavior can be studied, as an aid in planning succeeding experiments.

4 SYSTEM OPERATION

Figure 1 illustrates the inputs and outputs of CCSIM. The two double lines divide the figure into approximate thirds. At the left are shown the inputs that must be provided to initialize CCSIM at run time. The bottom of the figure shows the various inputs that can be applied at operator discretion during a run. At the right are the outputs produced by the system. Each block will be described briefly in the following paragraphs.

In the run-time inputs area of Fig. 1, the Network Description is a formatted file describing the size and topology of the network, as well as the size, type and nomenclature of the trunk groups connected to each node. The routing tables are applied in the call processing functions for each generated call as it traverses the network. The Traffic Matrices are estimates of the average busy-hour traffic in terms of "call intentions" for each source-destination pair (see the discussion of Fig. 2 below). For each call type (voice and data) there are five

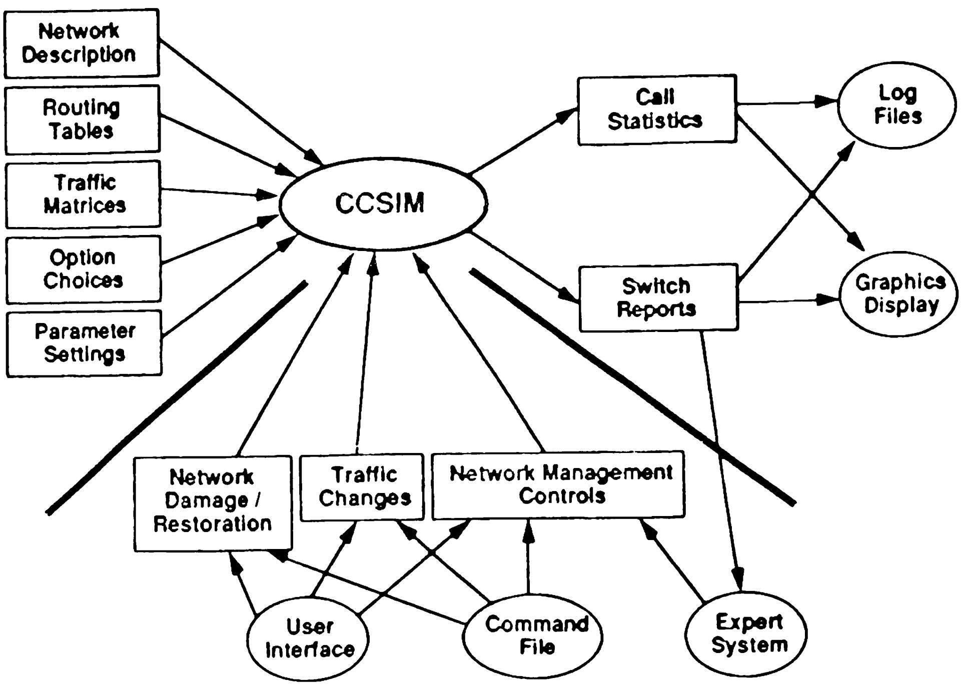

Figure 1: CCSIM Inputs and Outputs

such matrices, one for each of the call precedence levels from Routine through Flash Override. A source or destination means one of the switch nodes in the network, and the Traffic Matrices lump together the traffic for all the end users connected to the corresponding switch node. The elements of the Traffic Matrices are used to set the average call arrival rate for the Poisson (or other desired) call generation statistics models that run in CCSIM. The Option Choices and Parameter Settings blocks in Fig. 1 are extensive files of input variables that can be used to represent a wide variety of network models and conditions.

Discussion of the Expert System in the operator inputs section at the bottom of Fig. 1 will be deferred to a later paragraph. Note that a human operator can exert control over CCSIM during a simulation run via either an on-line user interface or a prepared command file. The former is a set of interactive mouse-sensitive graphics screens which are convenient for NM operator training and for experimentation, while the latter is particularly useful for executing and repeating complex scenarios under precise control. The three categories of inputs are (1) damage and restoration of selected network components such as switches and trunks; (2) localized or network-wide changes in traffic patterns and intensities; and (3) application and removal of network management control commands to remedy problem conditions in NM training scenarios, or to explore techniques in NM experimentation scenarios.

Two classes of reports are generated by CCSIM as a simulation run progresses, as shown at the right-hand side of Fig. 1. The Call Statistics are comprehensive records of counts, events and parameters selected by the operator from a large list of options. They are automatically dumped to files for analysis at leisure, and they are also available for viewing in color graphics displays designed for convenience in evaluation of the current experiment and planning of the next move, parameter changes, etc. The Switch Reports replicate the data that is presently reported live by DSN switches to the theater operations center: at regular intervals (five minutes in Europe, fifteen minutes at present in PAC) each switch sends a formatted message summarizing key counts and status indicators for the preceding interval. Because of limits on data reporting bandwidth and on real-time human comprehension capability, these Switch Reports are a small and carefully selected subset of the large range of operations measurements that could be provided by the switches.

The Expert System at the lower right in Fig. 1 was mentioned earlier; it was created by encapsulating in software the NM knowledge obtained through experimentation with CCSIM, as well as through knowledge engineering with available outside sources of NM expertise. A significant part of the development of the Expert System was done in the connection mode shown in Fig. 1, where the Expert System was actually managing the simulated network: the human operator could inject problems, the Expert System would diagnose them by analyzing the Switch Reports, and the Expert System would apply NM controls to CCSIM to remedy the problems. Since then the Expert System development has diverged, focusing on functioning in the real DSN, and it no can longer be used as in Fig. 1. Figure 2 illustrates call processing in CCSIM, and stepping through this flow chart will help to explain how CCSIM operates. There are five logical Call Generators for each source-destination pair in the network (one for each precedence level). A Call Generator is a random process (typically Poisson) whose average call rate is equal to the corresponding entry in the Traffic Matrix described above. After every 0.1-second clock tick, CCSIM notes which Call Generators (if any) are now announcing that a "call intention" has been created. The "call intention" concept represents a need for a user to complete a call to another user, of a randomly-selected (typically exponential) duration. One call intention could result in many call attempts, blocking or preemption events, and retries before the user manages to complete the call intention as desired, or gives it up. Thus the total number of call attempts observed in the switches of an actual network can be many times greater than the number of call intentions.

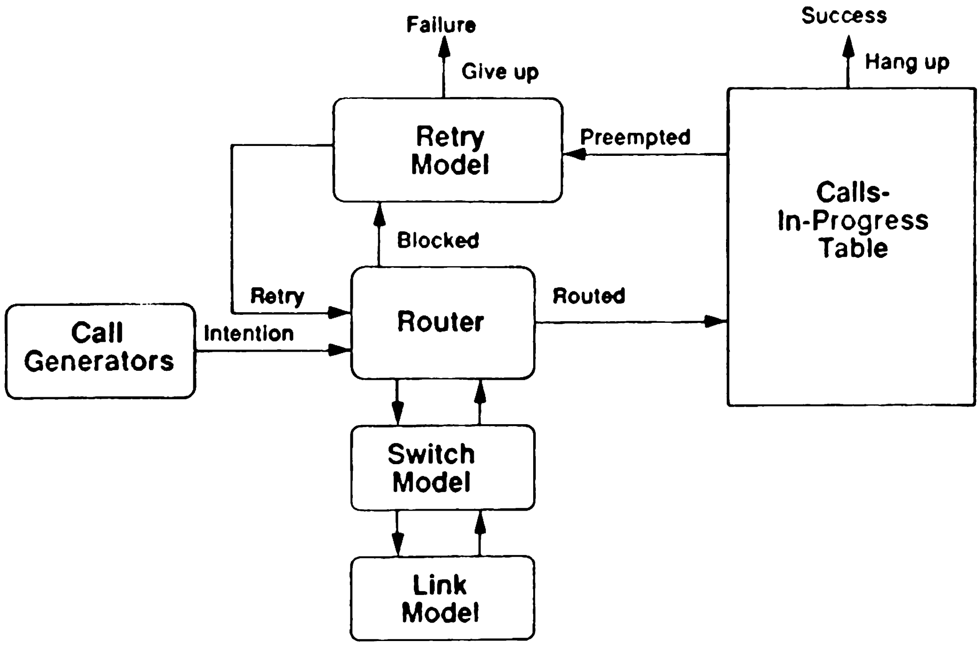

Figure 2: CCSIM Call Processing

The creation of traffic matrices is straightforward if one is designing a new network and postulating what its traffic will be. It can be very difficult, on the other hand, to work backwards and derive traffic matrices that will produce simulated call patterns in CCSIM that match observed traffic in a real network. The reason is that the call counts measured and reported by the real switches include all the retries (indistinguishable from original call attempts), and the retry behavior is a complex function of human psychology, traffic intensity, current call blocking levels on the trunk group being accessed, and other intangible factors. The derivation of traffic matrices from measured call data is a protracted cut-and-try operation, with no guarantee that one will converge on correct answers.

Each call intention produced by the Call Generators is passed to the Router in Fig. 2, which invokes a model of the route selection processing for each switch in the call path, together with models of the interswitch links involved. If the call is successfully routed it is entered in the Calls-in-Progress Table, which it can leave either by being preempted or by achieving its planned total duration. Calls which are blocked or preempted are retried in accordance with a user-selectable probabilistic model. CCSIM tracks every call intention from birth to death, keeping complete statistics for user analysis if desired.

5 SUMMARY

A telephone network simulator has been developed and matured over some ten years, and has been used for various purposes in supporting the development of the DSN. It is written in Fortran and C, and is compatible with UNIX operating systems. While its functions and features are tailored to military network requirements, it is highly flexible and can also be configured to suppress the military features and match commercial network characteristics. Its domain of applicability is small to medium-sized networks. It can be used for many purposes, including network design and engineering, network management training, and development of contingency and emergency plans.

REFERENCES:

Lincoln Laboratory Annual Report - Defense Switched Network Technology and Experiments Program, ESD-TR-82-167, 30 September 1982

Lincoln Laboratory Annual Report - Defense Switched Network Technology and Experiments Program, ESD-TR-83-228, 30 September 1983

Lincoln Laboratory Annual Report - Defense Switched Network Technology and Experiments Program, ESD-TR-85-172, 30 September 1984

Lincoln Laboratory Annual Report - Defense Switched Network Technology and Experiments Program, ESD-TR-86-006, 30 September 1985

Lincoln Laboratory Annual Report - Knowledge-Based System Analysis and Control Defense Switched Network Task Areas, ESD-TR-87-268, 30 September 1987

Lincoln Laboratory Annual Report - Knowledge-Based System Analysis and Control Defense Switched Network Task Areas, ESD-TR-88-314, 30 September 1988

Lincoln Laboratory Annual Report - Knowledge-Based System Analysis and Control Defense Switched Network Task Areas, ESD-TR-90-141, 30 September 1989

Lincoln Laboratory Annual Report - Knowledge-Based System Analysis and Control Defense Switched Network Task Areas, ESD-TR-90-151, 30 September 1990

Lincoln Laboratory Annual Report - Knowledge-Based System Analysis and Control Defense Switched Network Task Areas, ESD-TR-91-247, 30 September 1991

Lippmann, R. P. "Survivable Routing Procedures for Circuit-Switched Satelite-Terrestrial Networks", Proceedings ICC-83

Lippmann, R. P. "Routing Algorithm Development and Evaluation for Circuit-Switched Mixed Media Networks", Proceedings ORSA TIMS, Dallas, 1984

Lippmann, R. P. "New Routing and Preemption Algorithms for Circuit-Switched Mixed Media Networks", MILCOM Oct 1985

AUTHOR BIOGRAPHIES

HAROLD M. HEGGESTAD received the B.S., M.S. and Sc.D. degrees in Electrical Engineering and Communication Theory from the Massachusetts Institute of Technology. After service in the U.S. Navy he joined Lincoln Laboratory, where he has worked for three decades for various DoD sponsors in radar systems development and field operation, satellite communications, networking, and network management expert systems. He is currently Associate Leader of the Machine Intelligence Technology Group.

CONSTANCE K. MCELWAIN received the B.S. in Mathematics from Tufts University. She has been a Staff Member at Lincoln Laboratory since 1955, working in software systems design and implementation. She is the lead implementer and maintainer of CCSIM.

JAMES W. FORGIE received the B.S. in Electrical Engineering from the Massachusetts Institute of Technology. He has been associated with MIT Lincoln Laboratory since 1951, working in various areas of computer hardware and software system design and application. Currently he is a Senior Staff Member, working on application software for communication system fault diagnosis and network management.

THOMAS LAM received his B.S. in Electrical Engineering from George Washington Unversity, and his M.S. in Electrical Engineering from Renseelear Polytechnic Institute. He is completing his Ph.D. in Information Technology and Engineering at George Mason University. Mr. Lam's interests are in the areas of Artificial Intelligence (AI) technology, Decision Support Systems (DSS), and Network Simulation. He is currently with the Defense Information Systems Agency (DISA), Center For Engineering (CFE) and serves as Lead Systems Engineer for the development, implementation, and testing of network management capabilities for the Defense Information Systems Network (DISN) subsystems.