# AN-TTC-39 CIRCUIT SWITCH SIMULATION

R. Green and M. Fox
GTE Sylvania, Inc., Needham, MA


## Abstract

A GASP IV based discrete event simulation program has been written to model the performance of a state-of-the-art military telephone circuit switch under development at GTC Sylvania Incorporated.

The simulation model represents the switch as a finite source queueing system. GASP IV queues are used to model the switch program priority levels. The simulation employs a collection of tables to dictate the processing sequence and the resources required to service interfacing line types. This table structure allows for easy modification of model representation of switch software design. The model also uses GASP-IV queues and events to simulate the effects of both synchronous and random stimuli, to simulate delays in communicating with the hardware, to simulate human response delays, and to simulate the priority determined initiation of program execution.

Simulator outputs include indications of processing delays, measures of the extent of utilization of switch resources including CPU time and measures of switch throughput.

The simulator program user may vary design parameters such as instruction estimates, program priorities and switch resource levels and present varying traffic loads and termination mixes to the model. He may then examine the program outputs to determine whether the system design meets requirements and to evaluate design changes.


## Introduction and Background

The AN-TTC-39 Circuit Switch is an advanced computer controlled switching system being developed at GET Sylvania, Needham, MA for the U.S. Army Electronics Command, Fort Monmouth, New Jersey as part of the Joint Tactical Communications Program (TRITAC).[1,2]

Early in the TTC-39 design phase it was decided that it would be desirable to be able to evaluate the capability of the proposed hardware and software design to meet the system timing and response requirements as well as to evaluate reserve system timing capacity. Analytical techniques, though useful, were discarded as a prime tool since the complexity of the interactions and the non-standard statistical distribution of the events made standard techniques less tractable. Real-time loading of the actual fielded hardware, while the ultimate test of a design, has the disadvantage of requiring the final system after hardware-software integration. If a problem is found at this state of system development, changes to the hardware or software historically tend to be quite costly. Therefore, since it was deemed to be important to evaluate the ongoing design, a simulation was selected to provide this capability.

The completely developed simulation offers the further advantages of allowing relatively easy modelling and evaluation of proposed system architecture or interface modifications. In addition, such a simulation allows potential buyers to evaluate switch performance in differing deployments with differing interconnected devices and traffic mixes.

A similar simulation model using GASP has also been developed for the TTC-39 message switch but this paper is limited to the circuit switch application.

## Language Selection

One of the main considerations in selecting a language in which to develop the simulation was that it be capable of operating on a wide range of different computers operating throughout the government community. In particular, it was desired to be able to run it on a Burroughs B5700 at Fort Monmouth as well as the IBM 360/370 family of computers readily available to the development effort at GTE Sylvania. This dictated the choice of a simulation language which is written in a commonly used higher level language. GASP,[3,4] which is a FORTRAN based language was chosen as the simulation language since it met the above criteria and since GTE Sylvania had some experience in the language. GASP is up and running on both the Burroughs B5700 and IBM 370/145.

In the discussion below, we first describe the telephone switching problem, and follow with a presentation of the model synthesis. This discussion indicates the complexity of the system being modelled and the flexibility of the simulation tool in accounting for detailed interactions whose effects can not be easily entered into analytical analyses.

## The Telephone Switching Problem

The philosophy of switching is that connections between subscribers are both transitory and variable. The demand for the facilities needed to provide the communication link is determined from postulated traffic statistics. The design of a switching system involves analyzing this probabalistically distributed demand to size the quantities and capacities of those resources which can, in effect, be shared by many subscribers and thereby provide a cost effective system. In computer controlled switching systems one of the principal shared resources is the processing system. Based on anticipated call statistics, real-time processing capacity and storage requirements can be minimized, allowing for reduced processor speed and memory size requirements, thus reducing overall system cost. Two measures of the adequacy of the resource pools and switch design are the average time for the calling party to get "dial tone" and the average time from completion of dialing until "ringing" of the called party. Some of the factors influencing these response times are illustrated by the discussion of a simple analog call.

## A Simple Call

In Figure 1 we illustrate some of the steps in making a simple call.

A hardware device called a scanner is dedicated to every m lines entering a typical switch. The scanner periodically examines each line looking for supervisory signals from the telephone instrument. In a typical call the user deciding to make a call picks up the phone. The "off hook" action, called Seize, causes a frequency change to be detected by the scanner. The time of detection varies as a function of how often the scanner scans all m terminals and how long it must stay on each line to register a valid signal, i.e., dwell time. In the TTC-39 design this change of state is then passed to the Control Processor and the Switch Control Software (step 1, Figure 1). This processor must then assign a device called a sender, to transmit the familiar audible dial tone to the calling party, and a device called a receiver, to detect and receive the digits that are dialed by the calling party (step 2). Since the sender and the receiver are only required for a short period of any call, they may be provided in a pool to be assigned by the processor to the calls as the demand occurs. The size of the pool is derived on a statistical basis using standard telephone engineering techniques. The deviation of the actual call statistics from the projected statistics will determine the adequacy of the pool and reflect on the delay for the user to get dial tone. Once the user receives dial tone he starts to dial. Dialing time depends on his familiarity with the number and the length of number which may be normally 3 to 10 digits plus special codes. The processor receives these digits as they are dialed and after a suitable number of digits are received, determines the called party, finds a route through the switching equipment to the called party, and causes the commands to ring him if he is not busy (step 3). When the called party answers, the "off hook" signal processed by the scanner is interpreted by the processor as an "answer" and the two parties are connected, (step 4). They then talk for a statistically derived length of time. When one party hangs up, the "on hook" frequency change is detected again on the scanner (step 1) and is presented to the processor and the software goes through its process of disconnecting the parties.

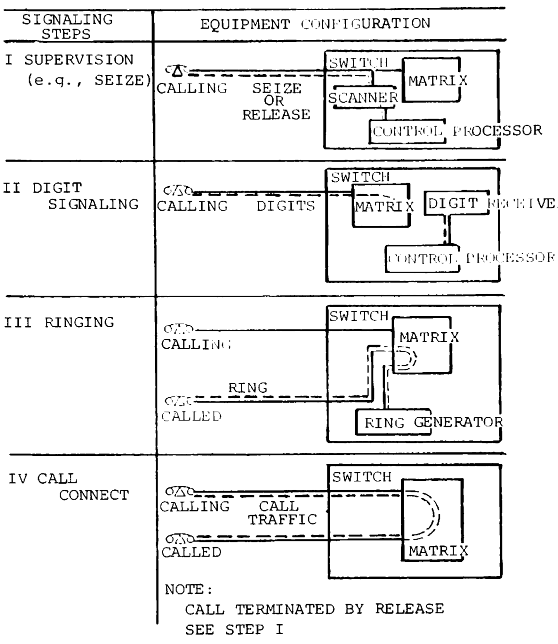


Figure 1 - A simple call

Note, this was a very simplified local call. An actual call involves more detailed interactions with the switching hardware, in particular the switching matrix, which have been eliminated from this discussion for the sake of clarity. These interactions are, however, largely provided in the simulation model. Calls can also be non-local and require signalling to other switches.

Software Control of Switch Resources

The Circuit Switch software controls the state of the switch hardware causing the appropriate signalling and supervision to be performed for a given termination. The software determines which connections and disconnections must be performed and which signals must be transmitted to an interfacing line type at each point in a processing sequence. It then correspondingly formats commands into an Input/Output (I/O) buffer and initiates the block transfer of commands to the associated interface. After a delay for transfer of I/O and for hardware response, status words are returned to a designated buffer from the interface indicating the disposition of the commands. The software design is a critical element in achieving the system response time requirements. Some discussion is necessary to understand the simulation model.

Software Structure and Timing

An AN-TTC-39 Circuit Switch programs may be divided into two categories:

(a) I/O and executive programs

(b) Call processing programs.

Each of these categories consists of a collection of program modules each of which is assigned an index called the 'level.' The execution of programs is initiated in priority order amongst those programs which are scheduled. The program's level dictates its priority. The scheduling of these program modules or levels occurs in response to hardware interrupts which may be synchronous or asychronous in nature. Synchronous interrupts are caused by two real-time clock timers which run out every 100 milliseconds (ms.) and 20 ms. respectively.

The executive program responds to the 100 ms. (major cycle) interrupt by scheduling all but two of the program levels and to the 20 ms. (minor cycle) interrupt by scheduling the remaining two levels. Asynchronous interrupts occur due to the filling of various input software buffers such as that used for scanner inputs and to the emptying of various output buffers, e.g., the line printer output buffer, and due to various other normal and abnormal equipment responses.

I/O related interrupts cause the scheduling of a high priority program level which "resets" the buffer to permit continuation of I/O, if possible.

The executive program initiates processing by the scheduled program with highest priority. Service by the program level continues until either it has completed all work required of it, (it then becomes no longer scheduled), or an interrupt occurs. An interrupt suspends operation by the

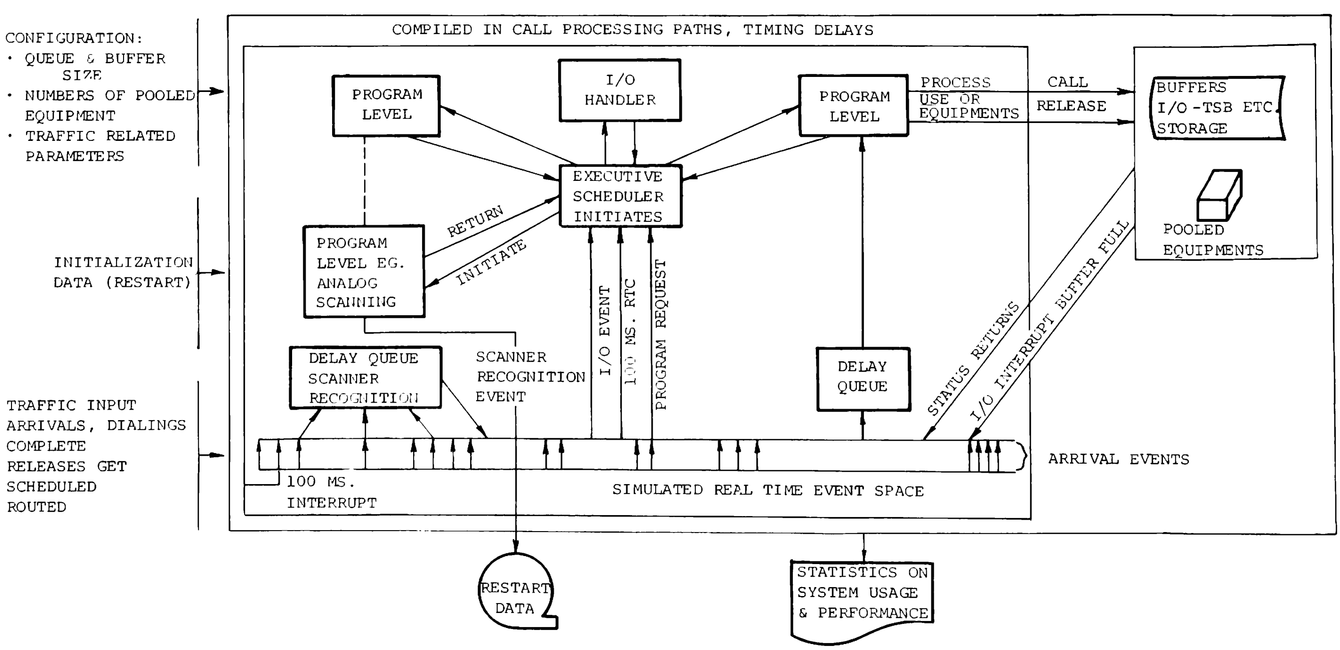


Figure 2 - System timing and interactions

program level and causes the executive to schedule one or more program levels. The executive then returns control to the highest priority level then scheduled. This may not be the program which was interrupted. However, when control is returned to a suspended program, it resumes processing.

Figure 2 illustrates those system interactions which affect timing as they are represented to the simulation.

## The Simulation Model

The Simulation model program has been developed in three parts. These perform the traffic generation, system simulation and post processor or report generation functions (see Figure 3).

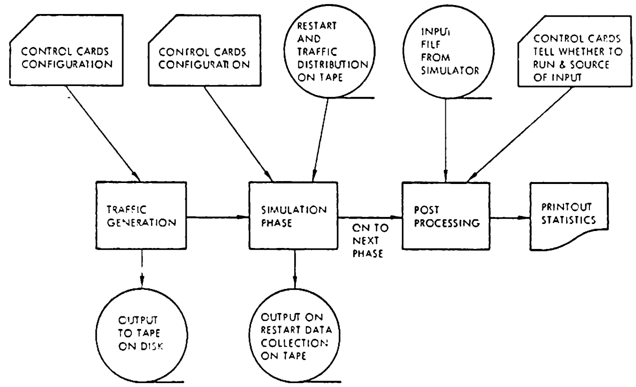


Figure 3 Three Phase System

## The Traffic Model

The traffic generation phase is designed to produce the stream of simulated requests for service which arrive at the switch. This traffic stream is written on a permanent file, and may then be presented to a variety of switch configurations as defined by the parameters specified to the system simulation phase.

The user characterizes the termination on the modelled switch by declaring the following information:

(a) type of interfacing device

(b) the distribution of interarrival intervals (intervals between calls)

(c) the distribution of holding intervals (length of calls)

The finite source nature of the system being modelled greatly complicates the use of the external traffic file. The exact time of the arrivals on a termination cannot be specified prior to running the simulation since it is not known if the termination will be available to initiate calls at the indicated times. Processing delays which we cannot precisely predict and which we in fact intend to derive from the model along with the interactions with calls from other subscribers might cause the termination to be busy at the indicated arrival time.

The approach taken is to associate with each arrival the time interval between it and its predecessor (time zero for the first arrival). This is accomplished by sampling a random deviate from the projected distribution. An approximate arrival time for use in determining when the information is to be made available to the simulation is also generated. The approximate arrival time is defined to be the value obtained by adding the dialing time, holding time and interarrival interval lengths to the approximate arrival time of the previous call on the termination. This value indicates the time at which the arrival would occur neglecting processing delays and calls from other subscribers.

When the information is read it is determined if no other arrivals are available for use with the termination and that the termination is idle. If so, then this arrival is scheduled, otherwise it is placed on a queue in order of approximate arrival time and arrivals from this queue are scheduled each time the termination becomes idle. The arrivals are scheduled to occur at a time interval equal to the interarrival interval after the last entry to idle state. The scheme described here makes the arrivals available in such a way as to avoid biasing the interarrival statistics and at the same time keeps the requirements for the number of readings of the traffic file and for program storage of traffic events within bounds.

## The System Performance Model

One GASP IV queue is assigned corresponding to each program level which is modelled and to each buffer from which the block transfer of I/O may be initiated. An additional queue is assigned to model and delays which may affect calls individually, e.g., for modelling dial delays and scanner recognition delays.

The events modelled are:

a. major and minor cycle interrupts

b. status return interrupts

c. buffer full interrupts

d. traffic file reading event

e. end of delay event

f. program initiation event

The consequences of the interrupts are modelled by setting entries in an array to indicate the scheduling of appropriate program levels and by placing entries on queues to signify the processing demands associated with servicing the interrupt.

The traffic reading event causes a buffer full of arrivals to be read from the traffic file as discussed previously.

The program initiation event code searches an array to find the index of the highest level program which is scheduled and calls a subroutine to process entries on the queue associated with that level. This program executes three modules in sequence to simulate the consequences of program level execution. Two of the queue entry attributes, set when the entry is placed on the queue, indicate the number of program instructions needed to complete processing the entry and the entry state index. First it is determined if there is sufficient simulated time until the next scheduled event to execute the necessary number of instructions. If not, the entry attribute specifying the processing time requirement is adjusted to reflect the use of all available time, the entry is replaced on the queue and a return is made to process the next event. In any case, a record is kept of the CPU time expended in the program level.

The next module determines if all resources required for completion of servicing of the entry are available. This module is entered only if the requirement for CPU time has been satisfied. The number of resources in use as recorded in a state vector is adjusted according to the satisfaction of demands. If this new level of utilization indicates that the capacity of an input buffer (scanner, receiver, trunk signalling) has been reached then a buffer full interrupt event is scheduled. If all required resources are not available, the entry state is changed to a new state and tables dictate the subsequent alteration in the call processing sequence.

The final module executed places entries on the next queue or queues in the call processing path. The entry attribute which defines its state is updated as is the processing time required attribute if the queue corresponds to a call processing level. If the queue index value exceeds the maximum index for a call processing, I/O, or delay queue, then it is used instead to direct a branch to code which does special processing. For example, the entry may have reached a point in its path which is the terminus of a path segment whose timing we are interested in, as for the time to traverse the path segment between seize and dial tone. The code to which the program branches will call GASP IV data collection routines to record the appropriate timing statistics. This structure is also used in modelling delays in the Call Processing sequence. One instance which illustrates this procedure is in handling the delay from dial tone to receipt of the first digit. The special code places an entry on the "delay queue" and schedules an end of delay event to remove the entry from the queue after an interval determined by a statistic on the time required to dial a digit.

When the event occurs the entry is removed from the delay queue and processed by the resource utilization and queuing modules. Delays in block transfer of I/O are modelled by placing entries on the unique queue which corresponds to the I/O buffer from which commands are to be issued. When the status return interrupt occurs, all entries are removed from the queue and again processed by resource utilization and queueing modules.

In the case of a call processing queue successive entries are serviced until either one entry encounters lack of processor time, as described previously, or until the queue is empty. If the queue becomes empty then the array indicating the scheduled program levels is modified and return is made to the program initiation event code to process the queue to the new highest priority program.

Simulation Experiments and Results

Statistics are collected during the simulation run primarily through the use of the GASP IV data collection routines. The resulting data arrays are written onto a permanent file and may then be reduced at user convenience by passing them through a post-processor phase which produces labelled reports.

The reports consist of indications of:

(I) Grade of Service Provided Subscribers in form of:

(a) histogram of time to dial tone from off-hook

(b) histogram of time to start of ring from completion of dialing

(c) histogram of time from release until subset marked available for further call initiation

(d) number of unsuccessful calls tabulated by cause — e.g., called party busy, blocking in space division matrix.

(II) Use of CPU Time

(a) histogram of % spare time per major cycle

(b) histogram of % of major cycle spent in each program level

(c) amount of simulated time spent in overhead functions.

(III) Use of Switch Resources, Pooled Equipment

(a) plots of number of calls in progress versus time (calls in progress is from off-hook to idle)

(b) for each pooled resource or I/O buffer a plot indicating the number in use (storage required) versus time

(c) a table indicating minimum, maximum, mean, standard deviation of numbers of users of each pooled equipment and of the level of storage in each buffer or queue.

Attached are some sample outputs from a short simulation run. The plot which has been produced using the GASP IV plotting routines indicates the level of utilization of several resources as a function of time.

The plot lines indicate numbers of:

(a) LKG's in use (LKG is a pooled security device) — LKG

(b) DTMF receivers in use — REDTM

(c) call registers in use — CALRG

(d) half paths through the space division matrix — SDMPT

(e) one word inputs into the scanner buffer received since the last interrupt — SCBUF

as a function of time. These plots serve as an indication that resource pools are adequately sized and also lend additional confidence in the accuracy of the simulation since the close correlation in such variables as the number of receivers and call registers in use is known independently of the simulation.

The first histogram indicates the extent to which the processor capacity is utilized, the measure being the distribution of the samples of spare time in each 100 millisecond major cycle.

The final histogram demonstrates the adequacy of the system in meeting one of the timing requirements. The histogram indicates the distribution of delays from the end of dialing until the ringing of the called party is initiated. Experience to date in running the simulation indicates a ratio of simulation program run time to the amount of switch operation time being simulated as indicated by:

$$1.5 \le \frac{\text{Simulation Program Executive Time}}{\text{Switch Operation Time Interval Modelled}} \le 2.0$$

The simulation programs have been used to model two successive design states for switches configured with a limited number of phone types. The preliminary results provide a high degree of confidence in the ability of the switch to meet the throughput and timing requirements. The results also indicate that resource pools have been adequately sized to meet the offered traffic, albeit from a limited selection of line types.

Present plans indicate that the simulation model tables will be enhanced to include calls between a wider range of line types, allowing the validation of switch performance in a considerable range of configurations.

References

1. "The Joint Tactical Communcation Program," Major General John E. Hower, USA SIGNAL, July 1975.

2. "The AN/TTC-39 Switching System: A Bridge to Digital Communications of the Future;" Jerome A. Blackman Proceedings of TELECOM 75 (COM SOC, G-BCTU) October 6-8, 1975.

3. GASP IV, by A. Alan B. Pritsker, Pritsker and Associates, Inc., West Lafayette, Indiana, 1973.

The GASP IV User's Manual, by A. Alan B. Pritsker, Pritsker and Associates, Inc., West Lafayette, Indiana, 1973.

4. Introduction to Queuing Theory, Robert B. Cooper, The Macmillan Company, New York, 1972.

5. Fishman, G. J., Concepts and Methods in Discrete Event Digital Simulation, New York: Wiley Inter-science, 1973.

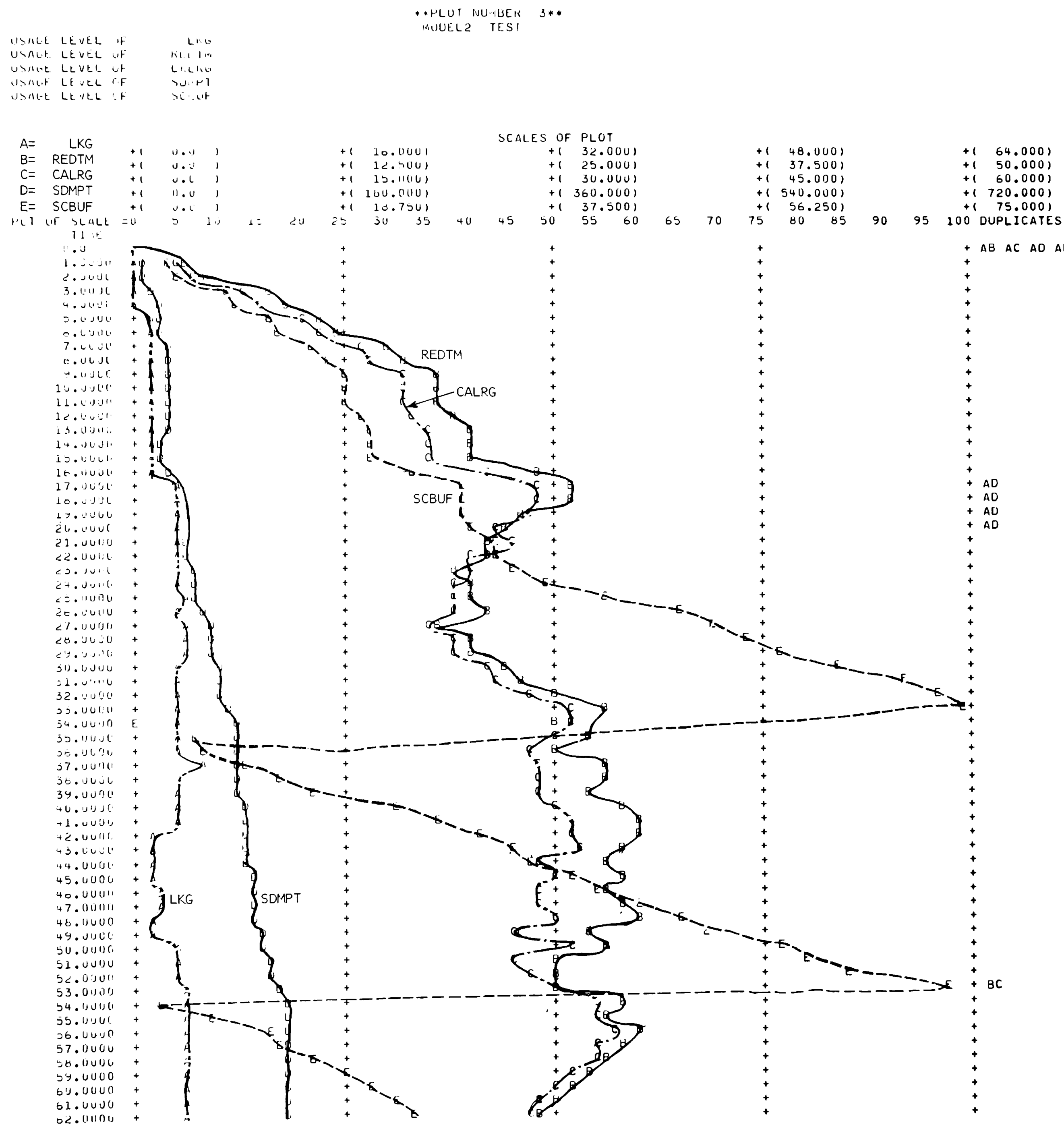


Figure 4 Plot of Pooled Equipment Usage Level

```
                     ::::HISTOGRAM NUMBER  1::::

MAJOR CYCLES VS SPARE TIME (MSEC)

OBSV  RELA  CUML    UPPER
FREQ  FREQ  FREQ CELL LIMIT 0         20        40        60        80       100
                            +    +    +    +    +    +    +    +    +    +    +
   0 0.0   0.0   0.1000E-05 +                                                 +
   1 0.001 0.001 0.1000E+02 +                                                 +
   2 0.002 0.003 0.2000E+02 +                                                 +
  11 0.011 0.014 0.3000E+02 +:                                                +
  44 0.044 0.058 0.4000E+02 +::C                                              +
 106 0.106 0.164 0.5000E+02 +:::::  C                                         +
 338 0.338 0.501 0.6000E+01 +:::::::::::::::::       C                        +
 498 0.498 0.999 0.7000E+02 +:::::::::::::::::::::::::                        C
   0 0.0   0.999 0.8000E+02 +                                                 C
   1 0.001 1.000 0.9000E+02 +                                                 C
   0 0.0   1.000 0.1000E+03 +                                                 C
   0 0.0   1.000    INF     +                                                 C
 ---                        +    +    +    +    +    +    +    +    +    +    +
1001                        0         20        40        60        80       100
```

Figure 5 - Histogram spare time in major cycle

```
PAGE 190
                     ::::HISTOGRAM NUMBER  3::::

CALLS VS TIME (SEC.) FROM END OF DIALING TO START OF RING

OBSV  RELA  CUML    UPPER
FREQ  FREQ  FREQ CELL LIMIT 0         20        40        60        80       100
                            +    +    +    +    +    +    +    +    +    +    +
   0 0.0   0.0   0.0        +
   0 0.0   0.0   0.1000E+00 +
   0 0.0   0.0   0.2000E+00 +
  27 0.329 0.329 0.3000E+00 +::::::::::::::::
  47 0.573 0.902 0.4000E+00 +:::::::::::::::::::::::::::::               C
   0 0.0   0.902 0.5000E+00 +                                            C
   0 0.0   0.902 0.6000E+00 +                                            C
   3 0.037 0.939 0.7000E+00 +::                                            C
   5 0.061 1.000 0.8000E+00 +:::                                              C
   0 0.0   1.000 0.9000E+00 +                                                 C
   0 0.0   1.000 0.1000E+01 +                                                 C
   0 0.0   1.000 0.1100E+01 +                                                 C
   0 0.0   1.000 0.1200E+01 +                                                 C
   0 0.0   1.000 0.1300E+01 +                                                 C
   0 0.0   1.000 0.1400E+01 +                                                 C
   0 0.0   1.000 0.1500E+01 +                                                 C
   0 0.0   1.000 0.1600E+01 +                                                 C
   0 0.0   1.000 0.1700E+01 +                                                 C
   0 0.0   1.000 0.1800E+01 +                                                 C
   0 0.0   1.000 0.1900E+01 +                                                 C
   0 0.0   1.000 0.2000E+01 +                                                 C
   0 0.0   1.000    INF     +                                                 C
 ---                        +    +    +    +    +    +    +    +    +    +    +
  82                        0         20        40        60        80       100
```

Figure 6 - Histogram indicating grade of service