1994

Object-Oriented Modelling of Military Communications Networks

Bailey, Michael; Kemple, William; West, Michael; Chase, Charles

Palgrave Macmillan Journals on behalf of the Operational Research Society

Bailey, Michael, et al. "Object-oriented modelling of military communications networks." Journal of the Operational Research Society 45.10 (1994): 1108-1122.
http://hdl.handle.net/10945/60787

J. Opl Res. Soc. Vol. 45, No. 10, pp. 1108–1122
Printed in Great Britain. All rights reserved
0160-5682/94 $9.00+0.00

Object-Oriented Modelling of Military Communications Networks

MICHAEL BAILEY, WILLIAM KEMPLE, MICHAEL WEST and CHARLES CHASE
Naval Postgraduate School, Monterey, USA

Military communications networks are large complex systems; their physical dynamics, composition, workload characteristics and operational goals make them very hard to model and analyse. It is particularly difficult to build a generalized model that can adequately fit each instance, generate a workload that exercises the network in a realistic way, and measure network effectiveness in terms of the operational goals. In this work, we describe our approach to building such a model for the US Marine Corps, and demonstrate the model's use by solving a particular equipment allocation problem. We combine object-oriented simulation with new concepts for workload modelling and measuring effectiveness that are based on the operational tasks the Marines perform.

Key words: object-oriented simulation, command and control, military operations research

INTRODUCTION

Military communications networks are large complex systems. This can also be said of their civilian counterparts, but there are several distinguishing characteristics that make military networks particularly difficult to model and analyse. Four key differences are their physical dynamics, their composition, their workload characteristics and their intended goal.

Physically, military communications networks are rapidly installed, removed, and reinstalled, and they vary in size (often by orders of magnitude) from network to network. Typically, they are heterogeneous networks of voice and data nets, where each net is like a party line—only one party (unit) can talk at a time, and all the other units on the net can listen. The message workloads that these networks process are neither stationary nor homogeneous; the type and tempo of operations determine the message sending requirements, but only indirectly. The intended goal of the network is not to provide communications services to the individual subscribers *per se*, but rather to enable them jointly to perform, in a timely manner, the operational tasks necessary for mission accomplishment—support to the force as a whole is more important than to independent subscribers.

As a consequence, it is difficult to build a generalized model that can adequately fit each instance, it is difficult to generate a workload that exercises the network in a realistic way, and it is difficult to measure network effectiveness in terms of the operational goal. A network's effectiveness cannot be measured directly by aggregating the traditional communications effectiveness measures (e.g. queue length and message waiting times) of its pieces. For example, a small mean message waiting time is of little value if the interdependent sequence of messages required for an artillery preparatory fire mission is not completed prior to an assault, and the fire mission is not conducted.

Much past effort has been expended evaluating the effectiveness of military communications networks, usually using analytic, approximation, Monte Carlo, or system simulation methods. All of these require some representation of the communications workload that the network must process. Typically, simulation-based evaluations have used stochastic models for this purpose. To a large degree, the choice of evaluation technology, the model development and implementation costs for the evaluation, and the degree of acceptance and usability of the end product, are determined by how closely the workload model reflects reality (its fidelity).

The low end of the fidelity spectrum is occupied by models that generate message-sending requirements using stationary (or even deterministic) arrival processes. They usually generate

Correspondence: M. Bailey, Department of Operations Research, Naval Postgraduate School, Monterey, California, USA

message-sending requirements from point A to point B according to a stationary Poisson process. This simple model is often selected because evaluation of the resulting communications traffic process is analytically tractable. This approach usually allows inexpensive development, but at the expense of model fidelity, usefulness of the results, and acceptance of the results by users. Examples of this approach are References 1 and 2.

At the other extreme, there are models that attempt to induce a realistic communications work-load by simulating the evolution of combat. Some modelling, experimental design, and computational drawbacks of this approach are readily apparent.

1. In order to generate realistic communications traffic, a high-resolution combat simulation model is needed. These models carry significant model development and programming costs and require voluminous, expensive input data. Confidence in the output is very tightly linked to confidence that data, and conclusions drawn from the results of high-resolution combat models, are valid only for the specific scenario used.
2. Because the communications workload is determined by the combat simulation, which can take many distinct turns for the same input data, many replications are required to distinguish differences in effectiveness due to changes in the factors from variability induced by the combat simulation.
3. Including extreme detail costs computational effort within each replication of the (obviously terminating) scenario; this results in extremely large computing requirements for the desired accuracy.

Examples of high resolution combat models for communications network effectiveness analysis are the Network Assessment Model[3], and a traffic simulator developed at the Naval Research Laboratory[4].

In this paper, we describe how we combined object-oriented simulation with new concepts for workload modelling and measuring effectiveness to build a flexible tool for modelling and analysing US Marine Corps communications networks. Both the workload and the effectiveness methodologies are based on the actual operational tasks that Marines perform.

The remainder of this section is US Marine Corps specific. We briefly describe a real-world command and control structure that logically connects the operational tasks that the Marines perform to communications requirements. This structure is referred to often. In the next section, it is used to build a communications traffic generation model that occupies the middle ground between the extremes of the stationary, point-to-point models and the high resolution combat models described above. We know of no other model that occupies this middle ground.

A similar command and control structure for other military organizations would be required to modify our model and analyse the communications networks. We note, however, that these data are but a subset of the data required to use a detailed combat model to generate message traffic.

The Marine Corps command and control structure that we use is described in detail in Reference 5. Items of particular interest to us are broad operational subtasks (BOSTs) and message exchanges. A BOST is a co-operative effort undertaken by a group of Marine Corps units to satisfy a single operational requirement. Examples of BOSTs are a request for helicopter evacuation of a wounded Marine and a request for artillery support by a forward observer (described below). All of the actions and communications required by Marine units to complete the operational subtask (BOST) are considered part of it. This includes a small precedence network of individual point-to-point message exchanges.

Many details are specified for each message exchange, including the units that transmit and receive the message, the doctrinal radio net that the message is usually passed on, and the message duration and priority. Since there is a logical relationship from mission to message, and we know these details, we can generate realistic, interdependent communications traffic by controlling the mission rates: we generate BOSTs with relative frequencies that are appropriate for the type of combat under analysis, and the BOST/message exchange structure provides the desired realism. (This process is described in detail below.) Furthermore, if there is a need, we can generate the BOSTs dictated by a combat-model-like script to get much of the realism of a combat model without the large development costs.

The remainder of the paper is organized as follows. Three sections are used to describe our object-oriented simulation model of the US Marine Corps communications networks. These are followed by a section that proposes a method for measuring the effectiveness of communications architectures and another that explores using this method to support a specific decision involving the deployment of frequency-hopping radios within existing Marine Corps communications architectures. The final section presents our conclusion and discusses future research. Statistical development is given in the appendix.

BASIC OBJECTS

Our model has four fundamental object types, BOSTs, the traffic generator, nets and units. In this section, we provide its salient details by describing the properties of these object types.

BOST objects

To assess the relative merit of a specific communications architecture, we must stress it in a realistic fashion. But, we want our conclusions to be independent of a specific scenario of events, so we provide the stress through the BOST/message exchange framework that was briefly described in the introduction. The tasks that the Marine Corps organization will undertake, and the messages that will have to be exchanged on the communications network have been identified and categorized in Reference 5. Consider the Artillery Direct Fire Support BOST shown in Figure 1. This BOST was initiated by a forward observer, and involves the co-operation of five other units. Message exchanges that must be accomplished to complete Artillery Direct Fire Support include the original call for fire, messages clearing the fire mission up the chain of command, messages relaying the clearance back down the chain, spotting and firing directions exchanged between the forward observer and the battery fire direction centre, and end-of-mission and surveillance messages.

There is some concurrency of message exchanges in this BOST, as well as a simple precedence structure between them. Messages are exchanged on four different radio nets. The specification for each message exchange includes the message format and content, the sender and receiver, the radio net to be used, and the message duration. The data structures used in the BOST object allow for any precedence structure and any number of simultaneous receivers, so we can construct BOSTs as complex as necessary to achieve realism.

The BOST objects also carry information that is used to help determine the effectiveness of the architecture modelled. Each BOST must be completed by a deadline. If it is not, the architecture is penalized a certain number of points per time unit late. Both the deadline and the penalty rate are determined by the type of BOST.

The traffic generator object

To generate traffic for the Marine Corps communications system, the traffic generator object generates a sequence of BOSTs and their originating units. Each BOST then generates its own sequence of message exchanges, with their associated requirements.

Each unit, j, in the organization initiates each BOST, i, at relative rate $\lambda_{i,j}$. That is, combination (i, j) initiates with this rate relative to the other BOST-unit combinations.

For efficiency and centralization of control, we generate the BOST-unit pairs in a central process shown in Algorithm 1: the heart of the central BOST generation process.

```
while (not TIME'S UP)
  sample DELAY with mean = 1/λ
  wait DELAY
  choose a BOST and UNIT
  tell UNIT to INITIATE BOST
end while
```

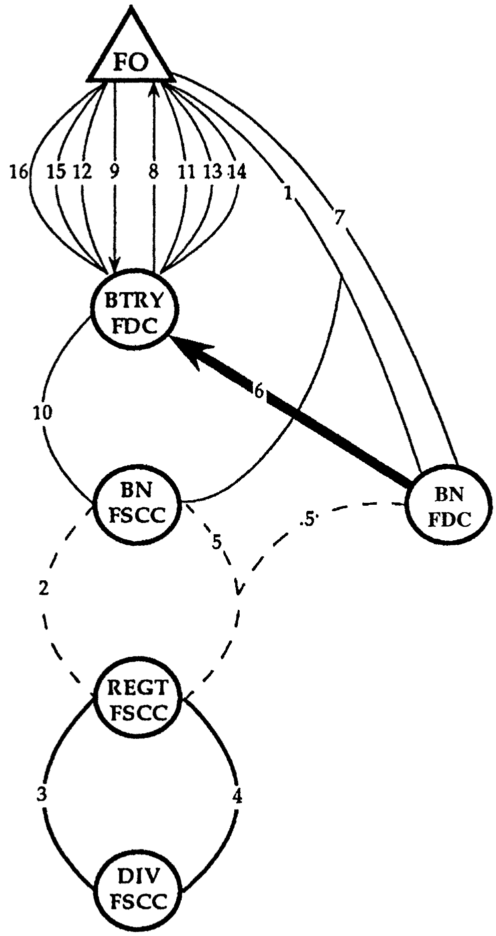

FIG. 1. *The artillery direct fire support BOST. Each arc represents one message exchange, the numbers give their sequence, and different line types indicate different radio nets. The triangle and circles are the units that must communicate to accomplish this BOST. They are the Forward Observer (FO); the Artillery Battery (BTRY) and Battalion (BN) Fire Direction Centres (FDC); and the Infantry Battalion (BN), Regiment (REGT), and Division (DIV) Fire Support Coordination Centres (FSCC).*

where $\lambda = \sum_{(i,j)} \lambda_{i,j}$. Given BOST type i and unit j, the BOST-unit combination (i, j) is chosen with probability $\lambda_{i,j}/\lambda$. If the central delays are chosen to be exponential, then each BOST-unit initiation is a filtered Poisson process. Otherwise, each time between BOST-unit initiations is a sum of a geometric number of independent and identically distributed (iid) delays. The distribution of BOST instances is pictured in Figure 2.

Thus, our detailed scenario will be the sample path of this BOST dissemination process. Note that it is possible to replace this stochastic traffic generator object with one that accepts a time series of BOST generation data from a detailed combat model or a Marine Corps exercise as input. This new object can still use the model's BOST objects to generate the realistic message workload. This ability to interchange traffic generators and produce a realistic workload without having to input a voluminous, detailed time series of messages arises from our modular, object-oriented design, including the logical mission-to-message relationship built into the BOST objects.

Net objects

Radio net transmission time is the only limited resource in our model. A net may be thought of as a one-talker-at-a-time party line. All units connected to the net, called subscribers, can hear every message transmitted on the net, but only one subscriber may transmit at any time.

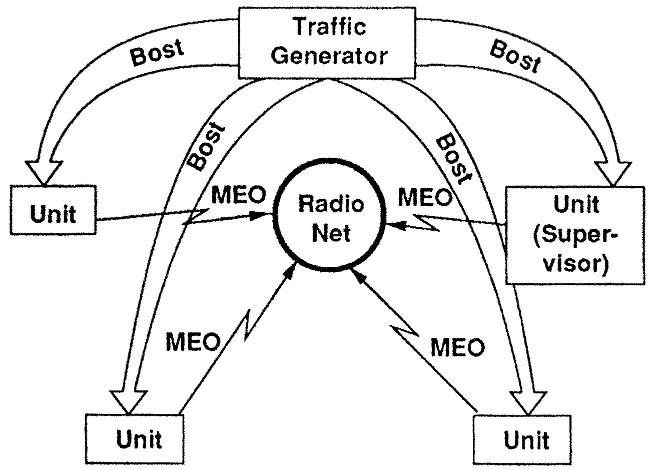

FIG. 2. *The relationship of the traffic generator, units, and the net resource for one net.*

The Marine Corps uses a four-tiered priority scheme (Flash, Immediate, Priority, and Routine) for formal message traffic on radio nets. The nets in our model use a highest-priority-first discipline. When an opportunity for transmission takes place, the net polls each of the subscribers and chooses a unit with a waiting highest-priority message at random. This message selection process may be a bit more democratic than the real system, as no attention is paid to the sender's rank and there is no net chit-chat. This discipline is easily varied by changing traffic selection methods of the net. (Methods of an object are executable routines that are part of the object.)

Nets are constituted at the beginning of the model run. This takes simulation time to model the administrative overhead involved in constituting a real radio net.

Radios are also subject to failure, and are capable of switching to different nets. If a unit experiences a radio failure on an important net, it may replace the radio with a spare, or switch a radio from a less important net. Each time that a radio attempts to enter an existing net, an administrative cost (time) is incurred and the net is not available for normal traffic. Later we will describe a specific use of our model to analyse high-technology radio equipment allocation schemes. The matter of net reconstitution and net entry becomes critical in this analysis.

Unit objects

The unit is the base object type from which all of the Marine Corps communication nodes are derived. Unit objects provide communications support for Marine Corps commands which range from a platoon (≈ 45 marines) to a division (≈ 19000 marines and sailors). Each unit owns radios which are housed in the unit's radio array, and each radio is connected to a radio net. Individual units are characterized by their place in the organizational hierarchy, their physical location, the composition of their radio array, the mean relative rates at which they initiate different BOSTs, and the net memberships of their radios.

The traffic generator stimulates the units by telling them to initiate BOSTs. The stream of BOST initiation instructions (which BOST to initiate and which unit to do it) is representative of what would be expected to occur in the type of combat chosen for analysis.

Each BOST is pursued via the execution of message exchanges between units. The unit that initiates the BOST consults it to determine the first message exchange, finds all of the units which must receive the message, determines the nets required to reach these units, and submits the message to the appropriate set of radios, one radio per net. The radios act as prioritized queues of messages and, when transmitting, initiate the busy periods of their attached nets. When a unit receives a message, it consults the BOST to determine the next message exchange, determines the appropriate nets, submits the new message to the radios, etc. This process continues until all of the message exchanges for this BOST are completed or until the BOST is cancelled.

There are circumstances under which the unit will not be able to reach some of the intended receivers on the nets specified in the message exchange, so each unit contains a complex routeing mechanism that can construct a sequence of units capable of relaying the message to the intended receiver.

TAILORING THE MODEL: JAMMERS, FAILING RADIOS, AND ROUTEING

In this section, we describe features of model objects that were added or modified for a specific radio allocation study. The study itself is described later. This section might be seen as an advertisement for object-oriented simulation modelling, as the characteristics described are add-ons; they were not part of the original model, but were easily incorporated.

Jammer objects

Significant radio jamming threats exist in several of the world's areas of interest. Our model incorporates these threats through jammer objects. Jammers are located in the area of operations by the modeller as part of the database. During the model run, the jammers search out jamming targets by monitoring transmissions and using direction finding equipment. Once a jamming target is found, the jammer attacks the receivers of the signal. Depending on the anti-jamming equipment available, the jammer's power relative to the signal, and the ranges of all of the transmitters involved, the jamming may prevent the signal from reaching the receiver. This may simply cause units trying to send messages to the receiver on this net to invoke their alternative routeing methods, or the receiver may recognize the jamming as intentional and issue a jamming alert report. In this case, the report, which is also radio traffic, is sent to all net members, adding to the network workload. The net may then opt to change frequency to elude the jamming signal, at the cost of reconstituting the net.

Radio failures

Radios fail in our model. Outgoing traffic at a failed radio is re-routed through the routeing sequence described below, and incoming traffic cannot be received. The radio stays broken for some time specified by the radio type, and is then allowed to re-enter the net to which it belongs. At present, there is no consideration of sparing and logistical support in our model.

Network congestion and routeing

A message exchange begins its existence when its sending unit receives traffic that stimulates the requirement to communicate. This stimulation may come from the traffic generator or from another unit attempting to complete a BOST. The unit determines the receivers for the message, as well as the nets on which the message exchange will occur. The sender then waits until s/he can capture the net, and tries to send the message.

Sending a message requires that the system successfully moves through three phases. All of the intended receiver units must acknowledge contact with the sender. Once this acknowledgement takes place, the message transmission takes place. Each of the intended receivers then confirms the receipt of the message, and the sender releases the net to the next sender. In an uncongested, pristine environment where radios have infinite range and no failures, this is how the process works.

However, the environment in which the US Marine Corps communicates is not so friendly. The acknowledgement process and the message transmission are susceptible to interruption due to jamming, radio failures or range problems.

When faced with an intended receiver who does not acknowledge contact, a radio operator goes through an extensive alternative routeing process to reach the receiver or otherwise push his/her traffic through the communications system. To be realistic, the model must imitate this process. It does so through the methods of the net object. Changes or enhancements can be adapted universally through minor modifications to these methods. We will spare the reader the intricate details of this process as an in-depth explanation can be found in the model documentation, or requested from the authors.

The unit will transmit the message to any subset of the receivers which acknowledge contact, so that the information is not excessively delayed. Once the message is transmitted, the receivers confirm receipt. If no receipt notification reaches the sender, the message is retransmitted. If the receiver still does not confirm, the receiver is pursued through the alternative routeing sequence.

Runtime environment

Our model is embedded in a runtime environment which includes

- database creation, manipulation, and maintenance;
- model control;
- model execution;
- output analysis.

The database consists of specifications of the units, nets and BOSTs which are to be examined. The user has the opportunity to adopt baseline force configurations, and then to revise these configurations to suit the precise system under study. The user can save the constructed system under a user-specified name. All of the database functions are menu-driven and self-explanatory.

Model control is the stage where the user may exert control on the behaviour of some of the objects. There are several ON/OFF choices to be made, such as whether to model radio failures, whether jammers are present for the run, the method of jammer target selection, and several other features. The user also specifies the duration of the model run, and the initial value of the traffic workload rate λ.

Model execution is the phase where the model constructs sample paths of the specified communications network. Due to its object-oriented construction, all of the objects in the simulation are dynamically constructed using the data found in the database. That is, the program itself has no nets, radios or units. The simulation contains only specifications of the behaviour of these objects. Thus, the program 'instanciates' the communications network, the BOSTs, and the traffic generator with whatever size, scope and relationships are described in the database.

During the construction of a sample path, the traffic generator informs the user of the BOSTs being generated and the model displays a graphic showing the simulation clock, completion statistics for the BOSTs in the system, and a dynamically constructed picture of the penalty accumulation from time zero to the present.

The output analysis stage of the process allows the user to construct a statistical report of the set of sample paths generated during model execution. Output analysis is described in more detail below.

MEASURING EFFECTIVENESS OF THE COMMUNICATIONS NETWORK

As mentioned above, each BOST has a completion deadline. Each generated instance of a BOST has an object called a timer attached to trigger the penalty process if necessary. The timer is created at the time the BOST is generated. It waits a BOST-specified amount of time called the `AllotedTime` of the BOST. During this time, the pursuit of the BOST is considered penalty-free. However, after `AllotedTime` has elapsed, the timer tells a system-level object called the penalty accumulator to assess a BOST-specified `OneTime-Penalty`. From this point forward, lateness of the BOST costs an additional BOST-specified `PenaltyRate` per unit time. This rate is assessed until the BOST is completed successfully, or it expires due to excessive lateness.

The penalty accumulator collects all of the penalties accrued over time to construct a penalty process. The long-run mean rate of penalty accrual reflects the degree to which the network is functioning properly. Note that the penalty process focuses on mission-essential tasks. It measures failures to accomplish operational tasks in a timely manner, not necessarily late message exchanges. If a large amount of penalty is being accrued constantly, the BOST deadlines are consistently being violated, i.e. operational tasks are not being accomplished on time. The sources of large consistent penalty accrual must be investigated. Network designers can then determine if the specified deadlines are unrealistic, if certain nets or units are consistently resource constrained, or if BOSTs should be redesigned by increasing task concurrency or changing task structure so that deadlines can be met.

The penalty process $p(t)$ is the superposition of all `OneTimePenaltys` and `PenaltyRates` that have been assessed up to time t. It is a monotone non-decreasing function of time. We are interested in the steady-state rate of penalty accrual, and in promoting architectures and systems which reduce this rate. As opposed to other traditional measures of effectiveness and performance often used by the communications modelling community, such as average delivery time of individual messages or blockage of individual circuits, penalty rate analysis takes the following into account:

- the relative importance of each BOST;
- the various natures of the cost of lateness for each BOST;
- the timeliness of communications;
- the variability of the system's performance;
- the possible non-existence of steady-state.

Several generic capabilities for analysing the penalty process output are built into our model.

For terminating scenarios (e.g. if we use a BOST generation scenario which resulted from a more complex combat model or from field exercises) the penalty rate is

$$p' = \frac{p}{T}, \tag{1}$$

where p is the penalty resulting from a run of length T. Developing methods to treat output when the model is run with steady-state, rather than scripted, traffic, posed some interesting statistical problems. In this case, the rate of penalty accrual,

$$p'(t) = \frac{\mathrm{d}p(t)}{\mathrm{d}t}, \tag{2}$$

gives the degree to which the architecture at hand fails to meet the demands placed upon it at around time t. We call $p'(t)$ the penalty rate process, and our output analysis centres around measuring the expected long run average value of p'. This process is discussed in detail in the appendix, and applied in the next section, where we explore the application of our model to a radio allocation problem faced by the US Marine Corps.

AN APPLICATION: SINCGARS ALLOCATION

In 1991, the US Marine Corps began purchasing the SINCGARS series of tactical radios. Due to budget and production constraints, it seemed unlikely that they would be able to execute a complete one-for-one swap with the currently deployed PRC-77 family of radios. They needed a method of gradually deploying the new radios throughout the Marine Corps that would maximize warfighting capability.

In this section, we describe how we used our model to address the SINCGARS allocation problem. This task, for which our model was initially designed, involves determining the superior positioning of SINCGARS among the network of PRC-77s. As compared with the PRC-77, the SINCGARS

- is nearly jam-proof when frequency hopping;
- has superior lifetime characteristics;
- takes longer to repair;

- takes longer to constitute a radio net;
- takes longer to change nets;
- can only frequency hop when subscribing to SINCGARS-only nets.

Thus, the SINCGARS is more resilient and less flexible on the battlefield.

We compare three allocation options:

(1) allocate NO SINCGARS (PRC-77 only);
(2) allocate SINCGARS toward the forward edge of battle area (FEBA-back);
(3) allocate SINCGARS from the top of the command structure to the bottom (top-down).

There is a nearly innumerable number of allocation possibilities. These represent allocations which are supportable from the logistics, training, and command and control points of view.

While the steady-state BOST/message exchange structure allows us to do our analysis without a particular combat scenario, we do require geographic references and some sort of force disposition. For the purposes of exposition, we examine the portion of the ground combat element involved in fire support operational tasks. It is distributed geographically over the Mawmee Mines region of the Twenty-Nine Palms Training Area, see Figure 3. The Marines are advancing diagonally, from the lower-right to the upper-left, so the four communications jammers are closest to the front line units. One of the issues to be studied is whether it is better to protect the radio nets closest to these jammers with the new radios, or to protect the higher headquarters' nets that are farther from the jammers (see the COCs, ACE and BSSG in Figure 3), but carry more important messages. The nets closest to the jammers are more susceptible, but the higher headquarters' nets may incur higher penalty rates when jammed.

There are several types of BOST that generate very different fire support communications evolutions. A subset of the BOSTs used is shown in Table 1.

This list is quite revealing, as a BOST like the Check Fire BOST must be accomplished quickly, and involves broadcasting a single message to many units spread over a wide-area, while a Standard Fire Mission BOST is a precise evolution which involves exchanging a large number of messages between a small number of forward-deployed units, along with the (silent) consent of higher command units. Thus, the Standard Fire Mission has a longer allotted time as well as a longer perishing time. However, its structure is very complex and its completion requires co-operation on several radio nets.

Figure 4 shows a set of penalty trajectories for our three allocation schemes, as well as a reference trajectory showing execution when there is no jamming in the environment. Note the similarities of the shapes of these trajectories. This is caused by our use of common environmental sample paths for the three runs:

(1) all of the BOSTs are generated
    - at the same locations;
    - at the same times;
(2) all of the jammer actions are the same and happen at the same times;
(3) all of the net behaviour is the same;
(4) all of the radio positions common to all of the allocations have synchronized failures and repairs.

The power of using this common environmental sample path is seen in our ability to accentuate the differences between the allocations' performances.

Results

We employed the Standardized Time Series[6] method described in the appendix to determine a suitable cut-off point to eliminate initialization bias, and thus provide a common setting for comparison. This led to truncation of observations [0, 500] from each penalty rate process. Next, we compared each of the allocations with the no-jamming scenario by subtracting no-jamming data points from allocation data points with identical timestamps. Figure 5 shows notched boxplots (see Reference 7) of this population of differences. In this graphical representation, the boxes contain the data's interquartile range (IQR), while the notches show the region $M \pm 1.57 \times IQR/\sqrt{n}$, where M is the median and n is the number

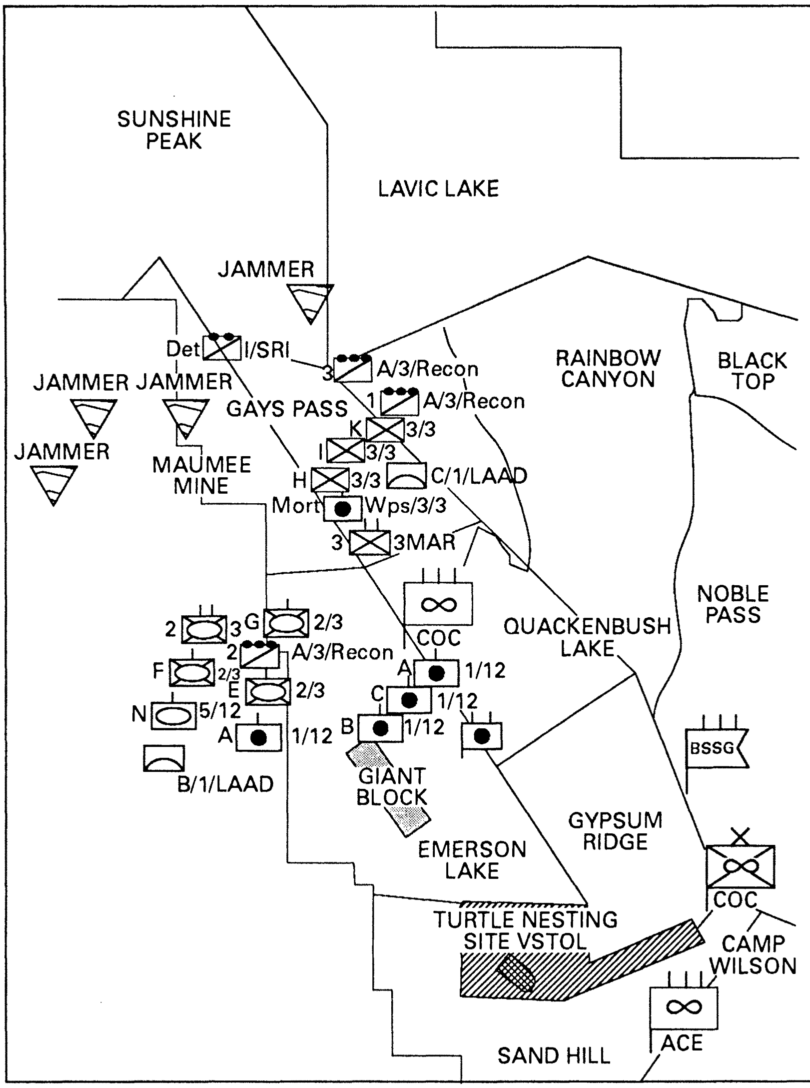

FIG. 3. *Map showing the units and jammer locations used in the sample scenario.*

TABLE 1. *Subset of BOSTs used*

Name	Allotted time	Perishing time	OneTimePenalty	PenaltyRate
Standard fire mission	65	200	100	3
Distribute GCE orders	18	80	10	2
Check fire	14	∞	125	10

of data points. The notches approximate non-parametric 95% confidence intervals for the true population median. Note that the upper notch of the FEBA-back population lies below the lower notches of the other allocations, telling us that we should reject any hypothesis of equal means and conclude that the FEBA-back allocation is superior. This conclusion was reached using an environment with a mean BOST intergeneration time of 15.0.

Among the many interesting data-journeys we have taken with the model, a very interesting one involved the sensitivity of this conclusion to the rate of BOST generation. (Recall that different levels of combat will be reflected in different BOST generation rates.) We ran studies where the mean intergeneration time varied in 10, 15, . . ., 35 time units. As we see in Figure 6, the allocations seem indistinguishable when the traffic rate is low; that is, when the mean intergeneration time between BOSTs lies above 30 minutes. In such a regime, there is

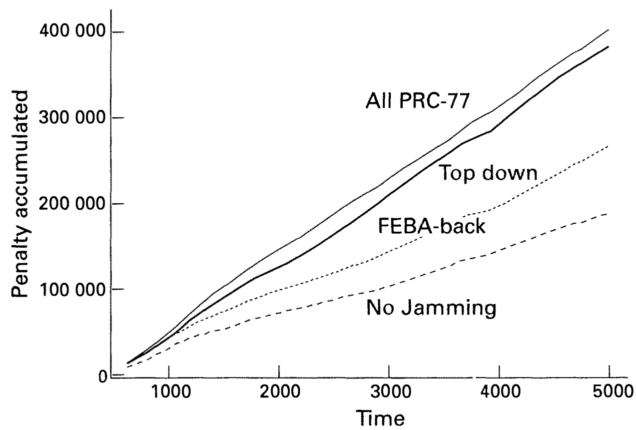

FIG. 4. *Raw penalty trajectories for the three allocations and the system when no jamming is present.*

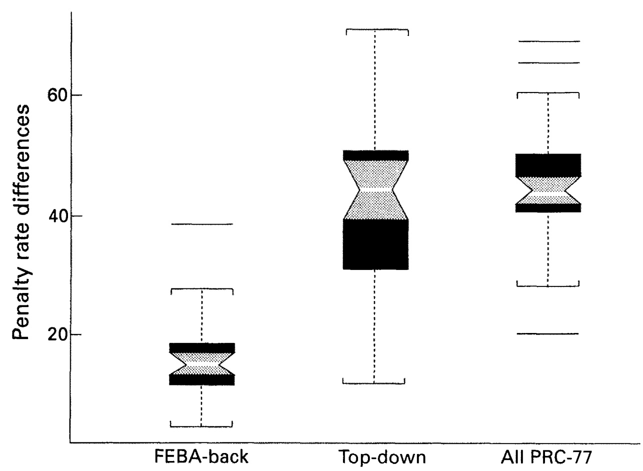

FIG. 5. *Notched boxplots showing the different allocations compared with the no-jamming scenario output.*

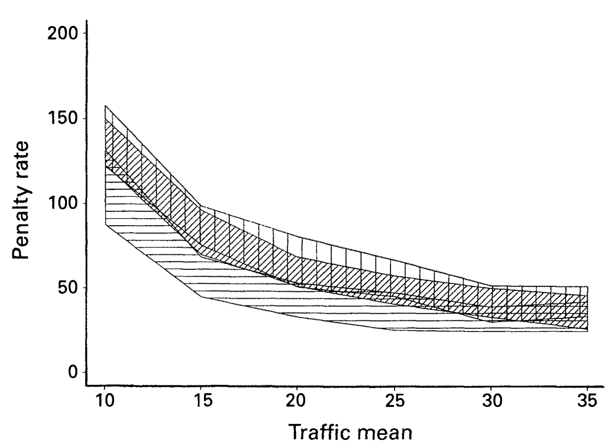

FIG. 6. *Confidence intervals for the mean penalty rate as the traffic rate varies. FEBA-back is shown with horizontal lines; Top-down in slanted lines; and All PRC-77 in vertical lines.*

enough communications equipment deployed for easy work-arounds to be arranged when jamming interferes. However, as the pace of traffic increases, the clear winner appears. The polygons in Figure 6 show the location and width of normal 95% confidence intervals for the mean penalty rate at each traffic level. We implemented the popular method of batch means sectioning[8] to remove autocorrelation between the estimates of $p'(t)$ for each traffic level before estimating the variance and forming the confidence intervals (see the appendix).

If we again compare by subtracting data points with identical timestamps, we see that the common environment allows us to make our conclusion stronger. Figure 7 makes this point very nicely; FEBA-back performs best, and does so in heavier traffic. As we would expect, the confidence intervals expand as we move from right-to-left: this is due to the system becoming less predictable as it becomes overburdened with traffic.

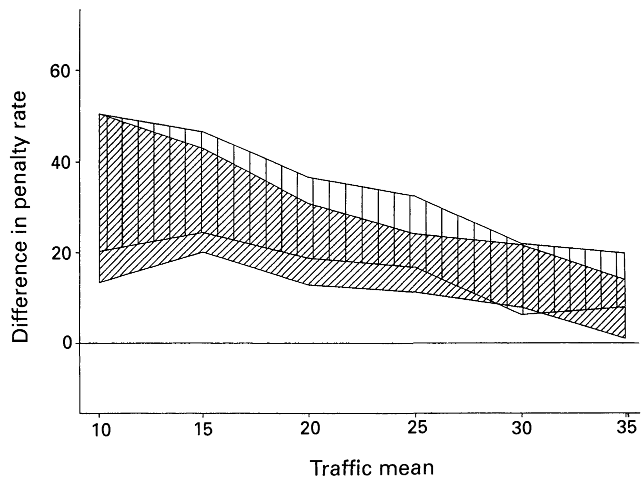

FIG. 7. *Confidence intervals for the difference between FEBA-back and the other allocations. PRC-77-only versus the FEBA-back allocation is shown in the polygon with vertical lines, while the Top-down to FEBA-back comparison is shown in slanted lines.*

CONCLUSIONS AND FUTURE RESEARCH

In this study, we have proposed a new paradigm for workload modelling in military communications systems which reflects the dynamics and dependencies of the actual system, while not requiring a complex, high resolution combat model. This workload model is facilitated by the BOST/message exchange structure described in Reference 5. The authors of that document unknowingly share in the credit for our model.

We presented an object-oriented model of the communications system which exploits the BOST/message exchange structure, measured the performance of the system through characteristics of a penalty accumulation process, and proposed methods for analysing the properties of this penalty process.

By using the model in concert with the database manipulation program, a communications analyst is afforded a rare opportunity to

- observe the effects of doctrinal modifications to routeing or net use;
- improve allocation of advanced technology radios throughout the US Marine Corps; and
- react to changing environments involving jamming and other threats within a pristine experimental environment.

In addition, with slight modifications to the model that are facilitated by its object-oriented nature, the analyst will be able to determine the overall capacity of an architecture to handle a mixture of data and voice traffic. We will modify the BOST definitions so that certain message exchanges act as digital transmissions (primarily adjusting the transmission time) rather than voice communications, and develop a digital radio net object that inherits all of our current net object's features except the alternative routeing scheme. The latter is necessary because digital nets use a simpler switching algorithm than human radio operators, are more persistent in communication attempts, and are less fault-tolerant. Thus, our model will be modified again, this time to perform all of the actions listed above for the emerging US Marine Corps voice–digital communications architecture.

APPENDIX: STATISTICAL TREATMENT OF THE PENALTY PROCESS

In this appendix, we describe the generic capabilities built into our model for analysing the penalty process output. The model can be run with scripted traffic from a wargame or combat exercise, or it can be run with steady-state traffic. While the treatment of output resulting from scripted traffic is simple to work with, the steady-state traffic poses some interesting statistical problems.

Terminating simulation analysis

If we use a BOST generation scenario which resulted from a more complex combat model, or from field exercises, we can develop a series of sample paths using the scenario as a script. If we repeat this script several times, we can construct a sample of n penalty rates p_i', $i = 1, 2, \ldots, n$ arising from

$$p_i' = \frac{p_i}{T}, \quad i = 1, 2, \ldots, n, \tag{3}$$

where p_i is the penalty resulting from the ith run of length T. The sources of variability in the p_i' are the jammer behaviours and the radio lifetime characteristics—the traffic load is deterministic. Univariate statistical methods can be used to analyse any set of these p_i'.

Steady-state traffic

In the steady-state traffic environment, BOSTs are generated in accordance with algorithm 1. If we generate a stationary sequence of BOSTs, the rate of penalty accrual,

$$p'(t) = \frac{\mathrm{d}p(t)}{\mathrm{d}t}, \tag{4}$$

gives the degree to which the architecture at hand fails to meet the demands placed upon it at around time t. We call $p'(t)$ the penalty rate process. Thus, our output analysis centres around measuring the expected long run average value of p',

$$p' = \lim_{t \to \infty} E[p'(t)]. \tag{5}$$

As with any examination of a steady-state quantity estimated by simulation, we must test for the existence of initial transients in the data, and remove them if they exist; and form confidence intervals for p' using (possibly) autocorrelated data.

Initial transients

When the system begins execution,

- none of the radios have messages waiting;
- all nets are idle and have no subscribers;
- all jammers are without targets; and
- no BOSTs are underway.

Thus, the rate of penalty accrual is zero at time $t = 0.0$. This state cannot be expected to last for too long unless the system is severely underutilized. Hence, samples of the penalty process taken near the beginning of the simulation are negatively biased, with expected values less than the long run average penalty rate. This phenomenon is known as initialization bias, and endures for the length of the initial transient.

Standardized Time Series (STS) has been successfully used to identify initial transients in steady-state behaviour in several documented instances[6]. Adapted for the penalty process, we have used this method to determine the duration of the initialization bias, τ. We can thus counter the effects of initialization bias by removing all samples from $[0, \tau]$ from the output, and use the remaining samples to construct confidence intervals on the value p'.

STS is a method of scaling and transforming a time series so that certain measurable characteristics have known distributions. In our application, we use the STS area estimator to estimate τ. Without reiterating much of the STS development, we know that we will generate a series of independent warm-ups of the simulation, each warm-up recorded on the interval $[0, T]$, where T is known to be larger than τ. Figure 8 shows ten warm-up trajectories on $[0, 700]$.

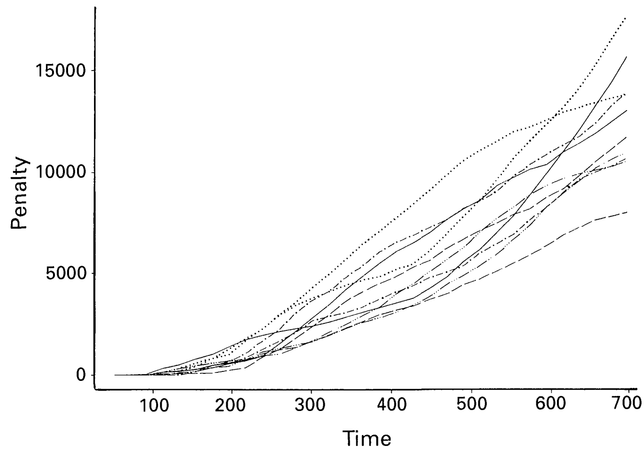

FIG. 8. *Plot of ten warm-up trajectories.*

Let t_i, $i = 1, 2, \ldots, m$ be an evenly-spaced set of points on $[0, T]$. Through a transformation, we can turn this set of samples into a series of functions f_{t_i} on $[t_i, T]$ which are Brownian bridges under the hypothesis that $\tau < t_i$, $i = 1, 2, \ldots, m$.

The STS area estimate $F_{v1,v2}(t_i)$ is roughly the area under the function f_{t_i} on $[t_i, T]$, and is distributed as an F random variable with v_1 and v_2 degrees of freedom if $t_i \geqslant \tau$. Figure 9 shows a plot of $F_{12,8}(t)$ produced by 20 independent warm-up runs. The model determined that the system was in steady-state at time ≈ 360.0 in this anaysis.

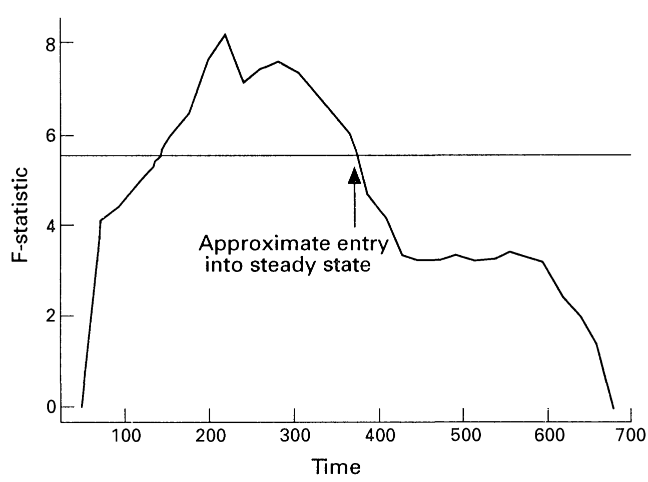

FIG. 9. *Plot of the STS area estimate used to determine the extent of the initial conditions with 20 independent warm-ups. The initial transient was determined to have vanished by time 360.0. The horizontal line shows the 95th percentile of $F_{12,8}$ as 5.67.*

Confidence intervals

Suppose that the simulation is now run for T time units. After truncating the penalty process to the interval $[\tau, T]$, we sample the penalty process at evenly spaced points t_1,

$t_2, \ldots, t_m$ producing the sample $p_1, p_2, \ldots, p_m$. The sample

$$p_i' = \frac{p_i + 1 - p_i}{t_i + 1 - t_i}, \quad i = 1, 2, \ldots, m - 1 \tag{6}$$

is produced by taking first differences of the original sample. It is obvious that $p_1, p_2, \ldots, p_n$ is a set of autocorrelated samples. In spite of this, $Ep_i = p'$, $i = 1, 2, \ldots, m - 1$. However, estimating the variance of p_i', and hence forming a confidence interval for p_i', requires some method of correlation removal. We implemented the popular method of batch means sectioning[8] to produce normal confidence intervals for p'.

REFERENCES

1. U. Madhow (1987) Bounds and asymptotic results for the performance of asynchronous frequency-hop packet radio networks. Defense Technical Information Center Technical Report DTIC AS-A182, University of Chicago.
2. A. L. Noel (1990) *Performance study of a marine expeditionary force radio system.* Master's Thesis, Naval Postgraduate School, Monterey, CA.
3. (1989) *Executive Summary: The Network Assessment Model (NAM).* Technical Report ND89-CBSF-0022, Ft. Gordon, GA.
4. R. R. Nair, M. A. Grimm and E. L. Althouse (1988) Discrete-event communication network simulator for the Naval Research Laboratory Test Bed System. NRL Memorandum Report 6360.
5. (1990) *Technical Interface Design Plan for Marine Tactical Systems (MTS TIDP), Volume II: Multiple Agency Message Exchange Sequences* [DRAFT].
6. L. Schruben, H. Singh and L. Tierney (1983) Optimal tests for the initialization bias in simulation output. *Opns Res.* **31**, 1167–1178.
7. J. M. Chambers, W. S. Cleveland, B. Kleiner and P. A. Tukey (1983) *Graphical Methods for Data Analysis.* Wadsworth and Brooks/Cole, Pacific Grove, CA.
8. P. D. Welsh (1983) The statistical analysis of simulation results, *The Computer Performance Modeling Handbook*, Academic Press, New York.