System Design and Assessment Notes

Note 19

June 1974

NETWORK ANALYSIS AND THE RELIABILITY ASSESSMENT OF SYSTEMS

by

WILLIAM P. DOTSON, JR.
Captain, USAF

Air Force Weapons Laboratory
Kirtland Air Force Base, NM 87117

ABSTRACT

A self-contained treatment for the reliability assessment of systems is presented. Testing and/or analysis is used to derive estimates of subsystem reliability. Network analysis is used to determine the system reliability from the subsystem reliability. Examples of a number of applications are presented for the ARPA computer network. The problem of optimum allocation of a fixed budget in system assessment/hardening/design problems is addressed, and approaches are proposed. These methods can be used to reduce the high cost associated with system level testing. Some problems that could lead to theoretical extensions are presented, and approaches are proposed.

While the technology discussed herein is slanted toward communications systems in application, it should be noted that it is in no way restricted to systems of that type. It may be applied equally well to the reliability assessment of any system which can be divided into statistically independent subsystems and described in terms which relate the system function(s) to the ability of the subsystems to perform their respective functions.

PREFACE

This report is the result of a part-time effort by the author to understand the technical issues pertinent to the reliability assessment of systems. The author has tried to write this report to reflect the evolution of his understanding. This evolution is reflected in the body of the note, the appendix, and in SDAN 16, 11 Jan 74, which are short reports on various aspects of the overall problem, written over a 2-year period. Consequently, the reader will find some lack of commonality in notation and language from the appendix and SDAN 16 to the body of this note. The reader is, therefore, urged to read the appendix and SDAN 16 as they are first referenced in the body of the note; and read for the concepts. The body of the note draws on the concepts contained in the appendix and SDAN 16 and attempts to summarize and extend them in consistent language.

The author is deeply indebted to a number of people who have directly or indirectly contributed to his understanding of the assessment problem. Major Austin Lyons deserves special consideration, not only for the free exchange and critical examination of ideas during our association, but also for contributions to this report. He introduced the author to Veitch diagrams and thereby deserves full credit for the germ of the idea in SDAN 16.

CONTENTS

ILLUSTRATIONS

ILLUSTRATIONS (Cont'd)

TABLES

SECTION I

INTRODUCTION

This report considers the problem of assessing the reliability of systems. By reliability is meant a measure of the connectivity probability in a hostile environment. An alternative meaning for reliability is the fraction of time the system is available for use. Still another meaning is a lower bounding estimate of the connectivity probability; or the fractional availability. The context of the problem being considered will always determine the appropriate meaning of the term "reliability." By system is meant a set of connected subsystems which function together to meet a system objective. For example, a communication network may be regarded as a system. Here the system objective is to permit the users to pass information to each other through a set of interconnected communications subsystems (equipment, facilities). Another example is a missile guidance system. The set of interconnected subsystems in this case would be the rocket thrusters, gimbals, sensors, inertial platform, computer, power, etc.

The assessment problem herein assumes that the system cannot be tested directly. Perhaps it is physically too large, as a continental or global communications system would be. It may be that the hostile environment to which the system must be assessed cannot be produced accurately without engaging in a general nuclear war, or economic considerations may indicate that direct testing is too expensive to warrant the expected return.

It is further assumed that a deterministic analysis of the system reliability is not possible. The complexity of the system and the state of the art in assessment are compelling reasons supporting this. Again, economic considerations may be overriding.

One must work then from a description of the system in terms of its interconnected subsystems. Section II gives an overview of the approach to be used in going from subsystem to system assessments. Subsections of section II, and the indicated references, go into considerable detail on the subject of subsystem assessment. Section III addresses the problem of aggregating results from subsystem to system level. A number of techniques are discussed, and the most efficient known to date is presented. Section IV is perhaps the most illuminating. It shows some of the more powerful applications of this technology by way of examples. The most important use of the technology is believed to be in the optimum allocation of fixed budgets in system assessment/hardening/design problems. Section V outlines some theoretical extensions and their potential applications.

While the technology discussed herein is slanted toward communication systems in application it should be noted that it is in no way restricted to systems of that type. It may be applied equally well to the reliability assessment of many systems. The system must only be divided into statistically independent subsystems; then described in terms which relate the system function(s) to the ability of the subsystems to perform their respective functions. It should also be noted that the technology can be extended in a straightforward manner. Extended applications would be the reliability assessment of a system operating in either a benign or hostile environment with global or local threats which can change with time.

The initial problem statement which motivated the research reported on here was to investigate the nuclear survivability/vulnerability of USAF Strategic Command, Control and Communications (C^3) systems, with special emphasis on electromagnetic pulse (EMP) effects. The problem as stated is extraordinarily broad in scope. Consequently a considerable amount of time was spent in an effort to better understand and define the technical issues pertinent to the problem. For an overview of this area the reader is referred to appendix A.

SECTION II

APPROACH

To summarize appendix A, one basic approach to assessing the reliability of a system is:

(1) Find the network (nodes and links) representation of the system. ("Network" and "System" as well as "Node" and "Subsystem" now become synonymous).

(2) Translate the system function into a connectivity requirement between one or more node-pairs in the network.

(3) Perform a network analysis to determine the probability (P) of satisfying the connectivity requirement in terms of the individual nodes probabilities (p_i) of survival.

The above leaves open the major technical area of determination of each node's survival probability. Figure 1 summarizes the technology areas, their interrelationship, and supporting tasks for each. It should be noted that each technology area is an extensive subject of itself, and a proper treatment is best done by a specialist in the area. Consequently, this report gives only a cursory treatment of the areas and their necessary integration in this section. Succeeding sections in this paper will be devoted to a more extensive treatment of the network analysis technology area.

2.1 NETWORK ASSESSMENT MODELS

Past efforts at subsystem assessment have typically resulted in qualitative judgments such as "hard" or "soft." In trying to assess a network of connected subsystems there is no way to aggregate hard/soft judgments at the subsystem level to any equivalent judgments at the system level. It appears then, that the final result of a network assessment must be quantitative in nature and address itself to questions asked by decision makers and high level management. Ideally the result would be a simple yes or no to the question of whether or not the network will do its job properly in the intended environment. Unfortunately such questions can seldom be answered in a dichotomous way. There may be several reasons for this inability. It could be, for example, that the parameters describing

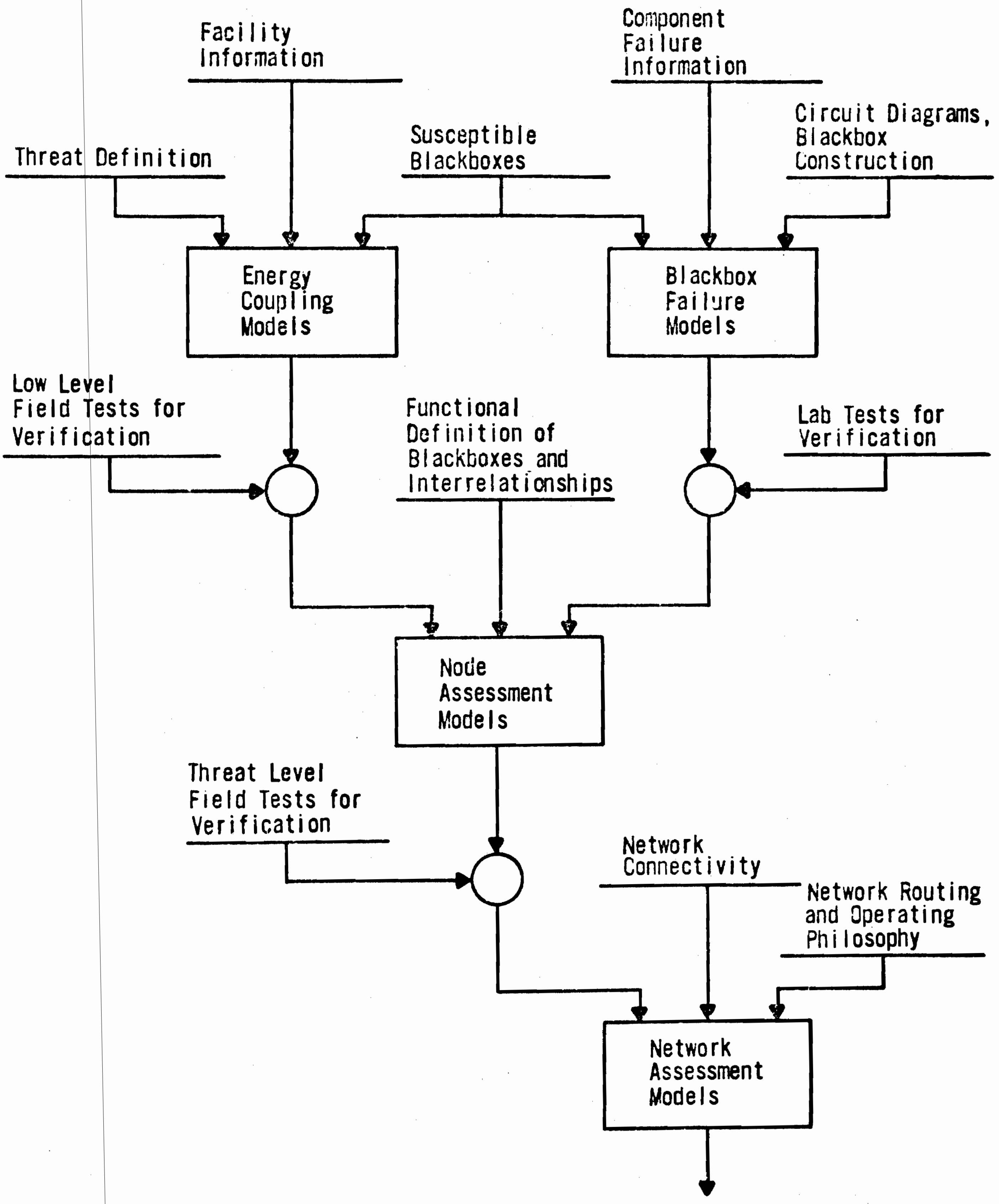

Facility Information
Component Failure Information
Threat Definition
Susceptible Blackboxes
Circuit Diagrams, Blackbox Construction
Energy Coupling Models
Blackbox Failure Models
Low Level Field Tests for Verification
Functional Definition of Blackboxes and Interrelationships
Lab Tests for Verification
Node Assessment Models
Threat Level Field Tests for Verification
Network Connectivity
Network Routing and Operating Philosophy
Network Assessment Models

Figure 1. Task Flow for C^3 Assessment.

the threat (environment) to the system are not known with precision. Instead the threat parameter(s) may be associated with some probability distribution. Another example involves the determination of the system or subsystems response (function/fail) to some given value of threat. Certainly a specific subsystem may be subjected to a specific threat and a yes or no decision can be made as to its ability to function after the fact. But what about the other subsystems (not to mention other threat values which can occur)? Here again the notion of probabilities enters the assessment problem in a natural way. While other examples could be cited, the point is made that a quantitative system assessment (with less than exhaustive knowledge of every aspect of the problem) can only be performed by appealing to probabilistic statements. These statements concern those aspects of the total problem which are not known in a deterministic way. The result of the assessment is likewise a probabilistic statement.

If it is assumed that the probability of survival of each node in the network is known, then network simulation techniques, such as were used in obtaining the results of appendix A, clearly apply in a straightforward way. Unfortunately, it turns out that the probability of survival of each node can only be estimated. In fact this estimate, for a single node, may range from 0.0 to 1.0, and different confidences may exist in the estimate depending on the specific value chosen. An illustration of the type of data that may be expected on the nodes for a quantitative assessment of a network is shown in figure 2. The interpretation of the graphical presentation is

p_i: The true, but unknown, value for the probability of survival of the i^{th} node.

$\hat{p}_i$: An estimate of p_i. This shall also be referred to as the reliability of the i^{th} node.

$C\ (p_i \geq \hat{p}_i)$: Confidence or "fiducial probability" that $\hat{p}_i \leq p_i \leq 1.0$

Data as shown in figure 2, for each node in the network, can be manipulated so that it is possible to define a confidence distribution for the estimate of the network connectivity probability (or terminal reliability), $\hat{P}$.

It has been shown that data in the format of figure 2 are all that can be reasonably expected from node assessment. Consequently, to develop an integrated system assessment technology, two major problems must be solved.

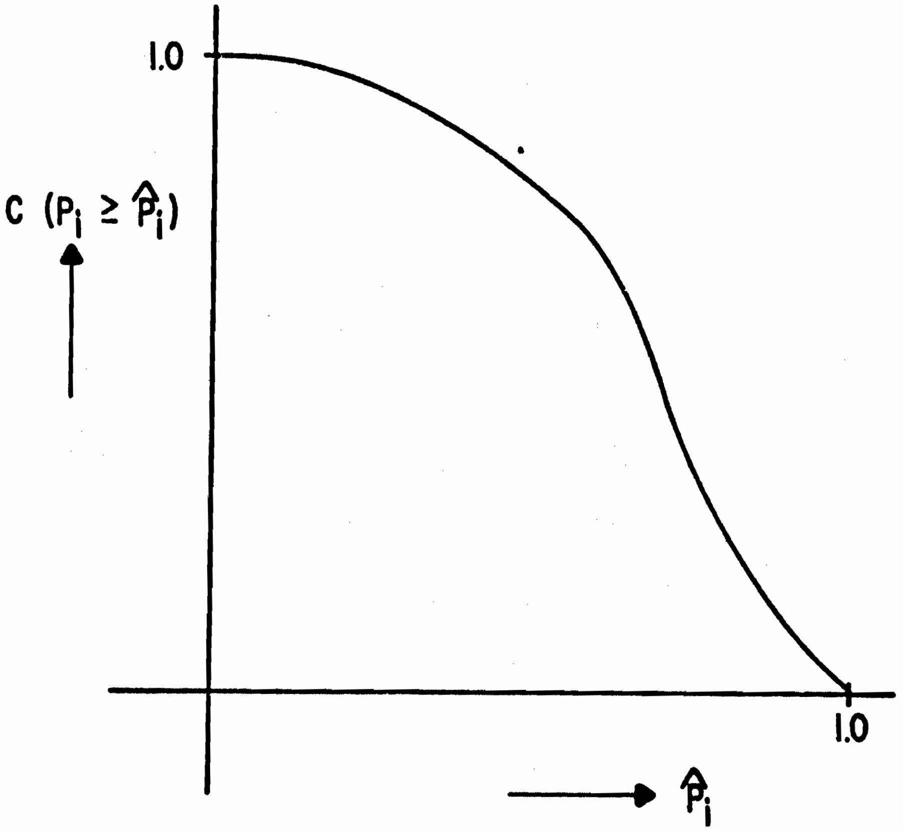


Figure 2. Confidence - Reliability Function for the ith Node.

(1) Network analysis algorithms capable of handling node data in the format of figure 2 must be developed.

(2) Procedures for obtaining the node data must be developed. Problem 1 will be treated in detail in section III. The second problem is now considered.

2.2 NODE ASSESSMENT MODELS

There are three basic approaches for obtaining node assessment data in the format of figure 2.

(1) Direct testing of the nodes.

(2) Analysis of the threat and susceptibility parameters pertinent to node survival.

(3) A mixture of testing and analysis.

Subsection 2.2.1 discusses the direct testing approach briefly and indicates a reference for additional reading. Subsection 2.2.2 discusses an analytical approach. Subsection 2.2.3 discusses a hybrid approach.

2.2.1 Direct Testing

Here the point of view is taken that the node may be subjected to as many tests as desired (conceptually infinite). It is assumed that the tests are truly representative of the environment in which the node must function. Given that this is true, the confidence reliability function for the node is (ref. 1):

$$C\ (p_i \geq \hat{p}_i) = \frac{\int_{\hat{p}_i}^{1} X^{S_i}\ (1-X)^{T_i\ -\ S_i}\, dX}{\int_{0}^{1} X^{S_i}\ (1-X)^{T_i\ -\ S_i}\, dX} \tag{1}$$

where the symbols not previously defined are:

T_i: Number of tests performed on the i^{th} node.

S_i: Number of tests which the node passed.

It is assumed that a dichotomous judgment (passed/failed) can be made after each test. It is also assumed that a good a priori distribution for the estimate is the uniform distribution in the interval [0,1].

2.2.2 Analytical Approach

Here the view is taken that the node may be represented as a collection of interconnected black boxes which work together to perform the higher order function of the node. For example, a node might be a radio relay which is made up of a receiver, frequency translator, transmitter, and a power supply. This subsystem may be represented as the series network of figure 3. The four nodes represent the four black boxes of the subsystem, and the total representation is series since all four black boxes must work in order for the sybsystem to function.

It might be suggested that each black box be independently tested as in subsection 2.2.1; then the results on each box could be aggregated to determine the estimate for the total subsystems probability of survival. This however is only half right. A real difficulty with this approach is the statistical dependence from one box to another. This dependence has its source in the calculation of the threat as seen by each box. To make this point clear, consider the following problem.

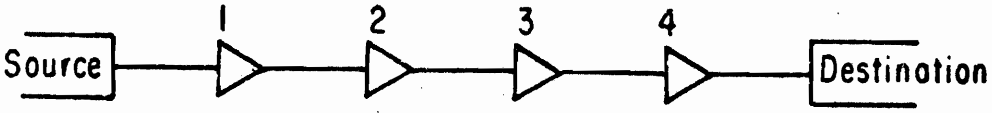


Figure 3. Network Representation of a One Way Radio Relay.

An ac-dc converter is to be assessed. The external threat is an EMP.

A semi-infinite ac power line is connected to the ac-dc converter. Strictly speaking, the free field EMP threat is a function of bomb design, burst altitude, and look angle from subsystem to burst. None of these parameters can be known a priori in a deterministic way and are therefore legitimately random variables. It is of course possible to decree a free field threat and direction of incidence in which case a Thevenin equivalent circuit for the pickup on the ac power line can be found (assuming linearity). The Thevenin equivalent will take the form shown in figure 4.

Both E_{th} and Z_{th} will be complicated functions of the physical structure of the power line and its surroundings, such as height above earth and the dielectric properties of the earth itself. If some or all of the descriptive parameters must be regarded as random variables, then E_{th} and Z_{th} are likewise random variables. They would also be dependent if the same parameter(s) appear in both functions. A good example of a parameter which should be regarded as a random variable in such a problem is ground conductivity. Certainly it can be measured precisely at some instant in time but rain (or a drought) will change it, and there is no way to determine that it will be when the threat environment occurs.

Consider now the ac-dc converter. A simple model for the purpose of determining its failure rate can be drawn as in figure 5. Here Z_L represents the driving point impedance of the ac-dc converter. For a nonlinear device (such as this ac-dc converter) the driving point impedance is defined in such a way as to allow it to be a (nonconstant) function of the level at which it is being driven. This representation can be determined from the design details of the black box. To do so however, the problem would normally be made piecewise linear by assuming that the sensitive component(s) in the box are near breakdown. It is well known that several components of the same type will exhibit breakdown at different points. The breakdown point for a given device type must therefore be treated as a random variable, and as a consequence, so must the associated failure current I_L and driving point impedance Z_L. A typical presentation of a failure model for a component is shown in figure 6 (ref. 2).

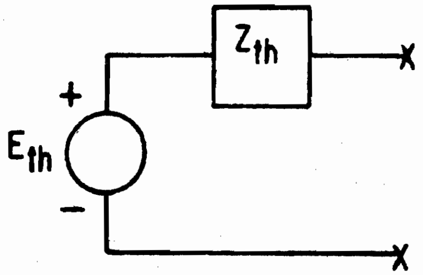


Figure 4. Thevenin Equivalent Circuit.

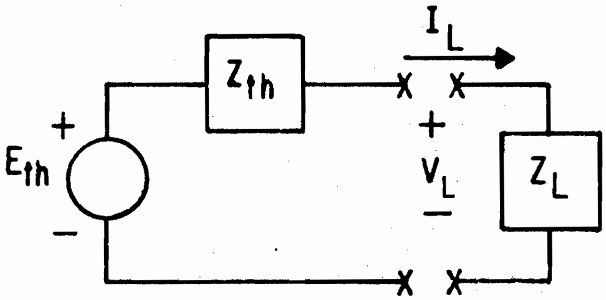


Figure 5. Model for Determining Failure Rate.

The reader has certainly observed that the problem as presented is oversimplified. The reason for doing so is simply to make the point that a statistical treatment of this type of problem is difficult at best and requires careful consideration of the parametric dependence involved. To keep this discussion simple the deposition time dependence indicated in figure 6 will be ignored.

Continuing the example, suppose that the failure model for the box as a function of (V_L, I_L) is found from the component(s) failure model(s) and the box circuitry. Furthermore the driving point impedance, Z_L as a function of I_L, of the box near breakdown is known from the preceding.. What is the failure probability of the ac-dc converter?

It is assumed that the failure data for the box is available in the formats of figures 7 and 8. It is also assumed that E_{th}, Z_{th} are known functions of, for example, ground conductivity σ. That is,

$$E_{th} = f(\sigma) \qquad (2)$$

$$Z_{th} = g(\sigma) \qquad (3)$$

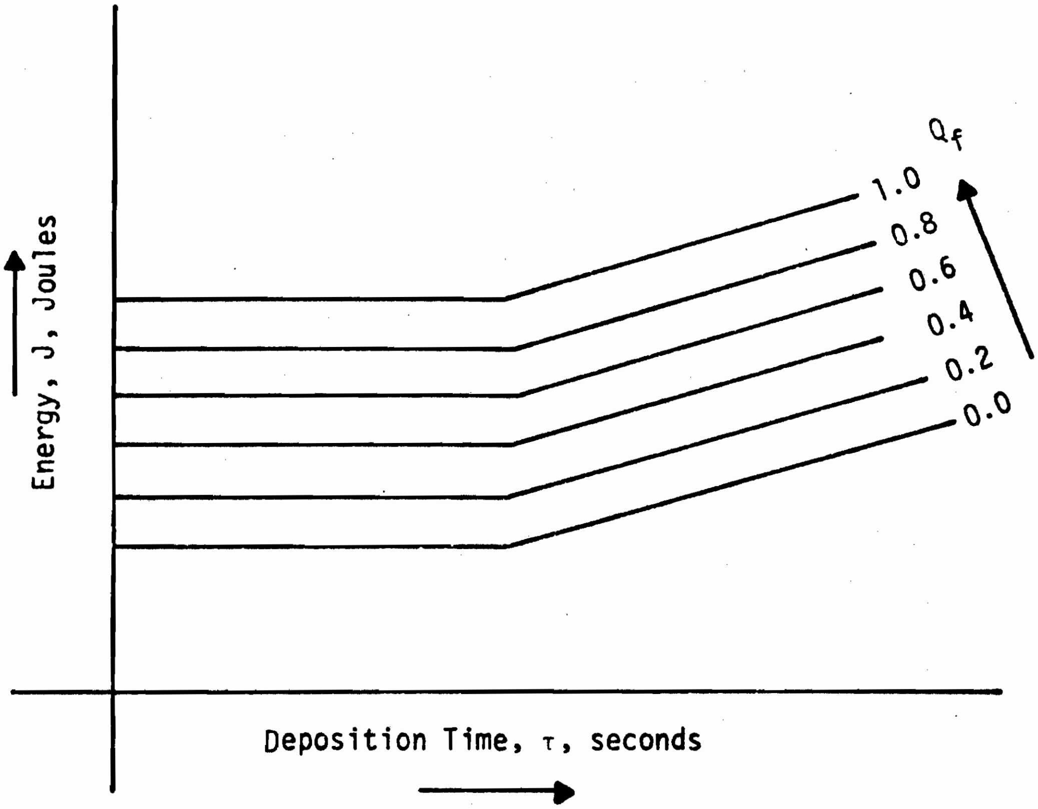


Figure 6. Probability of Function Failure, Q_f, Versus Energy and Deposition Time.

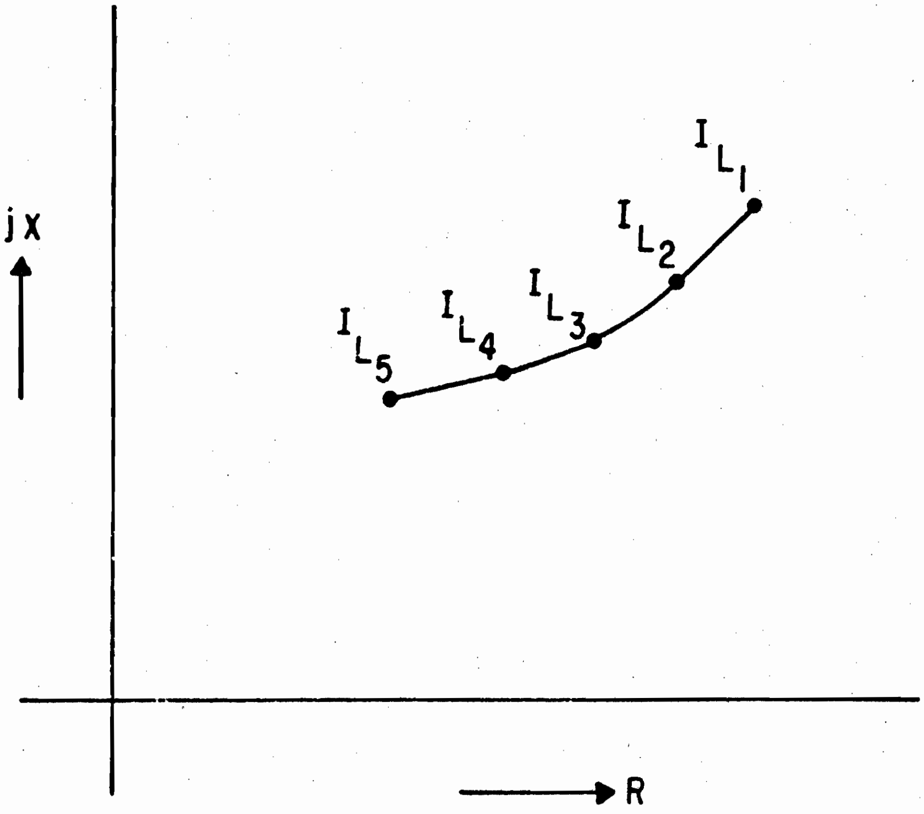


Figure 7. Z_L Versus I_L Near Breakdown (single frequency).

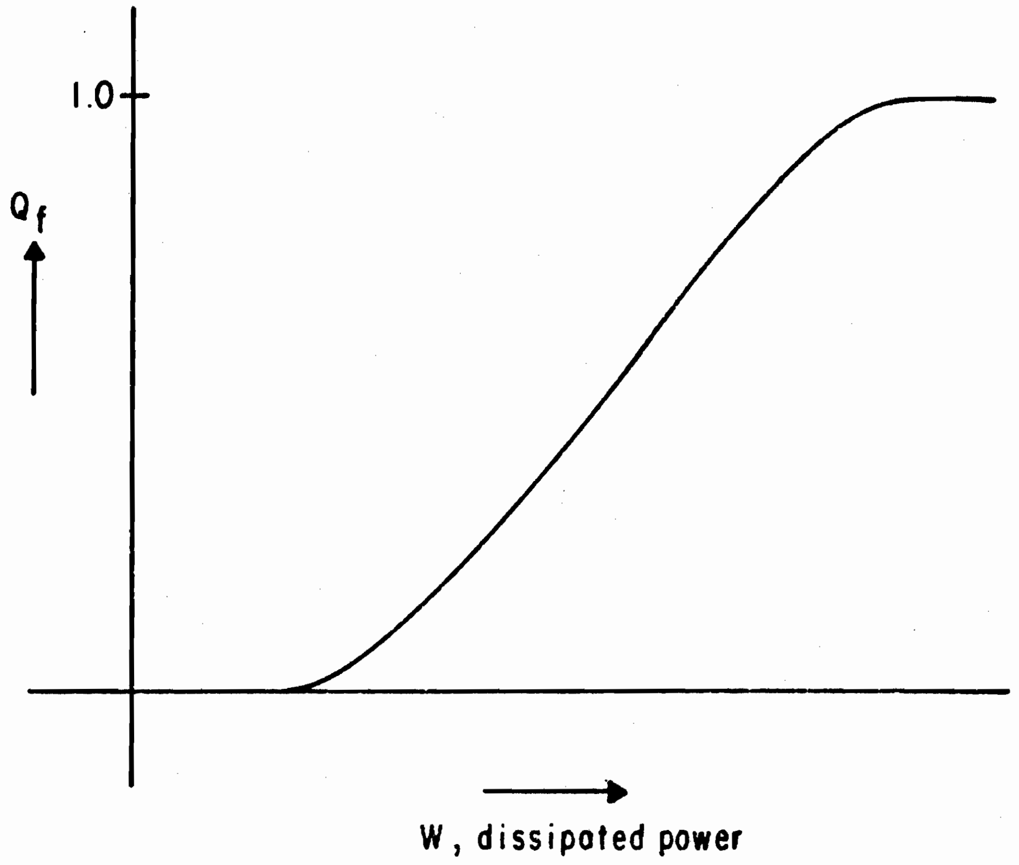


Figure 8. Probability of Failure, Q_f, Versus W.

and the distribution of σ is assumed to be discrete (an engineering approximation) and is given as in figure 9.

The following algorithm will solve for the failure rate of the box in this simplified case:

```
(1) $i \leftarrow 0$
    $Q_f \leftarrow 0.0$
(2) $i \leftarrow i + 1$
(3) $\sigma \leftarrow \sigma_i$
    $X_i \leftarrow p\ (\sigma = \sigma_i)$        (from figure 9)
    $E_{th_i} \leftarrow f\ (\sigma_i)$        (from equation 2)
    $Z_{th_i} \leftarrow g\ (\sigma_i)$        (from equation 3)
(4) Solve the circuit of figure 5 for $I_{L_i}$ subject to the driving point impedance constraint of figure 7.
    $I_L \leftarrow I_{L_i}$
    $Z_L \leftarrow Z_{L_i}$
    $W_i \leftarrow I_L^2\ \text{Re}\left[Z_L\right]$
(5) $Y_i \leftarrow Q_{f_i}$        (from figure 8, using $W_i$)
(6) $Q_f \leftarrow Q_f + Y_i\ X_i$
(7) If $\sigma < \sigma_{max}$go to (2)
(8) STOP
```

It should be noted that Z_L in figure 5 is a random variable which depends on the random variables E_{th}, Z_{th} according to the laws of circuit theory and the functional constraint as presented in figure 7. In turn, the failure probability is dependent on Z_L.

The reader can easily see that the problem is considerably complicated if the model for determining failure rate is as in figure 10. The complicating factor here, beyond the obvious additional computational problem, is that when a box fails, its driving point impedance will change radically. This, of course, introduces box-to-box dependencies in the failure rate model.

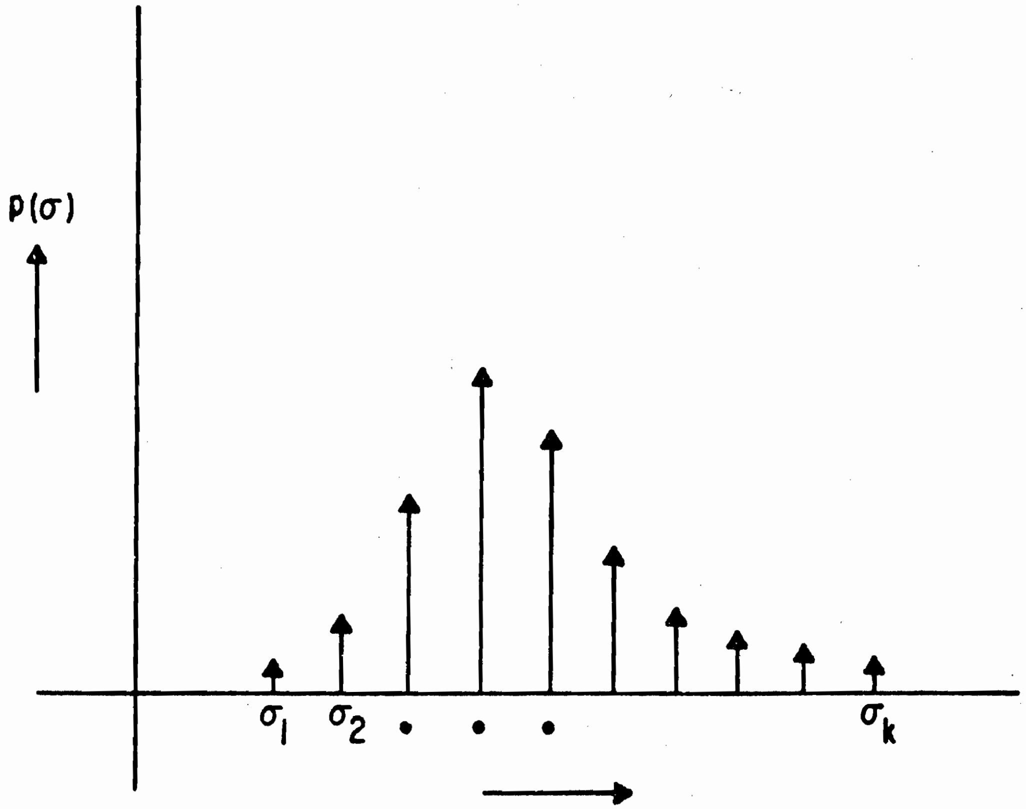


Figure 9. Probability Density of σ.

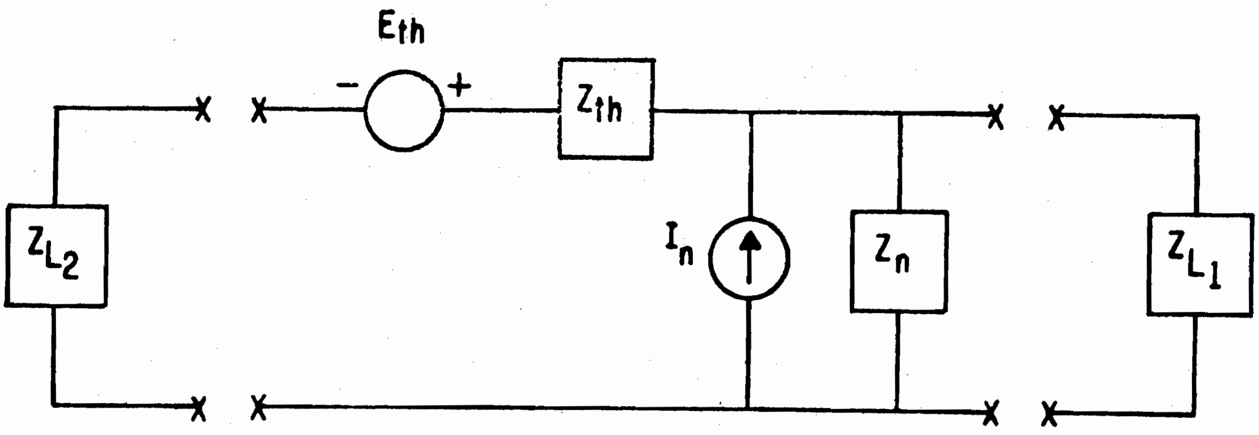


Figure 10. Model for Determining the Failure Rate of Two Electrically Connected Black Boxes.

As mentioned previously the problem has been oversimplified for the sake of discussion. Some of the items that require further attention are:

(1) How to obtain the distributions of the parameter(s) governing E_{th}, Z_{th}, and Q_f as a function of W. In fact only estimates of the statistical moments of the parameters can be obtained, and the moment estimates have confidences less than 1.0 associated with them according to the number of samples taken and the assumed form of their distribution. This adds a dimension to the overall problem.

(2) The algorithm above needs to be improved by adding the dimension of confidence. Explicit consideration needs to be given to the deposition time of figure 6.

(3) For systems of continental size, the algorithm should accommodate the free field EMP threat as a variable.

(4) Efficient techniques for finding coupling models (as in figures 4 and 10) from much more complicated lumped constant models should be developed. Ideally the techniques should allow variations in the free-field pulse shape, amplitude, direction of incidence, and polarization.

Some work in this direction is being done and can be found in reference 3.

Development of analytical techniques, as discussed previously, could be quite expensive. However, the potential expense should be carefully weighed against the alternative expense of overdesigning future systems or retrofitting existing ones in a knee-jerk response to a worst-case systems analysis. As the reader probably knows, a worst-case analysis will result in a failure threshold calculation which may be as much as two orders of magnitude less than the actual threshold and a threat calculation which may be as much as an order of magnitude higher than the actual. The combined effect of these two sources of pessimism (in addition to the initial worst casing of the threat pulse and direction of incidence) can easily lead the uninitiated to consider a system unreliable when, in fact, the reliability may be quite acceptable.

2.2.3 Some Concluding Remarks on Node Assessment, a Hybrid Approach

It was shown in the preceding subsections that the ideal result of an assessment would be a simple yes or no answer to the question: Will the node do its job in the intended environment? It was further shown that the next best objective (since the first cannot be obtained) would be to obtain the nodes probability of survival, p_i. Finally, it was suggested that the only information about p_i which could be practically obtained is an estimate of p_i, $\hat{p}_i$. This estimate ranges from 0.0 to 1.0 and takes the general from

$$C\ (p_i \geq \hat{p}_i) = f\ (\hat{p}_i, \ldots). \tag{4}$$

Two techniques for obtaining the indicated estimator were discussed briefly. These were direct testing and analysis. Hybrid techniques are clearly possible. Whatever assessment technique is used, considerable expense can be anticipated. For example, the budget estimate for assessing a single radio relay junction station in the AT&T long lines system by testing to the anticipated environment, is approximately 500,000 dollars. On the other hand, a pilot study of an analysis technique for assessment of a microwave repeater station (ref. 3) has cost the Air Force 100,000 dollars with a predicted production run estimate of 10,000 dollars per assessment. It seems reasonable to expect that hybrid techniques for assessing the entire set of nodes in a network would bring the average cost per node below any of the above estimates. Such hybrid techniques would involve:

(1) Simple worst-case analyses for culling out those nodes which can be shown to have high ($\approx$1.0) reliability even under a worst case set of assumptions.

(2) Design and performance of some reasonably straightforward direct testing (still using worst-case assumptions) on nodes to establish estimates on those which are borderline in (1).

(3) Design and application of hybrid test/analysis techniques to the nodes remaining after processes (1) and (2) have been applied. At this point worst-case assumptions should be discarded and replaced with the best estimates of the governing parameters that can be economically obtained.

Some work in this direction is being done and can be found in reference 4.

Finally, it is noted that the cost of node assessment must be weighed against the impact it has on network assessment. That is, how is the expected value of the commodity protected (or alerted) by establishing communications connectivity changed in response to the result (and expense) of node assessment? These points will be addressed after developing the necessary tools for network analysis in the next section.

SECTION III

NETWORK ANALYSIS

This section treats the problem of determining the connectivity probability, P, between two distinguished nodes in a network. The distinguished nodes will be called source and destination nodes, and it will be assumed that their survival probability is 1.0. The remaining nodes in the network are assumed to have data regarding their respective estimated probabilities of survival, p_i, in the format of figure 2. Such nodes are called weighted nodes. It is further assumed that the nodes are statistically independent. Specifically, it is assumed that any dependence in the network has already been considered via testing and/or analysis at the appropriate level of complexity (subsections 2.2.1, 2.2.2, and 2.2.3) and that these processes have allowed node definitions so that the assumption of node independence is valid.

It should be noted that data on the nodes, in the format of figure 2, is clearly a function of the level of threat under which the node was assessed. Consequently this data could be found as a function of the external (to the node) threat parameters. These parameters, in turn, can easily be made dependent on time (scenario) and space (node and burst coordinates). An efficient network analysis should be able to use all this information so that the inevitable question of, what if... this scenario?, can be answered quickly and efficiently. This can be accomplished only if the network analysis results in an explicit function, i.e.,

$$P' = g\ (p_1,\ p_2,\ \ldots,\ p_n) \tag{5}$$

relating the network connectivity probability to the survival probability of each node in the network. The function can then be evaluated at different times as the estimates for the nodes evolve in time.

Network, as used here, will mean a set of nodes and a set of links connected together in a well defined way. The network function is to establish connectivity between the network users (source and destination nodes). An example of a network is shown in figure 11. This example involves all symbols necessary to define an arbitrarily complex network. The symbols are defined in table 1.

3.1 NETWORK SIMULATION

Simulation is a technique which can be readily applied to almost any problem. Rather than depending on a formal analytical approach, simulation requires only a physical understanding of how the system works, knowledge of a programming language, and the patience to faithfully translate physical understanding into code. Consider the example of figure 11.

Suppose that the probabilities of survival of each weighted node are given in the set $\{p_1, p_2, p_3, p_4, p_5\}$. The task is to determine the probability of connectivity, P, from node 6 to node 7; i.e., P_{6-7}. Although it is an easy task to find P_{6-7} analytically for this problem it is apparent from the example that more complicated networks will not be so tractable. In this case simulation is frequently resorted to. To solve the network by simulation the following two "sub-algorithms" are required.

(1) RANDU: an algorithm which generates a random number uniformly distributed in the interval [0,1].

(2) CONNECT: an algorithm which examines a given network and makes a determination of the connectivity [1] or lack of it [0] between source and destination nodes.

A simulation algorithm to solve the network of figure 11 is;

```
(1)  NODES = number of weighted nodes in the network
(2)  LOOP = number of simulations desired
(3)  $P_{6-7} \leftarrow 0.0$
(4)  $j \leftarrow 0$
(5)  $j \leftarrow j + 1$
(6)  $i \leftarrow 0$
(7)  $i \leftarrow i + 1$
(8)  $Y = \text{RANDU}$
(9)  If $Y > p_i$ remove the $i^{th}$ node from the network
```

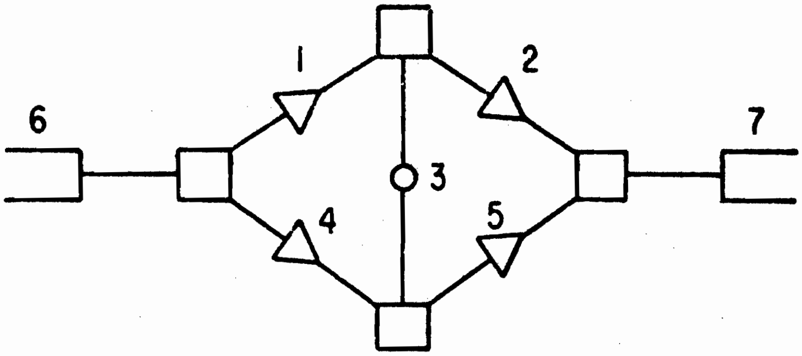


Figure 11. An Example of a Network

Table I

SYMBOLS

⊐ : Source or Destination node, survival probability = 1.0

□ : Node with survival probability = 1.0

○ : Undirected node, survival probability = p_i

▷ : Directed node, survival probability = p_i

—— : Undirected link, survival probability = 1.0

NOTE: A link with survival probability $\neq$ 1.0 can be represented as:

—○— : undirected

—▷— : directed

```
(10)  If $i < NODES$ go to (7)
      (At this point one possible realization of the network, e.g., figure
      12a, has been created)
(11)  $X = CONNECT$
      (The realization, e.g., figure 12a or 12b is examined for connectedness or disconnectedness; $X = 0$ or $1$)
(12)  $P_{6-7} \leftarrow P_{6-7} + X$
(13)  If $j < LOOP$ go to (v)
(14)  $P_{6-7} \leftarrow P_{6-7}/LOOP$
(15)  STOP
```

As the reader can see, simulation is a powerful tool which can be readily applied to network reliability problems even if the investigator is unable to begin a formal analytic approach. The method does have severe limitations. For example, exercising the preceding algorithm on some given set of node probabilities yields no insight at all into the solution under a different set of node probabilities. Or suppose that one wishes to examine the effect on network reliability of permanently removing one node (for example, node 3 in figure 11). The entire problem must be re-simulated. These situations can be handled by simulation but this approach can be quite inefficient and time consuming. Consequently, more efficient techniques are in order.

3.2 ELEMENTARY NETWORK ANALYSIS

The objective of this subsection is to develop some techniques which will, with a single simulation, yield some insight into how the terminal reliability changes as a function of the node probabilities. The technique to be discussed is prompted by the observation in the algorithm of subsection 3.1 that network realizations can be examined for connectivity.

3.2.1 Series/Parallel Network Analysis

Network analysis of series/parallel combinations of nodes is straightforward but can become tedious when large numbers of nodes are involved. Here, two analysis examples involving only series/parallel networks are considered.

Consider the series network shown in figure 13. The series configuration is indicated by the observation that only a single path exists between source (K+1) and destination (K+2). Hence there is only one way that connectivity between these nodes can be maintained. That is, all K weighted nodes must be functional.

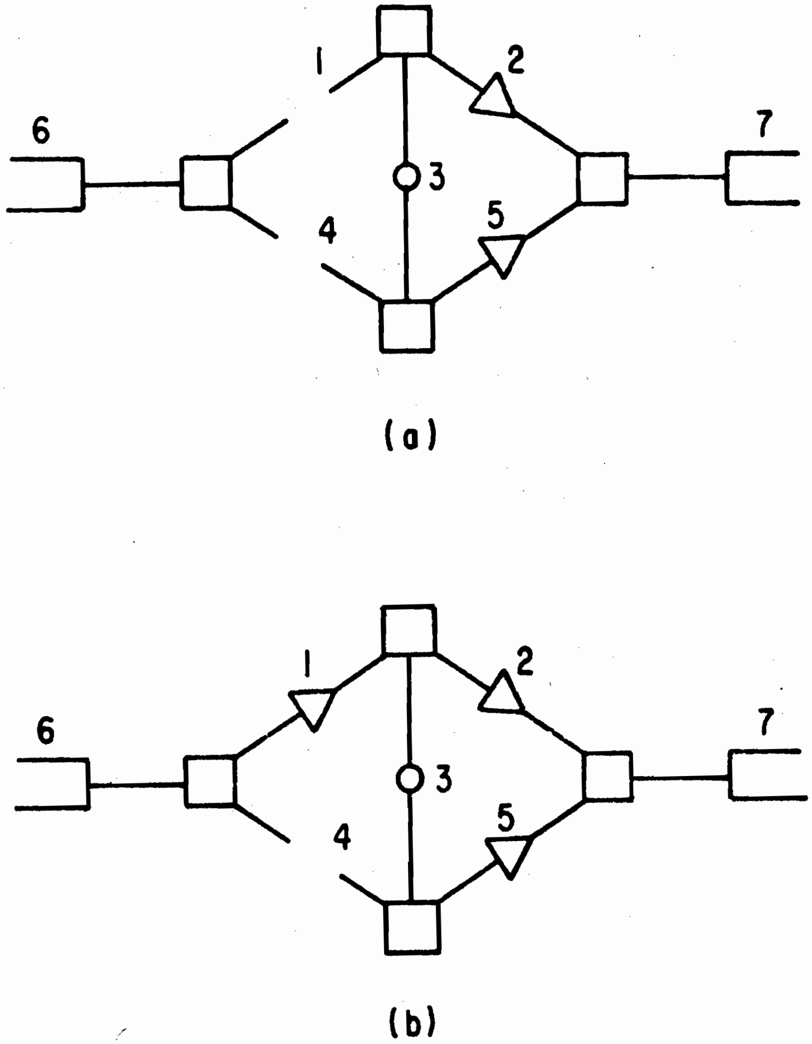


Figure 12. Realizations of the Network of Figure 11.

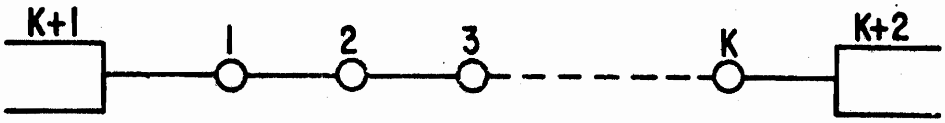


Figure 13. A Series Network.

Now if the probability that the i^{th} node is functional is given by p_i; then the probability of connectivity between the source and terminal nodes is just:

$$P = \prod_{i=1}^{K} p_i \tag{6}$$

Consider the parallel network shown in figure 14. The parallel configuration is indicated by the observation that only a single means exists for destroying connectivity between the terminals. That is, all K weighted nodes must be nonfunctional. The probability that the i^{th} node is nonfunctional is:

$$q_i = 1 - p_i \tag{7}$$

And the probability of not having connectivity between the terminals is:

$$Q = \prod_{i=1}^{K} q_i \tag{8}$$

hence, for this network

$$P = 1\text{-}Q = 1 - \prod_{i=1}^{K} q_i \tag{9}$$

Suppose now that there is a need for a network graph containing a large number of weighted nodes. Equations 6 and 8 can be used to decrease the complexity of such a graph if it contains any segments which are series/parallel. These equations, while somewhat trivial, thus have considerable utility. Unfortunately it is quite common to encounter networks which cannot be reduced by the methods of this section. See, for example, the network of figure 11.

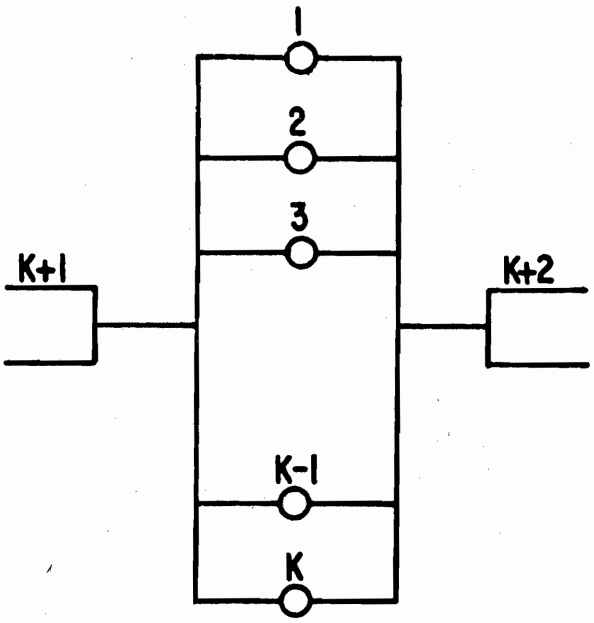


Figure 14. A Parallel Network.

In the next subsection the means for handling networks such as figure 11 are considered. The stepping stone will be another technique for analyzing small networks which are not necessarily series/parallel.

3.2.2 Analysis of Small Networks

Elementary network analysis can be expressed in terms of a logic table which contains all possible network realizations. Consider the example of figure 15. A logic table lists all elementary events that can occur in the network (an elementary event is a network realization). The logic table for figure 15 is shown in figure 16. The entries in the logic table are interpreted as: 0 - node does not exist and 1 - node exists.

Next a Boolean function is created from the logic table. The Boolean function is a listing of only those elementary events which will yield connectivity between the terminals. Hence the Boolean function for the example is as shown in figure 17, since events 1, 2, and 3 do not yield connectivity.

Consider the slightly more complicated example of figures 18 and 19. Note that the logic table is nothing more than a complete listing of all the binary numbers from 0 through $2^K - 1$; where K is the number of weighted nodes in the network. The Boolean function is created by examining each event in the logic table (in conjunction with the network realization implied thereby) and making a determination as to whether or not the particular event yields connectivity. The

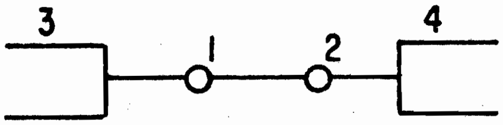


Figure 15. A Series Network.

		NODE	
		1	2
EVENT	1	0	0
	2	0	1
	3	1	0
	4	1	1

Figure 16. The Logic Table for the Network of Figure 15.

		NODE	
		1	2
EVENT	4	1	1

Figure 17. The Boolean Function for the Network of Figure 16.

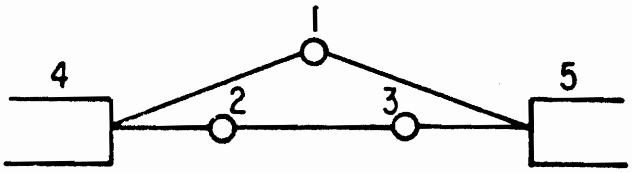


Figure 18. A Series/Parallel Network.

	NODE		
	1	2	3
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

Figure 19. The Logic Table for the Network of Figure 18.

Boolean function for figure 18 is as shown in figure 20, since events 1, 2, and 3 do not yield connectivity.

An analytic function for the terminal reliability can be created from the Boolean function by the correspondences:

$$0 \Longrightarrow 1 - p_i = q_i \qquad (10)$$

$$1 \Longrightarrow p_i \qquad (11)$$

and summing the probabilities of all events. Using these correspondences for the example of figure 18

$$P_{4-5} = q_1 \; p_2 \; p_3 + p_1 \; q_2 \; q_3 + p_1 \; q_2 \; p_3 + p_1 \; p_2 \; q_3 + p_1 \; p_2 \; p_3 \qquad (12)$$

where P_{4-5} is the terminal reliability. By letting all weighted nodes have the same existence probability:

$$p_i = p; \; i = 1,2,\ldots, K \qquad (13)$$

$$P_{4-5} = p + p^2 - p^3$$

3.2.3 Larger Networks

The technique exposed in subsection 3.2.2 clearly becomes unmanageable for networks with large numbers of nodes. This can be seen by considering

K = number of weighted nodes in the network (14)

$E = 2^K$ = number of elementary events in the logic table (15)

Thus, for example, a network with 24 weighted nodes would create a logic table with 16,777,216 elementary events. Clearly both time and computer storage requirements, using the technique of subsection 3.2.2, would be exorbitant. However, if the assumption of equation 13 is valid, an efficient technique can be found.

Consider the following approach. One merely acknowledges the existence of the Boolean function and finds only those parameters descriptive of the function which are required to find the terminal reliability. What parameters are important?

		NODE		
		1	2	3
EVENT	4	0	1	1
	5	1	0	0
	6	1	0	1
	7	1	1	0
	8	1	1	1

Figure 20. The Boolean Function for the Network of Figure 18.

From equation 13 the Boolean function (if one has it) can be used to get to an equation for the terminal reliability. It is also possible to get from an equation for the terminal reliability to the pertinent parameters of the Boolean function; since it is nothing more than the sum of products indicated by the Boolean function.

To illuminate this point consider again the example of figure 18. The Boolean function can be used to create an analytic function which ultimately yields the terminal reliability. Thus, the correspondence of figure 21 is created. Note that the correspondence indicated in figure 21 is event by event. Note also that the entry in the analytic function table is a "literal" interpretation of its corresponding entry in the Boolean function table. Consequently the word literal is often used in this context.

Event 3 yields the literal (by assumption 13), $(1-p)$ (p) $(p) = p^2 - p^3$; event 5 yields (p) $(1-p)$ $(1-p) = p-2\ p^2 + p^3$, etc. Then summing events 4 through 8, the equation for the terminal reliability is found.

There is a quicker way to arrive at the terminal reliability equation from the Boolean function. Since each row of the Boolean function creates a corresponding row in the analytic function with a well specified form, one needs to find only the form and the number of occurrences of that form. The form is dictated by the number of "1" entries in the row of the Boolean function. And of course each row in the Boolean function has the same number of entries; K, the number of weighted nodes. Now all rows of the Boolean function which contains the same number of "1's" will create the same analytic function row entry

Boolean function ⟷ Analytic function

NODE

EVENT

	1	2	3
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

⟷

NODE

	1	2	3
4	1-p	p	p
5	p	1-p	1-p
6	p	1-p	p
7	p	p	1-p
8	p	p	p

Figure 21. An Example of the Correspondence Between a Boolean Function and an Analytic Function; for the Example of Figure 18.

(since the column entries in the arithmetic function are commutative under the assumption that all weighted nodes have the same probability of survival p). Now let

j - number of "1's" in a particular row of the Boolean function (16)

m_j - number of rows in the Boolean function which contain exactly j "1's" (17)

A single event probability with exactly j surviving nodes is:

$$e_j = p^j \ (1-p)^{K-j} \qquad (18)$$

or,

$$e_j = \sum_{r=o}^{K-j} \binom{K-j}{r} \ (-1)^r \ p^{r+j} \qquad (19)$$

The sum of all event probabilities with exactly j surviving nodes is:

$$e' = \sum_{r=o}^{K-j} m_j \binom{K-j}{r} (-1)^r \ p^{r+j} \qquad (20)$$

It is now reasoned from the Boolean function that there will be rows containing (k) "1's," (k+1) "1's " etc. Further one can, by inspection of the network, determine the minimum number of nodes which must be operable to have connectivity, i.e.,

$$c \text{ - minimum number of nodes for connectivity} \tag{21}$$

Hence the minimum value of the subscript on m is c; and the maximum value of the subscript is, of course, K. Therefore the sum of all event probabilities indicated by the Boolean function (which is the terminal reliability of the network) is:

$$P = \sum_{j=c}^{K} \sum_{r=o}^{K-j} m_j \binom{K-j}{r} (-1)^r \; p^{r+j} \tag{22}$$

Observe now that the expansion of the series of equation 22 results in a polynominal in p, i.e.,

$$P = \sum_{i=1}^{K} a_i \, p^i \tag{23}$$

The a_i may be found by equating equations 22 and 23.

$$a_i = \sum_{j=c}^{i} m_j \binom{K-j}{i-j} (-1)^{i-j} \tag{24}$$

Consider now the means for finding m_j. The physical meaning of m_j in network connectivity terms is the number of elementary network realizations in which j of the K weighted nodes exist and the terminals are connected. The total number of elementary network realizations in which j of the K weighted nodes exist (whether connectivity results or not) is

$$K_j = \binom{K}{j} \tag{25}$$

It follows that the number of elementary cut events involving j nodes is just:

$$n_j = \binom{K}{j} - m_j \tag{26}$$

A technique for finding m_j, given j and K is:

(1) Find the value of K_j (equation 25)

(2) Form an L_j length sequence of random elementary network realizations with exactly j of the K nodes in the original network existing.

(3) Find the decimal percentage, k_j, of those networks from (2) which are connected.

(4) The approximate value of m_j is:

$$m_j \approx k_j K_j \tag{27}$$

Clearly this technique can be optimized by proper selection of the sequence lengths L_j, in step (2). This would be done by forcing a direct correspondence between the L_j and the K_j; K_j being a known variable for different values of j according to equation 20. The general concept used is known as stratified sampling. This is discussed in some detail in references 5 and 6.

3.3 GENERAL NETWORK ANALYSIS

The object of this subsection is to provide the reader with a brief review of several analysis techniques. The techniques will be successively more powerful and will all hinge on the concept of an event space.

3.3.1 The Event Space

A network with K weighted nodes is given. The corresponding event space is a K-dimensional hyperspace. Each one of the K dimensions corresponds one-to-one with one of the nodes in the network. It was seen earlier that a given network realization is a depiction of the network in terms of the state of each node (it exists or it does not exist). It was also pointed out that there are precisely 2^K possible realizations of the network. To complete the correspondence between a network and its event space it is required that each axis of the event space have only two labels, i.e., for the i^{th} axis the labels are: 0 - the i^{th} node does not exist and 1 - the i^{th} node exists. A convenient picture to hold in mind is figure 22, a region in a plane with 2^K admissible points in the region. Each point has a label which describes the network realization it pertains to.

Each point has an associated probability according to its corresponding network realization. The j^{th} point then has probability:

$$e_j = \prod_{i=1}^{K} f_i\ (p) \tag{28}$$

where

$$f_i\ (p) \begin{cases} = p_i \text{ if the } i^{th} \text{ node exists} \\ = q_i = 1 - p_i \text{ if the } i^{th} \text{ node does not exist} \end{cases} \tag{29}$$

It is axiomatic that:

$$\sum_{j=1}^{2^K} e_j = 1 \tag{30}$$

since by the definition of the event space each event is disjoint and the sum of all possible events is indicated in equation 30.

The object of a general network analysis can be stated in terms of this event space; viz., to sort out the events which are favorable to network connectivity from those events which are not.

3.3.2 Sum of All Elementary Events

Supposing for the moment that the sorting is already accomplished and that G of the e_j have been found favorable and renamed as g_i; the probability of network connectivity (terminal reliability) is:

$$P = \sum_{i=1}^{G} g_i \tag{31}$$

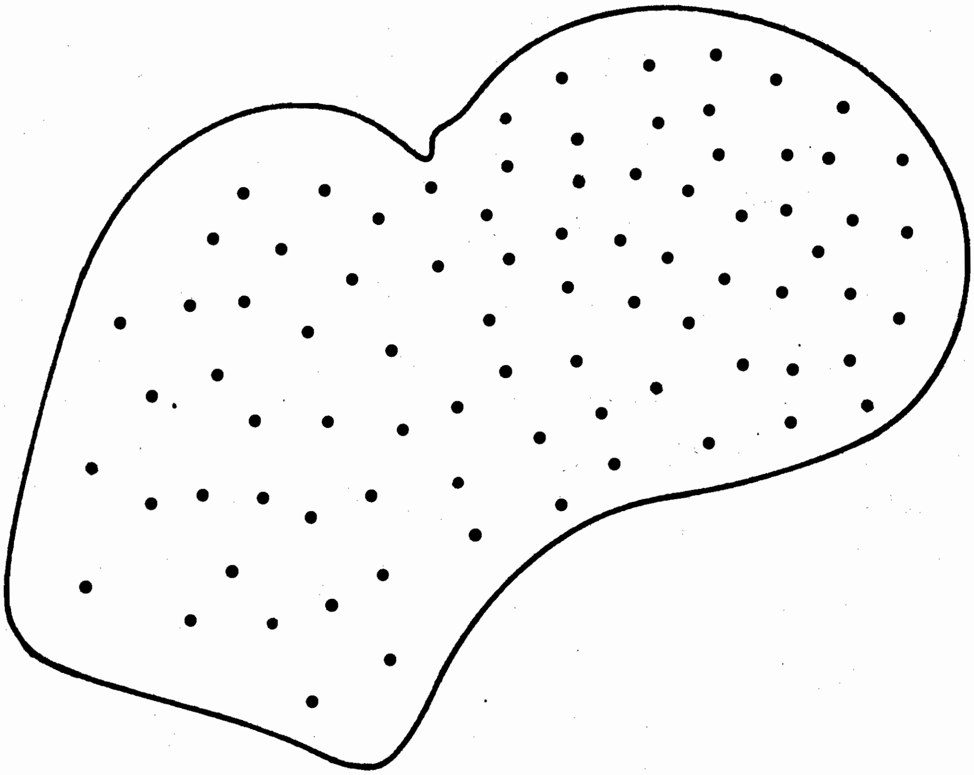


Figure 22. An Event Space.

Calling the unfavorable events b_k then the terminal unreliability is:

$$Q = \sum_{k=G+1}^{2^K} b_k \tag{32}$$

It is apparent that

$$P + Q = 1 \tag{33}$$

Now consider the sorting. As was shown in subsection 3.2.2, one technique for doing this consists of the following steps:

(1) List (or generate in a binary sequence) all possible elementary events in the hyperspace.

(2) Examine the network realization corresponding to each elementary event.

If the network realization provides connectivity

$$g_j \leftarrow e_i$$

If the network realization does not provide connectivity

$$b_k \leftarrow e_i$$

Another technique depends on first examining the network for all simple paths, $[\Gamma_r; r = 1,2,\ldots, L]$(ref. 7). The following steps are then exercised (ref. 8):

(1) as above

(2) If $e_i \in \Gamma_r$; $r = 1,2,\ldots, L$

then $g_j \leftarrow e_i$

If $e_i \notin \Gamma_r$; $r = 1,2,\ldots, L$

then $b_k \leftarrow e_i$

Consider an example. The simple paths of the network of figure 11 can be written as the literals:

$$\Gamma_1 \Leftrightarrow p_1\ p_2$$

$$\Gamma_2 \Leftrightarrow p_4\ p_5$$

$$\Gamma_3 \Leftrightarrow p_1\ p_3\ p_5$$

$$\Gamma_4 \Leftrightarrow p_2\ p_3\ p_4$$

Note that the literals above do not necessarily specify the state of every node in the network, only those on the path indicated. Literals such as these may be thought of as subspaces each of which may contain many elementary events. Now suppose that the 11th elementary event is being considered.

$e_{11} \Leftrightarrow 0_1\ 1_2\ 0_3\ 1_4\ 1_5$ (Binary 11); or

$e_{11} \Leftrightarrow q_1\ p_2\ q_3\ p_4\ p_5$ (the literal)

Now

$$e_{11} \not\in \Gamma_1$$

$$e_{11} \in \Gamma_2$$

$$\therefore \; g_j \longleftarrow e_{11}$$

Or suppose that the 13th elementary event is being considered.

$$e_{13} \qquad 0_1 \qquad 1_2 \qquad 1_3 \qquad 0_4 \qquad 1_5 \qquad \text{or}$$

$$e_{13} \qquad q_1 \qquad p_2 \qquad p_3 \qquad q_4 \qquad p_5$$

Now

$$e_{13} \not\in \Gamma_1$$

$$e_{13} \not\in \Gamma_2$$

$$e_{13} \not\in \Gamma_3$$

$$e_{13} \not\in \Gamma_4$$

$$\therefore \; b_k \longleftarrow e_{13}$$

Clearly, performing this examination for all the elementary events will sort the set

$$\left\{e_i ; \; i = 1,2,\ldots \; 2^K\right\}$$

into the two desired and disjoint sets. The set of favorable events

$$\left\{g_j ; \; j = 1,2,\ldots \; G\right\}$$

yields the terminal reliability by equation 31.

The set of unfavorable events

$$\left\{b_k ; \; k = G + 1, \; G + 2,\ldots \; 2^K\right\}$$

Yields the terminal unreliability by equation 32.

The basic difficulty with this technique is that the event space contains 2^K elementary events which require inspection by the algorithm. K is typically a large number and 2^K events quickly become too large and too time consuming to allow serious consideration of this technique.

3.3.3 Inclusion - Exclusion

Here it is presumed that the set of all simple paths is available as a set of literals $\{\Gamma_r\}$. It was observed in subsection 3.3.2 that the set of simple paths could be used to determine if a given elementary event belonged to the set $\{g_i\}$ or the set $\{b_k\}$. It should also be observed that a single simple path literal may be considered as a subspace which contains many elementary events. For example, the literal $p_1\ p_2$ in the network of figure 11 contains eight elementary events:

$$\begin{array}{ccccccccccc}
p_1 & p_2 & q_3 & q_4 & q_5; & (1 & 1 & 0 & 0 & 0) \\
p_1 & p_2 & q_3 & q_4 & p_5; & (1 & 1 & 0 & 0 & 1) \\
p_1 & p_2 & q_3 & p_5 & q_4; & (1 & 1 & 0 & 1 & 0) \\
 & & \vdots & & & & & \vdots & & \\
p_1 & p_2 & p_3 & p_4 & p_5; & (1 & 1 & 1 & 1 & 1)
\end{array}$$

A convenient picture to bear in mind here is shown in figure 23.

It would be ideal to simply add all the simple path literals to obtain the terminal reliability. Unfortunately, as is shown in figure 24, the same elementary events may belong to several different subspaces (path literals). So, if two path literals are summed to obtain terminal reliability one must be careful to subtract out the set of elementary events common to both literals (ref. 9). The overlap, or set of elementary events common to two literals, is found by

$$O_{i-j} \Leftarrow \Gamma_i \cap \Gamma_j \qquad (34)$$

For example, the overlap between the literals $p_1\ p_2$ and $p_4\ p_5$ is

$$O_{1-2} \Leftarrow p_1\ p_2\ p_4\ p_5$$

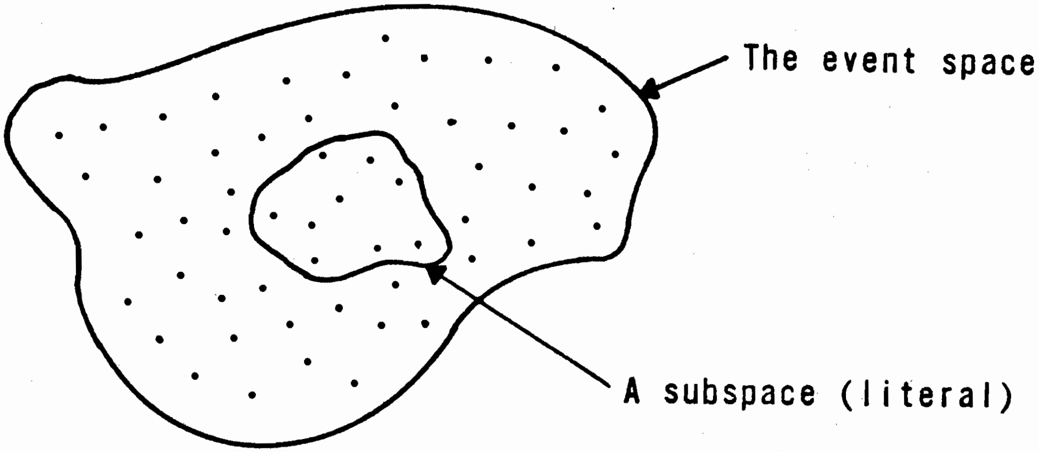


Figure 23. The Event Space and a Subspace.

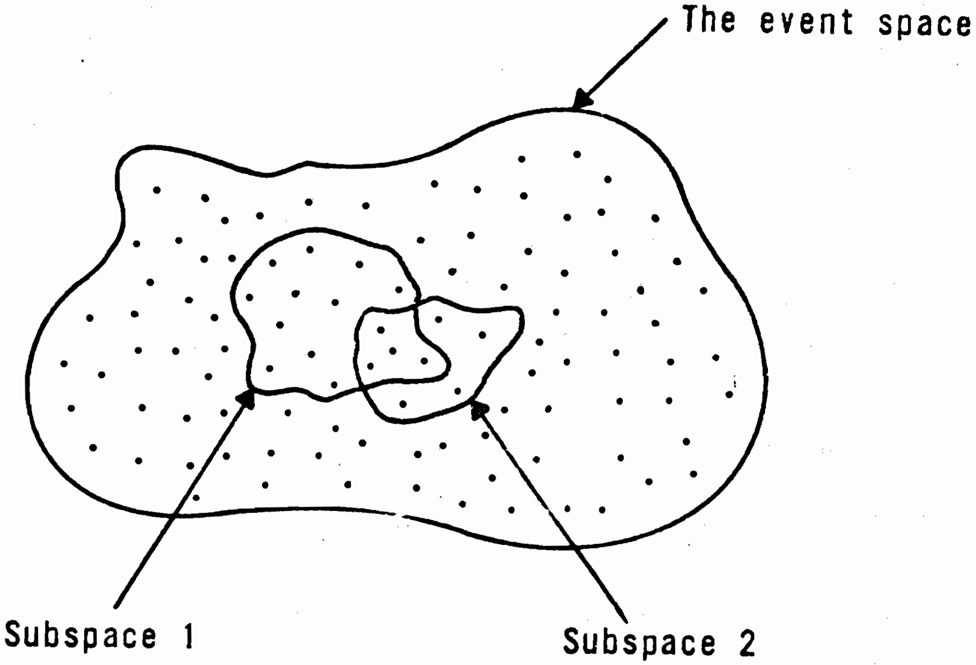


Figure 24. The Event Space and Two Overlapping Subspaces.

Now if the set of simple paths in a network is two in number, for example, Γ_1 and Γ_2 then clearly:

$$P \Leftarrow \Gamma_1 + \Gamma_2 - (\Gamma_1 \cap \Gamma_2) \tag{35}$$

If the P so found is imagined to describe a boundary for a single region in the hyperspace a recursive algorithm for finding the terminal reliability when the network contains L simple paths $(\Gamma_1,\Gamma_2,\ldots,\Gamma_L)$ is:

```
(1)  $i \leftarrow 1$
(2)  $P \leftarrow \Gamma_i$
(3)  $i \leftarrow i + 1$
(4)  $P \leftarrow P + \Gamma_i - (P \cap \Gamma_i)$
(5)  If $i < L$ go to (3)
(6)  Stop
```

This technique also has serious difficulties. The number of terms in the reliability expression is dependent on L, the number of simple paths. It can be shown that the number of terms, including empty sets, in P generated by this algorithm is:

$$T_L = 2^L - 1 \tag{36}$$

Since networks can (and have been) designed which have more simple paths than weighted nodes, the technique of subsection 3.3.2 may often be more efficient than this one.

3.3.4 Disjunction via Boolean Operations

Just as the principle of inclusion-exclusion can be used to form an expression for the terminal reliability from the set of all simple paths, so can Boolean operations. This technique, unlike that in the preceding subsection, always forms strictly additive terms in the terminal reliability expression. Suppose, as before, that one is given the set of simple paths in the network.

$$\{\Gamma_1,\Gamma_2,\ldots,\Gamma_L\}$$

Clearly, if there is only one simple path in the network

$$P \Leftarrow \Gamma_1 \tag{37}$$

Since there is more than one simple path it is reasonable that

$$P \geq \Gamma_1 \tag{38}$$

The "greater than" implies that events outside the subspace of Γ_1 contribute to the terminal reliability. Going down the list of simple paths, clearly events in the subspace Γ_2 so contribute. It would be desirable to add those events, but one must be sure that they are not already included in the subspace Γ_1. This can be accomplished by:

(1) finding those events <u>not</u> in Γ_1, i.e., $\overline{\Gamma}_1$.

(2) finding the intersection of the events in Γ_2 and in $\overline{\Gamma}_1$, $\overline{\Gamma}_1 \cap \Gamma_2$.

(3) Adding these events to the terminal reliability expression. The result of the foregoing operations for a network containing two simple paths, Γ_1 and Γ_2, yields the result:

$$P \Leftarrow \Gamma_1 + \overline{\Gamma}_1 \cap \Gamma_2 \tag{39}$$

Extending this reasoning to the case of a network containing L simple paths results in the following algorithm for terminal reliability:

```
(1)  $i \leftarrow 1$
(2)  $P \leftarrow \Gamma_i$
(3)  $i \leftarrow i + 1$
(4)  $P \leftarrow P + \overline{P} \cap \Gamma_i$
(5)  If $i < L$ go to (3)
(6)  Stop
```

This technique is discussed in some detail in reference 10. It is very powerful method which can be implemented on a digital computer in a reasonably straightforward manner. As with all network analysis algorithms though, it will not terminate for networks containing large numbers of nodes. In this event the algorithm will stop with a lower bound for terminal reliability.

The technique to be discussed in the next subsection has the same merits as above; with the additional features of greater speed, and an upper bound is also provided.

3.3.5 Direct Labelling of the Event Space

The ideas exposed so far in subsection 3.3 have depended on either subtracting overlap or forming disjoint sets of events in the hyperspace. The disjoint sets were formed by using some simple properties of the space itself and the non-disjoint simple paths. Here, a direct method is explored. The set of all simple paths is not required; only the network itself and the concept of the event space is needed.

The event space is imagined as consisting of 2^K admissible points and one seeks the largest possible subset of these points which are favorable to network connectivity. This subset is described by the literal for the shortest path through the network. Call this literal F_1. One can then conclude:

$$P \geq F_1 \tag{40}$$

The "greater than" is intended to express the idea that other events, in $\overline{F}_1$, may also contribute to the terminal reliability. Now one can view the created literal(s), $\overline{F}_1$, as describing "sub-networks;" each of which is easily made disjoint from the others. That is, $\overline{F}_1$ may be written as a sum of several disjoint literals as indicated in figure 25. Further, each subnetwork may be examined for its short path in the same manner as the original network. This process creates literals which, when intersected with the created literals in which they occurred, create disjoint and additive terms in the terminal reliability expression.

At some point in the process a literal will be created which describes a subnetwork in which no path exists. Call this literal C_1 (a cut literal). One can conclude then that:

$$Q \geq C_1 \tag{41}$$

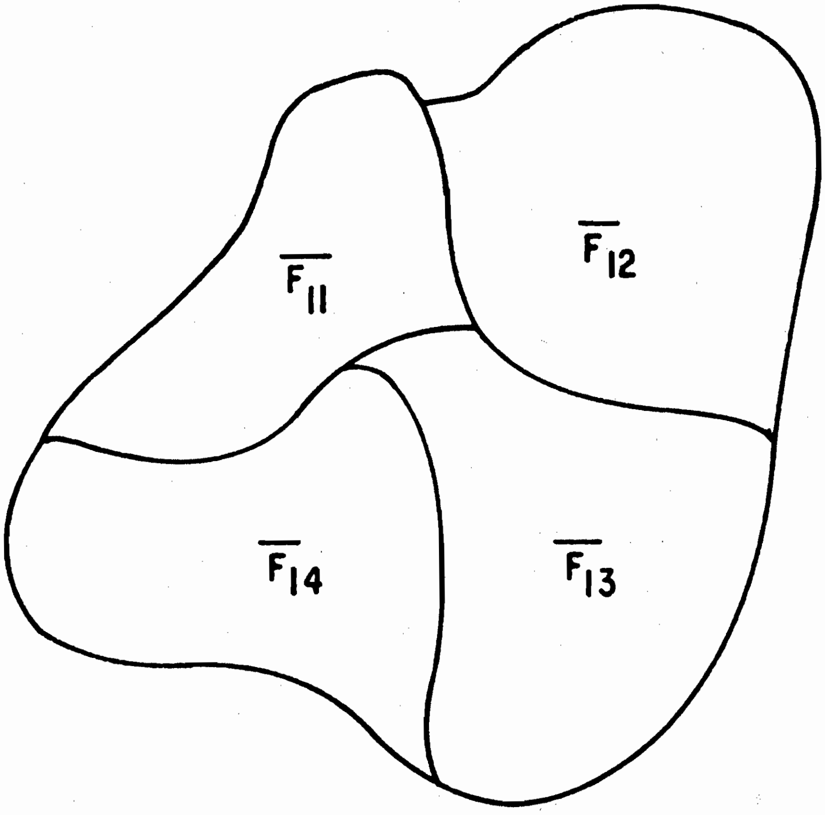
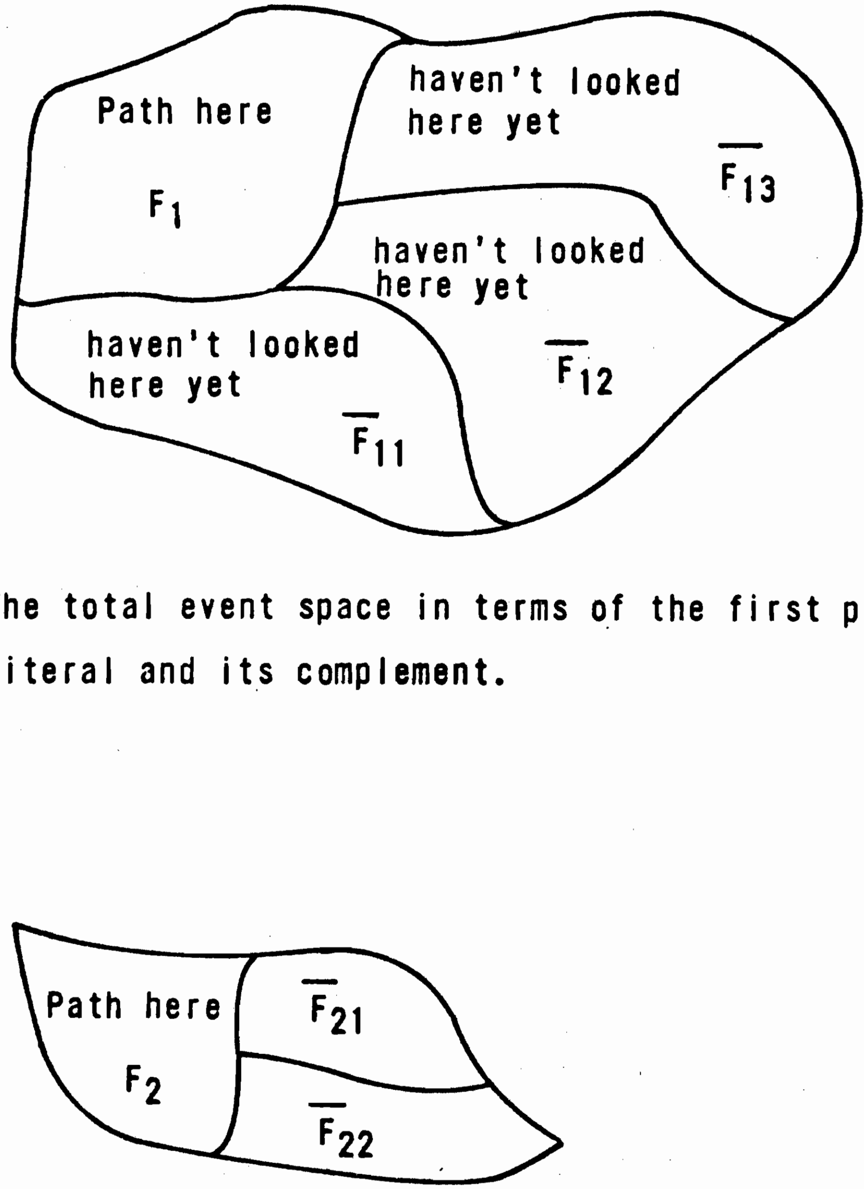


Figure 25. The Subspace for $\overline{F}_1$ Shown as the Sum of the Disjoint Subspaces $\overline{F}_{11}$ through $\overline{F}_{14}$.

The "greater than" is intended to express the idea that other literals later on in the process may also be unfavorable to network connectivity. Naturally enough, the list of created literals along with the list of cut literals and the list of path literals intersected with their associated created literals provides a complete covering of the event space at any stage in the process. A convenient pictorial presentation of the process is shown in figure 26.

A full exposition of these ideas is contained in SDAN 16. The algorithm (from SDAN 16), insofar as is known, is most efficient means for finding the symbolic form of the terminal reliability function. As a by product it also creates the terminal unreliability function. As was pointed out at the beginning of section III, the symbolic function is extremely valuable in scenario- dependent communications network analysis. As shall be shown in section IV and subsequently, it is also valuable in gaining insight into network reliability questions concerning resource distribution, node valuation, and network design.

Before addressing these issues, consider a basic part of the direct labelling algorithm.



(a) The total event space in terms of the first path literal and its complement.

(b) The subspace $\overline{F}_{11}$

Figure 26. A Pictorial Presentation of the Direct Labelling Process.

3.3.6 An Algorithm for Finding the Shortest Path in a Network

The analysis process discussed in subsection 3.3.5 depends on finding a path through a network. It can easily be shown that a short path contains a larger "area" of the total event space than a longer path. The short path also creates fewer literals to cover the remainder of the event space. Thus, while the process of the preceding subsection will work by finding any path, the shortest path causes it to work more efficiently. Note also that the efficiency of the process of subsection 3.3.5 is directly related to the efficiency of finding a single short path through a network. The question of the most efficient way of finding a short path could use additional research. One algorithm for this problem is offered here.

Suppose that one has the connectivity matrix for the network. The rows and columns of the matrix are labeled with their respective node labels. The entries in the k^{th} row of the matrix are either "1" or "0." A "1" is used if the k^{th} node can transmit to the node identified with the column in question; a "0" is used otherwise. The connectivity matrix for the network of figure 11 is shown in figure 27.

In addition to the connectivity matrix, the source and terminal node labels are also required. The algorithm requires two subroutines. The first subroutine will be called PATH. The second is called EXIST. The function of PATH is to task EXIST and to keep a record of the locations where EXIST performed successfully. EXIST takes the information from PATH concerning source node, terminal node, and maximum allowable path length (MAPL), then decides if a path does exist under these constraints.

Subroutine EXIST works as follows:

```
(1)  Path length (PL) ← 0

(2)  PL ← PL + 1

(3)  Build a list of all nodes PL moves away from source node.

(4)  Is the terminal node in the list (3)?
     If so, set FLAG ← 1, set MAPL ← PL, and return to PATH
     If not, go to (5)

(5)  Is PL ≥ maximum allowable path length (MAPL)?
     If so, set FLAG = 0 and return to PATH
     If not, go to (2)
```

$$\begin{array}{cc|ccccccc}
 & & \multicolumn{7}{c}{\text{NODES}} \\
 & & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
\text{N} & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\text{O} & 3 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
\text{D} & 4 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
\text{E} & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\text{S} & 6 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

Figure 27. The Connectivity Matrix for the Network of Figure 11.

Subroutine PATH works as follows:

(1) MAPL ← (some large number).

(2) Call EXIST; pass over MAPL, source node, and terminal node. EXIST returns FLAG and MAPL ← PL.

(3) MAPL ← MAPL - 1.

(4) Build a list of all nodes one move away from source node. Index this list (I).

(5) Is the terminal node in the list (4)? If so, stop. If not go to (6).

(6) I ← 0

(7) I ← I + 1

(8) Source node ← I^{th} node in the list of (4).

(9) Call EXIST

(10) If FLAG = 0, go to (7).
If FLAG = 1, write the source node in a list of nodes in the short path.

(11) Go to (3).

SECTION IV

APPLICATIONS AND EXAMPLES

The material in the preceding sections can be summarized in three fundamental results:

(1) The confidence-reliability function for the ith node can be found from direct testing and takes the form:

$$C\ (p_i \geqq \hat{p}_i) = \frac{\int_{\hat{p}_i}^{1} X^{S_i}\ (1-X)^{T_i-S_i}\ dX}{\int_{o}^{1} X^{S_i}\ (1-X)^{T_i-S_i}\ dX} \tag{42}$$

where

p_i The true but unknown probability of survival of the ith node.

$\hat{p}_i$ The estimate of p_i.

C( ) Confidence, or fiducial probability, that the argument ( ) is correct.

T_i Total number of tests performed on the ith node.

S_i The number of tests in the test series that the ith node passed.

(2) A network reliability analysis can be performed to yield:

$$P \geq \sum_{i=1}^{A} \alpha_i \tag{43}$$

$$\alpha_i = \prod_{j=1}^{K} f_i(p_j) \tag{44}$$

$$f_i(p_j) = \begin{cases} p_j & \text{if the } j^{th} \text{ weighted node exists} \\ q_j = (1 - p_j) & \text{if the } j^{th} \text{ weighted node does not exist} \\ 1 & \text{if the } j^{th} \text{ weighted node is irrelevant} \end{cases} \qquad (45)$$

where

P Literal for the terminal reliability of the network

A Number of terms in the series

K Number of weighted nodes in the network

p_j Literal for the probability of survival of the j^{th} weighted node

(3) The same network reliability analysis as (2) will yield:

$$Q \geq \sum_{i=1}^{B} \beta_i \qquad (46)$$

$$\beta_i = \prod_{j=1}^{K} g_i(p_j) \qquad (47)$$

$$g_i(p_j) = \begin{cases} p_j & \text{if the } j^{th} \text{ weighted node exists} \\ q_j = (1 - p_j) & \text{if the } j^{th} \text{ weighted node does not exist} \\ 1 & \text{if the } j^{th} \text{ weighted node is irrelevant} \end{cases} \qquad (48)$$

where

Q Literal for the terminal unreliability of the network

B Number of terms in the series

A few of the potential applications of these results are developed by way of examples.

4.1 CONFIDENCE RELIABILITY FUNCTION FOR A NETWORK

Consider a network in which the probabilities of survival of the nodes are not known. It is assumed that each node in the network has had a limited number of tests performed on it. This of course casts the reliability information on each node into the mold of equation 42. The implication of this is that each node may have any probability of survival in the range [0,1]. But there are different confidences in different values in this range according to the confidence distribution of equation 42. Now of course it can be argued that a single true value of probability of survival exists for each node. Then likewise a single true value of terminal reliability exists for the network. Unfortunately these can never be known. The best that can be done is to form the confidence distribution for the terminal reliability of the network.

Consider the network of figure 28. This is a simplified version of the ARPA computer net. Suppose, for the sake of illustration, that each weighted node in this network has been subjected to 10 tests. Each test simulated a hostile environment of concern. Information is desired on what the confidence-reliability distribution is between the terminals UCLA and CMU in the hostile environment. The test results for the weighted nodes are assumed to be:

Node (i)	T_i	S_i
2	10	10
3	10	8
4	10	9
5	10	9
6	10	8
7	10	10
8	10	10
9	10	8
10	10	9
11	10	9
12	10	8
13	10	8

The algorithm to solve this problem is:

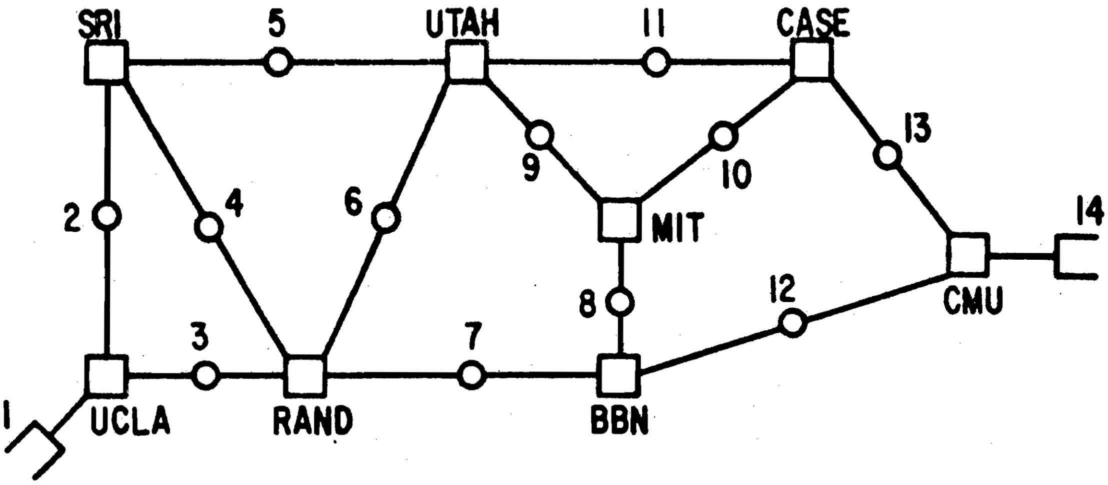


Figure 28. The ARPA Computer Network after Series Parallel Reduction

(1) Find the terminal reliability function, equation 43, using the algorithm of SDAN 16.

(2) Build the confidence reliability function(s) for the nodes (See equation 42.).

(3) j ← 0

(4) j ← j + 1

(5) Form a single realization for the estimate of the probability of survival of the i^{th} node; i.e., $\hat{p}_i$ = 1,2,... K. (Here conventional numerical techniques are used to map a random variable uniformly distributed in [0,1] to a new random variable $\hat{p}_i$ distributed in [0,1] according to the confidence reliability distribution for the i^{th} node formed in step (2).).

(6) Form a single realization for the estimate of the terminal reliability, $\hat{P}$, using results of steps (5) and (1).

(7) Place the results of step (6) into a left accumulated histogram.

(8) If j < j max, go to (4).

(9) Normalize the result of (7).

(10) Stop.

Figures 29, 30, and 31 show the results of exercising step (2) of the above algorithm. Figure 32 shows the confidence reliability function between the

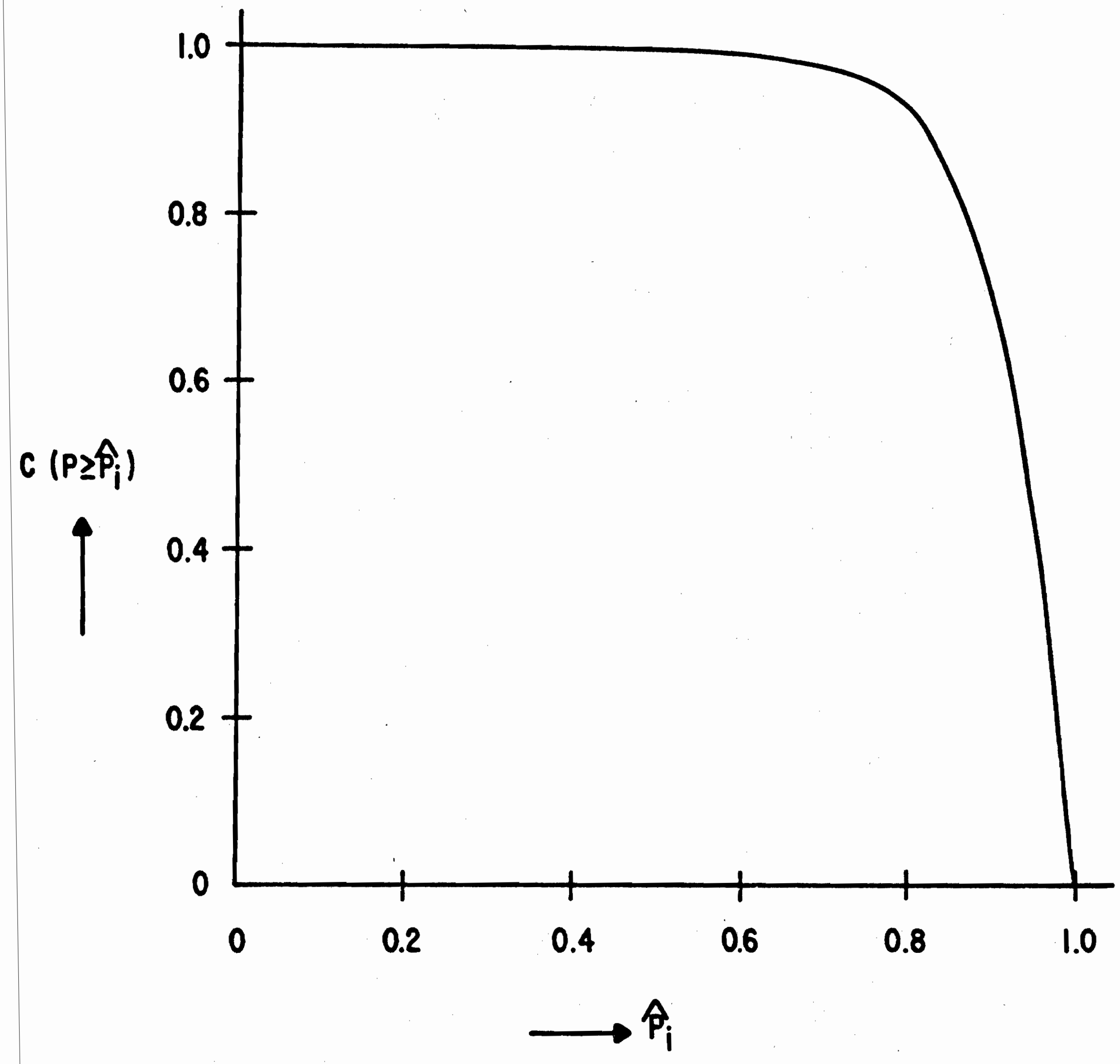


Figure 29. Confidence Reliability Distribution for the i^{th} Node; i = 2, 7, and 8.

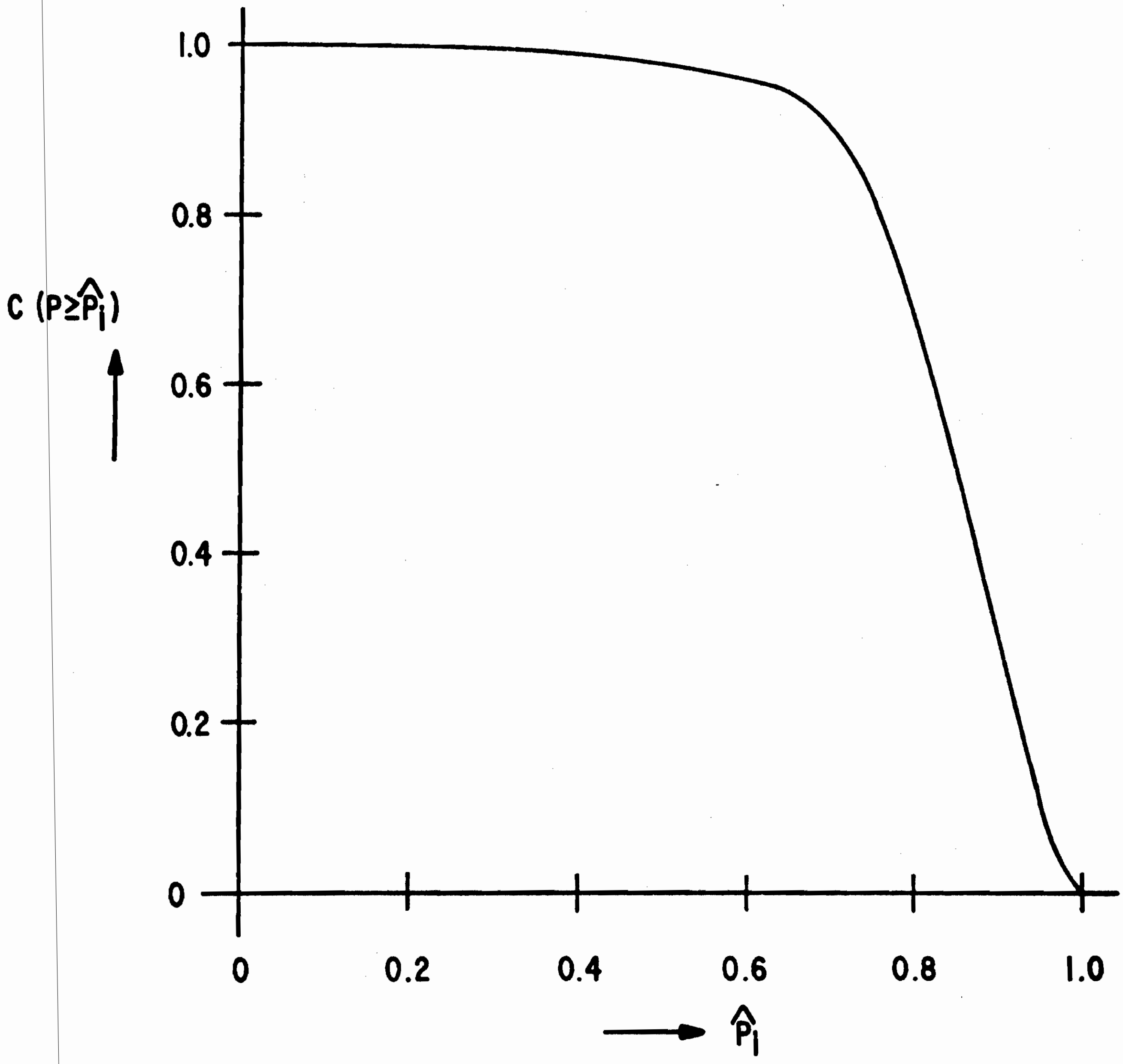


Figure 30. Confidence Reliability Distribution for the i^{th} Node; i = 4, 5, 10, and 11.

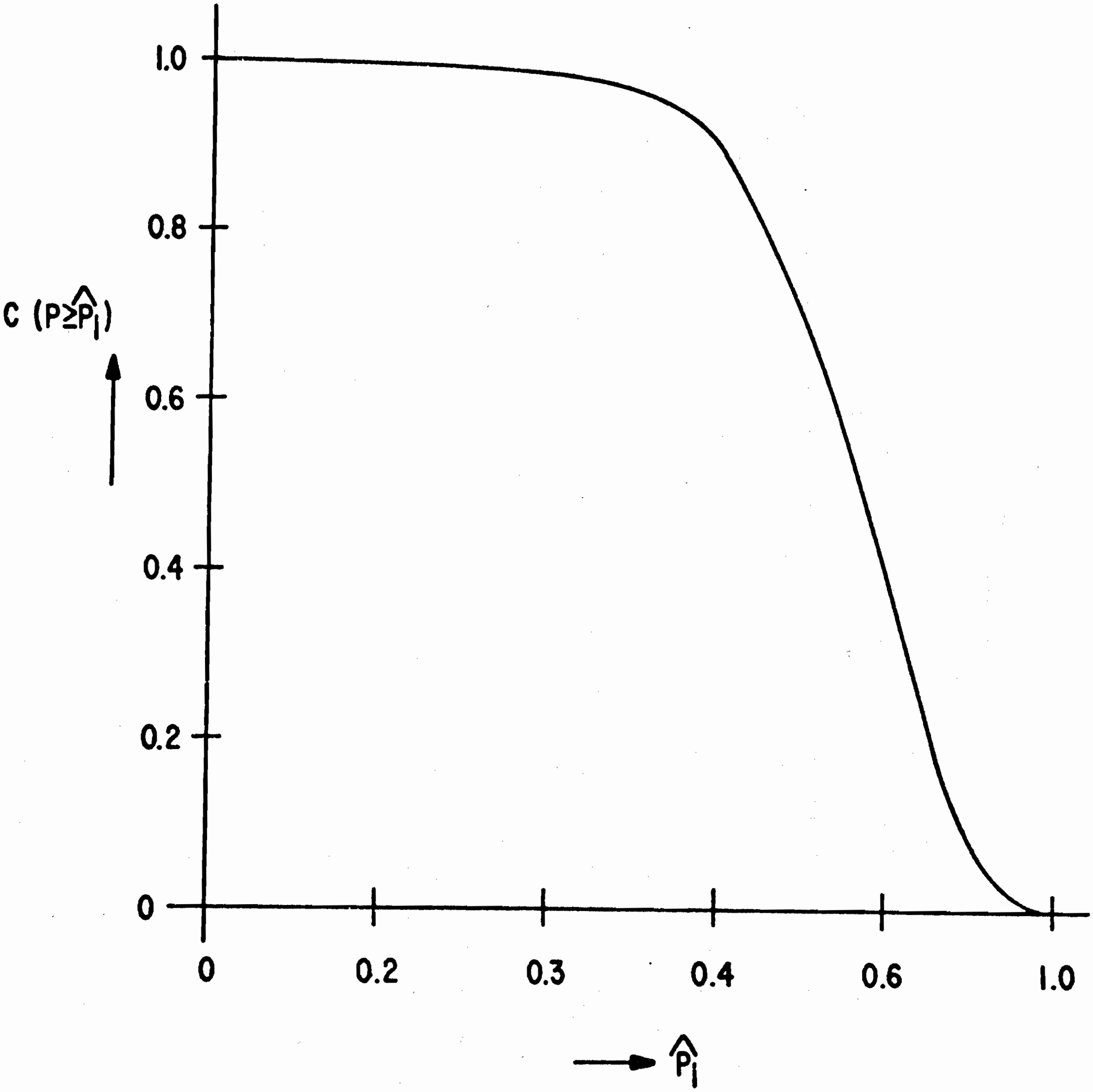


Figure 31. Confidence Reliability Distribution for the i^{th} Node; i = 3, 6, 9, 12, and 13.

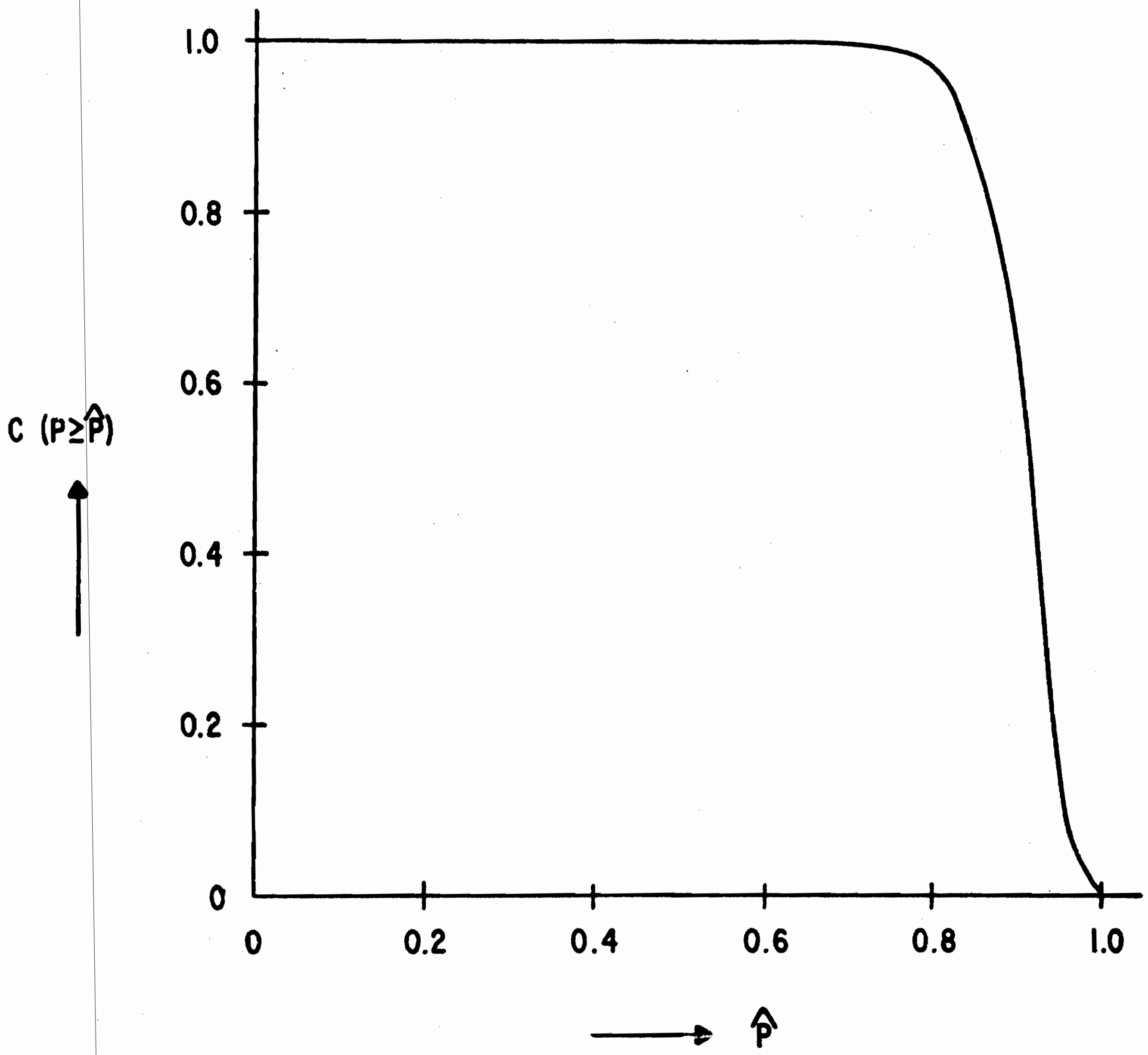


Figure 32. Confidence Reliability Distribution for the Terminals, UCLA and CMU.

terminals UCLA and CMU. This output exercises the entire algorithm. The algorithm is extremely fast since it requires only a single solution to the network analysis problem, resulting in a symbolic solution for terminal reliability which is solved repetitively in a Monte Carlo sense in steps (4) through (8). The result of figure 32 shows, for example, that one can have 90 percent confidence that the terminal reliability, in the hostile environment, is equal to or greater than 0.83.

4.2 NETWORK ASSESSMENT WITH AN ATTACK SCENARIO

This subsection considers the problem of finding the confidence reliability distribution for a network under an attack which evolves with time. A cookie cutter approach to the problem is not taken. Rather, it is assumed that the effect of the individual weapons on node probability of survival is known. Specifically, it is assumed that this effect is known through simulation testing (subsection 2.2.1, and equation 42). For the sake of showing the flexibility of the network analysis algorithm it is also assumed that individual weapon effects are local in nature. That is, a weapon directed at node i does not affect the reliability of node j ($i \neq j$). It is further assumed that the attack scenario does not allow repair of any nodes in the network.

Consider again the network of figure 28. Suppose that the individual nodes (2 through 13) have teen tested to the weapons effect of interest with results as indicated in subsection 4.1. Further suppose that the attack scenario consists of three waves of weapons at times T1, T2, and T3. The individual weapons are targeted at the different times on the nodes indicated below:

Node	T1	T2	T3
2	1	0	1
3	1	1	1
4	1	0	0
5	1	1	1
6	1	0	0
7	1	1	0
8	1	1	1
9	1	0	0
10	1	0	1
11	1	1	0

Node	T1	T2	T3
12	1	1	0
13	1	0	0

How does the estimate for the terminal reliability behave under this scenario?

The algorithm of subsection 4.1 will solve this problem with only a minor modification. The modification consists of using the expedient of reconfiguring the network in accordance with the attack scenario. This reconfiguration does not affect the symbolic form of the terminal reliability solution; only its interpretation. Consider the following. The nodes in the network may be interpreted as a probability; namely the probability that the node still functions after a single attack. If the node has been attacked n times (without repair); then the probability that it still functions is the same as the probability that a series string of n nodes, each attacked once, will function. Using this expedient, the algorithm of subsection 4.1 is augmented so that the evaluation of the symbolic form of the terminal reliability now incorporates a count of the number of weapons delivered to a given node as a function of the attack scenario. If the i^{th} node has been attacked n times, the realization of $\hat{p}_i$ (step (5)) is found from:

```
(1) $\hat{p}_i \leftarrow 1.0$
(2) $j \leftarrow 0$
(3) $j \leftarrow j + 1$
(4) Form a single realization for the estimate of the probability of
    survival of the $i^{th}$ node at the $j^{th}$ attack; $\hat{p}'_{ij}$
(5) $\hat{p}_i \leftarrow \hat{p}_i * \hat{p}'_{ij}$
(6) If $j < n$, go to (3)
```

The result of using the modified algorithm of subsection 4.1 under the assumed scenario and test results is shown in figure 33.

Without going into detail, it should be clear that a straightforward extension of the expedient used in this subsection will allow multiple effects assessment. For example, the reliability under ambient conditions could be folded in with a single weapons effect; or multiple weapon effects. Node repair could also be easily accommodated.

It should also be noted that a way has been shown to obtain the entire confidence reliability distribution. Normally only the expected value of the terminal reliability estimator is found; particularly when time dependence is involved. The reason for this is simply that previous algorithms have been too slow to effectively find the entire distribution. This algorithm can be simplified to work only with expected values with a significant increase in speed. For example, the result of figure 33 was found in 174 seconds; if expected values only are used the solution time reduces to 7 seconds (both solution times are from using the AFWL CDC 6600 computer).

4.3 VALUATION OF NODES

The notion that some nodes in a network may be more important than others is a rather old and intuitively agreeable idea. Unfortunately little exploitation of this notion has been possible in the past. References 11 and 12 are good examples of past efforts. This problem is now addressed as an application of the algorithm of subsection 3.3.5.

One additional concept is needed for this treatment. This is simply that the network itself has value. If one considers the network as a resource it is possible to pose objective functions which maximize the expected value of the resource. The expected value will depend on the nodes in the network. Consider for example the telephone network. This network has value to AT&T inasmuch as revenue is realized from the users of the network. If some of the nodes in the network do not operate (or are unavailable due to traffic congestion) then some potential users are denied the use of the network. In turn, AT&T is denied the revenue from that potential use and the expected value of their resource (the network) is decreased. A more fundamental example from a military point of view involves warning networks. Here one could think of the value of the network as being the value of the resource which must be alerted via the network in case of impending attack. For example, this value may be the equivalent of the capital investment involved in a wing of B-52 bombers.

Other examples could be given, but the point is made that a value or set of values may be placed on a network. This can be done from the point of view of, for example, a corporation which owns the network and derives revenue from a multiplicity of users; or from the point of view of an individual user who depends on the network to protect some commodity of value to him. This last problem shall be treated here.

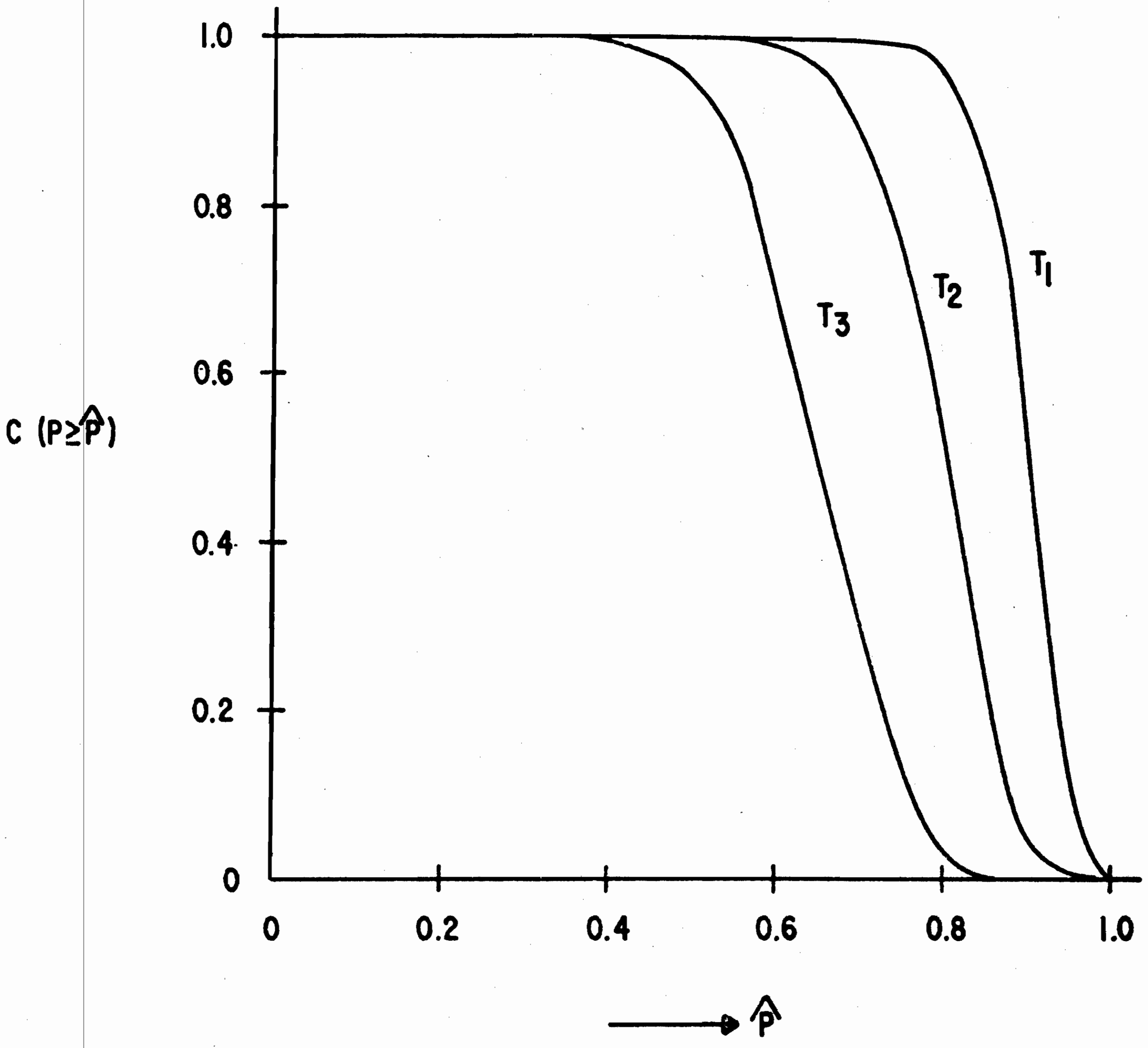


Figure 33. Drawdown of the Confidence Reliability Distribution for the Network of Figure 28.

4.3.1 An Insurance Strategy

The concept to be exploited in this subsection is that a network user determines the revenue he will obtain from use of the network if it is perfectly reliable. Call this revenue V_T. The user has access to data whereby he can determine the expected value of the reliability estimate for each node in the network in an ambient environment. From this data he determines his expected revenue from use of the network.

Now this user also knows that occasionally one of the nodes in the network is removed (perhaps for repair or maybe due to hostilities) for an extended period of time. He knows intuitively that during this period his expected revenue from use of the network will decrease. There is a company in town that sells insurance to protect against this happening. The policies are written so that, regardless of the face amount, only his real losses or the face amount (whichever is less) are covered; but at the same time the premium goes up as the face amount does. Clearly this man has a node valuation problem. The face amount of the policy he buys on each node will reflect the value of that node.

Suppose that a series of random checks have been made on the availability of the i^{th} node in a network. By equation 42, the confidence reliability distribution for the i^{th} node may be found. Alternately, the confidence reliability density function may also be found as

$$\mathcal{C}(\hat{p}_i) = \frac{\partial}{\partial \hat{p}_i} C(\hat{p}_i \geq p_i)$$

or

$$\mathcal{C}(\hat{p}_i) = \frac{\hat{p}_i^{\,S_i}(1-\hat{p}_i)^{T_i-S_i}}{\int_0^1 X^{S_i}(1-X)^{T_i-S_i}\, dX} \tag{49}$$

From equation 49 the expected value for the reliability of the i^{th} node is:

$$E\hat{p}_i = \frac{\int_0^1 \hat{p}_i^{\,S_i+1}(1-\hat{p})^{T_i-S_i}\, d\hat{p}_i}{\int_0^1 X^{S_i}(1-X)^{T_i-S_i}\, dX} \tag{50}$$

The integrals involved are the beta integrals (ref. 13) and the solution to equation 50 is:

$$E\hat{p}_i = \frac{S_i + 1}{T_i + 2} \qquad (51)$$

The reader should note that the maximum likelihood estimate for the reliability of the i^{th} node is not being used. Instead, the mean (or 50 percent confidence estimate) is used because of its useful property in estimating the mean terminal reliability. This property is embodied in the following (ref. 13):

From equations 43, 44, and 45:

$$E\hat{P} \geq \sum_{i=1}^{A} \gamma_i \qquad (52)$$

$$\gamma_i = \prod_{j=1}^{K} h_i(\hat{p}_j) \qquad (53)$$

$$h_i(\hat{p}_j) = \begin{cases} E\ \hat{p}_j & \text{if the } j^{th} \text{ weighted node exists} \\ 1-E\ \hat{p}_j & \text{if the } j^{th} \text{ weighted node does not exist} \\ 1 & \text{if the } j^{th} \text{ weighted node is irrelevant} \end{cases} \qquad (54)$$

Now the network analysis algorithm of subsection 3.3.5 solves for the symbolic form of equations 52, 53, and 54. Therefore one needs only substitute numerical values for each node, according to equation 51 to find $E\hat{P}$. It follows that the expected value of the network is:

$$EV_T = V_T * E\hat{P} \qquad (55)$$

For the insurance problem one needs only to note that equation 55 can easily be evaluated under two different conditions:

(1) The "normal" or ambient condition.

(2) Remove the i^{th} node; i.e., set $\hat{p}_i = 0.0$.

The expected value of the i^{th} node then is:

$$EV_i = EV_{T(1)} - EV_{T(2)} \qquad (56)$$

This expected value, EV_i, is the face amount of the insurance policy the network user should buy on the i^{th} node.

Consider again the example of figure 28. The user is UCLA who obtains revenue by communicating with CMU. Problem data are:

$$V_T = 1000$$

Availability of the nodes is estimated from the measurements:

Node (i)	T_i	S_i
2	998	949
3	998	899
4	998	974
5	998	974
6	998	899
7	998	949
8	998	949
9	998	899
10	998	974
11	998	974
12	998	899
13	998	899

Implementing the equations of this section in an algorithm and solving, the results are:

Node (i)	$EV_{T(1)}$	$EV_{T(2)}$	EV_i
2	984	890	94
3	984	939	45
4	984	982	3
5	984	977	7
6	984	983	1
7	984	975	9
8	984	980	5
9	984	983	1

Node (i)	$EV_{T(1)}$	$EV_{T(2)}$	EV_i
10	984	982	3
11	984	977	7
12	984	894	90
13	984	893	92

To demonstrate a point which may not be completely obvious, suppose that the availability of the nodes was estimated from a different set of measurements:

Node (i)	T_i	S_i
2	998	974
3	998	949
4	998	899
5	998	899
6	998	949
7	998	974
8	998	974
9	998	949
10	998	899
11	998	899
12	998	949
13	998	949

The results under these measurements are:

Node (i)	$EV_{T(1)}$	$EV_{T(2)}$	EV_i
2	995	947	48
3	995	962	33
4	995	991	4
5	995	989	6
6	995	992	3
7	995	984	11
8	995	993	2

Node (i)	$EV_{T(1)}$	$EV_{T(2)}$	EV_i
9	995	992	3
10	995	991	4
11	995	990	5
12	995	938	57
13	995	947	48

Clearly the value of each node depends on the reliability of each of the other nodes in the network, as well as the network topology. This can be seen by a careful inspection of equations 52 through 56. As a remark on the efficiency of the algorithm used, the above problem was solved in 7 seconds on the AFWL CDC-6600 computer.

Finally, one notes that there are many variations on this type of problem. For example, multiple users with different values, multiple node removals, etc. One could not hope to address them all. However, it is believed that reasonably straightforward extensions of the concepts of this subsection will handle most, if not all, of the problems of real interest.

4.3.2 Optimum Allocation of a Fixed Assessment Budget

The value system used in this case hinges on the fact that the budget to assess the network is a fixed quantity. Note from equation 51 that the expected value for the reliability estimate of the i^{th} node depends on the number of tests (and the results) performed on that node. A test involves the expenditure of part of the fixed budget. Since the budget is fixed, clearly the number of tests that can be performed is likewise fixed. Suppose that one is budget constrained to perform no more than the V_T tests. The number of tests on the i^{th} node is T_i, then:

$$V_T = \sum_{i=1}^{K} T_i \qquad (57)$$

The objective of this subsection may be stated as finding that allocation of tests, $\{T_i;\ i = 1, 2, \ldots, K\}$, which will maximize the expected value of the terminal reliability estimate, $E\hat{P}$.

As was seen in subsection 4.3.1, the valuation of each node in the network depends on the expected value of the reliability estimate for all the other nodes in the network. This indicates that one must make some assumption about the test results on each node before resource (test) allocation is possible. Using this assumption it is possible to define an optimum allocation which is indeed optimum as long as the assumed results are borne out by testing. If testing results depart from anticipated results, the testing results may be used to anticipate future results, which in turn will indicate a re-allocation of the remaining resources. As the reader can see, the value of each node in a fixed assessment budget situation is reflected in the number of tests performed on the node.

The objective function is to find $\{T_i; i = 1, 2, \ldots, K\}$ such that $E\hat{P}$ is a maximum. Now T_i is constrained by equation 57; and $E\hat{P}$ (equations 52, 53, and 54) is dependent on $\{E\hat{p}_i; i = 1, 2, \ldots, K\}$. In turn, $E\hat{p}_i$ is dependent on T_i, S_i (equation 51). Thus, assume, or anticipate, test results on each node in order to allocate the budget. Call the assumed value of $E\hat{p}_i$ for the i^{th} node $A\hat{p}_i$. This allows the assignment of:

$$S_i = A\hat{p}_i \ (T_i + 2) - 1 \qquad (58)$$

If the S_i so found is not integer (as it must be) then set S_i to the nearest integer or T_i, whichever is less.

Now given an arbitrary set $\{T_i; i = 1, 2, \ldots, K\}$, satisfying the constraint of equation 57 and the set $\{A\hat{p}_i; i = 1, 2, \ldots, K\}$ the set $\{S_i; i = 1, 2, \ldots, K\}$ can be generated. Using the arbitrary set $\{T_i; i = 1, 2, \ldots, K\}$ and the generated set $\{S_i; i = 1, 2, \ldots, K\}$, one can easily solve for $E\hat{P}$ by using equations 51, 52, 53, and 54. The question now is: Does another set $\{T_i'; i = 1, 2, \ldots, K\}$ exist which will yield $E\hat{P}'$ greater than $E\hat{P}$? In case the reader has some philosophical problems at this point with the set $\{A\hat{p}_i; i = 1, 2, \ldots, K\}$ note again that this set can be changed at will during the process of testing nodes to be in accord with past results. It is necessary only to make some assumption to initialize the allocation process.

To the best of this writer's knowledge, the above problem has never been solved. It is believed however, that a solution is now possible; all that is required is the time to code an algorithm which will operate on an arbitrary initialization set $\{T_i; i = 1, 2, \ldots, K\}$ to generate an optimum set $\{T_i^\circ; i = 1, 2, \ldots, K\}$. This optimum set must have the property of generating an $E\hat{P}^\circ$ greater than $E\hat{P}'$ generated by any other set $\{T_i' ; i = 1, 2, \ldots, K\}$. The problem can be posed as a nonlinear programming problem subject to the single constraint of equation 57. The method of gradients appears promising since one can show that P (equation 43) is a monotone increasing function in the hyperspace $\{p_i; i = 1, 2, \ldots, K\}$. Furthermore, one has the algorithm of subsection 3.3.5 to generate the symbolic form for P or $\hat{P}$, i.e.,

$$\hat{P} = f(\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_k) \tag{59}$$

It would be a simple matter to write an algorithm to find:

$$\frac{\partial \hat{P}}{\partial \hat{p}_i} = \frac{\partial}{\partial \hat{p}_i} f(\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_k) \tag{60}$$

One might suppose that equation 60 could be used directly to apportion a certain percentage of V_T to each node according to its influence, $\frac{\partial \hat{P}}{\partial \hat{p}_i}$. Certainly this is a necessary intermediate step. However, it is obvious that the value of $\frac{\partial P}{\partial \hat{p}_i}$ is dependent on the values in the set $\{E\hat{p}_j; j = 1, 2, \ldots, K \mid j \neq i\}$. $E\hat{p}_j$ in turn depends on T_j, S_j. Consequently it is believed that the following recusive algorithm is appropriate to this problem:

(1) Set up $\{A\hat{p}_j; j = 1, 2, \ldots, K\}$

(2) Set up $\{T_j; j = 1, 2, \ldots, K \mid T_j = V_T/K\}$

(3) Solve for $\{S_j; j = 1, 2, \ldots, K\}$ using equation 58

(4) Solve for $\{E\hat{p}_j; j = 1, 2, \ldots, K\}$ using equation 51

(5) Solve for $E\hat{P}$ using (4) and equations 52, 53, and 54

(6) Solve for $\left\{E \frac{\partial \hat{P}}{\partial \hat{p}_j}; j = 1, 2, \ldots, K\right\}$ using equation 60 and the properties used to generate the equations 52, 53, and 54.

(7) Solve for $\{\Delta E\hat{p}_j; j = 1, 2, \ldots, K\}$ where

$$\Delta E\hat{p}_j \stackrel{\bullet}{\equiv} E\hat{p}_j\Big|_{T_j \leftarrow T_j+1} - Ep_j\Big|_{T_j \leftarrow T_j} \tag{61}$$

(See equations 51 and 57.)

(8) Solve for $\{V_j;\ j = 1,\ 2, \ldots,\ K\}$ where

$$V_j = \left(E\frac{\partial \hat{P}}{\partial \hat{p}_j}\right)\left(\Delta E\hat{p}_j\right) \tag{62}$$

(9) Normalize $\{V_j;\ j = 1,\ 2, \ldots\ K\}$ such that

$$\sum_{j=1}^{K} V_j = 1 \tag{63}$$

(10) Set up $\{T_j';\ j = 1,\ 2, \ldots,\ K \mid T_j' = V_j * V_T\}$.

(11) Solve for $E\hat{P}$ using (10)

(12) If $E\hat{P}' > E\hat{P}$

Then $\{T_j \leftarrow T_j';\ j = 1, 2, \ldots,\ K\}$ and go to (3)

If not, $\{T_j^\circ \leftarrow T_j;\ j = 1,\ 2, \ldots,\ K\}$ and stop

This algorithm is at present only a heuristic. It has not been coded and exercised, and no proof with regard to its convergence characteristics is available. Some research on this problem would be valuable both from a theoretical and a practical standpoint. Possible variations on the problem would include:

(1) Account for the fact that all tests may not be of equal cost so that equation 57 would become:

$$V_T = \sum_{i=1}^{K} D_i \tag{64}$$

$$D_i = \lambda_i T_i \tag{65}$$

That is, D_i is the number of dollars spent on the i^{th} node, which allows T_i tests to be performed according to the cost per test, λ_i.

(2) Allow for the more realistic case of multiple users of the network and maximize the sum of the expected network returns over all users.

4.3.3 Optimum Allocation of a Fixed Hardening Budget

This problem, like that of subsection 4.3.2, has not yet been solved. However, a method of solving it seems clear. In fact, the algorithm proposed in subsection 4.3.2 appears to be generally applicable.

The constraint that applies to this problem is the fixed hardening budget. V_T. As before, postulate allocating D_i dollars to the i^{th} node so that:

$$V_T = \sum_{i=1}^{K} D_i \tag{66}$$

It would be required to define the functional relationship between dollars spent on the i^{th} node and the expected value of the node reliability estimate, i.e.,

$$\hat{p}_i = f\ (D_i, \ldots) \tag{67}$$

Some similar functions could be constructed; for example, one can imagine plotting a cost curve for several variations of some specific function module versus its advertised Mean Time Between Failures (MTBF). This module may be a node in some system that may be designed for maximum MTBF at a fixed cost. In general however, one cannot at the present time define cost functions for equation 67. Presume though that such functions will be available in the future. We now sketch their potential usefulness.

Simply stated one desires to allocate V_T over a set $\{D_i;\ i = 1, 2, \ldots, K\}$ so that the resultant $E\hat{P}$ is greater tnan any other $E\hat{P}'$ resulting from any other allocation $\{D_i';\ i = 1, 2, \ldots, K\}$. The author believes that the algorithm of subsection 4.3.2 is applicable to this problem. Some fairly obvious changes are, of course, necessary. For example, in step (1) of the algorithm one needs to initialize with the set $\{E\hat{P}_i;\ i = 1, 2, \ldots, K\}$; where $E\hat{P}_i'$ is the expected value for the reliability estimate of the i^{th} node before any resources are spent on that node. The gradient calculations, steps 7 and 8 would have to be performed according to equation 67, i.e.,

$$d\hat{p}_i = \left(\frac{d}{dD_i} f(D_i, \ldots)\right) dD_i \qquad (68)$$

SECTION V

SOME EXTENSIONS AND THEIR APPLICATIONS

In this section two network analysis algorithms which are extensions to the basic algorithm of subsection 3.3.5 are discussed. The first extension discusses a proposed algorithm which should be an even more efficient technique for finding the upper bound on terminal reliability than the algorithm of subsection 3.3.5. The second extension outlines a recursive algorithm for network analysis. This technique would be extremely valuable in the optimum evolution (in a reliability sense) of a communications system.

5.1 A DIRECT LABELING TECHNIQUE USING CUT

The reader is referred to subsection 3.3.5. Recall that the concept used there was the disjoint partitioning of the event space. The partitioning was carried out by using an algorithm called PATH to discover the single literal which covered the largest possible area in the space. The remainder of the space was then described in terms of the complement of this literal.

Now PATH was designed to find a literal which was favorable to connectivity. It also found, quite by accident, literals which were unfavorable. The dual to PATH, which will be called CUT, can use the same event space partitioning operations as in the algorithm referenced in subsection 3.3.5. Hence the dual to the algorithm using PATH will be essentially the same algorithm but using CUT.

CUT will operate by finding a set of nodes which will disconnect the source and destination nodes. If the reader will recall how PATH operates, it can be seen that CUT is a simple modification to PATH. The modification involves:

(1) Use PATH until the first node is found in the short path. If no node is found the algorithm is through.

(2) Place the first node in the short path in a cut list. Remove the node from the network graph.

(3) Go to (1).

An algorithm using CUT will converge on Q faster than one using PATH, since it deliberately looks for cutsets. Now for some large networks it will not be possible to find the exact solution for either P or Q, using either algorithm, in a reasonably short period of time. Therefore, the utility of having both algorithms lies in the observation that different convergence rates on the upper and lower bound for terminal reliability can be expected from the two algorithms. The applications of the preceding sections can be treated as well in an upper and/or lower bounding sense if it is not possible to force an exact solution for P and Q.

5.2 A RECURSIVE NETWORK ANALYSIS ALGORITHM

Suppose that one is given some network graph and the symbolic solutions (P, Q) for terminal reliability and unreliability of the graph. This can be obtained with the algorithm of subsection 3.3.5. Now it is decided to augment the given graph by adding one new node to the graph and connecting this node to some of the nodes previously in the graph. This creates a new graph and it is desired to find the new terminal reliability and unreliability, P' and Q'. Furthermore, it is decided to take advantage of previous knowledge, P and Q. The solution outline follows.

Denote the given network by $G(\{N\}, \{L\})$ with solutions P and Q.

where $\{N\}$ is the set of nodes and $\{L\}$ is the set of links.

Denote the added node by the index $(n + 1)$ and the added links by the set $\{\ell\}$. All the links in the set $\{\ell\}$ are incident on node $(n + 1)$. The augmented graph will be denoted by $G'(\{N, n + 1\}, \{L, \ell\})$. Using the graph-event space correspondence developed in subsection 3.3, the event space of $G(\{N\}, \{L\})$ is described as G. The event space of $G'(\{N, N + \ell\}, L, \ell\})$ will be called G'.

Now G' can be written as the sum (or union) of two disjoint subspaces.

$$G' \Longleftarrow (G \cap (n + 1)) \cup (G \cap (\overline{n + 1})) \tag{69}$$

where:

$(n + 1) \Longrightarrow$ the $(n + 1)^{st}$ node exists

$(\overline{n + 1}) \Longrightarrow$ the $(n + 1)^{st}$ node does not exist

Now make the identifications:

$$(G \cap (n + 1)) \Longleftrightarrow G'_1 \tag{70}$$

$$(G \cap (\overline{n + 1})) \Longleftrightarrow G'_2 \tag{71}$$

Consider the network implied by G_2' . This is nothing more than the given network since the $(n + 1)^{st}$ node is assumed not to exist. Therefore, the literal for all events favorable to connectivity in G_2' is found from the solution to the given graph intersected with the assumption $(\overline{n + 1})$; i.e.,

$$P_2' \Longleftarrow P \cap (\overline{n + 1}) \tag{72}$$

or

$$P_2' \Longleftarrow P\ q_{n+1}$$

Consider the network implied by G_1'. This is the given network with the added node (assumed to exist) and incident links. The literal for all events favorable to connectivity in G_1' is found from P intersected with the assumption $(n + 1)$; plus the literal for all events in G_1' which strictly depend on the assumption $(n + 1)$, i.e.,

$$P_1' \Longleftarrow P\ p_{n+1} + (E - E \cap P) \tag{74}$$

where:

E The literal(s) for those events favorable to connectivity in G_1' which depend on node $(n + 1)$ existing.

$E \cap P$ The possible overlap events in the subspaces P and E.

Now, since G_1' and G_2' are disjoint:

$$P' = P_1' + P_2' \tag{75}$$

and the solution for the graph G' $(\{N, n + 1\}, \{L, \ell\})$ is:

$$P' = P\ p_{n+1} + (E - E \cap P) + P\ q_{n+1} \tag{76}$$

or

$$P' = P + E - E \cap P \tag{77}$$

since

$$p_{n+1} + q_{n+1} = 1 \tag{78}$$

Equation 77 can be rewritten as:

$$P' = P + E \cap (1 - P) \tag{79}$$

or

$$P' = P + E \cap Q \tag{80}$$

P and Q are both known. Therefore find only the literal(s) for E. This point will be returned to later. Now consider Q′.

The solution for Q′ is found from equation 80 and the observation that:

$$P' + Q' = 1 \tag{81}$$

From equations 81 and 80

$$Q' = 1 - P' = 1 - P - E \cap Q = Q - E \cap Q \tag{82}$$

This is not a convenient form, as allowance must be made for the possible case of a lower bound solution for Q. If this should occur, equation 82 will not subtract all the overlap. Hence, rewrite equation 82 as:

$$Q' = Q \cap (1 - E) = Q \cap \overline{E} \tag{83}$$

Now consider E and $\overline{E}$. As stated before E is the literal(s) for favorable event(s) in the graph G′ ({N, n + 1}, {L, ℓ}) which depend on node (n + 1). E and $\overline{E}$ can be found from the same basic algorithm as in subsection 3.3.5. The only difference is that the short path sought by PATH is constrained always to contain node (n + 1). Consider an example:

Given the graph in figure 34 with solutions:

$$P = p_1p_2+q_1p_3p_4+p_1q_2p_3p_4$$

$$Q = q_1q_3+q_1p_3q_4+p_1q_2q_3+p_1q_2p_3q_4$$

Find the solution to the graph in figure 35. E depends on node 5. The first literal for E is the short path through 5 or:

$$E_1 \Longleftrightarrow \{1\ 4\ 5\}$$

$\overline{E}_1$, the remainder of the event space which must still be investigated is:

$$\{\overline{1\ 4\ 5}\} \Longrightarrow \{\overline{1}\} + \{1\ \overline{4}\} + \{1\ 4\ \overline{5}\}$$

Investigating $\{\overline{1}\}$ results in the second literal for E:

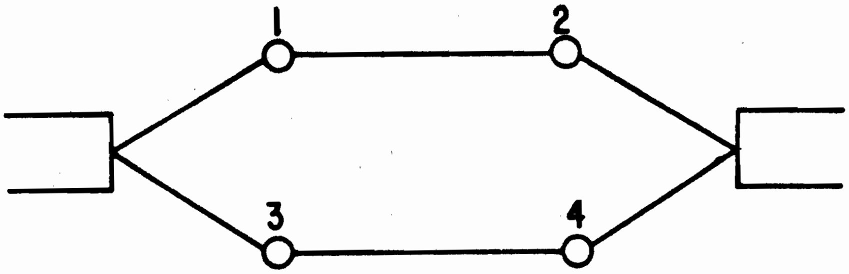


Figure 34. A Network Graph G ({N}, {L}).

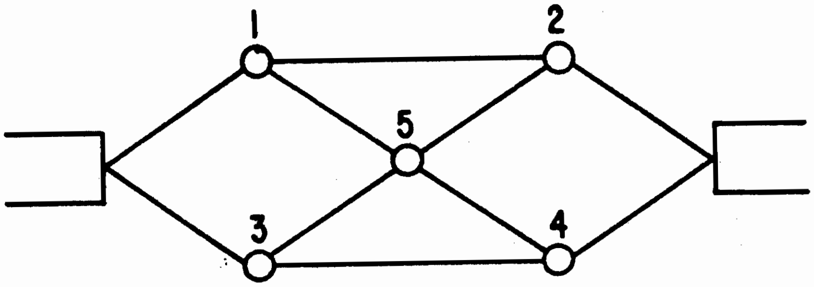


Figure 35. The Augmented Graph G´ ({N, 5}, {L, ℓ}).

$$E_2 \Longleftrightarrow \{\bar{1}\ 2\ 3\ 5\}$$

Investigating $\{1\ \bar{4}\}$ results in a negative finding. (There is a path but it does not depend on 5.) Therefore, the first literal for $\bar{E}$ is:

$$\bar{E}_1 \Longleftrightarrow \{1\ \bar{4}\}$$

Investigating $\{1\ 4\ \bar{5}\}$ results in the second literal for $\bar{E}$;

$$\bar{E}_2 \Longleftrightarrow \{1\ 4\ \bar{5}\}$$

The remainder of the event space is described by those events in $\{\bar{1}\}$ and in $\overline{\{2\ 3\ 5\}}$ (See E_2.). Therefore the remainder of the event space to be investigated is:

$$\{\bar{1}\ \bar{2}\} + \{\bar{1}\ 2\ \bar{3}\} + \{\bar{1}\ 2\ 3\ \bar{5}\}$$

Investigating each of these results in:

$$\bar{E}_3 \Longleftrightarrow \{\bar{1}\ \bar{2}\}$$

$$\bar{E}_4 \Longleftrightarrow \{\bar{1}\ 2\ \bar{3}\}$$

$$\bar{E}_5 \Longleftrightarrow \{\bar{1}\ 2\ 3\ \bar{5}\}$$

To sum up:

$$E \Longrightarrow \{1\ 4\ 5\} + \{\bar{1}\ 2\ 3\ 5\}$$

$$\bar{E} \Longrightarrow \{1\ \bar{4}\} + \{1\ 4\ \bar{5}\} + \{\bar{1}\ \bar{2}\} + \{\bar{1}\ 2\ \bar{3}\} + \{\bar{1}\ 2\ 3\ \bar{5}\}$$

Thus, using equations 80 and 83

$$P' = p_1p_2+q_1p_3p_4+p_1q_2p_3p_4+p_1q_2q_3p_4p_5+q_1p_2p_3q_4p_5$$

$$Q' = p_1q_2q_3q_4+p_1q_2p_3q_4+p_1q_2q_3p_4q_5+q_1q_2q_3+q_1q_2p_3q_4$$
$$+q_1p_2q_3+q_1p_2p_3q_4q_5$$

The procedure exposed here can very readily be turned into a computer algorithm. This however, will be left to future research. Such an algorithm would be extremely valuable in any situation where a communications network is evolving

evolving (as they always are). The utility, of course, is that in a one-node-at-a-time evolution the new node placement can always be chosen to maximize the reliability of the resultant graph. This would be accomplished by evaluating the present graph, then trying the new node at several locations to determine the best placement. Each placement would require an individual solution, but each solution would be extremely fast since it draws on the solution to the present graph.

SECTION VI

CONCLUSIONS

The basic technology required to assess the reliability of a system in a hostile environment has been exposed in this report. Network analysis is the foundation of the assessment technique. Support efforts involving testing and/or analysis are used to perform node assessments. The direct testing approach is covered in reference 1. Analytical and hybrid approaches are discussed only briefly herein; detailed treatment can be found in references 3 and 4.

The technology discussed herein is applicable to the reliability assessment of any system which can be given a network representation. It happens, however, that this research was motivated by a need to assess the reliability of USAF Command, Control, and Communications (C^3) in an electromagnetic pulse (EMP) environment. Consequently some concluding remarks on this subject are made.

One could, both in principle and practice, perform an EMP assessment today of the USAF C^3 system using the technology herein and that of references 1, 3, and 4 for node assessment. However, it would be imprudent to do so without first determining the economic impact of implementing such an assessment. The cost of performing such an assessment is not clear at this time. What is clear, however, is that the C^3 system used by the USAF represents a capital investment of tens of billions of dollars. This system in turn protects and controls a weapons system investment which is likewise a resource valued at tens of billions of dollars. Clearly one must consider an assessment program on such a system even though it may cost large sums of money. Assurance is needed, before beginning such a program, that the potential return justifies the investment.

As a consequence, the latter subsections of this report (4.3.2, 4.3.3, and 5.2) were devoted to theoretical problems in optimizing the allocation of fixed budgets for assessment/hardening/design problems. These are difficult problems which merit considerable research both on academic and practical grounds. It is regretted that only partial solutions and heuristic approaches to these problems could be offered at this time.

In spite of the present open-endedness of the optimization aspects of the technology discussed, it should be clear that immediate application is possible on smaller systems; that is, systems not continental in size. In fact, for smaller systems, this technology could materially reduce present assessment program costs.

APPENDIX A

AIR FORCE WEAPONS LABORATORY IN-HOUSE MODELLING PLANS

William, P. Dotson, Jr., Captain, USAF

This appendix is a transcript of a speech delivered at the Air Force Weapons Laboratory (AFWL) Command, Control, and Communications (C^3) Program Review Meeting on 21 June 1972.

Good morning. My name is Bill Dotson. My job with the C^3 group at the Air Force Weapons Lab is to try to find ways to assess the effects of high-altitude electromagnetic pulses (EMP) on the ground-based portion of Air Force Command, Control, and Communications (C^3) Networks.

It's an extremely large job but we have made some progress. For example, it is now generally accepted that it would not be cost effective to assess the ground-based C^3 networks with the simulator shown in figure A.1.

Headquarters USAF has levied a requirement on the Air Force Weapons Lab to do this job, though. It now becomes incumbent on us not only to figure out how to do the job but exactly what the job is. What does it mean to assess a C^3 network?

What I shall do this morning then is tell you what C^3 assessment means to the C^3 group and what our in-house modelling plans are to do this assessment.

To determine what C^3 assessment means we can begin by considering three critical missions of C^3. These are:

1. Communications for tactical warning

2. Communications for USAF support of the NCA

3. Communications to implement USAF portions of the SIOP options.

What we are interested in, of course, is whether or not these missions can be performed in a hostile environment. Or perhaps it would be better to say that we are interested in the reliability of performing these missions.

These missions will be carred out by passing information through a network of communications systems. Whether the information is digital or analog we would be interested in: (a) can the information arrive at the intended destination?

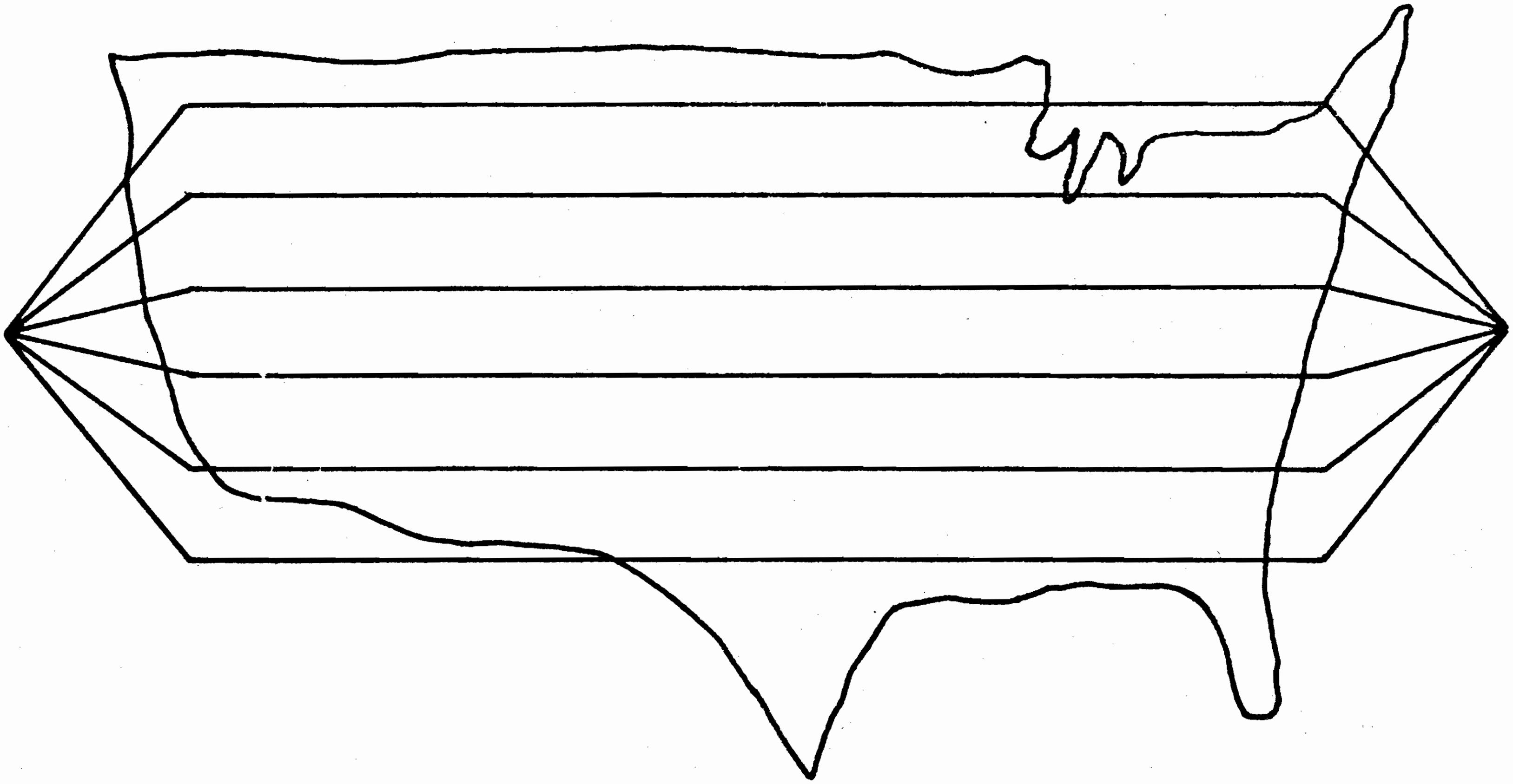


Figure A.1. From the Files of Rube Goldberg.

(or the probability that it can), and such things as (b) bit error rate (BER), signal-to-noise ratio (SNR), is the message intact? Will it require retransmission? Is retransmission possible? What time delays are involved?

So, when we speak of assessing C^3 networks on the basis of the probability of being able to perform some critical mission, we can just as well speak of those functions a communications network must perform in order to fulfill that mission. What we are interested in then is:

1. The reliability of a C^3 network in establishing connectivity between specified users.

2. The "quality" of the information transfer, i.e., SNR, BER, time delays, retransmissions, etc. To assess a C^3 network then we go through these steps: (1) define the mission, (2) define the network, and (3) define the functions to be performed in support of the mission.

In measuring the degree of success with which a network fulfills its mission we have decided that what must be measured is the probability that those functions, necessary for mission accomplishment, are performed. We believe that this is handled in two phases, failure and degradation; or another way of putting it: (1) Analyze the reliability of establishing connectivity between specified users, and (2) Analyze the quality of information transfer.

At the present time we are restricting ourselves to the first phase--trying to assess the probability that a C^3 network will be able to establish connecttivity. In doing this, the equipment of the network must be defined, and a decision must be made on what level to base the assessment. Bearing in mind that our objective is to perform an assessment at the network level we define these levels of complexity.

1. Component - a device such as a resistor, capacitor, inductor, tube, transistor, etc. Clearly it would be an impossible task to go directly from component data to a network analysis.

2. Black Box - a collection of components designed to perform some function, such as power supply, or to provide a desired input-output characteristic. Examples would be a receiver, transmitter, modem, etc.

Again, if we consider the scope of our final objective, assessment of a network, it is clear that the amount of data required for the assessment, if done directly from the black box level, would be crushing. As an aside, we should note that work has been done which indicates the feasibility of determining the functional reliability of a black box from the reliability of its components. We believe that an extension of this type of approach is vital to success in network assessment. We should learn how to predict the reliability of a subsystem from reliability data on its black boxes.

3. Subsystem - a collection of black boxes designed to perform some higher order function such as computation, computer interface, switching for system self-healing, etc.

Even at this level there would be an enormous amount of data to be worked into a network analysis, and while it might be feasible, we believe that it would be advantageous to work from the next level.

4. System - a collection of subsystems performing a variety of functions resulting in two major functional groups (from a communications point of view).

a. Switching to route information from one point to another, and

b. Transmitting, or carrying, information between points.

An example of the first would be a switching center; examples of the second would be land lines and microwave links.

5. Network - a collection of systems designed to transmit information between users in the network. Examples would be AUTOVON, AUTODIN, etc.

A network might look like this (figure A.2).

The systems in the network are nodes and links (or switching centers and trunk lines).

At this point we should restate our <u>initial</u> C^3 assessment objective--we want to determine the probability that the network provides connectivity between specified users (or nodes) in the network when the network is built from less than perfectly reliable systems (nodes and links).

We recognize that in this form C^3 assessment is so oversimplified that we can actually find a quantitative measure of network reliability. And, while this is a long way from the final answer on C^3 assessment, it does provide us with a starting point and a framework to build on.

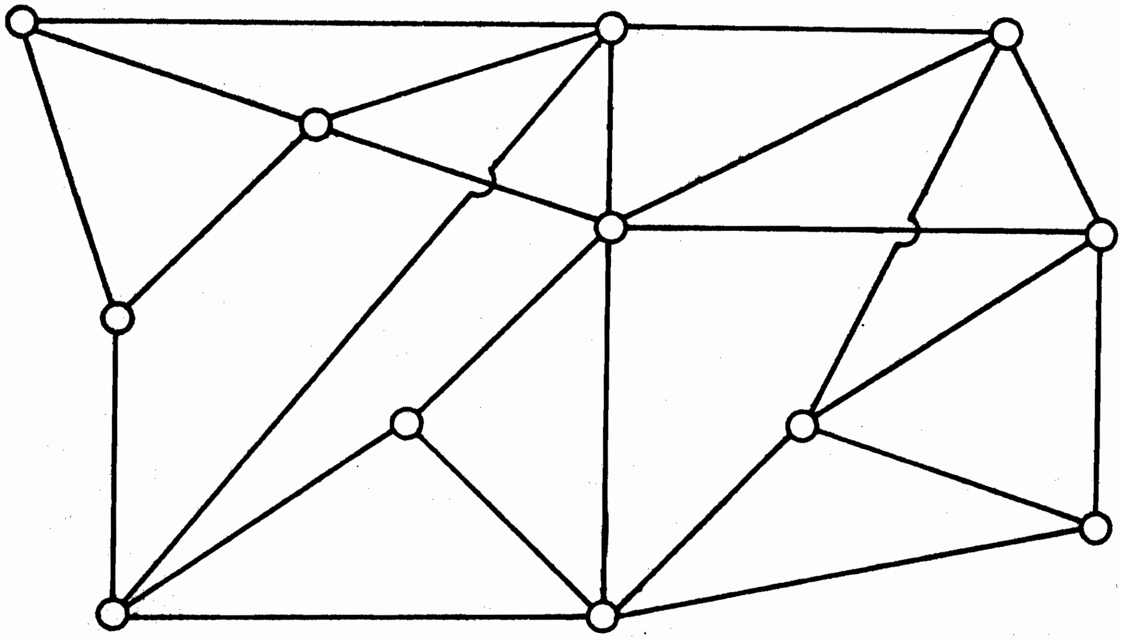


Figure A.2. A Network Graph

With this as background let me tell you what we presently have working in the way of a network model. After that it will be easier to explain what our future plans for in-house modelling are. Conceptually, the model looks like this (figure A.3). The heart of the model is a routing algorithm which uses data on the network topology, reliability of systems within the network, and the function or functions the network must perform, to determine the network reliability in performing those functions.

The network data is entered into the model on a node-by-node basis. For each node in the network we enter this data.

1. Its geographical location (in matrix coordinates). The matrix coordinates are used for compactness and simplicity in reading data into the model. What we do is break a large geographical area up into a grid pattern, then code each grid element with a matrix I.D.

2. The geographical location of each node in the network to which it normally has a direct connection.

3. The "class" of the link connecting the nodes.

4. The "class" of each node to which the node is connected.

The connections, or links, between nodes may be either bilateral or unilateral. The "class" referred to is a slot deliberately left open for later

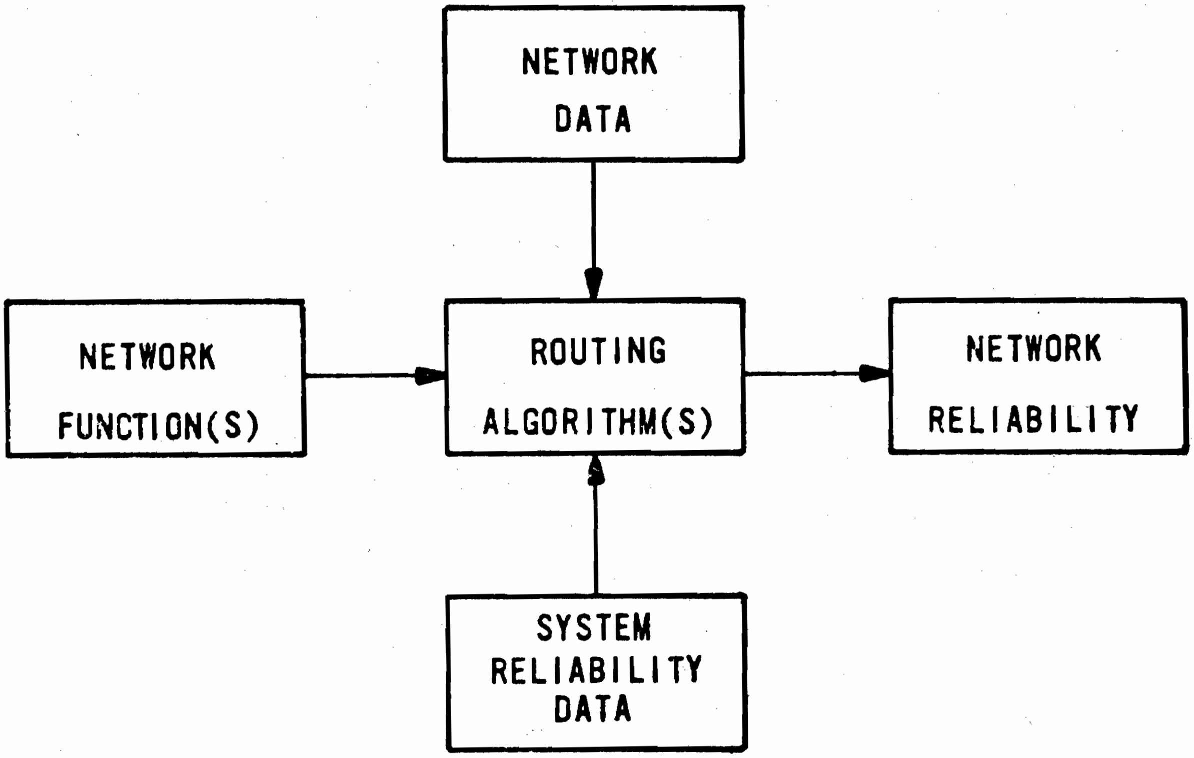


Figure A.3. Network Model.

use in the model. It will allow us, if necessary, to specify a reliability for each system in the network on an individual or group basis. A simple example of this would be if we found that microwave links had a different reliability than buried coaxial links.

System reliability data is a number, or set of numbers, interpreted to be the probability of functional failure of the systems within the network. This can be entered into the model as a function of some other variable if desired. For example we might tell the model that a node of class two will have a probability of failure versus external E-field amplitude as shown here (figure A.4). A node falling in a different class would have a different reliability curve. The system reliability data is used to control the physical degradation of the network. For instance, if the model were told that all nodes in the network have a 0.5 probability of failure, then the model would, on a random basis, degrade the network by denying the routing algorithm the use of some nodes in performing the network functions.

The network function is straightforward. The model is given an origination node, a destination node, and the task of establishing connectivity between them.

The routing algorithm contains the logic used in searching through the network to establish a path between these two nodes. If the node to which a call would normally progress in the next step of establishing a path is not functioning, or the link has been broken, the routing algorithm makes alternate choices if possible.

The result of all this is that we find what the network reliability, or probability that it will be able to perform the network function, is. As one might expect, network reliability, depends on the topology of the network, the specific function to be performed, how "smart" the routing algorithm is, what the probabilities of failure of the systems within the network are, and other physical limiations on the network, such as the maximum allowable number of tandem links that may be used to establish connectivity.

I should mention that the model is intended to be somewhat flexible. It's a very simple thing to change the network function; the system reliabilities can be changed easily; a different network topology means punching a different data deck; and the routing algorithm is written so that it can be modified to a minor degree very easily or replaced in its entirety if necessary.

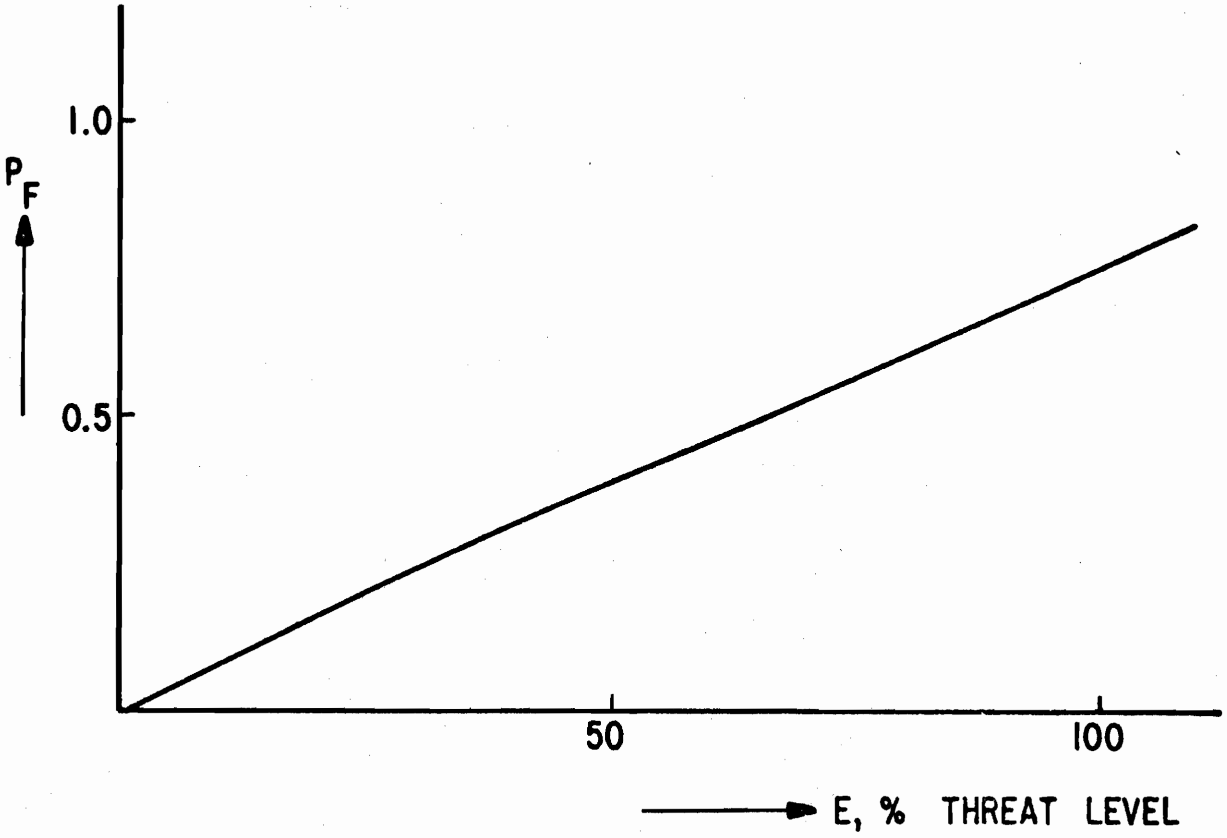


Figure A.4. System Reliability Data
$p_f = g(E, \text{type}, \text{class})$.

Let's take a look at a specific network now and see what the model can tell us about its reliability and perhaps what we could most effectively do next in a C^3 assessment and hardening program.

This a graph of the AUTOmatic Blast Analysis Network (AUTOBAN) (figure A.5). The network is fictitious. Any resemblance between this network and any real network is intentional, but we make no claims concerning the validity of applying results of simulation runs on this network to any real world network.

The AUTOBAN network consists of 58 nodes (or switching centers) located at various points in the Continental United States and about 400 links (or trunk lines) connecting the switching centers in a polygrid pattern.

The routing philosophy used in establishing connectivity in the network tries for direct routing if possible, forward routing as a next choice, and lateral routing if necessary. This results in a given switching center having from six to nine choices of centers for transferring a particular call to. By a slight modification to the routing algorithm we can permit a call to be routed

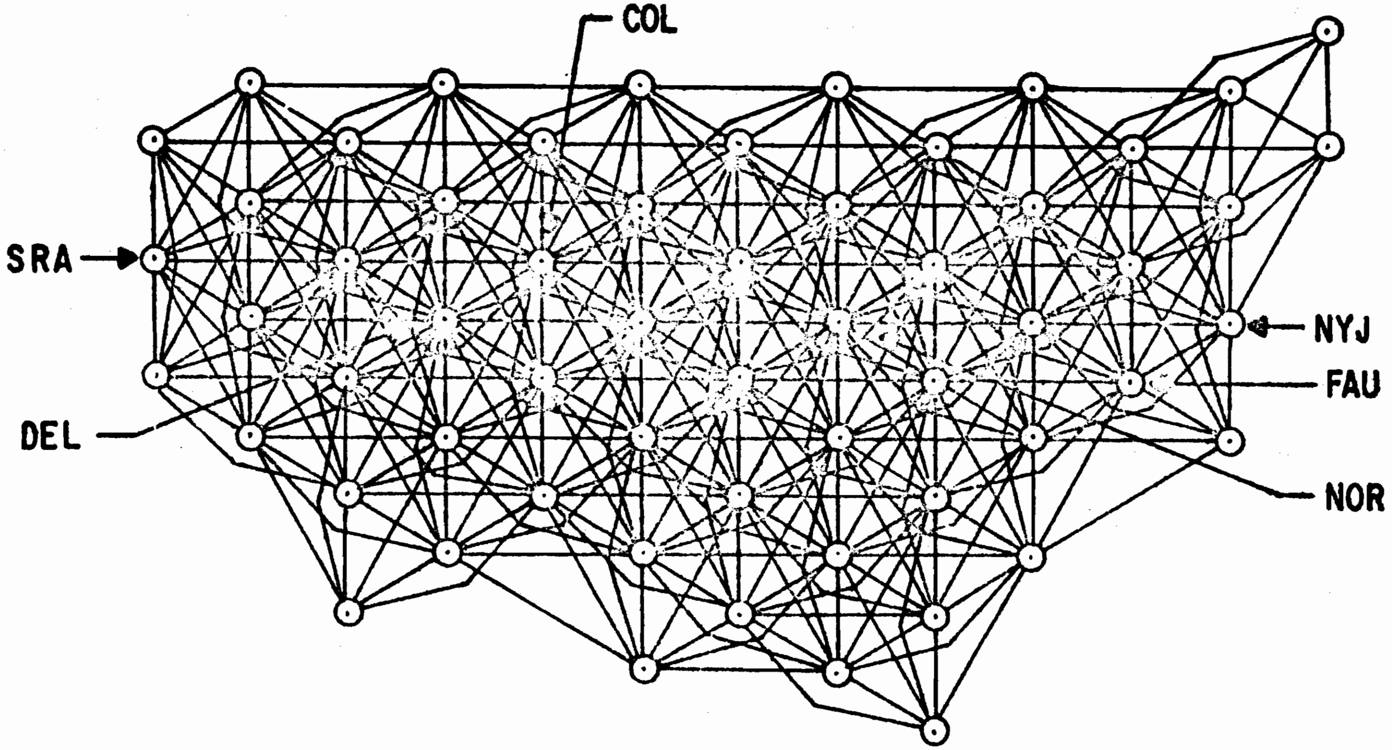


Figure A.5. Automatic Blast Analysis Network

in a backward direction if none of the other choices are possible. This increases a switching center's options to about fourteen in most cases.

There are also restrictions in the routing process. Spill forward routing is used, and shuttle is prohibited. That is, if a call has been transferred from center A to center B and it develops that B is blocked, the call cannot be sent back to A. There is a maximum total allowable number of trunks in tandem that can be used to establish connectivity. When a call has progressed to being within a certain distance of the destination center there is a separate limitation applied to the allowable number of tandem trunks. These are limitations applied to conserve network capacity.

The routing algorithm is also designed to help conserve network capacity by tying up as few trunks as possible in placing any one call. The basic objective function of the routing algorithm is essentially like this: At whatever switching center the call is, at an instant in time, the algorithm must choose the next switching center to which the call will be passed in such a manner that the distance remaining to the destination center is minimized.

Let's look at some results from the model. Suppose that one of our critical C^3 missions requires that our fictitious network be able to establish connectivity between Faulkner and Colorado switching centers. The reliability with which the network can perform this function is shown here (figure A.6) as being dependent on the reliabilities of the intermediate systems within the network. (Orgination and destination centers are considered perfectly hard). Other variables, such as routing philosophy, are implicit in the problem; or assumed away, such as traffic--here we assume that the call is of sufficient priority to seize whatever network resources are required to establish connectivity.

This graph tells us primarily one thing. If, say, a network reliability of 95 percent is considered by the mission planners as being an acceptable trade-off point between reliability and the cost to achieve it, then the systems within the particular network now have a firm reliability requirement to meet: 70 percent reliability or 30 percent failure probability. It is not necessary to keep on arbitrarily hardening the systems. We now have an aid in performing a marginal analysis of the cost effectiveness of hardening to a given point.

On the other hand, such results could also serve as an aid in determining the amount of effort to be expended in finding out what the actual system reliability is. To illustrate this, let's suppose that by expending X amount of

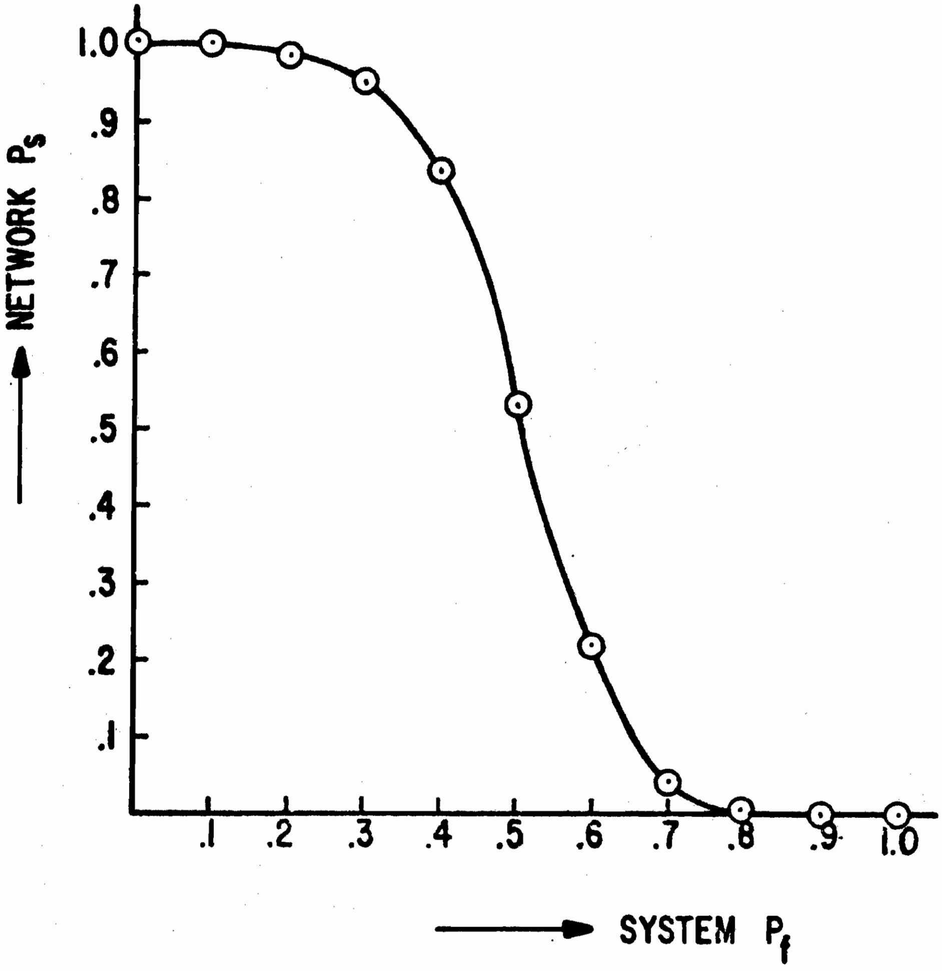


Figure A.6. Network P_s Versus System P_f, Nodes and Links FAU to COL

effort we have determined that the system reliability is 80 ± 20 percent. In other words, we're not certain just what the system reliability is, just that it's somewhere between 60 and 100 percent. If it's 60 percent our network reliability is 83 percent--which does not meet our postulated 95 percent reliability requirement. So we dump another delta of effort into system assessment and come back with a revised estimate on system reliability of say, 82 ± 12 percent. This time the lower bound on system reliability of 70 percent, indicates that we can meet out network reliability requirement. Further effort on system assessment would reduce our error bar still more--but would yield diminishing returns in terms of network reliability assessment.

Another question we can ask the model runs like this: Suppose that of the two types of systems in the network, centers and trunks, it is only possible to harden one type. Which would yield the greatest return in increased network reliability? These curves (figure A.7) indicate a possible answer to that question. One curve shows results for network reliability with perfectly "hard" centers and "soft" trunks. The other is for perfectly "hard" trunks and "soft" centers. We can see that it is more effective to harden the trunks in our fictitious network. This might indicate a preference in new installations for buried coax over microwave links. That of course is not the complete answer. We could also need to figure the cost of each option, but we can get the marginal return in terms of an increase in network reliability for each option.

Other, more sophisticated questions are possible. Our next speaker, Dr. Frank, will tell you (among other things) about minimum cut-sets. The use of minimum cut-sets, from an aggressor point of view, is to find the most cost-effective location(s) in the network to break and deny you mission performance. From our point of view, the same location(s) would be the most cost-effective in terms of resource allocation to harden the network and assure mission performance.

In a completely different vein, we can also ask such questions as: Are there any changes in network operating philosophy that we could make to improve its reliability without hardening the systems within the network?

This next graph (figure A.8) is representative of an answer to such a question. Here we vary the maximum allowable number of tandem trunks permissible in establishing connectivity while holding other variables constant.

We can see that allowing eight trunks in tandem results in better network reliability than seven, but going beyond eight we get swiftly diminishing returns. In fairness, we should mention that allowing eight trunks in tandem rather than seven would decrease the amount of traffic the network could handle. The next question then is: By how much and is it worth it? Is there still enough capacity to handle those mission critical calls in a wartime situation?

Changing the routing philosophy used in the network can also result in an increase in reliability. In figure A.9 we have results for three different routing philosophies. We can see that allowing backward routing, if absolutely necessary, results in an increase in network reliability without hardening the systems within the network.

Unfortunately there is one gaping hole in any network reliability assessment we can make today. And that is, we simply don't know what the system reliabilities are or how to measure them without destructive testing of large numbers of

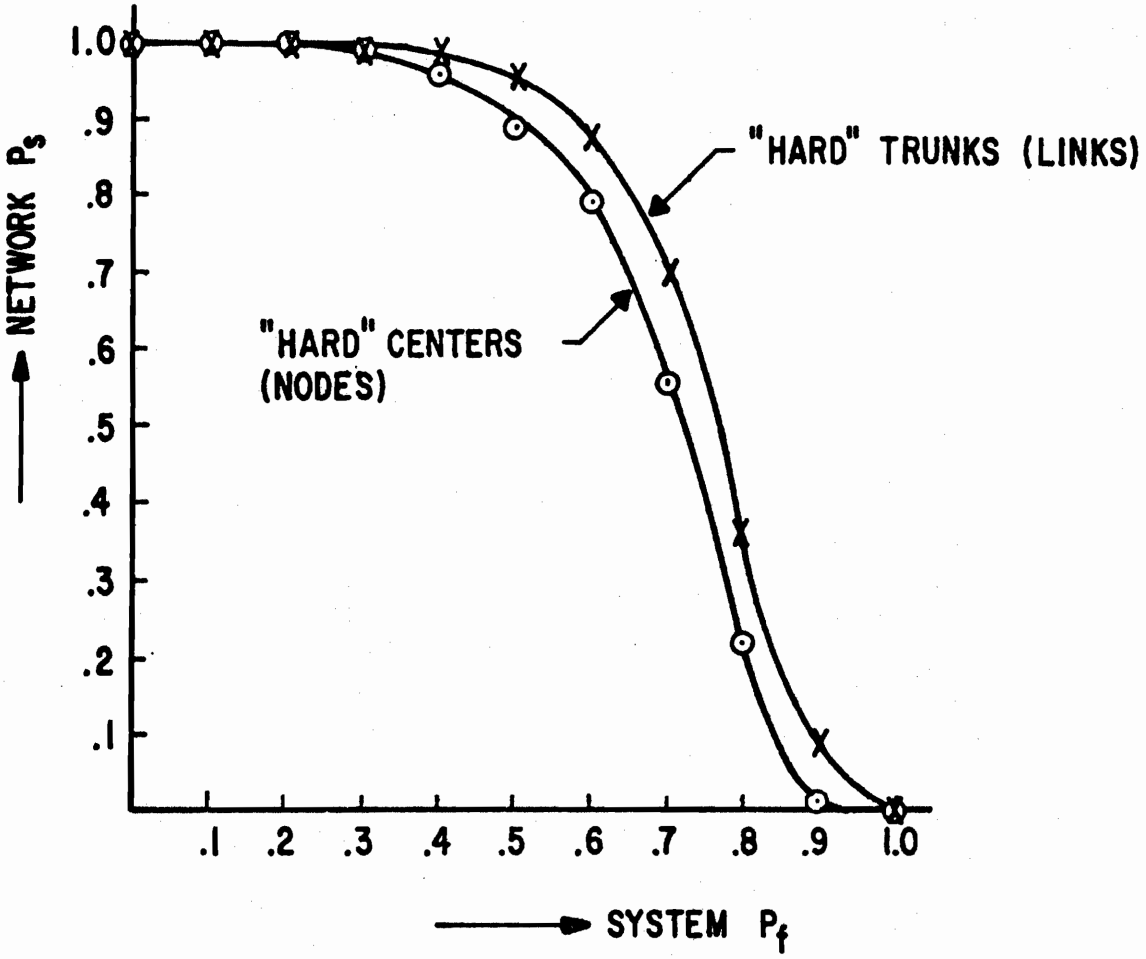


Figure A.7. Network P_s Versus System P_f, Nodes or Links (FAU to COL)

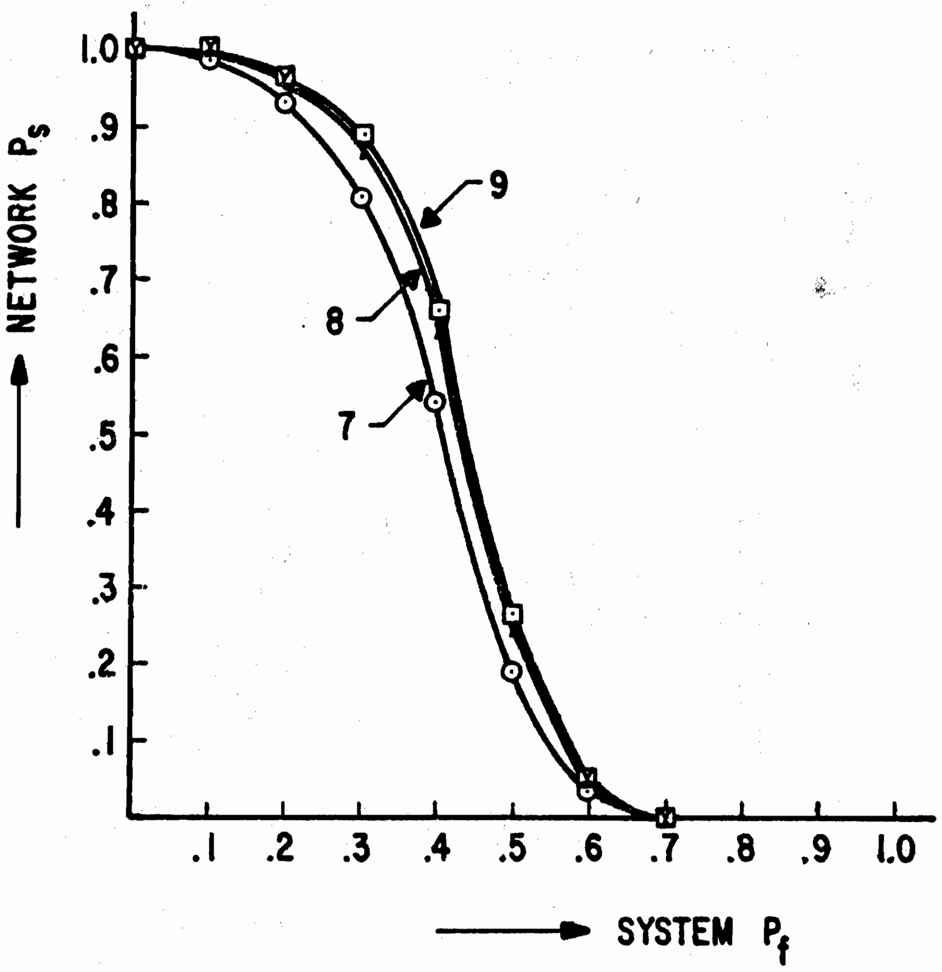


Figure A.8. Network P_s Versus System P_f and Allowable Number of Tandem Trunks (links) SRA to NYJ.

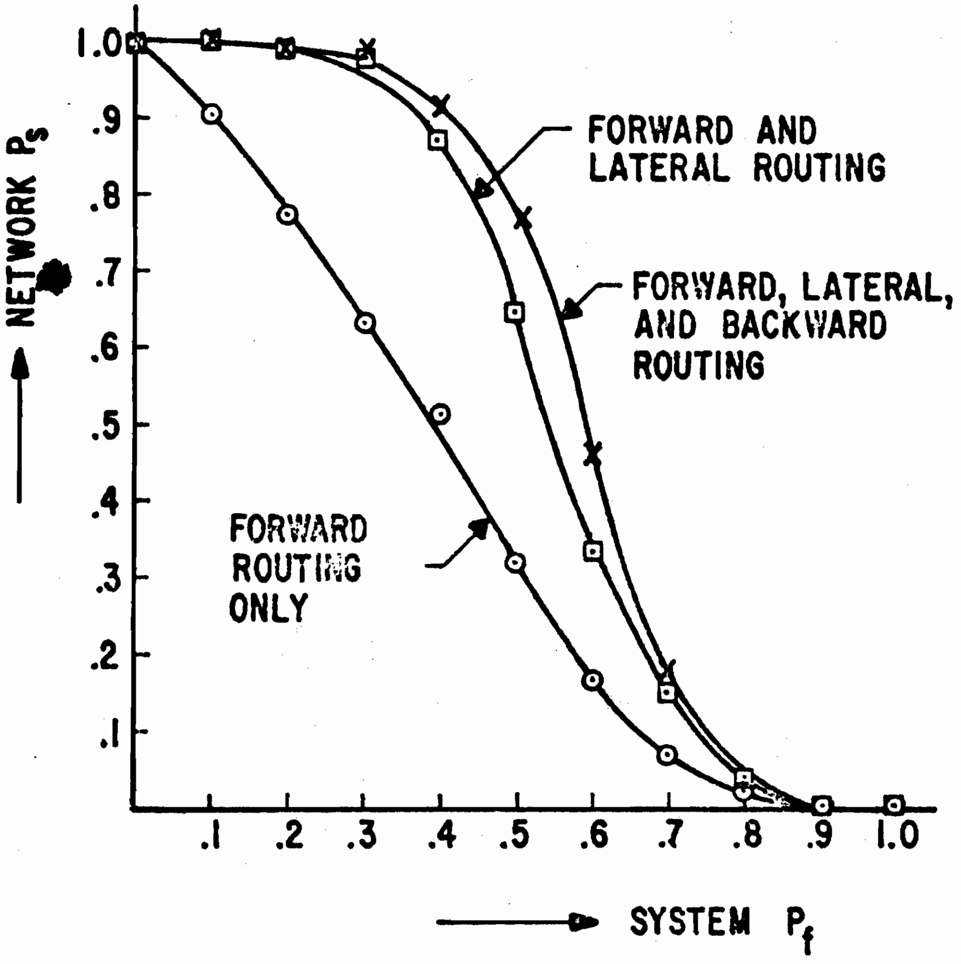


Figure A.9. Network P_s Versus System P_f and Routing Philosophy DEL to NOR.

systems. A network model can only tell us how reliable the systems must be to perform mission critical C^3 functions; or it can tell us what the network reliability is if, and only if, we can input to the model what the reliabilities of the systems within the network are.

This will be a major part of our future in-house plans-- an effort to close this gap--to find effective ways of assessing system reliabilities. A qualitative statement that a system is pretty hard or pretty soft just won't cut it. Our network modelling can help us in this effort by telling us the form system reliability data should take.

What we would like is to determine the probability that a given system will be able to perform its intended function, such as switching or carrying information, as a function of external field amplitude. We recognize that it's really not that simple. There are many variables in an EMP assessment. In reality the system reliability would be a function of many parameters, such as: E-field amplitude, direction of incidence, polarization, the system geometry, shielding, subsystem thresholds, cabling, points of entry, etc. We suggest though, that it should be possible to "worst-case" such variables as direction of incidence and polarization; and to make implicit the variables under control of the designers such as shielding, cabling, and points of entry.

What we really want to do then is to tie together functional reliability and external field amplitude as focal points for a large number of investigators, each of which would be doing assessment in his area of expertise. In this manner we would have a common point in disseminating data without getting buried in it, while the individual assessor could retain control and use of those implicit variables which are important in his particular area.

The "big picture" we have in mind for C^3 network assessment is essentially a layering of the problem so that a hierarchial approach is established (figure A.10). One would be taking a "macroscopic" to "microscopic" view of the problem depending on where he was in the loop.

Those working at the network level would be dependent on systems level people for reliability data to input to their studies. The network level people would determine the necessary form for their input data, such as system reliability as a function of EMP level; and the systems level people would make the determination on techniques to be applied and what inputs they would require.

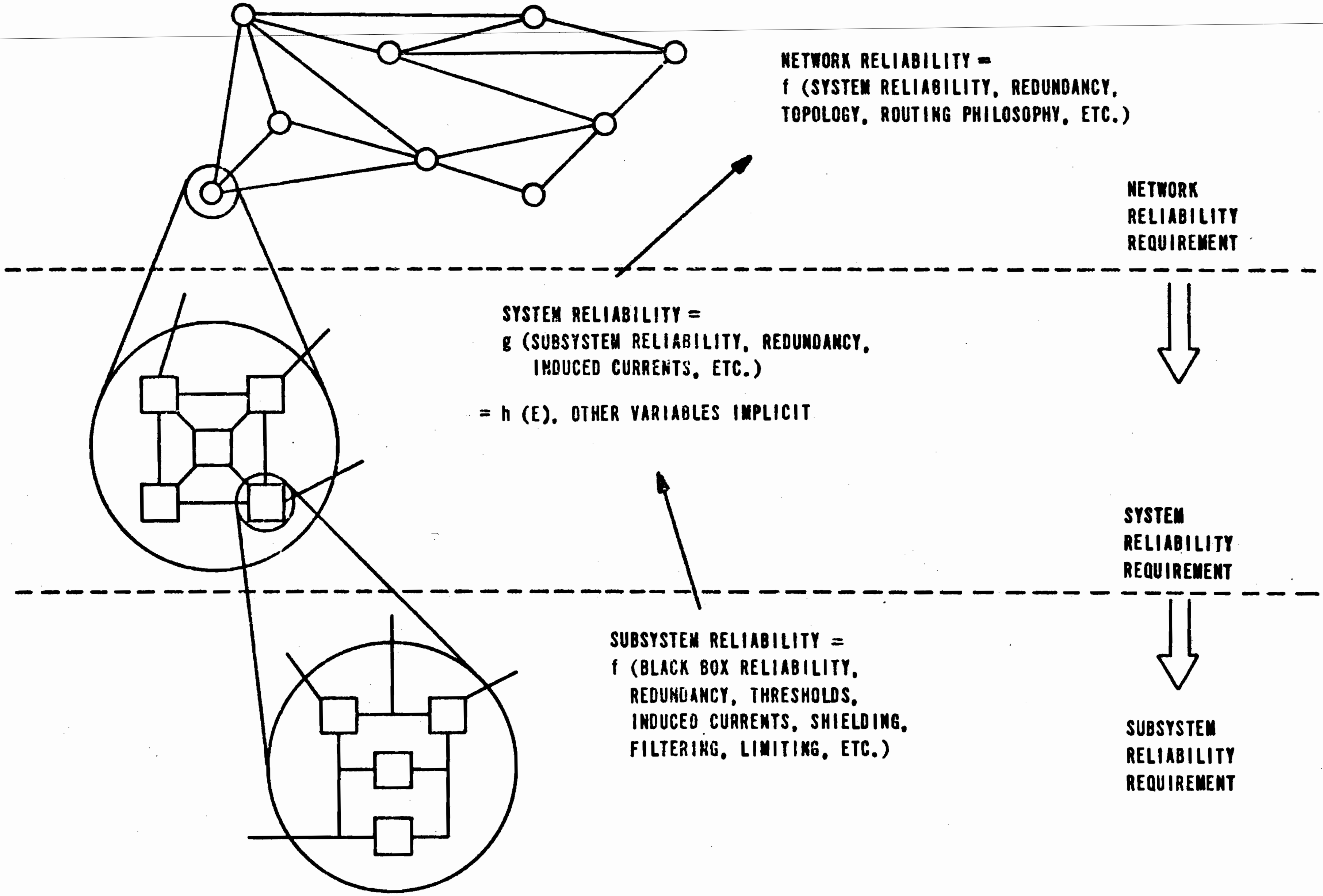

NETWORK RELIABILITY =
f (SYSTEM RELIABILITY, REDUNDANCY,
TOPOLOGY, ROUTING PHILOSOPHY, ETC.)
NETWORK
RELIABILITY
REQUIREMENT
SYSTEM RELIABILITY =
g (SUBSYSTEM RELIABILITY, REDUNDANCY,
INDUCED CURRENTS, ETC.)
= h (E), OTHER VARIABLES IMPLICIT
SYSTEM
RELIABILITY
REQUIREMENT
SUBSYSTEM RELIABILITY =
f (BLACK BOX RELIABILITY,
REDUNDANCY, THRESHOLDS,
INDUCED CURRENTS, SHIELDING,
FILTERING, LIMITING, ETC.)
SUBSYSTEM
RELIABILITY
REQUIREMENT

Figure A.10. C^3 Assessment Methodology.

Inputs to the systems level people would be outputs from those working at the subsystem level. Here again the systems level could best determine what form their input should be in, and the subsystem level determines how best to do it.

The process also allows for cost-effective hardening. Network reliability requirements that cannot be effectively met by operational changes at the network level can be translated into minimum system reliability requirements through network simulation techniques.

Since the systems level people are in the business of assessing system reliabilities in terms of field amplitude and implicit variables at the system level; they have only to work the problem from the other end to meet a reliability requirement. A similar situation would hold at the next level--the subsystem.

We have oversimplified the problem. We recognize that it's an extremely complex interdisciplinary problem, and this is part of the problem. The fact that it is interdisciplinary will require us to adopt some sort of methodology that will allow for good and meaningful information flow among all the parties involved.

So what we have attempted to do is outline the problem, from what assessment means to us to a thumbnail sketch of how we believe a network assessment can be accomplished.

We think that the problem is a critical one, and one of the most challenging we've ever seen, and we recognize that there are a great many uncertainties in EMP assessment at any level of complexity, from blackbox to network. Because of this we think that the best approach to the problem today is to admit our uncertainties in the form of statistical reliability data and get on with an analysis even if we do have to live with large error bars for now. After all, in the words of Josh Billings "It ain't what a man don't know that makes him a fool. It's what he does know that ain't so."

REFERENCES

1. Ashley, C., "Confidence and Reliability in an Infinite Population," System Design and Assessment Notes, Note 3, 7 October 1971.

2. Conference Proceedings, Component Degradation from Transient Inputs, U.S. Army, MERDC-X-3, April 1970.

3. Analysis of Communications Systems, Air Force Weapons Laboratory Technical Report to be published.

4. Leased Circuit Assessment, A Status Report, Air Force Weapons Laboratory Technical Report to be published.

5. VanSlyke, R. and Frank, H., "Network Reliability Analysis: Part 1," Networks, Vol. 1, No. 3, pgs 279-290, 1972.

6. Frank, H. and Frisch, I. T., Communication, Transmission and Transportation Networks, Reading, Massachusetts, Addison-Wesley Publishing Company, 1971.

7. Kroft, D., "All Paths Through a Maze," Proceedings of the IEEE, Vol. 55, pgs 88 - 90, January 1967.

8. Brown, D. B., "A Computerized Algorithm for Determining the Reliability of Redundant Configurations," IEEE Transactions on Reliability, Vol. R-20, No. 3, pgs 121 - 124, August 1971.

9. Apostol, T. M., Calculus of Several Variables With Applications to Probability and Vector Analysis, Volume II, New York, New York, Blaisdell Publishing Co., 1962.

10. Fratta, L. and Montanari, U. G., "A Boolean Algebra Method for Computing the Terminal Reliability in a Communication Network," IEEE Transactions on Circuit Theory, Vol. CT-20, No. 3, pgs 203 - 211, May 1973.

11. Krakowski, M., "Some Considerations in Assignment of Values to Elements of Communications Networks," a working paper for the U.S. Department of Commerce, National Bureau of Standards, September 1967.

12. "Valuation of Telecommunications," National Bureau of Standards Report 9977; Progress Report for FY68.

13. Dwass, M., Probability and Statistics, New York, New York, W. A. Benjamin, Inc., 1970.

14. Montgomery Phister, Jr., Logical Design of Digital Computers, Chapter 4, "The Simplification of Boolean Functions and the Veitch Diagram Simplification Method," John Wiley & Sons, Inc., 1961.