THE MINUTEMAN COMMUNICATION NETWORK SIMULATOR

W. V. Neisius and E. D. Katz
TRW Systems Group
Redondo Beach, California

This is a case history of a very versatile and complex Simscript simulation which has been completed and successfully used since December 1965. Several points are of interest: first, the original approach took maximum advantage of Simscript by concentrating upon the development of a quick and simplified model which could be expanded as additional system details were made available; second, the analysis reports went through a number of versions as experimentation goals were modified; and, finally, the simulation was responsible for uncovering potentially significant system problems.

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 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. 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 about 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 3

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 not 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.

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.

During the development of this program a number of different output reports evolved for analysis purposes. These were as follows:

- Statement of message origination.
- Statement of message reception.
- Chronological record of message flow.
- Summary of message activity by stations.
- Summary of message activity by message type.
- Overall summary of interrogation/response successes.

The last four of these could be inhibited or called for on a selective basis. The last report provides a one page summary of the complete run which gives a quick "yes" or "no" answer to the success of the run. If the answer is "no", that is interrogation/response cycles were not successfully completed, then the other reports will supply sufficient details to isolate the cause.

Figure 4 shows a flow chart of the program. The circles containing "D" numbers represent delays which are, in some cases, fairly complex random functions. Figure 5 provides a brief description of the routines used in the program.

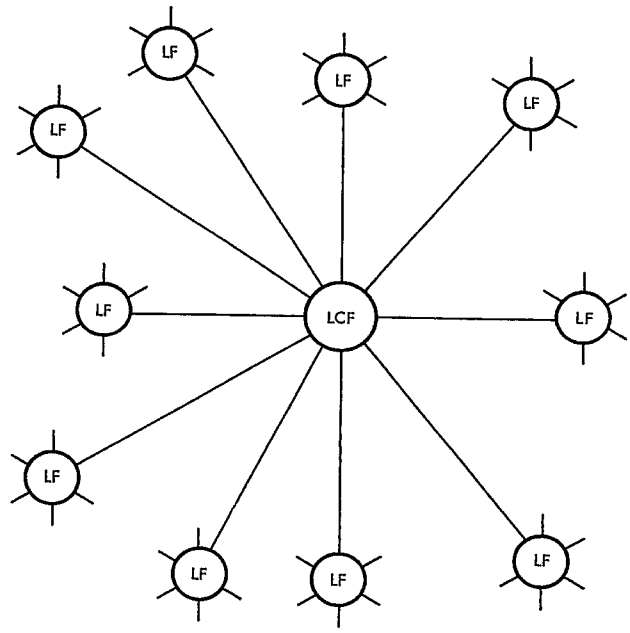

FIGURE 1

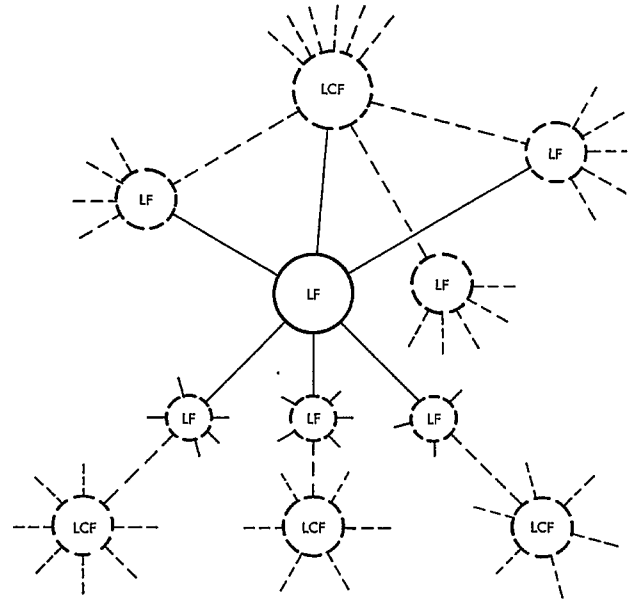

FIGURE 2

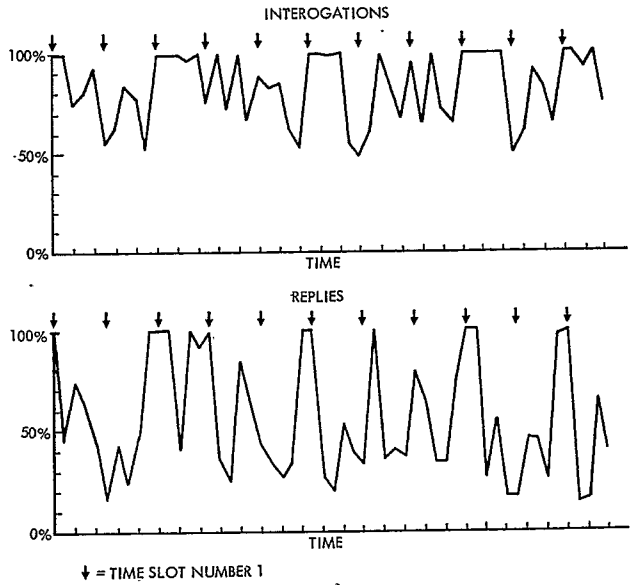

FIGURE 3

START	THIS ROUTINE IS USED TO START THE SIMULATION AND IMMEDIATELY CALLS READ.
READ	THIS ROUTINE READS AND INTERPRETS THE INPUT DATA DECK AND PERFORMS THE APPROPRIATE FUNCTIONS. IT USUALLY FORMATS DUMMY MESSAGES FOR FILING IN TEMP SO THAT TSLOT WILL LATER CAUSE THE CORRECT EVENTS TO OCCUR.
TSLOT	THIS EVENT NOTICE INTRODUCES NEW MESSAGES INTO THE SYSTEM AND CONTROLS THE PRINTING OF REPORTS, ETC. TSLOT IS CONTROLLED BY THE ITEMS STORED IN THE SET TEMP. ASSOCIATED WITH EACH OF THESE ITEMS IS A TIME OF OCCURRENCE. THE ITEMS IN TEMP FOR DEFERRED ACTION ARE STORED BY THE SUBROUTINE READ.
COPY	THIS IS AN EVENT NOTICE WHICH IS CALLED WHENEVER A STATION WISHES TO TRANSMIT COPIES OF MESSAGES IT IS RECEIVING OR ORIGINATING. IT IS CALLED BY RECVE, REPLY, TSLOT, AND STATC.
DELAY	THIS EVENT NOTICE IS CALLED BY COPY AND FORMATS THE MULITPLE COPIES OF THE MESSAGE IT IS RECEIVING OR ORIGINATING.
RECVE	THIS EVENT NOTICE CONTROLS THE SCANNING OF MESSAGES IN RECR. THE CALLING OF THE ROUTINE RECVE IS CONTROLLED BY THE STATUS OR RECVN (SEE THE DISCUSSION OF RECVN).
ALINE	THIS EVENT NOTICE IS CALLED BY RECVE 56 MS (OUT) AFTER THE LAST MESSAGE BIT HAS BEEN RECEIVED. IT IS USED TO UNLOCK THE LCS SO THAT IT CAN RECEIVE MESSAGES FROM ALL STATIONS EXCEPT THE LAST STATION IT RECEIVED FROM. ALINE ALWAYS CALLS LLINE AFTER AN ADDITIONAL 44 MS (LOUT).
LLINE	THIS EVENT NOTICE IS CALLED BY ALINE. ITS FUNCTION IS TO UNLOCK THE LAST STATION AS IDENTIFIED BY THE ATTRIBUTE LOCK. THE UNLOCKING IS ACCOMPLISHED BY SUBTRACTING LOCK (LLINE) FROM BUSY (WHO(LLINE)).
SEIZE	USED BY RECVE, THIS FUNCTION DETERMINES THE NUMBER OF BITS LOST BY A MESSAGE PRIOR TO LINE SEIZE.
SEND	THIS FUNCTION, USED BY RECVE, DETERMINES THE NUMBER OF BITS TO BE RETRANSMITTED BY A STATION.
LCFCT	THIS FUNCTION, USED BY RECVE, DETERMINES IF THERE WILL BE A LCF TIME SLOT CUTOFF OF A MESSAGE.
FOLLOW	THIS FUNCTION, USED BY RECVE, FORMATS "FOLLOW-ON" MESSAGES.
LFCUT	THIS FUNCTION, CALLED BY RECVE, DETERMINES IF THERE WILL BE A CUTOFF OF A RETRANSMITTED MESSAGE BY A CSR SIGNAL.
RECPT	CALLED BY LFCUT. THIS ROUTINE FORMATS REPLY MESSAGES AND CALLS REPLY.
REPLY	WHEN LFCUT RECOGNIZES THAT A STATION WILL GENERATE A REPLY, IT CALLS SUBROUTINE RECPT. RECPT IN TURN CALLS THE EVENT REPLY. REPLY SENDS THE REPLY MESSAGE AND CALLS COPY AND ULOCK.
ULOCK	THIS EVENT NOTICE IS USED TO UNLOCK THOSE STATIONS WHICH ORIGINATE A MESSAGE OR A REPLY. THE UNLOCKING IS ACCOMPLISHED BY SUBTRACTING 200 FROM BUSY (WHO (ULOCK)). ULOCK IS CALLED BY REPLY AND TSLOT.
STATC	THIS EVENT NOTICE IS CALLED BY READ. IT IS USED TO INTRODUCE STATIC INTO THE SYSTEM.

FIGURE 5

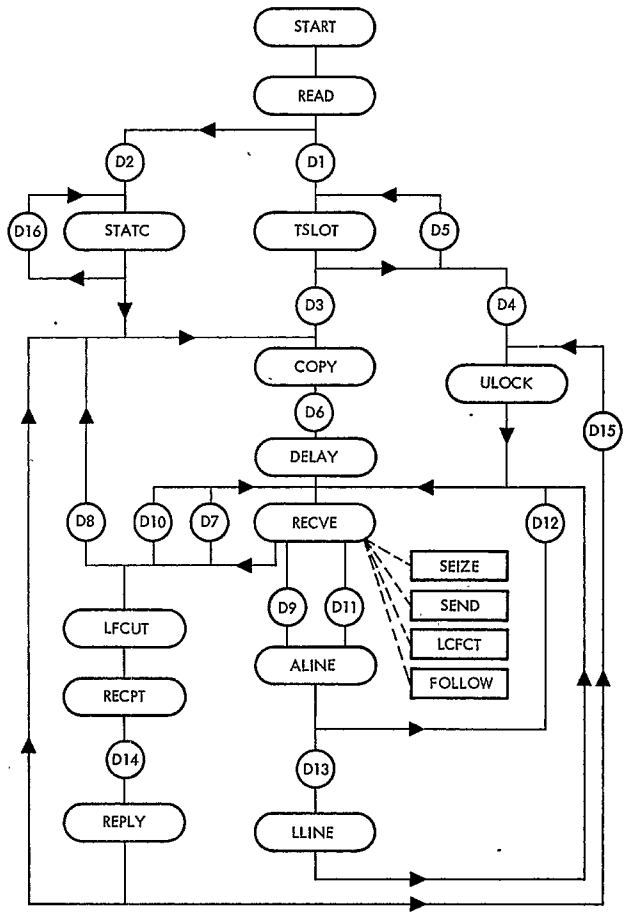

FIGURE 4