T-4676

AN ANALYSIS OF THE ADVANCED FIELD ARTILLERY TACTICAL DATA SYSTEM (AFATDS)

by

PAUL W. GAASBECK

ProQuest Number: 10794087

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

ProQuest 10794087

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

T-4676

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mathematical and Computer Sciences).

Golden, Colorado
Date 16 Jan 95

Signed: ____________
Paul W. Gaasbeck

Approved: ____________
Dr. Ruth A. Maurer
Thesis Advisor

Golden, Colorado
Date 1/16/95

____
Graeme Fairweather
Professor and Head
Mathematical and Computer
Sciences Department

ABSTRACT

This thesis analyzes the U.S. Army Field Artillery Tactical Data System (AFATDS). This system is scheduled to replace the existing digital fire support system currently in use by the Army's Field Artillery known as TACFIRE. In AFATDS a digital data system is used to transmit requests for field artillery and other fire support assets from units engaged in direct combat. The method of transmission is FM tactical radios. An initial version of this new system has been designed and built, but problems were experienced during initial testing. Because of these initial poor results, the Army desired that a simulation study be conducted. The Army requires a system capable of the managing the massive amounts of information concerning targets and their engagement that the future battlefield will present.

After gathering all available information and performance data to date, a model was designed of a U.S. Army Heavy Brigade's Fire Support Radio structure. A one hour peak combat period during a brigade operation was selected as the scenario for the simulation. Three statistics that could gauge the system's capabilities were collected and analyzed during the study. These statistics were the mean number of sensory inputs (fire requests and other information) to the brigade, the mean percentage rate at which fire requests were translated into fire orders to a field artillery platoon, and the mean time for fire requests to reach a field artillery platoon. The result of this analysis is that the Advanced Field Artillery Tactical Data System was found to be more than capable of meeting the challenges of providing fire support during future combat.

TABLE OF CONTENTS

Page

CHAPTERS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

The author wishes to thank the members of his thesis committee for their guidance and support. Dr. Gene Woolsey has been a valuable mentor and professor. His leadership prepared the author for facing the challenges this project presented. Also, thanks to Dr. Ruth Maurer for introducing the author to simulation. Because of Dr. Maurer's teaching skills the author had the confidence to take on this project. Dr. Warren Spaulding (Colonel, USA, Retired) was instrumental in relating his experience to the project.

The author would also like to express his gratitude to LTC Stephen Moore for providing resources to make this project possible.

Finally, thanks to all the members of the Operations Research Guild who provided assistance during this project. A special thanks goes to Erdem Ince for his expert computer advice.

CHAPTER 1
INTRODUCTION

Currently, the U. S. Army Field Artillery uses a digital FM tactical fire support system that was developed in the 1970s. This system, known as the Tactical Fire Direction System or TACFIRE, has been adequate but was not designed to handle the information requirements of today's battlefield. The TACFIRE system consists of extremely large, bulky components using older technology. All data is transmitted at 1200 bits per second over wire or more commonly FM tactical radios. The main use of the system is to process requests for artillery fire. It is not particularly fast and has limitted planning, command and control capabilities. In addition, it needs relatively large amounts of electrical power to operate. As computers have become more and more advanced, a small micro computer can now perform all of the operations of a headquarters component of the TACFIRE system. This may not seem significant, however these components fill a tactical shelter on the back of a five ton truck. Furthermore, as warfare and weapons systems have become more complex, the ability of the Field Artillery to engage targets is directly related to the information processing systems in use.

Rather than just "catch up" with today's technology, the U. S. Army wanted to field a system that could deal with information in the environment of the future. Already the battlefield is saturated with information. Armed forces of the future will collect and process even more information as sensory and communications devices become more advanced. Based on these predictions for the future, the U. S. Army decided to specify and procure an advanced system. The Army needed a system that could handle an increased information flow. This new system would not only need to be capable of processing large amounts of information, but it would have to process hundreds, if not thousands, of requests for fire support (mortars, artillery, close air support and other assets) during combat. Further requirements would be that all of this information and fire requests would have to be processed in a timely manner. Unnecessary delays were not an option.

Pursuant to this goal, the Department of Defense contracted Magnovox Electronic Systems Company in 1989 to design and build a new tactical data system to be used to conduct fire support operations. Once fielded, this system, called the Advanced Field Artillery Tactical Data System or AFATDS, will be used by both the U. S. Army and Marine Corps. All units, from platoon to corps level, that conduct fire support operations will use this system. It will be used to conduct command, control, and support functions for all phases of fire support operations. Moreover, AFATDS will be required to operate in accordance with AirLand Battle Doctrine, the U. S. Army's method of conducting warfare.

As one could imagine, the specification for AFATDS contained many requirements. The system had to be hardened for military use (physical characteristics) and had to meet certain requirements for information processing (processor characteristics). But, these specifications for the processing portion of AFATDS were not based on today's battles. They were based on the battlefield of tomorrow. Perhaps the most impressive of these requirements was that the system had to successfully process 120 requests for fire support (mortars, artillery, close air support, etc.) per hour at the Brigade Headquarters level. This is a difficult requirement to visualize by today's standards. The author has participated in combat and in large scale combat training exercises and, based on his experience, today's well trained brigade combat team might be capable of processing 40 requests for fire support in a three or four hour battle. And not only would all of these 120 requests have to processed, they would have to be done so quickly; each one would have to be processed in a matter of seconds. Another important task was the processing of all types of messages. A brigade level headquarters would be required to receive a total of 701 different messages during this same hour long period.

After the design was finalized, an initial version of AFATDS was built. This initial version would replace the headquarters components of the TACFIRE system. Each headquarters element in the U. S. Army fire support structure would now have the equivalent of a personal computer instead of a large assembly of aging TACFIRE components. The AFATDS system at each headquarters would consist of a central processing unit (CPU), a monitor, a mouse, and the appropriate number of modems to communicate on the required number of radio nets for that headquarters. The modems are similar in size to an external modem that one would use with a personal computer. Not only would AFATDS process requests for fire like TACFIRE does, the system would also be used in all aspects of fire support operations to include command, control, and logistics.

This initial version underwent some limited field testing. Due to a number of reasons, the AFATDS system had some difficulty processing requests for fire in a timely manner during this testing. Because of these difficulties, the U. S. Army wanted someone to analyze the system and its capabilities to determine if it was capable of performing at the required level. As a result, the AFATDS project manager contacted the Colorado School of Mines to undertake this project. The U. S. Army wanted to know if AFATDS as designed could process in a timely manner 120 requests for fire per hour in a heavy information flow scenario.

CHAPTER 2
LITERATURE REVIEW

There is a wealth of information concerning the simulation of communications. Most of the current articles deal with the simulation of bit error rates or a specific characteristic of a radio or a radio network. Another large group of articles deals with simulating networks. These articles, however, deal with Local Area Networks (LANs), involving communications between computers. Still another group of articles covers the areas of simulating cellular communications and other personal communications devices. Not a great deal has been written about simulating an entire network, especially a tactical FM radio network. Many of the articles that have been written, however, contain principles that can be applied to any communications network.

Computer simulation and computer aided techniques are extremely valuable in solving analysis problems. One article entitled “Simulation of Communication Systems,” (Kosbar and Tranter, 1994) describes different examples of modeling communications systems. Model validation is dicussed and it is the authors’ suggestion that each sub-system of any simulation model be validated first. Once each component is validated, then a validation should be conducted at the system level. Another interesting topic addressed in this article is the use of redundant simulation. Because of the relatively inexpensive cost of today's advanced computers, more than one simulation team should work simultaneously on model building, simulating, and analyzing. This is the approach taken by the author of this thesis and fellow members of the project team who built and simulated another model using different tools.

Another key feature of this article is a brief checklist for developing a simulation. The first step should be to develop the model. Next, the signal processing operation necessary for each part of the subsystem needs to be identified. The next step is to define the simulation products (required outputs). Finally, the last step should be the selection of a software package.

Perhaps the best piece of advice given by Kosbar and Tranter is to start out simple. The initial model design should incorporate only what is absolutely needed to conduct the simulation. As model development advances and as each piece is verified and validated, then enhancements can be made.

If a computer simulation is to be performed then a software package must be selected. Many different types of simulation software that are currently available are discussed in "Simulation Software for Communications Networks," (Law and McLomas, 1994). The authors submit a list of what to look for in a communications simulation package. Model flexibility is emphasized as probably the most important feature in a package. A program or system should be able to model a wide range of communications systems and features, not just one type. Another feature to look for is ease of model development. The modeler should not have to go to extraordinary lengths to replicate a system or subsystem. Other aspects to consider are execution speed and statistical capabilities. Related to these features are output reports. The modeler should be able to collect the appropriate statistics and have them available in an output function. A drawback to this article and many others, from the standpoint of this project, is that only LANs are discussed. No mention is made of other communications networks.

In any communications system access to the system's resources is a primary concern. In "Simulation Analysis of a Communication Link with Statistically Multiplexed Bursty Voice," (Habib, Tarek, and Saleh, 1993) the authors discuss the use of computer simulation to evaluate an access control scheme for multi-media communications, a topic of growing popularity. This article is not directly related to FM communications, but the idea of controlling access to the network is. Access to a particular radio net plays a large role in the operation of AFATDS. This topic will be discussed further later in this thesis.

In their multi-media application the authors chose the simulation language SIMSCRIPT II.5 as their simulation tool. They give a general description of their model with diagrams and code. Although the author of this thesis is not familiar with SIMSCRIPT II.5, the article is able to convey the basic idea of their model. A series of queues and decision logic was used to replicate the control scheme. Performance of the access control system was gauged through the use of SIMSCRIPT II.5 global variables. Using these global variables the total number of generated message "packets" and blocked packets were recorded in order to determine a performance level.

Many simulation models deal with users or customers competing for a limited resources. Some interesting ideas are presented in an article entitled "Semaphore Queues: Modeling Multilayered Window Flow Control Mechanisms,"(Fdida, Perros, and Wilk, 1990). The authors present the modeling of a LAN that is operating with the use of tokens. A token is a symbolic representation of a resource or some other process where customers, messages or entities compete to use that same process. In this case, when a station has a token it can use a resource or utilize a service. Magnovox Electronic Systems Company has experimented with various token schemes to control access to FM networks.

These authors go on to present the idea of a semaphore queue. This is a pair of queues. One queue holds available tokens and the other is a queue of waiting customers or entities. If tokens are available when a customer or entity arrives at the queue, the customer takes the token and is then free to use the resource. If tokens are not available, the customer enters the waiting queue and waits for one to become available. This scheme is very similar to the idea of a resource in the simulation language SLAM II. In SLAM II a resource is specified as well as the number of resource units available. In SLAM II entities must gain control of a resource in order to complete a service activity. Later in the article the authors discuss the use of single and multiple semaphore queues. This article presented a unique idea of a common situation in simulation. It was very interesting to see someone else's perspective. This teaches the modeler not to have a mindset concerning the replication of a network or a process.

Often the structure or hierarchy of a network will, to a degree, define the method of access to the network. In “Modeling and Simulation of Networks Using CSIM,” (Edwards and Shankar, 1992) the modeling of various token ring networks is discussed. The term "ring" is used to refer to the structure of the network and the fact that all of the stations can communicate with other stations on the network either directly or through a relay. Round-robin token ring networks are also discussed. This is extremely important because this type of network attempts to allow all stations on a network equal access to the network. This is directly related to the topic at hand because equal access to the FM radio net is vital in order for AFATDS to operate efficiently.

In addition, the article discusses the simulation language CSIM. It is described as a process oriented, general purpose language that is written with C language functions. The advantages of CSIM are also addressed.

More importantly, the authors of this article present a general algorithm for the analysis of a network. The first step is to activate a network with "n" nodes. The second step is to transmit a packet or message on the medium according to the medium protocol. The third and last step is to keep statistics on the network for the parameters of interest. Although this algorithm is by no means detailed, it does provide the focus to model a complicated communications process. Sometimes in the midst of development the modeler can get off track without such a focus.

As previously stated the bulk of communications simulation literature deals with LANs. One such representative article is “Development of Design Guidelines for Local Area CSMA/CD Networks,” (Cobb, Mansfield, and Mellichamp,1992). The authors of the article describe the main features of a LAN. They are traffic related-features, protocol-related features, and configuration-related features. Although these features are not related to FM communications, they are food for thought for the modeler in general. The authors give a good description of the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol. This protocol is often referred to as an "ethernet". In this protocol the stations monitor the net to determine if any other station is transmitting a message. If the network is not in use, a station will wait a specified amount of time and then transmit. In addition, the transmitting station listens while transmitting to determine if there are possible message "collisions" caused by other stations transmitting.

The article also presents a simulation that uses the GPSS V version 1 simulation language. Particular advantages or disadvantages to GPSS V version 1 are not addressed, but the authors do describe particular aspects of their model using GPSS terminology. Since the author of this thesis has no experience with GPSS, a brief description of language capabilities and why they facilitated their model would have been informative.

An important topic in this article is the discussion of different validation techniques. One such technique is extensive message or entity tracing to ensure the model is behaving as intended. Another technique is to used fixed values. In this technique, input values or parameters are selected in such a manner that system performance can be inferred. Actual performance of the model can then be compared to expected performance. Lastly, comparing model output to another model's output is another option. This is feasible if the model used for comparison has been validated or if modeling is being used to predict system performance for a system that does not yet exist.

CHAPTER 3
METHODOLOGY

Pritsker, Sigal, and Hammesfahr describe the role of simulation in the decision making process (Pritsker, Sigal, and Hammesfahr, 1994, 1). Often leaders can make decisions after reviewing information gathered as a result of direct experimentation on a process or operation. Alternatively, another branch of the decision making process can include model building, model simulations, and analysis. This path can be taken in cases where direct experimentation is not possible or is too expensive. In today's world, where high costs often prevent any building, designing, or fielding in an experimental mode, computers often make simulation possible and desirable.

A simulation model can be made of almost any process or operation. Models can be extremely complicated, such as those used in the nuclear energy field, or as simple as one used to model a walk-up bank teller operation. No matter what process or operation is being modeled, there are six steps in solving a problem using simulation (Pritsker, Sigal, and Hammesfahr, 1994, 15). These steps are (1) formulate the problem, (2) specify the model, (3) build the model, (4) simulate the model, (5) use the model and (6) support decision making.

These six steps can be illustrated using the simple walk-up bank teller example. A bank needs to make a decsision concerning its walk-up teller service. The decsion maker, the bank president, is concerned with the bank's ability to service customers. Perhaps he has a new system he wants to implement. However, when the bank tested the new system problems were encountered. In this example there is one walk-up window and one teller.

Customers arrive and are served one at a time. If customers arrive while another is being served, they wait in line in order of arrival.

1. Formulate the Problem

This first step is crucial to the whole process. The exact nature of the problem must be specified and goals must be established. In our example the problem could be that customers are waiting too long. The bank wishes to minimize the waiting time so that customers will be satisfied. The bank teller process would need to be defined to include each step of the process, including what constitutes waiting time.

2. Specify Model

Once the modeler understands the inner workings of the system he or she can then specify a model that captures the most important aspects of the system without including unnecessary information. The modeler must keep in mind the problem to be solved. For the bank teller system a modeler could specify one server (the teller), customers and a waiting line or queue.

3. Build Model

This step consists of three parts: (1) develop the model, (2) collect data, and (3) define experimental controls. Before the model is developed, a simulation tool must be selected. Choices include hand computations (a table or other record of some sort) or a computer. If a computer is chosen, then a specific software and/or hardware package must be selected. Once the modeler has chosen his or her simulation option, the model must be developed. The modeler must divide or group the model into basic elements, decide on inputs and maintain clear documentation.

The next step is to collect data that describes the system (such as the arrival rate of customers to the bank teller window). The modeler must also collect data on the performance of the system. This could be the number of customers served during a particular time period, how long it takes to serve each customer, or how long a customer waits in line. Also, data from an alternate example could be obtained. For instance, if, in our bank teller example, the bank one day employed two tellers or another competing bank used a varying number of tellers to meet increased demand, data from these situations might be helpful.

The final step of building the model is to define experimental controls. These controls include over what time period will the experiment be conducted. The bank teller simulation could be conducted for one hour, one day, or just for a peak time period during the busiest hours. In addition, the degree of accuracy of the test statistics must be determined. Most importantly, the number and type of test statistics to be collected, as well as the number and type of output reports, must also be decided.

4. Simulate Model

The components of this step are running the model, verifying the model and validating the model. Running the model includes performing an iteration of the simulation using the means or device selected. Inputs must be made in accordance with the design of the model and the output must be collected.

Verifying the model is the process of ensuring that the model operates in the manner the modeler intended. All elements of the model must operate in the correct manner and all these components must interact in the proper way. In the bank teller example, if the modeler designed a first-in-first-out (FIFO) queue to represent the waiting customers, the queue in the model should operate that way during the simulation.

Validating the model involves ensuring that the designed simulation represents the system or process being studied in a realistic or useful way. If the bank teller takes a five minute break every hour and this makes the customers wait longer, then the simulation should include this characteristic. Often this step will require the collection of more data. The person doing the simulation will want to get other samples that are separate from those that were used to construct the model. Sometimes, however, the system being modeled has not yet been implemented so the collection of data cannot occur because there is nothing yet to observe. In this case validation can be accomplished by comparison with other models. If the system being simulated has undergone limited testing or trials, then the data from those tests can be used to perform validation.

5. Use Model

This step is comprised of running the model the required number of times and analyzing the results. Alternate versions of the model would also be simulated if desired. In the bank teller example this might include running the model first with one teller and then with two tellers. Final results would then be evaluated. Sometimes this evaluation might indicate that changes to the model were needed to be made. Results might not make sense and this should cause the modeler to reverify or revalidate the simulation model. Changes would then be made and the iteration and analysis process repeated.

A part of this step is determining how many runs of the model to conduct. The modeler wants the simulation output to adequately represent the performance of the actual system being studied. The number of required runs can be determined based on how certain and how accurate the modeler wants to be. The output of a simulation run will normally contain a numerical value of a performance measure. The person conducting the analysis of the simulation will want to determine with what confidence the simulation results actually reflect the real process being studied. The test statistics can be used to form a numeric interval which includes the population parameter. This confidence interval (CI) is usually expressed as a 90, 95, or 99 percent confidence interval. This is the probability that the actual parameter of interest falls in the interval. The value α is then obtained by the formula $CI = 1 - \alpha$. For example a 99 percent confidence interval would result in $\alpha = .01$. Further, the modeler wants to specify his sample statistics with a certain degree of accuracy. This accuracy is usually expressed to the nearest unit that is being measured. In the bank teller example, the modeler might want to express service times by the bank teller to the nearest second. This accuracy is denoted by ε. The following steps are detailed by Banks and Carson (Banks and Carson, 1984, 426-427).

Determining Required Number of Runs with a Specified Degree of Accuracy and Confidence Interval

Step 1

Complete a set of initial runs, R_0. Banks and Carson suggest this number of replications is usually at least five.

Step 2

Compute S_0^2, an initial estimate of the population variance, using the formula:

$$S_0^2 = \frac{1}{R_0 - 1} \sum_{r=1}^{R_0} \left(Y_r - \overline{Y}\right)^2 \qquad (1)$$

where Y_r is the value of the statistic for each run and $\overline{Y}$ is the mean of these values:

$$\overline{Y} = \frac{1}{R_0} \sum_{r=1}^{R_0} Y_r \qquad (2)$$

Step 3

Compute R, the required number of runs:

$$R \geq \left(\frac{t_{\alpha/2, R_0 - 1} S_0}{\varepsilon}\right)^2 \qquad (3)$$

where $t_{\alpha/2, R_0 - 1}$ is the value of a Student's t distribution with $R_0 - 1$ degrees of freedom and $\alpha/2$ level of significance.

Step 4

Next conduct $R - R_0$ additional replications.

Step 5

Determine the confidence interval for θ, the population parameter:

$$\overline{Y} - \frac{t_{\alpha/2, R-1} S}{\sqrt{R}} \leq \theta \leq \overline{Y} + \frac{t_{\alpha/2, R-1} S}{\sqrt{R}} \qquad (4)$$

where R is the total number of runs determined in equation (3) and S is computed using equation (1). The value $t_{\alpha/2,R-1}$ is from a Student's t distribution with R-1 degrees of freedom and $\alpha/2$ level of significance.

6. Support Decision Making

This step is the most important. The results of the simulation must be presented to the decision maker in a clear and concise manner that he or she can understand. The best model and simulation in the world is of little use if the decision maker can not understand the model or its outputs. The simulation result should also be presented in such a manner as to reflect clearly the original nature of the problem.

CHAPTER 4

SIMULATION

4.1 Problem Definition

When the project manager for AFATDS contacted the Colorado School of Mines to request the conduct of a simulation, he was primarily concerned with the communications aspects of AFATDS. He did not want this project to involve the actual processor that would perform calculations and maintain the database. A central assumption then became that the processors for AFATDS, the fire support operational facilities (OPFACs), were fully capable of meeting their requirements. The larger concern was AFATDS's ability to transmit large amounts of data via tactical radios which have a relatively slow transmission rate by today's standards.

The basic question then became could the AFATDS at a U. S. Army Heavy Brigade headquarters process 120 requests for fire via FM tactical radios in a one hour period? Testing had been done by Magnovox, but it had been conducted in a medium similar to a LAN. The testing had involved the 120 fire requests per hour level and higher. Significant testing had not been performed, however, on FM tactical radios at the 120 fire mission per hour level. Also actual field testing of AFATDS had occurred with FM radios but not at a high level of radio traffic density.

A secondary part of this problem was at what rate could these fire missions be processed? There was a concern that even if 120 fire requests could be processed by the brigade Fire Support Element (FSE) in an hour, they would be processed to the field artillery delivery units at a slow rate. The concern was that it would take too much time for the fire request to be processed and converted to a fire order. The fire order would have to reach a delivery unit in a timely manner.

Lastly, another objective was to determine if the brigade FSE could process a total of 701 sensory inputs during this same hour long period. A sensory input is defined as some kind of information input that requires some kind of response or action such as the dissemination of intelligence or a fire request.

4.2 Data Gathering

As there is no current US. Army unit using AFATDS in a normal role other than testing, there was no way of measuring the actual processing of 120 fire requests. Data could be collected, however, on most of the components and features of the system. The starting point for the whole system of transmitting messages digitally was then the messages themselves. The different message types were collected as were their characteristics. The most important characteristic was message length. The length of each message would determine the length of each radio transmission. The length, in bits, divided by the radio transmission rate in bits per second would provide the transmission length for each message. Table 1 provides a brief summary of each message type and its purpose.

It was also neccessary to determine the data transmission rates for AFATDS. When AFATDS is first fielded there will be two classes of devices which will be used to transmit digital messages. One class of devices is the type currently in use throughout the Army. These are all components of the current TACFIRE system and are not scheduled to be replaced when AFATDS is first used. These devices are only capable of

TABLE 1: Message Descriptions

Message Type	Description
Fire Request	Request for Fire Support
Radar Report	Report from Counter Battery/Counter Mortar Radar
Intelligence Report	Intelligence Sent by Observer
Survey Report	Record a Target as a Known Point
End of Mission	Mission is Terminated by Observer
Location Report	Observer Reports His Location
Message to Observer	Mission Specifics Sent by Firing Unit to Observer
Shot	Sent by Firing Unit, Rounds Have Been Fired
Splash	Sent by Firing Unit to Notify Observer of Impact
Frag Order	New or Supplementary Orders
Intelligence Summary	A Summary of Enemy Activities/Capabilities
Intelligence Report	Report on Recent Enemy Activities
Fire Support Coordination Measure	Changes to Graphical/Map Control Measures
Acknowledgment	Message to Sender that Message was Received

transmitting messages at 1200 bits per second. The second class of devices is the AFATDS devices which will be fielded with the system. These devices can transmit data at a rate of 4800 bits per second. A further constraint to the system is that if a radio network is comprised of both TACFIRE and AFATDS devices, then that radio network is restricted to data transmission at the 1200 bits per second rate. A listing of message transmission lengths is provided in Table 2. The message transmission lengths listed reflect a data transmission rate of 4800 bits per second.

The next piece of information needed was the arrival or creation rate for these different types of messages. The Field Artillery Board at Fort Sill, Oklahoma provided these data for a scenario involving 60 requests per hour. These rates then had to be doubled to attain the 120 fire requests per hour. It was assumed that the messages would arrive on the radio nets at a rate λ per hour. If these messages are treated as Poisson arrivals, then the inter-arrival times can be considered to be distributed exponentially with mean equal to $1/\lambda$ (Banks and Carson, 1984, 160). Table 3 contains time between creation rates for all messages. Messages not listed in Table 3 do not need to be created because they are only created in response to the messages listed. In addition to the 120 fire request per hour level, rates are listed for 200, 250, 300, and 350 fire requests per hour.

Other information needed was the delay at each node in a radio network due to processing time. Even though the OPFACs were assumed to be capable, the processing time still needed to be incorporated into this simulation. One sample of processor times was available. It consisted of 87 observations. While this was by no means an exhaustive sample, it sufficed for this model. These sample times, rounded to the nearest tenth of a second, are listed in Table 4.

TABLE 2: Message Transmission Times

Message Type	Length (Bytes)	Length (Bits)	Transmission Time (Seconds)
Fire Request	492	3136	.820
Radar Report	300	2400	.500
Intelligence Report	300	2400	.500
Survey Report	508	4064	.847
End of Mission	222	1776	.370
Location Report	300	2400	.500
Message to Observer	306	2448	.510
Shot	228	1824	.380
Splash	228	1824	.380
Frag Order	300	2400	.500
Intelligence Summary	300	2400	.500
Intelligence Report	300	2400	.500
Fire Support Coordination Measure	300	2400	.500
Acknowledgment	100	800	.167

Transmission Rate - 4800 Bits Per Second

TABLE 3: Message Creation Rates

		Time Between Message Arrival (seconds)				
Message	λ (Messages / Hour)	120	200	250	300	350
FIST Fire Request	36	100	70	55	45	35
FO Fire Request	44	80	60	40	35	30
Radar Fire Request	28	120	60	50	40	35
COLT Fire Request	8	500	250	250	250	250
OH-58 Fire Request	4	900	500	500	500	500
Radar Report	22	160				
Intelligence Report	7	500				
Survey Report	26	140				
Location report	21	170				
Frag Order	1	3600				
Intel Summary	1	3600				
Intel Estimate	1	3600				
Fire Support Coordination	10	400				

(Acronyms are defined in section 4.3)

TABLE 4: OPFAC Processor Times (Seconds)

1.2	2.1	2.7	3.4	3.8	4.3	4.6	5.1
1.2	2.2	2.7	3.4	3.8	4.4	4.7	5.2
1.5	2.3	2.8	3.4	3.9	4.4	4.7	5.3
1.5	2.3	2.8	3.7	3.9	4.5	4.8	5.4
1.5	2.3	2.8	3.7	4.0	4.5	4.9	5.4
1.5	2.5	2.8	3.7	4.0	4.6	5.0	5.4
1.5	2.5	3.0	3.7	4.1	4.6	5.0	5.4
1.6	2.6	3.1	3.7	4.1	4.6	5.0	5.6
1.6	2.6	3.1	3.8	4.2	4.6	5.0	5.6
1.7	2.6	3.2	3.8	4.3	4.6	5.0	6.3
1.7	2.6	3.3	3.8	4.3	4.6	5.1	

Assumed Minimum Value = 1.5 Assumed Maximum Value = 5.5 Mode = 4.6

When no data or limited data are available, one may use a triangular distribution to represent a distribution of data if assumptions are made about the minimum, maximum and mode value of the data (Banks and Carson, 1984, 134). Using the data in Table 4 as a basis for assumption, the minimum value for an OPFAC processor time was assumed to be 1.5 seconds. Likewise, the maximum value was assumed to be 5.5 seconds. When the data from the sample is rounded to one decimal place, the value 4.6 seconds appears seven times. This value is assumed to be the mode for the triangle distribution.

4.3 Model Building

SLAMSYSTEM was the simulation chosen to build and simulate this model. The author assumes the reader has a basic understanding of SLAM. Examples of basic elements of SLAM II are shown in Appendix 1. SLAMSYSTEM is a simulation package that uses the SLAM II simulation language. SLAM II consists of nodes and branches. Branches, or activities as they are called in SLAM II, can involve a processing or delay time. Nodes are connected before or after activities to facilitate routing or decision points. There are numerous types of nodes, including queues. Entities flow through a SLAM II network and can represent anything that is processed. Each entity can have an attribute set to describe the entity.

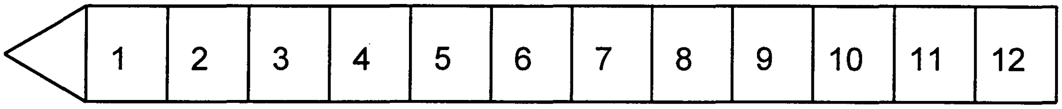
SLAMSYSTEM has a graphical design feature that allows relatively easy construction of networks. Different radio stations, or subscribers to a radio net, are easily modeled as <u>queue</u> nodes. Since only one station can transmit at one time on an FM radio, the use of the radio net is readily modeled as a <u>service</u> <u>activity</u> with one server. The SLAMSYSTEM <u>assign</u> nodes facilitate decision logic for each radio network. Addresses, transmission times and other message characteristics can easily be assigned to entities that represent messages. Figure 1 contains the description of the attribute set of the entities in this model.

The U. S. Army Heavy Brigade fire support digital radio net structure consists of many radio networks. Figure 2 and Table 5 depict this net structure. Ten of these nets are of particular interest. Three maneuver battalions normally make up a maneuver brigade. Each battalion headquarters has a battalion fire support element (FSE). The battalion FSE will be equipped with an OPFAC. The three battalions can be a combination of two mechanized infantry battalions and one armor battalion or two armor battalions and one mechanized infantry battalion. The mechanized infantry battalion has four infantry companies. Each infantry company has a Fire Support Team (FIST) which is equipped with the FIST Digital Message Device (FIST DMD). This is a TACFIRE device. In addition, each infantry company has three platoons. Each of these platoons has a forward observer (FO) who is equipped with a Forward Entry Device (FED), also a TACFIRE device. In the mechanized infantry battalion there are a total of twelve FOs and four FISTs. The armor battalion has a similar structure except there are no platoon FOs. There is only a total of four FISTs.

For this model, there will be two mechanized infantry battalions and one armor battalion. This structure was chosen because there are more subscribers and generally more radio traffic with this structure as opposed to one that has two armor battalions and one mechanized infantry battalion.

Each of the three battalions operates on a digital radio net known as the Battalion Mortar net. These nets are utilized using the previously described TACFIRE devices with the exception of the battalion FSE which will have an AFATDS OPFAC. For this

. FIGURE 1: Attribute Set of SLAM II Entities



SLAM II Entity

Attribute	Description
1	Mark Attribute (Records Time of Creation)
2	Message Type
3	Originator of Fire Request
4	Message Priority
5	Sending Station
6	Receiving Station
7	Temporary Address
8	Firing Unit
9	TACFIRE Message Transmission Time
10	AFATDS Message Transmission Time
11	Used for Message Concatenation
12	Pointer to Batched Entity

FIGURE 2: Brigade Fire Support Radio Net Structure

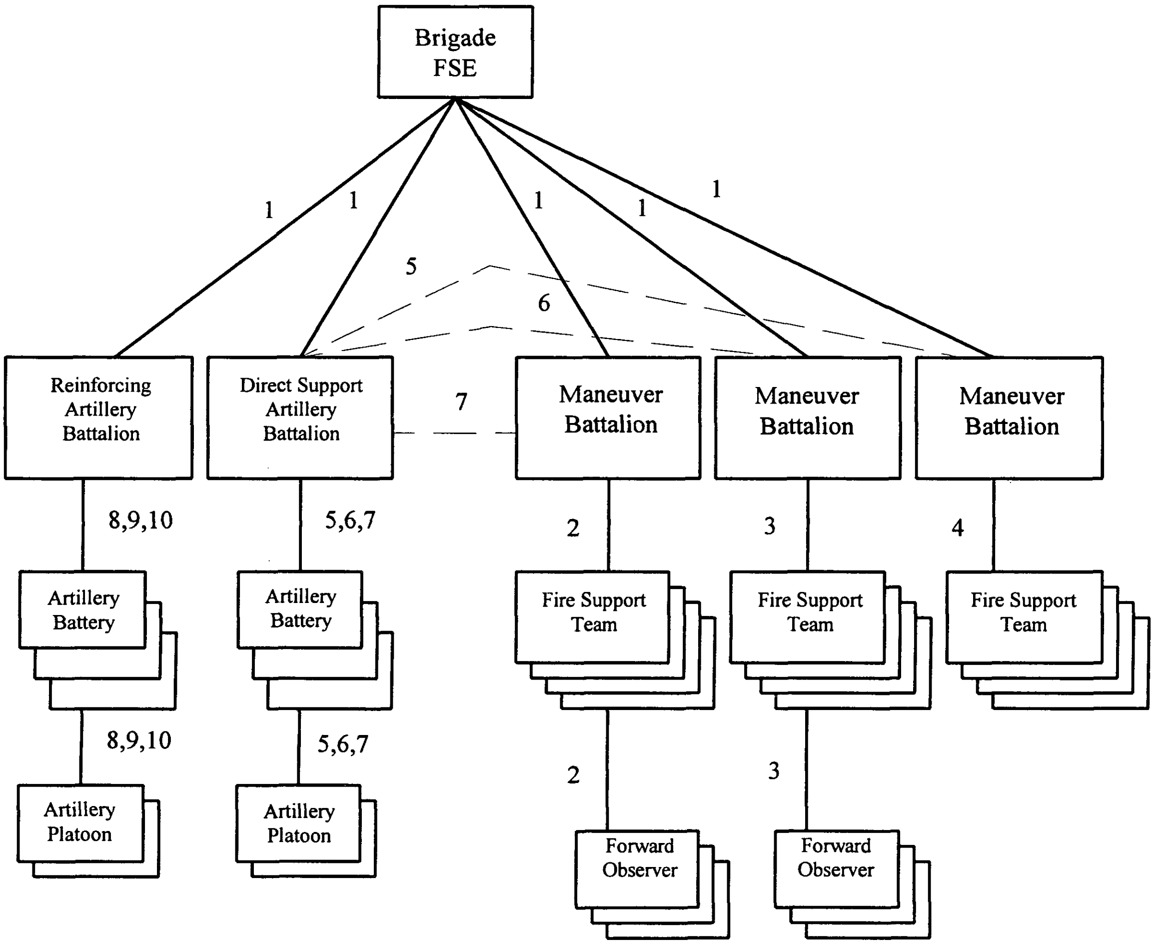


TABLE 5: Legend for Figure 2

Legend

Number	Radio Net	Abbreviation	Meaning
1	Brigade Operations	Bde Fse	Brigade Fire Support Element
2,3,4	Maneuver Battalion Mortar	R BN	Reinforcing Artillery Battalion
5,6,7	Direct Support Artillery Battalion Fire Direction	DS BN	Direct Support Artillery Battalion
8,9,10	Reinforcing Battalion Fire Direction	MN BN	Maneuver Battalion
		BATT	Field Artillery Battery
		FIST	Fire Support Team
		PLT	Field Artillery Platoon
		FO	Forward Observer

reason, data transmission is limited to 1200 bits per second. On these radio nets the fire requests are generated by FOs and FISTs. Another aspect of these nets is that because of the DMD, all messages are 75 bytes in length. Furthermore, these nets feature direct acknowledgments. This means that after a message is transmitted, the receiving station must send a reply to the sending station acknowledging receipt of the message before any other message is transmitted over the net. Therefore, transmissions on the Mortar nets can be modeled as the initial message plus the acknowledgment message. This whole process takes .5 seconds for the initial message (75 bytes transmitted at 1200 bits per second) plus another 2.2 seconds for the receiving station to sense the end of the transmission, wait to ensure the net is clear, and then transmit a 24 byte acknowledgment. The battalion FSEs can also transmit on one other digital net. Each maneuver battalion will operate on one of three fire direction nets maintained by the direct support artillery battalion. These nets will be discussed in detail later. The first maneuver battalion will talk on the first fire direction net, the second maneuver battalion on the second fire direction net and the third on the third fire direction net.

The most important net in the heavy brigade is the Brigade Operations net. This net is composed of the brigade FSE, the three maneuver battalion FSEs, the direct support artillery battalion headquarters and the reinforcing artillery battalion headquarters. In this configuration data is transmitted at 4800 bits per second. Unlike the Mortar nets, this net does not feature direct acknowledgments. Acknowledgments can be sent at a later time. An immediate response is not required.

Other units on the battlefield communicate with the brigade on digital nets, but they do so on nets that are external to the brigade. Three Combat Observation Lasing Teams (COLTs) generate fire requests that the brigade processes. The initial request will arrive at the brigade on an external radio net. Once the brigade receives the fire request, it will then transmit the order to fire to the artillery units. Subsequent messages will be relayed by the brigade to the COLTS on the external net. Similarly, two OH-58 helicopters will normally transmit messages to the brigade. In addition, the brigade FSE will receive counter mortar and counter battery fire requests from at least one counter mortar and/or counter battery radar. The missions are routed from the brigade FSE to the artillery in this particular scenario.

The direct support and reinforcing artillery battalions also have radio nets to communicate with each of their three firing batteries. Each artillery battalion has three fire direction radio nets to talk to its respective batteries. Each battery in turn has two firing platoons. Each platoon communicates on the battery's fire direction net. For example, Battery A communicates with the field artillery battalion on fire direction net number 1. Each of Battery A's platoons also utilize this fire direction net. These nets are also able to transfer data at the rate of 4800 bits per second.

Another important aspect in the operation of these nets is message concatenation. Message concatenation is the combining of several messages into one radio transmission. While a station is waiting to send its next radio transmission, it combines messages into one transmission until its next turn to transmit. AFATDS, as developed, will utilize message concatenation. As an additional informal experiment, a separate model would be developed that did not use message concatenation. The results of this model could be compared to the original model.

Perhaps the most important aspect of any digital FM radio system is the protocol used by the processors and communications devices. Most protocols use some sort of scheme to determine which subscribers on the net can gain access and in what order. AFATDS will use a Net Access Delay scheme to control access. These types of schemes require each station on the net to wait a specified period of time after another station's transmission to access the net and transmit. In the most commonly used scheme of this type, the number one station on the net would be assigned a delay time of .5 seconds. The number two station would be assigned a delay time of 1 second, the third 1.5 seconds and so on until each station was assigned a delay time. Magnovox initially tried this type of scheme and it did not work well in high density message traffic scenarios. What typically occurs is that the number one and two stations end up dominating the net because their access times are shorter than the other stations. As a result they always gain access to the net while those stations with higher delay times seldom gain access. The higher numbered stations in effect become locked out of the net. To counter this situation Magnovox developed a scheme called Deterministic Adaptable Priority Network Access Delay. In this scheme, there is a continuous recalculation of delay times for each station on the net in a cyclic manner. During the first iteration of message transmissions, station number one gets the first opportunity to transmit messages, followed in order by the rest of the stations. During the second iteration, station number two gets the first opportunity to transmit and so on. An example to illustrate this scheme might involve four stations on a radio net. The sequences of stations transmitting their messages would be: 1,2,3,4, 2,3,4,1, 3,4,1,2, 4,1,2,3. In this manner each station on the net would get an equal opportunity to transmit. SLAMSYSTEM's user insert capabilities allows the introduction of FORTRAN programs for use in a network. This enabled a FORTRAN program to replicate this process of net access control. Each of the non-TACFIRE nets uses this scheme to gain control of the radio net.

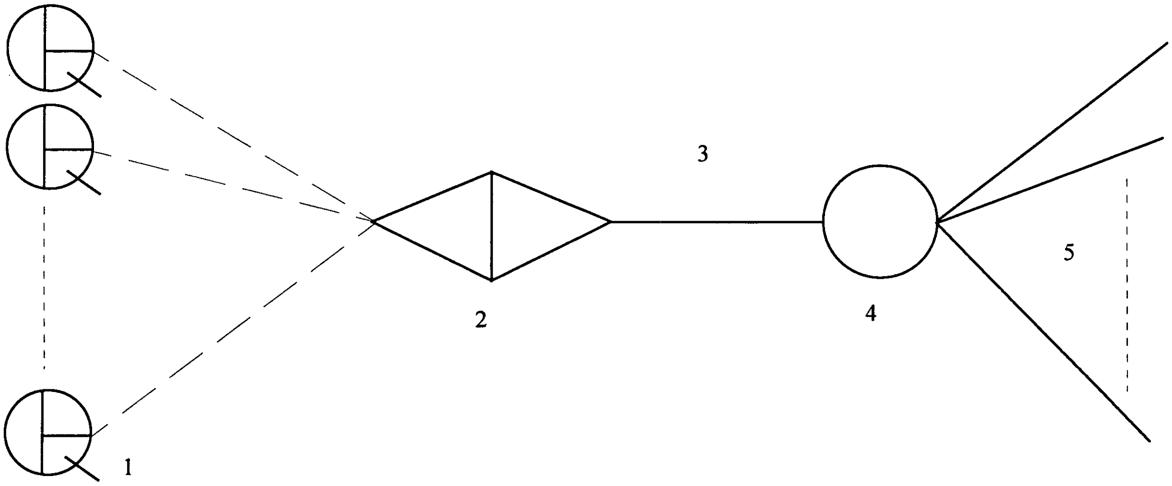
Once these basic themes were explored, the layout of the SLAM II network could begin. The actual network would be composed of six sub networks: one each for the three maneuver battalion Mortar nets, one for the Brigade Operations net, and one for each artillery battalion's fire direction nets, for a total of six sub networks. Each station, or subscriber on a net would be represented by a queue node. The actual radio net is represented by a service activity. A select node was used for each service activity. This select node would chose the queue from which the next entity would come. This entity would traverse the service activity or, in other words, the message would be transmitted. By using the user insert in the select node, the FORTRAN replication of the net access scheme could be used to select the stations to transmit. Assign nodes were used to assign characteristics to each entity in the system. These characteristics, such as the sender and designated recipient of a message, could be examined at each decision point to ensure proper routing of entities or messages.

4.4 Network Logic

The three maneuver battalion sub networks all operate in a similar manner. Messages are created at the appropriate times and routed to the appropriate sender. For example, an FO fire request is created at a create node. Then it is sent to one of twelve FO queues with equal probability. Messages are created in a similar manner and sent to FIST queues for transmission. Before arriving at the intended queue, entity (message) attributes are given certain characteristics by assign nodes. At the assign node, attributes are set for the correct message type and transmission times as well as sender and receiver addresses. After the attributes have been set, the entity is then sent to the appropriate queue which represents the transmission sender. The select node then selects the queue that will transmit a message. Once the message, or entity, is selected, it traverses the service activity. The duration for this activity is based on the contents of either attribute nine or ten, where the transmission time for that message is stored. Attribute nine is used for nets that use TACFIRE devices and attribute ten is used for AFATDS nets. Once a message is transmitted it reaches a decision node. By placing conditions on activities emanating from a decision node, entities are routed to the correct address. By further testing for different message types, entities are processed in the appropriate manner. If the direct support artillery battalion receives a fire order from the brigade FSE, then the order must be transmitted to one of its firing platoons. On the other hand, if the same unit receives an acknowledgment of a previous message, then that entity can be terminated because no further action is necessary after receiving an acknowledgment. Another example would be the transmission of an intelligence summary. This message, once received by a particular station, would be transmitted to all subordinate units. A generalized diagram of this network flow is depicted in Figure 3. A SLAM II network printout of a portion of one of the networks is included in Figure 4.

The seven other nets all operate in this same manner; however, there are some additional features. The other nets have the capability to transmit one message addressed to all other stations on a net. An example would be an intelligence summary transmitted by the brigade FSE. One transmission is made and all stations will receive the message. This is modeled by cloning the message after it has passed the service activity representing the radio transmission. Once the message is cloned, it is routed to all the queues representing stations, except the one sending the original message. The message is then processed like any other message.

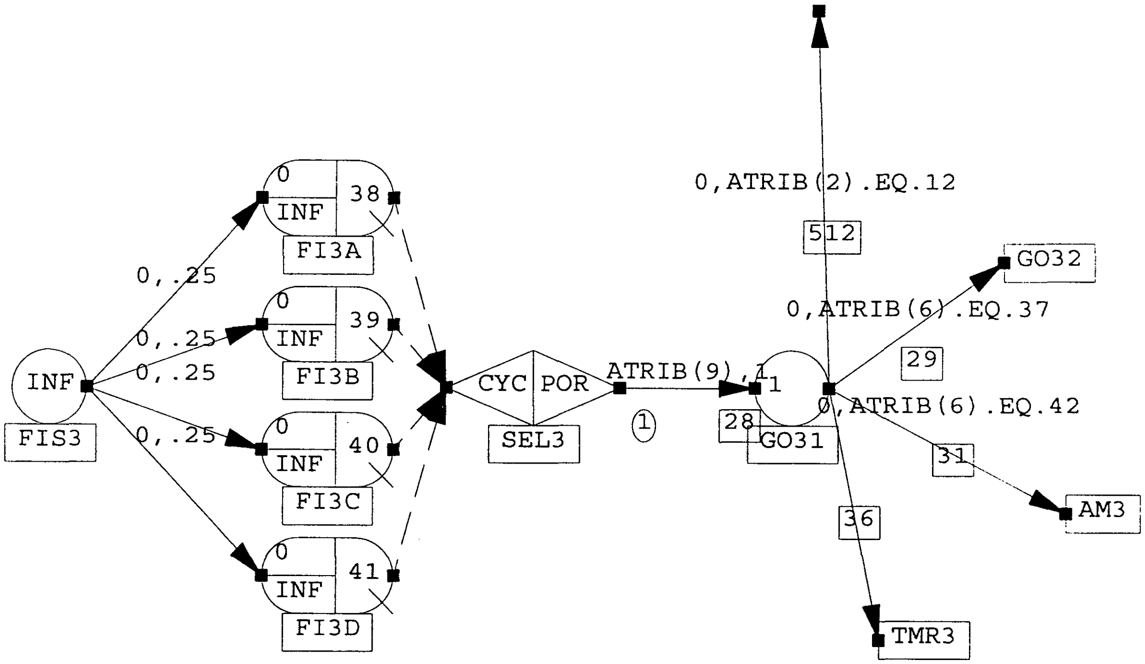
FIGURE 3: Generalized Network Diagram



Legend

Number	Description
1	Queues Representing Stations
2	Select Node
3	Service Activity
4	Decision Node
5	Conditional Routing

Figure 4: Portion of SLAM II Network



Another difference on the AFATDS nets is message concatenation. This combining of messages into one transmission is performed using the SLAM II batch node. Messages are accumulated in a batch node before going to the queue representing the transmitting station. The messages are combined until the total transmission time is ten seconds, the current doctrinal limit on radio transmission times, or until it is the station's turn to transmit a message. A station will not miss its turn at transmitting because it has a concatenated message duration that is not yet ten seconds. In other words, messages do not have to be fully batched at the ten second limit before they are sent to the queue representing a station. Entities are routed in such a manner as to cause the batch to be released even though the batched message transmission time is less than ten seconds.

4.5 Verification

The model when fully designed is fairly large. The network consists of approximately 1000 activities, 76 queues, and numerous other SLAM II nodes. Because of the size and complexity of the network, conducting verification of the model was man-hour intensive. Each decision node and branch had to be tested to determine if the network logic was correct. Not only did each decision node need to be tested, but each possible branch had to checked. This was accomplished by forcing entities to travel each part of the system. In addition, each create node had to be checked to see that entities were being created at the correct times. Assign nodes had to be verified as well. The proper characteristics had to be assigned at the proper place and time. Needless to say, numerous changes were incorporated throughout the process. Once entities were flowing through the network in the proper manner, simple message transmission times were assigned to entities. Then one entity or message of each type could be sent through the system to ensure that the entities traveled through the network in the proper time frame. By using simple values for message transmission times, a hand calculation could be conducted to compare the time values of actual entities at various stages in the model. An FO's fire request, for example, would be routed from the FO to the FIST, then from the FIST to the battalion FSE, then from the battalion FSE to the brigade FSE, then from the brigade FSE to the artillery battalion, and then to the firing platoon for a total of five transmissions. If a value of one was assigned to the attribute for message transmission time when it was created at time zero, then the entity should reach the firing platoon at time five. Once entities were checked using this procedure, then the delay times at each node due to processor time was inserted.

4.6 Validation

This task proved difficult to accomplish. A partial validation was conducted in two ways. As previously mentioned, other members of the project team built and ran a model using a different software package. Output of trial runs for the two models were constantly compared for differences in output. Initially, there was a wide gap between the timeliness of the processed missions. This model was showing that it was taking in excess of four minutes for fire requests to be processed, while the other model was producing output that indicated it was taking under thirty seconds. Upon examining this model an error was discovered in the message transmission process. When this error was corrected, fire request processing times only varied by plus or minus five seconds. This time difference results because the other model does not have a user interface that can replicate the net access scheme used by AFATDS. That model uses a protocol that is similar to one used by a LAN. Although this was not a stringent validation, it was adequate for a system that has not yet been implemented.

In addition to the above validation process, an informal comparison was made with limited data available from the Field Artillery Board at Fort Sill, Oklahoma. The board had used a combination of OPFACs and FM radios and other simulators to replicate a scenario in which 170 fire requests were submitted in a one hour period. All of the fire requests were processed in that one hour, but no other types of messages were transmitted simultaneously. This did not provide a perfect validation but as shall be explained later, these results do provide some evidence that the SLAM II model is a valid one. Further testing will be accomplished at Fort Sill in the future and more exact data will be available for comparison. Likewise, when AFATDS is fielded, more comparisons can be made if the model is to be used further.

4.7 Simulation

Before conducting any runs with the model, a decision had to be made concerning which statistics to collect. Based on the problem definition, three areas were deemed most important. First, the rate of fire requests processing had to be collected. This would be the number of fire orders received at the firing platoons divided by the number of fire requests created. This was accomplished by counting the number of entities traveling along the activities directly following the fire request create nodes and by using a SLAM II <u>collect</u> node to count all the fire orders received at the firing platoons.

The second important piece of information that would be collected was the average time for all fire orders reaching the firing platoons. This would be an important measurement of AFATDS's ability to process fire requests in a timely manner. The same collect node that counted fire orders would calculate the average time for all fire orders. This was accomplished by comparing the SLAM II variable TNOW, which is the current simulation time, with the time stored in the first attribute of the message entity, the mark attribute, or time of its creation. In this manner the time interval for all fire requests from origin to receipt at the firing platoon was calculated. SLAMSYSTEM then calculated the mean time for all firing orders to be received.

Once the decision concerning test statistics was made, some initial test runs were made. First, comparisons were made between a model that used message concatenation and one that did not. Initial runs were made at different levels of fire requests per hour. The different levels were 120, 200, 250, 300, and 350 fire requests per hour. These rates were picked to see if there were differences at different fire request rates. A summary of these initial results are shown in Table 6 and Table 7. An analysis of these results indicates that the model using concatenation performed better than the one that did not use it. Firing processing rates differed slightly, but a bigger disparity is evident in the timeliness of the order to the firing platoons. At 250 fire missions per hour, the average time for fire orders to reach the firing platoons starts to increase for the model that does not use message concatenation. As the fire request rate increases, the timeliness of the fire orders starts to decline. Based on this observation, no further testing of the model that does not use concatenation was warranted. The model is still available if further research in this area becomes necessary.

A third measurement would be the number of messages processed by the brigade FSE in the one hour time period. This would be accomplished by counting the number of entities that reach the queue representing the brigade FSE. This count would reflect all types of messages except acknowledgments. Acknowledgment messages are terminated once they are transmitted across the net. This is not a problem because acknowledgments are not considered a sensory input.

Analysis of the ten service activities from these initial runs indicated that these activities (which represented the ten radio nets) were under-utilized for fire request rates

Table 6: Mean Fire Order Times for Concatenated
and Non-Concatenated Test Runs

Fire Request Rate	Concatenation	Mean	Std Deviation	Minimum	Maximum
120	N	28.1	11.7	8.4	58.0
120	Y	22.4	10.5	6.76	103.0
200	N	26.5	11.0	7.99	69.5
200	Y	20.7	7.60	7.05	61.7
250	N	33.7	13.9	8.16	79.4
250	Y	21.5	7.90	7.13	52.3
300	N	94.2	34.9	12.4	179.0
300	Y	22.0	8.49	7.79	73.8
350	N	298.0	176.0	7.92	593.0
350	Y	22.1	8.51	7.87	66.8

Table 7: Fire Request Processing Rates and Brigade Inputs

for Concatenated and Non-Concatenated Test Runs

Fire Request Rate	Concatenation	Fire Requests Created	Fire Requests Processed	Rate (%)	Brigade Inputs
120	N	119	119	100	389
120	Y	120	120	100	389
200	N	195	192	98.5	532
200	Y	195	192	98.5	532
250	N	251	248	98.8	632
250	Y	251	248	98.8	632
300	N	296	283	95.6	694
300	Y	296	293	99.0	704
350	N	348	290	83.3	740
350	Y	348	346	99.4	796

below 300 fire requests in an hour. For each run, SLAMSYSTEM calculates the utilization rate for each service activity. In addition, for the lower rates the total number of inputs to the brigade FSE was well below the 701 input level. These net utilization rates and brigade input counts are listed in Table 8. Based on these observations, the author decided to conduct more extensive testing at the 300 fire request per hour level. Fire request rates, mean times for fire orders to the platoons, and brigade inputs could all be calculated using one set of runs.

Table 8: Net Utilization for Test Runs

Net Number and Utilization Rate

Fire Request Rate	1	2	3	4	5	6	7	8	9	10	Bde Inputs
120	.243	.257	.138	.181	.051	.045	.042	.029	.029	.036	389
200	.294	.257	.183	.249	.061	.059	.051	.045	.045	.050	532
250	.325	.400	.215	.277	.076	.078	.074	.061	.047	.048	632
300	.379	.446	.239	.335	.083	.081	.080	.074	.065	.048	704
350	.475	.466	.295	.359	.097	.112	.098	.070	.062	.070	796

The next step was to conduct an initial set of runs at this 300 fire request per hour level. Five iterations of the model scenario were executed. The results of these runs are listed in Table 9. Using a confidence interval of 99 percent ($\alpha = .01$) and a degree of accuracy of either one second ($\varepsilon = 1$) or one percent ($\varepsilon = .01$) depending on the statistic, the required number of runs was calculated. With a level of significance of .005 and

Table 9: Results of Five Initial Test Runs

Run	Mean Fire Order Time	Fire Requests Created	Fire Requests Processed	Processing Rate	Brigade Inputs
1	25.2	300	298	.993	730
2	24.1	325	320	.985	735
3	25.2	293	291	.993	694
4	24.9	318	316	.994	768
5	24.3	313	311	.994	716
Mean ($\overline{Y}$)	24.74			.992	
Variance (S_0^2)	.263			.00396	
R≥	5.565			3.318	

$R_0 - 1 =$ 4 degrees of freedom, a value from a Student's t distribution of $t_{.005,4} = 4.60$ is obtained. Using equations 1, 2, and 3, the required number of runs, R, is calculated. The required number of runs for the mean fire order time was 6 and for the processing rate the required number was 4. Instead of doing six runs, however, it was determined that twenty runs should be made. Twenty iterations of the simulation would result in better confidence in the results.

After completing fifteen more iterations of the simulation, equations 1, 2, and 4 were used to complete means, variances and confidence intervals. The degrees of freedom, R-1, now became 19, resulting in a value from a Student's t distribution of $t_{.005,19} = 2.78$. The results of all twenty runs and the resulting statistics are listed in Table 10.

Table 10: Results of Simulation Runs

Run	Fire Order Mean Time	Fire Requests Created	Fire Requests Processed	Processing Rate	Brigade Inputs
1	25.2	300	298	.993	730
2	24.1	325	320	.985	735
3	25.2	293	291	.993	694
4	24.9	318	316	.994	768
5	24.3	313	311	.994	716

Table 10: Results of Simulation Runs (Continued)

Run	Fire Order Mean Time	Fire Requests Created	Fire Requests Processed	Processing Rate	Brigade Inputs
6	24.6	331	329	.994	770
7	25.4	319	317	.994	753
8	24.3	324	322	.994	741
9	24.3	345	342	.991	766
10	24.4	319	317	.994	756
11	25.3	336	335	.997	776
12	23.3	317	314	.991	731
13	24.6	355	354	.997	830
14	24.2	317	313	.987	771
15	24.4	360	357	.992	806
16	24.2	320	317	.991	771
17	23.1	290	290	1.000	696
18	24.6	326	322	.988	778
19	24.6	334	331	.991	799
20	23.7	295	291	.986	703
Mean ($\overline{Y}$)	24.435	321.85	319.35	.992	754.5
Variance (S_0^2)	.3719	18.8604	18.7933	.003752	36.5506
Confidence Interval	24.045, 24.825			.990, .994	

CHAPTER 5
RESULTS / CONCLUSIONS

5.1 Results

A number of observations can be made about the results of this simulation. First, the simulation of this model indicates that AFATDS is certainly capable of processing 120 fire requests in one hour. In fact, it is capable of processing over 300 fire requests in an hour. During twenty runs in a 300 fire request per hour scenario, the system processed 99.2 percent of the requests for fire it received. This is certainly an acceptable rate. One cause for the rate not being 100 percent is that some fire requests were probably generated late in the scenario, preventing the fire orders from reaching the firing platoons before the one hour time limit. There is no other evidence to suggest fire missions are being lost or delayed for an exceedingly long period of time. The answer to the basic question of will AFATDS process 120 fire requests in an hour is "yes".

The results of the simulation has also shown that AFATDS should be capable of processing these requests for fire in a rapid fashion. In over twenty iterations the average time of all fire orders to the firing platoons was under 26 seconds. This means that from the time the forward observer presses the transmit button on his communication device until that request is turned into a fire order at a field artillery platoon, on average, only an astonishingly short twenty six seconds has elapsed. If the artillery platoon receiving this mission is a well trained unit, fast and effective artillery fire will be delivered. This is the whole objective of fielding AFATDS. This simulation result is extremely important, therefore, because it demonstrates AFATDS's ability to process a large volume of fire requests and other message traffic with relatively little effect on the timeliness of the transmission of fire orders to the artillery. As future battlefield fire requests increase in frequency and number, AFATDS will still be able to complete its mission. This simulation demonstrates that AFATDS performance will not deteriorate due to an increase of fire requests and other information.

Another important simulation result is that AFATDS should be able to perform in this superior manner at the required level of sensory inputs to the brigade FSE. The average number of inputs to the brigade FSE over the twenty runs of the simulation was 754. The brigade FSE will be able to process 120 fire requests in conjunction with the high sensor input. An interesting piece of data related to this issue is the utilization of the Brigade Operations net. In examining the SLAMSYSTEM output for the twenty runs of the simulation, the author noted that the highest utilization during the runs was under 42 percent. This means that for the entire twenty runs, 58 percent of the time the Brigade Operations net was idle. This would seem to suggest that AFATDS should have even more capacity than originally thought. The AFATDS system should be capable of meeting the three challenges that were specified in the problem definition and examined during this study.

One final observation is needed, however. No matter how good the system is, the fog of combat and combat training has a strange effect on people and equipment. In the heat of battle, soldiers can be compelled to do things they otherwise might not do. Despite the rigors of combat and combat training, soldiers and their leaders will have to employ the system correctly, have confidence in its capabilities, and trust its operation. This will only occur through long term training and familiarization. Soldiers are, by their nature, skeptical of new systems of any kind. Leaders will have to become experts on the system and train their soldiers well. An unused or untrusted system will only have a detrimental effect on a combat force. If AFATDS is properly integrated into the units of the U. S. Army, it should enjoy a reputation as a superior digital fire support system. More importantly, it will fill the Army's need for a fast and efficient way to deliver fire support to units engaged in a future conflict on an advanced battlefield.

5.2 Future Research

This study only focused on three aspects of AFATDS's communications link. Many more areas can be studied. A future consideration for study might involve net saturation since at some fire mission request rate level things may slow down. Another area that could be examined is the maximum number of stations per radio net. An increased number of stations per radio net could have an impact on the fire request processing rate or the timeliness of fire orders to the firing platoons.

Once the AFATDS system is actually fielded, more data concerning the system will be available. This information could be incorporated into this model or a new model could be developed. Models need to be constantly upgraded if they are to be used in future decision making.

A larger scale simulation could involve alternate net structures. Different radio net structures could be examined or several net structures could be pitted against each other in order to find the most efficient one. The Army currently has different types of units that have different missions. One particular net structure or configuration of AFATDS might be better for a particular unit than another. In addition, as weapons systems become more complicated and armed force structures evolve, alternate net hierarchies may become evident. Clearly, alternate configurations and/or protocols of AFATDS could yield better performance, depending upon circumstances.

REFERENCES CITED

Banks, Jerry, and John S. Carson, II. 1984. Discrete-Event System Simulation. New Jersey: Prentice-Hall, Inc.

Cobb, Richard, Edward R. Mansfield, and Joseph M. Mellichamp. 1992. Development of Design Guidelines for Local Area CSMA/CD Networks. Simulation. 58 (4): 270-279.

Edwards, G., and R. Shankar. 1992. Modeling and Simulation of Networks Using CSIM. Simulation. 58 (2): 131-136.

Fdida, Sege, H.G. Perros, and Andrzej Wilk. 1990. Semaphore Queues: Modeling Multilayered Window Flow Control Mechanisms. IEEE Transactions on Communications. 38 (3): 309-317.

Habib, Ibrahim W., Tarek, N. Saadavi, and Mohsen A. Saleh. 1993. Simulation Analysis of a Communication Link with Statistically Multiplexed Bursty Voice. IEEE Journal on Selected Areas in Communications. 11 (3): 432-442.

Kosbar, Kurt L., and William H. Tranter. 1994. Simulation of Communication Systems. IEEE Communications. 32 (7): 26-35.

Law, Averill M., and Michael G. McLomas. 1994. Simulation Software for Communications Networks. IEEE Communications. 32 (3): 44-50.

Pritsker, A. Alan B., C. Elliot Sigal, and R.D. Hammesfahr. 1994. SLAM II: Network Models for Decision Support. Scientific Press.

SELECTED BIBLIOGRAPHY

Jeruchim, Michel C., Philip Balaban, and K. Sam Shanmugan. 1992. Simulation of Communication Systems. New York: Plenum Press.

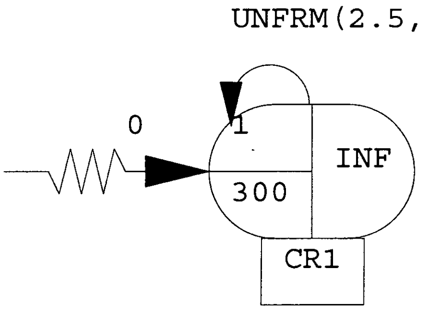
Magnovox Electronic Systems Company. 1993. System Segment Specification for the AFATDS Version 1. Fort Wayne, Indiana: Magnovox Electronic Systems Company.

Mendenhall, William, and Terry Sincich. 1992. Statistics for Engineering and the Sciences. San Francisco, California: Dellen Publishing Company.

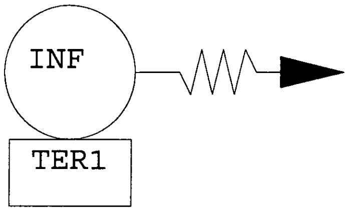
Pritsker, A. Alan B. 1986. Introduction to Simulation and SLAM II. West Lafayette, Indiana: Systems Publishing Corporation.

SLAMSYSTEM 4.2. 1993. West Lafayette, Indiana: Pritsker Corporation.

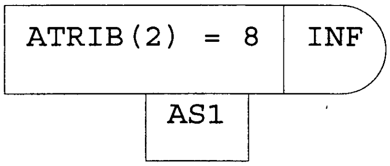
APPENDIX A
BASIC SLAM ELEMENTS



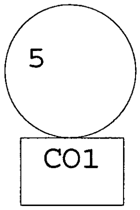
CREATE NODE



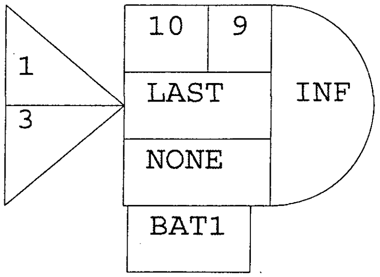
TERMINATE NODE



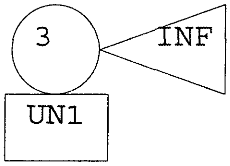
ASSIGN NODE



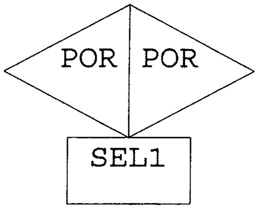
GOON NODE



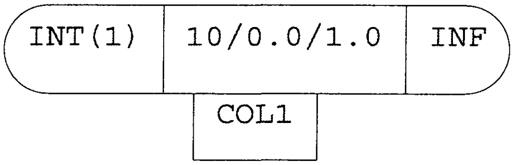
BATCH NODE



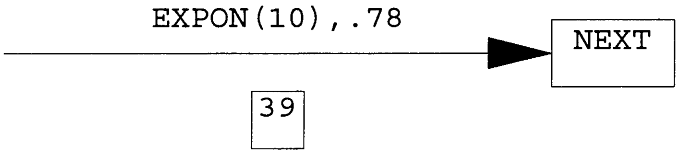
UNBATCH NODE



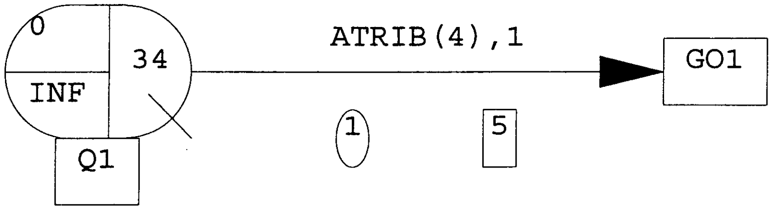
SELECT NODE



COLLECT NODE



ACTIVITY



QUEUE NODE SERVICE ACTIVITY