The Space Congress® Proceedings

1969 (6th) Vol. 2 - Space, Technology, and Society

Apr 1st, 8:00 AM

GENCOM - A Generalized Communication Model Concept

W. V. Neisius

TRW Systems

GENCOM - A GENERALIZED COMMUNICATION MODEL CONCEPT

W. V. Neisius
TRW Systems Group
Redondo Beach, California

ABSTRACT

The paper describes the capabilities of a General Purpose Communication Network Simulation Program called GENCOM and traces the background leading up to its development. GENCOM, as a versatile, flexible, relatively simple program was conceived by the analysis of a complex, special purpose program written for the evaluation of the Minuteman Communication Network.

The simulation of the Minuteman Communication network was written in Simscript and modelled in detail the complex procedures for the flow of messages through the system. Unexpectedly, the simulation showed that certain second order effects could cause the system to "ring" in such a manner as to seriously interfere with successful message transmissions. This phenomena would not have occurred in the system as originally designed and could be traced to a "minor" system modification. In addition to uncovering this problem, the simulation provided the means of investigating possible solutions and identifying the most efficient one.

The success of this study spotlighted the desirability of using a simulation during the development of a system rather than after the design is frozen. The special purpose Minuteman program was analyzed to identify general techniques which might be applicable in a wide variety of communication networks. The results of this analysis led to the development of a completely General Purpose Communication Network Simulation program called GENCOM.

Although GENCOM, as originally derived from the Minuteman Simulation, was specifically oriented toward military tactical communication networks, it has been shown to be applicable in many other areas as well. Examples include the modelling of the communications network of a large metropolitan police department and the data management system of a space probe. In addition, a modified version of GENCOM has been developed for evaluating the operating procedures of complex command and control networks.

GENCOM is a program, written in SIMSCRIPT I.5 designed to facilitate the simulation of a wide variety of possible communication networks. Although this paper will attempt to describe GENCOM in some detail, it is believed that the background leading up to the development of GENCOM is even more interesting than GENCOM itself.

The idea to develop GENCOM occurred in September of 1967. In less than one man month of effort, the specifications were prepared, the program written, test cases were developed and checked out, and GENCOM completely documented. Obviously, to accomplish this rather non-typical time scale for the development of a computer program of this complexity, there must have been a reason. The fact is that the essential elements of GENCOM had unknowingly been written almost two years before. In September of 1967 it was merely recognized, identified, and documented. This was accomplished by taking a previously written, extremely complex, sophisticated communication network simulation and deleting all special network features. The skeleton program remaining, spotlighted the necessary specifications for a Generalized Communication Model Concept.

The original communication simulation was written for the Minuteman program during the last six months of 1965. It was a simulation designed to investigate the flow of messages in a Minuteman Squadron.

A Minuteman flight consists of a Launch Control Facility (LCF) and ten Launch Facilities (LF). A communication line connects each LF to its "parent" LCF (See Figure 1). Each LF, in addition to being connected to one LCF, is also connected to a maximum of five other LFs. Typically, it would be connected to two other LFs within its own flight, and three LFs in other flights (See Figure 2). A squadron consists of five flights. Interrogations are sent from LCFs to LFs, and responses are sent from LFs back to LCFs by means of a network "saturation" process. That is, the LCF originating an interrogation addressed to a particular LF will not just send the message to that one LF, but will send the same message, in parallel, to all LFs. Each LF, in turn, will immediately begin retransmitting the message on all of its available lines and in a fraction of a second the message will exist on all 350 lines of the squadron. The purpose of this is to insure that even with many broken

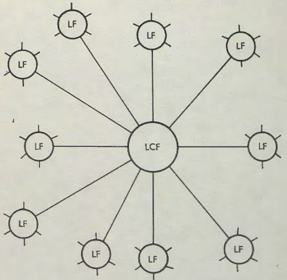

FIGURE 1

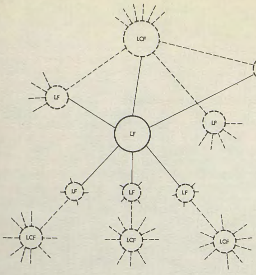

FIGURE 2

cables and possible down LCFs and LFs, any LCF will have the maximum probability of communicating with any LF still operable. A basic interrogation/response cycle requires approximately 1.5 seconds. This interval of time is called a "time slot" and five time slots is called a "frame". Time is divided into frames and each LCF is assigned one of the five time slots during which it may send out messages (See Figure 3).

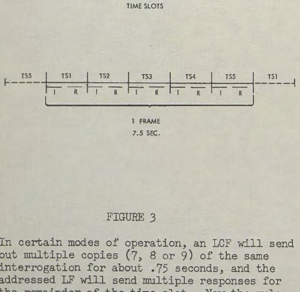

FIGURE 3

In certain modes of operation, an LCF will send out multiple copies (7, 8 or 9) of the same interrogation for about .75 seconds, and the addressed LF will send multiple responses for the remainder of the time slot. Now the rules of operation begin to get complicated. Any LF, receiving a message chain from an LCF, will retransmit the entire multiple message train (up to a maximum of about 1.5 seconds). However, if it receives the message from another LF, it will stop retransmitting as soon as one complete message has been decoded and will enter a period of "Lock Out", approximately equal to the time to transmit one complete message. At the end of this Lock Out period, it will again look at incoming communication lines, and if there is a message on a line (other than the last line on which a message was received) retransmission will again continue until one complete message has been decoded. This overly simplified description has ignored many system complexities which are also included in the simulation.

- There are hardware delays associated with each item of equipment.
- Initial bits are lost while equipment is establishing synchronization with an incoming message.
- The actual message selected for retransmission is a function of a scanning commutator position.
- If a station misses the initial sync pattern of a message, it may add noise bits to the end of a message due to "tone detector hang-on".
- The actual start of a message is subject to random delays.
- There may not be a positive timing sync between LCFs permitting "time slots" to gradually overlap.
- A seventh position on each LF commutator scans for radio messages.
- Each LCF monitors all transmissions by other LCFs, and if it does not observe the expected flow, it may assume that stations are not operating and attempt the interrogations itself.
- An LF, when retransmitting a message received directly from an LCF, will insert an additional eight bit preamble in front of the message.

The original simulation was written in Simscript during the month of July 1965, and took four man weeks. The program consisted of less than 400 statements and completely represented the network as originally described to the programmer analysts. However, reviewing this program as a descriptive document of the system, the engineers introduced additional refinements which had been omitted from the initial simulation.

By December of 1965, after many conferences between systems engineers and simulation programmers, the Minuteman Communication Network Simulation program contained over 1500 statements and quite accurately incorporated these and many other complexities. A number of validation runs indicated the expected performance. However, when multiple messages were sent out by an LCF, occasionally an unexpected phenomenon would occur - stations would come out of "Lock Out" while message fragments were still being transmitted and cause a "ringing" situation. This "ringing" would create a random noise situation so severe as to interfere with the proper transmission of messages. Figure 4 illustrates the effect of the "ringing". At first, doubts were cast upon the validity of the simulation model, because extensive mathematical analyses over a two year period had conclusively "proved" that while other problems might exist, almost certainly "ringing" could not!

A study of the simulation showed that "ringing" arose from certain second order random effects which were not considered in the original mathematical analysis. Furthermore, the system as originally conceived, would <u>not</u> have had ringing; however, minor engineering modifications had opened up the possibility which was not detected until the completion of the simulation. The happy ending of this story is that the simulation provided the tool to test for possible fixes and came up with an optimum solution which could be installed with confidence. Since this problem was detected before the installation of equipment in the field, the cost of modification was a small fraction of what it might have been otherwise.

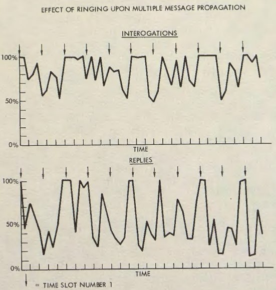

FIGURE 4

Since that time, the Minuteman model has been used as a Simulation model should be used--that is, to check out proposed system changes, rather than verify system performance after modifications have been made. Furthermore, it was clear that if the simulation had been prepared earlier in the Minuteman program, the possibility of ringing would probably have been uncovered when the modification which made it possible was proposed.

This leads to the interesting conclusion that ideally the simulation for any proposed new system should be written even before the specifications of the new system are available. To attempt to simulate a system even before it is designed is obviously ridiculous; however, GENCOM is an attempt to do just that for certain classes of communication networks. This brings us up to September of 1967 when GENCOM was created--discovered--uncovered--recognized--or however you might describe its origination.

At that time our company was planning a study involving the development of a rather complex tactical command communication network. The question was asked, "To what extent is the Minuteman Communication Network Simulation applicable to this problem?" The first quick answer was "Not at all!" The Minuteman simulation was a large, complex program incorporating many details of a sophisticated system designed with the capability of almost overnight responses to proposed system changes.

A careful look was taken at the Minuteman model. Each of the features peculiar to Minuteman were deleted until there was only a skeleton program left. However, this skeleton did suggest, most definitely, how a "general purpose" communication network simulator should look! Literally, in a matter of hours, GENCOM came into existence.

GENCOM consists of a basic simulation program, together with a collection of alternate subroutines which may be used as needed. This approach is used in order to achieve an optimum balance between programming effort and computer running time.

Certain assumptions have been made in GENCOM. GENCOM assumes that any network can be described as a collection of STATIONS interconnected in some complex fashion. STATIONS are of two basic types, NODES and TERMINALS. A NODE is defined as a switching STATION, i.e., a STATION which can interconnect other STATIONS, either NODES or TERMINALS. A TERMINAL is defined as a STATION which can transmit, receive, or store a message. Thus a telephone is a TERMINAL, while a switchboard or dial system is a NODE. A radio transceiver is a TERMINAL if it can operate only on a single fixed frequency, but is a combination TERMINAL and NODE if multiple frequencies are possible.

Suppose we assume our TERMINAL is a teletypewriter. Typical operation of the TERMINAL might be as follows:

Messages are delivered to an operator and placed in one of several incoming slots. Each slot represents a particular priority. The operator takes the first message and attempts to contact addressee. If contact is made, then message is sent (or appropriate two-way conversation conducted). If connection cannot be established, operator time stamps message and decides to try again after some minimum interval (maybe 20 minutes or sooner, dependent upon priority of message).

The operator next looks at incoming slots and picks oldest of highest priority messages and repeats above steps. At some point in time he will complete a message and find that new messages are still queued up and it is time to retry old messages. In this case, he will pick the highest priority from the two groups and attempt to send message. He will continue in this fashion until he is either able to send a message or has exhausted the stack of new messages and old messages scheduled for retry.

The flow chart (Figure 5) shows the sequence of operator decisions. In addition, incoming messages can also tie up the station if the same equipment is used for both sending and receiving messages.

If we considered the equivalent operations for a telephone, radio, or even on-line computer, we find that except for possibly different delay values, the same chart is applicable.

GENCOM has been written in SIMSCRIPT I.5. Some of the basic GENCOM routines will now be described.

GENCOM PROGRAM ELEMENTS

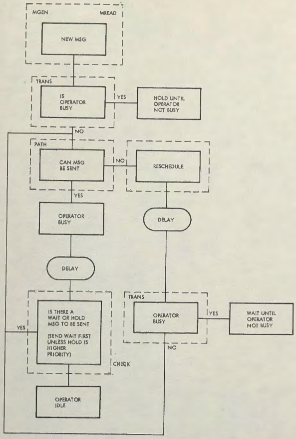

FIGURE 5

The Endogenous Event TRANS is used to handle the transmission of messages between two selected points. The Subroutine PATH automatically selects the available path between the two points which will pass through the minimum number of NODES, and the Subroutine CHECK determines if a TERMINAL, which has just completed a message, has other messages waiting for transmission. The input routine, MREAD, is capable of permitting extremely versatile operations. For example, although it introduces messages at a specified time, each message introduced can initiate a sequence of parallel and follow-on messages into the system. Thus, the equivalent of the following instructions could be input by MREAD:

"TERMINAL 1 should transmit instructions to TERMINAL 2 at 10:00 or as soon thereafter as possible. TERMINAL 2 should repeat message to TERMINALS 5 and 6. All TERMINALS, except 2, should send a short acknowledgement to 1 after a specified delay for message processing."

By this procedure, complex chains of messages can be introduced and the delays in propagation, due to message interference, analyzed. Another routine, called MGEN is used to randomly produce some level of system load, while MREAD is used to introduce message of significant interest.

ENDOG EVENT TRANS

When the Event TRANS is called, the associated message bears the name TRANS. The variable FROM (TRANS) specifies the origination TERMINAL identification number. A message may possess a MODE number from 0 through 4. The following is a description of the significance of each MODE.

(MODE 0) This message has just entered the system. This is the first attempt to transmit. If the TERMINAL is not busy and connection with the Final Station can be established, then transmission will start, mode will be changed to 4, and appropriate loading placed upon all TERMINALS, NODES, and LINES used for transmission. TRANS will be called again when transmission is complete. If originating TERMINAL is busy, file message in QIN, change MODE to 1, and do not call TRANS until TERMINAL is again available. If originating TERMINAL is not busy, but a suitable connection cannot be established to Final station, then file message in QCON, change MODE to 2 and call TRANS again when RETRY is scheduled. In every case, accumulate the proper statistics for output reports.

(MODE 1) Subroutine CHECK has determined that this TERMINAL is now available and has a message waiting in QIN. Remove message from QIN and accumulate total queue time waiting for TERMINAL. If connection to Final station is available, change MODE to 4, etc., as in MODE 0. If connection cannot be established, file message in QCON, change MODE to 2 and schedule RETRY at suitable time.

(MODE 2) This means a message in QCON has now been scheduled for RETRY. If originating TERMINAL is busy, change MODE to 3 and do not call TRANS until TERMINAL is available. If TERMINAL is not busy, but connections cannot be established, schedule next RETRY. If TERMINAL and Connection are available, compute time in QCON, remove from QCON, change MODE to 4 and transmit as in MODE 0.

(MODE 3) Subroutine CHECK is called whenever a TERMINAL changes from the busy to the available state. CHECK looks at messages in QIN and in QCON with MODE equal 3. In this case there was a MODE 3 message in QCON with a priority higher than (or equal to) any message in QIN. If connection can be established, then message is removed from QCON and wait time computed. MODE is changed to 4 and message transmitted, etc. If connection cannot be established, MODE is changed back to 2 and a RETRY scheduled.

(MODE 4) This means that the transmission of a message has been completed. The appropriate TERMINALS, NODES, and LINES are released. The message is checked (FMESS) to see if a reply or follow-on message was requested. If so, this message is introduced into the system after an appropriate delay. If the PMESS attribute of the follow-on message is not zero, then a parallel message is also introduced into the system. Again PMESS for this message is checked, and additional parallel messages introduced until a zero PMESS is reached. Each of these messages, in turn, can initiate additional follow-on and parallel messages.

Subroutine Path

This routine (which may be modified for other systems), represents one manner in which a path can be selected, between two terminals, in a complex system. In this case, the path is selected which passes through the minimum number of NODES.

Subroutine Check

Each time the originating and final TERMINALS are released, QIN and QCON are checked to see if other messages are waiting for transmission.

Analysis Reports

The routines for the analysis of performance are as follows:

- Utilization of each TERMINAL, NODE, and LINE
- Queueing while waiting for TERMINAL
- Queueing while waiting for connection at TERMINAL
- Detailed analysis of "flagged" messages

Of course, special reports can be added to collect other data about system performance.

System Connectivity

Figure 6 shows a simple system.

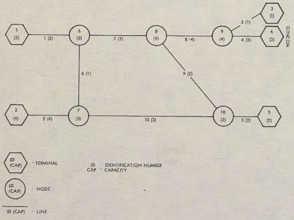

FIGURE 6

TERMINALS 1, 2, 4 and 5 could be thought of as groups of teletypewriters with operators. Thus there are two units at TERMINAL 1, four units at 2, etc. TERMINAL 3 has one unattended printer. Although 3 could be collocated with 4, it is handled differently. For this example, the LOAD for each message is one unit and the capacities of LINES and NODES indicate the number of simultaneous messages they can handle. This connectivity and capacity can be described by a series of tables stored in the simulation program. These tables are as follows:

JOIN TABLE — JOIN (I,J)

STATION	CONNECTIONS		
1	6		
2	7		
3	-		
4	9		
5	10		
6	1	7	8
7	2	6	10
8	6	9	10
9	3	4	8
10	5	7	8

The JOIN table is a double subscripted array, stored as a "Ragged Table" to minimize memory. In general it is referred to as JOIN (I,J) where I is the station number and J represents the connection position, thus JOIN (9,2) is equal to 4. The fact that 9 is connected to 4 by means of the "second" connection is used to determine LINE numbers. Thus we have a companion table called LNUM (I,J).

LINE NUMBER TABLE - LNUM (I,J)

STATION	LINE NUMBER		
1	1		
2	2		
3	-		
4	4		
5	5		
6	1	6	7
7	2	6	10
8	7	8	9
9	3	4	8
10	5	10	9

Having determined that JOIN (9,2) is equal 4, we find that LNUM (9,2) equals 4, meaning that STATIONS 9 and 4 are connected by LINE number 4.

The remaining special tables are quite simple. We have a capacity table for STATIONS (CAP) and a capacity table for LINES (LCAP), thus:

STATION CAPACITY		LINE CAPACITY	
STATION	CAPACITY	LINE	CAPACITY
1	2	1	2
2	4	2	4
3	1	3	1
4	3	4	3
5	2	5	2
6	3	6	1
7	5	7	3
8	4	8	4
9	4	9	2
10	3	10	3

Whenever a LINE or STATION is in use, the corresponding LOAD of the message is subtracted from the appropriate CAP and LCAP values. When a message is completed, then LOAD is added back. The availability of a STATION or LINE is determined by testing to see if the capacity is greater than LOAD.

Two other tables complete the descriptive statistics for the network. RETRY indicates the time an operator will wait before retrying a message when the lines are busy, and DLAY represents the additional delay in establishing connections through a particular NODE.

TERMINAL NUMBER	RETRY TIME (Min)	NODE NUMBER	DLAY TIME (Sec)
1	2.0	6	5
2	2.0	7	2
4	2.0	8	3
5	2.0	9	1
		10	5

Voice Communication

Although this system has considered only teletype messages, suppose that our system also included a telephone hookup. This could be run as an independent set of STATIONS and LINES; however, phone messages could initiate parallel and follow-on teletype messages and vice versa, so that a realistic system interplay could be simulated. Furthermore, although the digital equipment would be independent of the voice equipment, it is possible that some common use of LINES could exist. This would be accomplished by using identical line numbers in both sections where this is possible. Thus, suppose that LINE 7 could carry three teletype messages or one voice message. In the voice portion of the simulation, each voice transmission would always require three units, and on those occasions when LINE 7 is used for voice, it would not be available for teletype messages.

Some Special Cases

Lest it be assumed that only teletype or voice can be handled, consider a more complex situation. Suppose we have thirty airplanes, within range of a single station, all operating on the same frequency. This is equivalent to a TERMINAL with a capacity of thirty connected to a NODE by a line with a unit capacity of one. If they can operate on multiple frequencies, then this is the equivalent of adding more lines. Further, if we assume that planes move out of the range of certain stations and within the range of others, this is accomplished by transferal of numbers between TERMINAL capacities. On the other hand, if we wish to keep track of individual planes, then a separate TERMINAL number could be assigned to each one.

GENCOM MREAD Routine

The MREAD routine makes it possible to input complex sequences of messages. This is accomplished by designating Parallel, Follow-on and Alternate message paths. A few simple examples will illustrate the technique. Letters, rather than numbers, will be used to designate stations in order to avoid confusion with message identification numbers. For the first example, consider a simple sequence of a message from A to B, then B sends to C and C sends to D. This would be accomplished as follows:

MESSAGE ID	FROM	TO	PARALLEL	FOLLOW-ON	ALTERNATE
1	A	B		2	
2	B	C		3	
3	C	D			

In this case, message 2 would not be initiated until after the completion of message 1, and 3 would not start until the completion of 2.

In the next example, Station A wants to send messages in parallel to B, C and D, and B, C and D will send replies back to A after each has received the original message.

MESSAGE ID	FROM	TO	PARALLEL	FOLLOW-ON	ALTERNATE
1	A	B	2	4	
2	A	C	3	5	
3	A	D		6	
4	B	A			
5	C	A			
6	D	A			

It should be noted that the three messages to B, C and D will be sent in parallel only if A has sufficient available capacity to send them, otherwise they will queue up and may be sent sequentially.

As will be noted, by means of Parallel (PMESS) and Follow-on message (FMESS) coding, complex sequences of message can be input. Whenever a message is scheduled for entry into the system, all Parallel messages are also scheduled. In the example shown, only the input data fields associated with message routing have been indicated. There are, of course, other descriptive parameters input at the same time. These include the length of time for message transmission (LONG), the type of message (TYPE), the loading of the message on the system (LOAD), the priority of the message (PRIOR), and an important parameter STIME. STIME provides a delay time to take care of message handling or processing. Thus the message is not immediately introduced into the system but is delayed by the amount of time stored in STIME.

Alternate Message Option

In this case we can specify one or more alternate paths or destinations for a given message. Thus consider the following:

MESSAGE ID	FROM	TO	PARALLEL	FOLLOW-ON	ALTERNATE
1	A	B		6	2
2	A	C		4	3
3	A	D		5	
4	C	B		6	
5	D	B		6	
6	B	A			

In this case, Station A wants to send a message to Station B. If A cannot send to B, it will attempt to send to C. If neither B nor C is available, it will attempt to send to D. If all three stations are tied up, then the message will be sent to the first free station and no further attempts will be made to send the other two alternates. Note, for example, that if the message is sent to C, then C's follow-on message is to B and messages 1, 3 and 5 are ignored and never enter the system. Message 6, in any event, will be introduced as soon as B has received a message by any one of the three alternate paths. (Note, in this case, we could think of C and D as "Store and Forward" NODES).

Another interesting possibility is also available by means of Alternate messages. As previously mentioned if, for example, three teletypewriters are located at a single terminal, we simplify our program by considering this a single terminal with a "Capacity" of three rather than three separate terminals. However, suppose that there are significant differences in the speed of the three operators, and we wish to consider each one as a separate terminal so that instead of a single terminal with a capacity of three, we now have terminals A, B and C, each with unit capacity. For each terminal we would use a corresponding SPEED factor to modify the length of transmission. Thus SPEED (A) might be 1.00, SPEED (B), .80, and SPEED (C) equal 1.30. However, messages sent from these three terminals must be handled by the first operator available. This is accomplished as follows:

MESSAGE ID	FROM	TO	PARALLEL	FOLLOW-ON	ALTERNATE
1	A	D	-	4	2
2	B	D	-	4	3
3	C	D	-	4	-
4	D	A	-	-	5
5	D	B	-	-	6
6	D	C	-	-	-

Thus A, B or C will send the original message to D, and D, after a delay for message processing, will send a reply back to either A, B or C.

Conference Calls

This requires that one station make connection with a number of stations and transmission will not start until all connections are made. This is accomplished by a special message type and parallel messages.

Conditional Follow-on

In this case, station A will send Interrogations to B, C and D and will wait for a response from all three stations before proceeding with the next message.

Hot Line Messages

Here a message establishes a path and keeps it open for an indefinite period. The path is released by a special message type which in turn may place a normal message on the selected "hot line."

Message Errors

Messages of a certain type may be subject to bit errors requiring retransmission. Upon receipt of such a message, the probability of successful transmission will be checked. Randomly, certain messages will be considered to be in error and instead of proceeding with the normal Follow-on message, a subsequence of a request for retransmission followed by a retransmission of the original message will take place.

Redundant Paths

It is possible to specify that an important message be sent over multiple paths for reliability. In this case the maximum number of paths would be specified, but the message will still be sent even if the maximum number of paths is not available.

GENCOM MGEN Routine

The MGEN routine provides a means of repeating the message chains introduced by MREAD, in a random manner with a specified mean interarrival time. This is accomplished by using a non-zero value of STIME for the first message in the input sequence. STIME is then interpreted as the mean exponential interarrival time and that message is then repeated randomly.

By establishing a normal, random load on the system by MGEN, it is possible to determine the effect of specific messages, input by MREAD, upon system operation.

Special Options

By means of a special data input routine, the parameters used by MGEN can be changed to reflect time variant changes in the normal system load. Also, the capacities of stations and lines can be modified to reflect equipment malfunctions, line failures, operators off duty, etc.

A Sample Problem

Copies of a sample computer run are shown for the example which has been described. Table 1 shows a printout describing the network. The network for this example is shown in Figure 6. This is a combined printout of the values stored in JOIN, LNUM, CAP, and LCAP. It should be emphasized that these values are part of an input data deck and completely independent of the GENCOM program. Thus no reprogramming is required to test any other configuration regardless of size or complexity of connections. Subject only to memory capacity, we could have any number of stations and there is no limit to the number of stations a station can be interconnected to. The table is read as follows:

"Station 1 has a capacity of 2; it is connected to Station 6 by means of Line 1. Line 1 has a capacity of 2, etc."

Table 2 shows the printing of an advisory comment every time the transmission of a message begins. Thus, the first line says,

"At Time 0.0 minutes, a message to type 1, priority 1, load 1, and length of 2.5 minutes, was sent from Station 1 to Station 5 and went through nodes 6, 7 and 10."

Before describing the other reports it would be well to look at Tables 6 and 7 which lists the input data deck. Table 6 is a list of the initialization cards. This deck inputs the values for station connectivity (JOIN and LNUM) capacities (CAP and LCAP). ISCAP and ILCAP are used to keep track of the initial capacities for report purposes. RETRY times for TERMINALS and DLAY times for NODES is also initialized.

The second part of the Data Deck, Table 7 controls the input of messages and the printing of reports. The headings at the top of the page refer only to the third through twentieth cards. The first card calls Exogenous Event Number 1, which is START, at Time 0. The second card calls Exogenous Event Number 2, which is MREAD at time 0 and specifies that the next 18 cards will contain a description of the messages to be input. Thus, message Number 1 says input a message of type 1, priority 1, loading 1, with length of 2 minutes, 30 seconds into the system with no delay (since STIME equals 0). This message is to be transmitted from station 1 to station 5. In parallel with the transmission of this message, parallel message number 2 is also scheduled for entry.

When the transmission of message 1 to station 5 is complete, GENCOM will then inspect the follow-on message number 3. Message 3 will be introduced into the system after a 1 minute 30 second delay. Parallel message number 4 will be delayed 45 seconds, parallel message number 6 will enter the system immediately, and parallel message number 8 will be delayed 2 minutes. In a similar fashion, the remaining messages would be input.

The last three cards are interpreted as follows: the card with "3 10" says call Exogenous event 3, which is ANALYZ, when simulation time equals 10 minutes. The next card calls ANALYZ at 30 minutes, and the final card calls Exogenous Event ENDSIM (number 4) at 30 minutes to stop the simulation run.

ANALYZ is the routine which calls for the printing of statistics about the run. Thus Table 3 presents queue statistics for each TERMINAL. It should be noted that although messages are introduced into the system, they are not necessarily transmitted until both the originating terminal and connection to the final station are available. Thus we note that there was a delay at terminal 4, while waiting for a connection, and station 5 had delays both for the terminal and connection. Table 4 summarizes the percentage of STATION utilization and Table 5 does the same thing for LINES. The remaining pages print the statistics for the interval between 10 and 30 minutes.

Summary

In summary, GENCOM is a program capable of realistically representing a wide variety of typical communication networks. It can be quickly configured to approximate many types of TERMINALS and can handle complex interconnections. Its modular structure also facilitates the addition of modifications peculiar to a specific system. GENCOM's evolution presents an unusual case history in the manner in which an extremely versatile simple program was developed from the study of a complex special-purpose program.

TABULATION OF STATION CONNECTIVITY

STATION		CONNECTIONS -- SN = STATION NUMBER, LN = LINE NUMBER, LC = LINE CAPACITY																	
NUMBER	CAPACITY	SN	LN	LC	SN	LN	LC	SN	LN	LC	SN	LN	LC	SN	LN	LC	SN	LN	LC
1	2.	6	1	2.															
2	4.	7	2	4.															
3	1.	TERMINAL RECEIVES ONLY																	
4	3.	9	4	3.															
5	2.	10	5	2.															
6	3.	1	1	2.	7	6	1.	8	7	3.									
7	5.	2	2	4.	6	6	1.	10	10	3.									
8	4.	6	7	3.	9	8	4.	10	9	2.									
9	4.	3	3	1.	4	4	3.	8	8	4.									
10	3.	5	5	2.	7	10	3.	8	9	2.									

TABLE 1

```
AT   0.    MESSAGE TYPE 1, PRIORITY 1, LOAD  1, AND LENGTH  2.50 WAS SENT FROM 1 VIA.  6   7  10   5
AT   0.    MESSAGE TYPE 1, PRIORITY 1, LOAD  1, AND LENGTH 10.   WAS SENT FROM 1 VIA.  6   8   9   3
AT   3.45 MESSAGE TYPE 1, PRIORITY 1, LOAD  1, AND LENGTH  8.   WAS SENT FROM 5 VIA. 10   7   2
AT   4.20 MESSAGE TYPE 1, PRIORITY 1, LOAD  1, AND LENGTH  4.33 WAS SENT FROM 5 VIA. 10   8   9   4
AT   8.68 MESSAGE TYPE 1, PRIORITY 1, LOAD  1, AND LENGTH  3.50 WAS SENT FROM 5 VIA. 10   7   6   1
AT   9.68 MESSAGE TYPE 1, PRIORITY 1, LOAD  1, AND LENGTH 10.   WAS SENT FROM 4 VIA.  9   8  10   7   2
```

TABLE 2

QUEUE STATISTICS BY TERMINAL AT TIME 0 HRS, 1[illegible] MINS, [illegible] SECS

TERMINAL	WAIT FOR TERMINAL			WAIT FOR CONNECTION		
NUMBER	AVERAGE NO.	AVERAGE TIME	MAXIMUM	AVERAGE NO.	AVERAGE TIME	MAXIMUM
4	0.	0.	0.	0.25	0.84	3.
5	0.40	3.98	1.	0.73	7.30	1.

TABLE 3

PERCENT STATION UTILIZATION

STATION NUMBER	PERCENT
1	70.08
2	17.16
3	100.
4	16.
5	75.25
6	46.72
7	21.76
8	37.
9	37.
10	51.22

TABLE 4

PERCENT LINE UTILIZATION

LINE NUMBER	PERCENT
1	70.08
2	17.16
3	100.
4	16.
5	75.25
6	40.16
7	33.33
8	37.
9	24.
10	36.27

TABLE 5

(Initial Conditions and Data Deck)

```
$DATA
1          38     60     60                                   40
   1      0 R                                                      10
   2      0 R                                                      10
   3   29 1  Z     10     1
  30      0  Z
  31      1 R      10     1                                     10(M3.2)         RETRY
  2.     2.              2.      2.                                                 RETRY
  32      1 R      10     1                                     10(M4)           DLAY
                             5    2    3    1    5                                  DLAY
  33      1 R      10     1                                     10(D3)           ISCAP
  2.   4.   1.   3.   2.   3.   5.   4.   4.   3.                                   ISCAP
  34      1 R      10     1                                     10(D3)           ILCAP
  2.   4.   1.   3.   2.   1.   3.   4.   2.   3.                                   ILCAP
  35      2 R      10     1                    R                10(I6)           JOIN
     6                                                                          1    JOIN
     7                                                                          1
   0                                                                            1
     9                                                                          1
    10                                                                          1
     1      7      8                                                            3
     2      6     10                                                            3
     6      9     10                                                            3
     3      4      8                                                            3
     5      7      8                                                            3
  36      2 R      10     1                    R                10(I6)           LNUM
     1                                                                          1    LNUM
     2                                                                          1
   0                                                                            1
     4                                                                          1
     5                                                                          1
     1      6      7                                                            3
     2      6     10                                                            3
     7      8      9                                                            3
     3      4      8                                                            3
     5     10      9                                                            3
  37        R                                                   50
  38      1  Z     50   37
                DATA CARDS FOLLOW
  1              THIS CARD STARTS SIMULATION AT TIME 0
  2              18   THIS CARD INPUTS THE FOLLOWING 18 CARDS AS MESSAGES
```

TABLE 6

ID	TO	FROM	Type	Prior	Long	Load	Amess	Pmess	Fmess	Stime
1	1	5	1	1	2.30	1		2	3	
2	1	3	1	1	10.	1				
3	5	4	1	1	4.20	1		4	5	1.30
4	5	2	1	1	8.	1		6	7	.45
6	5	3	1	2	6.	1		8		
8	5	1	1	1	3.30	1			9	2.00
5	4	5	1	2	10.	1		10		.30
10	4	3	1	2	4.	1		11		.45
11	4	1	1	1	2.30	1		12	13	.10
12	4	2	1	1	10.	1			14	1.00
7	2	3	1	3	5.30	1		15		
15	2	1	1	1	7.10	1		16		1.30
16	2	5	1	1	10.25	1		17	18	
17	2	4	1	1	3.00	1				
9	1	3	1	3	5.10	1				.30
13	1	3	1	3	2.00	1				
14	2	3	1	3	1.00	1				
18	5	1	1	1	6.00	1				

3 10 CALLS FOR PRINTING OF REPORTS AFTER 10 MINUTES
3 30 CALLS FOR PRINTING OF REPORTS AFTER 30 MINUTES
4 30 STOPS SIMULATION AFTER 30 MINUTES OF SIMULATION

TABLE 7