Derivation and use of a survivability criterion for DDP systems*

by RICHARD E. MERWIN and MOHAMMED MIRHAKAK

The George Washington University
Washington D.C.

INTRODUCTION

With the advent of low cost compact computing systems, there has been a natural tendency to bring the computer to the job rather than the job to the computer. The geographical distribution of computing also leads to a necessity to interconnect installations so that they can share data, provide back up computing support, and permit rapid transfer of messages between sites. We are now seeing the emergence (1,2) of networks of computers interconnected by communication facilities. There are numerous benefits to be achieved by this dispersal of computing facilities including more reliable overall operational capability, better overall service to geographically separated sites, and the advantages of being able to utilize a wide range of software and hardware facilities available through telecommunication systems.

Inherent in a distributed data processing (DDP) system is to a varying degree some interdependence of each computing site and the interconnecting communications system. Failure of a particular DDP network computer site, hereafter referred to as a node, will have a negative effect on the overall distributed data processing system. In a similar vein, failure of communication links will reduce the performance of the system. A study was initiated to examine these failure modes and develop criteria to measure the performance of a DDP system, including its associated data distributions, in terms of individual equipment failure modes and associated probabilities. The term survivability index is used as a performance parameter of a DDP system and an objective function has been defined to provide a measure of survivability in terms of the nodes and links of a network and their failure probabilities, data set distributions, and weighting factors for network nodes and computer programs.

Having derived an objective function to measure DDP performance, alternative data set distributions and network architectures can be evaluated. Criteria can be included such as addition or deletion of communication links, movement of programs among nodes, duplication of data sets, etc. Constraints can be introduced which limit the number and size of files and programs that can be assigned to a node and the maximization of the objective function will be subject to these constraints. One of the key concerns in our derivation of an objective function is its computability. The present algorithm exhibits exponential growth with the number of nodes and links. Studies are in process to find more efficient computational algorithms to quantify the survivability index.

This paper is organized into eight sections. The next three sections will provide background on communication network survivability, DDP survivability concepts, and a definition of a survivability index $S$. An objective function is derived to quantify the DDP survivability index $S$ in the fifth section, followed by a section presenting an example computation of $S$ for a four node network. The next section describes the evaluation of $S$ using a computer program for nine node networks representing seven architectures with three data distributions. Some comments and conclusions are presented in the final section.

BACKGROUND: COMMUNICATION NETWORK ANALYSIS

The study of communication network survivability (3,4,5) has been divided into two nearly disjoint areas: deterministic and probabilistic. This activity was aimed at determining optimal communication network architectures to better withstand either wartime attack or the impact of natural disasters. Since it serves as a starting point for the research activity described below for a DDP system, a brief review of the underlying concepts of these studies is presented here.

Deterministic survivability (3,6,7,8,9) presumes a fixed communication network architecture. In considering survivability from wartime attack, it is assumed the adversary has full knowledge of the network architecture. Survivability is measured in terms of maximal connected subnetwork components surviving after damage is inflicted on the original communications network. Criteria include the number of nodes still in communication after removal of nodes and links. Selection of network architectures which maximize the connected subnetwork components are the goal of these studies (3,4,5).

The approach to defining probabilistic survivability of a
communication network can be further divided into networks exhibiting random structure (3,10,11,12,13) or fixed structure (3,14,15,16,17). For networks with random structure it is assumed that any two nodes are connected with a known probability. This probability can be a function of whether the network is subjected to attack or a natural disaster. Survivability is measured in terms of the probability of survival of a connected network, i.e., one in which some specified number of nodes are still in contact. Mathematical techniques of cut sets and path analysis are used to determine survivability of a damaged network.

The fixed structure approach assumes a fixed and known network architecture and assigns a probability of failure to each node and link. Again these probabilities can be raised or lowered as a function of being subject to wartime attack or natural disasters. Survivability is measured in terms of the probability of criteria such as “all nodes can communicate with one another” or that a “percentage of nodes both survive the attack and remain in contact with the largest single group (component) of surviving nodes” (3).

**DISTRIBUTED DATA PROCESSING SYSTEMS**

DDP systems can be constructed in a number of ways. A very simple example occurs when a data processor serves as a “front end” for a larger processor and handles input-output functions. A somewhat more complex example is an array of interconnected processors (18,19). More complexity is added when geographically remote processors are interconnected by a telecommunications network (1). It will be this concept of a DDP which is considered in this paper.

Adding data processing and associated data sets at the nodes of a telecommunication network to create a DDP system greatly complicates the analysis of survivability. If it is further assumed that not all data sets are co-resident with the processor executing a program at a node, the quantification of DDP survivability becomes even more difficult. It is this latter implementation of a DDP system which is the basis for the research reported here.

When an executing program at a node needs access to another node within the DDP, two failure modes must be considered. One is that at least one link or node along all paths interconnecting the needed data to the executing program have failed thus breaking communication between the executing program and its required data. The second is that the node at which the required data is resident has failed. Either of these failure modes prevents a program at some node in the DDP from executing. We will refer to this situation as a DDP system with remote data requirements.

A final failure mode is that the node at which a program is to execute fails, i.e., the data processor is inoperative. Any of the three failure mechanisms noted above can cause a program at a node to be unexecutable. Probabilities for failure “q” and being operational “p” can be assigned to each node and telecommunication link. It is assumed here that these probabilities are independent which is consistent with the assumptions for survivability analysis for telecommunications networks.

**DDP SURVIVABILITY CRITERIA**

A simple quantitative measure of the survivability of a DDP is the number of programs that remain operational after some combination of nodes or links have failed. For a given network architecture there are a large number of subarchitectures that occur because of failures of nodes and links. Each of these subarchitectures has a probability of occurring and for each the number of operational programs can be determined based upon the data set requirements for programs executing at operable nodes and the data set distribution across nodes. A survivability criterion can be generated by taking the expectation of the number of programs operable for each subarchitecture leading to a summation of the proportion of programs operable for a subarchitecture times the probability of its occurrence. This criterion is designated as $S$ and provides a quantitative measure of DDP survivability as a function of initial network architecture, a given data set distribution, and the data set requirements for each program at each node. A brief mathematical derivation of $S$ is presented in the next section. An example of the use of this criterion for a simple four node network is described in the following section.

The mathematical approach to quantifying DDP survivability is computationally costly. For a DDP with $N$ nodes and $L$ links, $2^{N+L}$ possibilities exist for subarchitectures. The requirement to access data at remote nodes greatly reduces the number of cases that must be considered but computationally we are quite limited as to the size of a DDP system that can be analyzed. At present we can handle DDP systems with nine nodes and from 10 to 12 links. Better mathematical approaches are being developed, but haven’t been implemented in a computer program which calculates $S$ for a given DDP and data distribution.

Other factors that can be introduced, which will impact $S$, are the assignment of weights to programs and nodes, and the introduction of constraints on the data set distribution. Weighting factors are introduced to indicate the relative importance to the function of a DDP system of individual programs or nodes. These weights would have to be assigned by the DDP designers and provision is made for incorporating these weights in the computer program which computes $S$.

The data set distribution constraints would again be design decisions and represent situations where only so many data sets could be resident at a node, or due to update frequencies, it would not be feasible to make duplicate copies of a data set which is generated at a particular node. In the DDP survivability analysis for various network architectures described below, both program and node weighting factors are held constant and equal while data set constraints are not considered.

**SURVIVABILITY INDEX COMPUTATION ALGORITHM**

An enumerative technique has been selected to evaluate the survivability index $S$ for a given DDP system architecture.
and data set distribution. The algorithm executes in four phases. The first phase determines the connected components, i.e., subarchitectures of the DDP network. The computation in this step is exponential in determining execution time and constitutes the limiting factor on DDP networks that can be analyzed for survivability. The second phase introduces the data set distribution and a further selection is made of those subarchitectures for which programs can execute, i.e., have access to the required data sets. A node is considered as surviving if at least one program executes at this node. The third phase assigns weighting factors to programs and nodes, while in the last phase the survivability index is computed by a procedure to be described below.

Given a DDP network consisting of \( N \) nodes and \( L \) links, define stochastic variables \( E_{ij} \) for every link \((i,j)\) and \( E_i \) for every node \( i \). These variables assume the values \{0,1\}. The value one indicates that the component is operational, while the value zero indicates it is nonoperational. The distribution of \( E_{ij} \) and \( E_i \) is:

\[
P(E_{ij} = 1) = p_{ij} \quad P(E_{ij} = 0) = q_{ij} \]
\[
P(E_i = 1) = p_i \quad P(E_i = 0) = q_i \]

Assume that a DDP network consists of \( N \) nodes and \( L \) links. With \( N + L \) binary stochastic variables, \( 2^{N+L} \) elementary events have to be considered. Each event corresponds to a subgraph, i.e., subarchitecture, of the graph corresponding to the network architecture. Each event's probability, since it is assumed all variables are independent, is the product of probabilities of the corresponding node and link probabilities. For event \( j \), assume a combination of programs survives at each node. Let \( a(i,j) \) represent the combination of programs surviving at node \( i \) for event \( j \) and let \( b(j) \) represent the combination of nodes that survive for this event. Because of the dependence of survivability on the combination of programs and nodes surviving, the survivability is defined for event \( j \) and then expanded to find the total survivability for the whole network. For event \( j \), the survivability index would be the weighted sum of the survivability of nodes as follows:

\[
S^j = \sum_{i=1}^{N} W_{i}^{p(i)} S_i^j \quad (1)
\]

where \( S_i^j \) is the survivability of node \( i \) for event \( j \) and \( W_{i}^{p(i)} \) is the weight of node \( i \) for combinations of surviving nodes \( b \) that depends on event (subgraph) number \( j \). \( S_i^j \), the survivability index of node \( i \) for event \( j \), is the sum of the weights of the surviving programs at node \( i \), as follows:

\[
S_i^j = \sum_{k=1}^{m_i} W_{i,k}^{a(i)} \quad (2)
\]

where \( m_i \) represents the number of programs at node \( i \), \( W_{i,k}^{a(i)} \) is the weight of program \( k \) located at node \( i \) for combination of surviving programs \( a_i \) that depends on event number \( j \). By substituting for \( S_i^j \) from (2) into (1) we would have:

\[
S^j = \sum_{i=1}^{N} W_i^{p(i)} \sum_{k=1}^{m_i} W_{i,k}^{a(i)} \quad (3)
\]

The total survivability is by definition

\[
S = \sum_{j=1}^{2^{N+L}} P(A_j) \cdot S^j
\]

where \( A_j \) represents the event \( j \) and \( P(A_j) \) represents the probability that \( A_j \) happens. The example showing the calculation of \( S \) in the next section will further clarify the details of the algorithm for computing the DDP system survivability.

COMPUTATION OF \( S \)

An example of the computation of survivability index \( S \) for a simple DDP system is presented in this section. The DDP network shown in Figure 1 consists of 4 nodes and 4 links. Assignment of files and programs to nodes is as shown. FA denotes the files available while FN denotes the files needed to execute the programs at a node. PM designates the programs to be executed at a node. For example, at node 2 we have two programs, \( X_{2,1} \) and \( X_{2,2} \). Files 2 and 4 are

![Figure 1—Four node DDP.](From the collection of the Computer History Museum (www.computerhistory.org))
There are 42 cases out of 256 (2^8 = 256).

Figure 2—Example of computation of S.
required to execute program \( X_{2,1} \), and are designated as \( S_{2,1} = [2,4] \).

If a program is executable at a node, then that node exhibits some degree of survival. In the DDP system shown in Figure 1, no program can execute at a node without access to a file resident at some other node. Thus for program \( X_{2,1} \) to execute at node 2, it must have access to file number 4 at node 3 or 4. This implies that data link \( E_{12} \) and node 1 be operational. In addition, either data link \( E_{13} \) and node 3 or data link \( E_{14} \) and node 4 must be operational. As can be seen, the same conditions exist for program \( X_{2,2} \) to be executable at node 2.

For the particular DDP shown in Figure 1 there are 256 different configurations ranging from everything having failed (all nodes and links) to everything being operational. For many of the possible configurations of operable nodes and links no program can execute at any node. The first phase of our computational algorithm selects all operable network subarchitectures for which it is possible for one or more programs to operate at a node. For Figure 1 there are 42 such configurations as shown in Figure 2 which illustrate the computational steps for \( S \). When availability of data sets are included as a selection criteria, then only 26 of the 42 operable subarchitectures have nodes with an executable program. The probability of occurrence of each of the 26 subarchitectures times their weighting functions is shown on the right side of Figure 2. Survivability index \( S \) is the summation of these terms.

For purpose of illustration of this computation, the probability of survival for a node or link was set at the representative values at 0.9, and for failure at 0.1. All nodes and programs were assumed to be of equal weight. Since the sum of the weights of all nodes must equal one, the weight for each of the four nodes is \( \frac{1}{4} \). Likewise, since the sum of the weights of the programs at each node must equal one, the program weights \( W_i \) are: \( \frac{1}{2} \) at node 1, i.e., 2 programs resident at this node; \( \frac{1}{4} \) at node 2; \( \frac{1}{3} \) at node 3; and 1 at node 4. Using the definition of survivability \( S \) as noted above, we get \( S = 0.7658 \) for the DDP system shown in Figure 1.

**DDP NETWORK SURVIVABILITY COMPARISON**

A computer program has been generated which computes \( S \) for a given DDP network, and program and data set distribution. The use of this program is restricted to DDP systems where the sum \( N + L \) is less than 20. For DDP of this complexity the running time of this program on an IBM 370/148 is several minutes. The addition of a node or link increases the running time by factors approaching two. No attempt has been made at this time to improve the efficiency of the program to enable computation of \( S \) for larger DDP networks. As noted above, other mathematical algorithms to reduce computation time and significantly increase the size of a DDP system for which the survivability index can be computed are being developed as part of an ongoing research program.

Seven DDP network geometries were investigated as shown in Figure 3. Some familiar DDP architectures which have been evaluated for survivability include the ring (3f), star (3e), grid (3a), and linear Z (3b). A single program is assumed resident at each node and a set of data files required to run each program is specified. Three distributions of data files were specified. Two were selected at random while the third was chosen to minimize the distance in terms of links between a program and the data files required to execute the program. Table I shows the three data-file node assignments and indicates for each node what data must be fetched from a remote node.

The corresponding value of survivability index \( S \) for each of the test cases (21 in all) were computed. Table II contains a tabulation of values of \( S \) for each test case.

As was expected, the data file assignment can have a large impact on DDP system survivability, thus demonstrating the importance of this factor in the design of distributed computer networks. Individual DDP architectures varied widely in terms of survivability when the data set distribution was changed. As would be expected, architectures with the most links, i.e., Figure 3a—grid, had better survivability. Also as was expected, the star architecture has high survivability because the distance between a program and its data set never exceeds two links. The various ring architectures rank moderately high on survivability. The string architectures, i.e., linear Z and swastika, have fewer links (8) and correspondingly lower survivability. This reflects the greater distance that can occur between data sets and programs and the lack of parallel or redundant communication paths between nodes.

**CONCLUSIONS**

The research directed toward deriving a survivability criteria in this paper for computer networks is still in a very preliminary stage. A survivability index \( S \) has been defined which measures survival in terms of the number of programs which remain executable in the DDP after some nodes or

![Figure 3a-Grid](image-url)
TABLE I.—Data file distribution

<table>
<thead>
<tr>
<th>Node Number</th>
<th>Program Files Needed</th>
<th>Distribution 1</th>
<th>Distribution 2</th>
<th>Distribution 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Resident Files</td>
<td>Needed Files</td>
<td>Resident Files</td>
</tr>
<tr>
<td>1</td>
<td>1,3,8</td>
<td>1, 3,8</td>
<td>3, 1,8</td>
<td>1,3</td>
</tr>
<tr>
<td>2</td>
<td>1,4,6</td>
<td>4, 1,6</td>
<td>7, 1,4,6</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>2,3,7</td>
<td>2,7,3</td>
<td>2,5, 3,7</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>1,4,6</td>
<td>1,6, 4</td>
<td>1,4,6</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>5,7,8</td>
<td>1,5, 7,8</td>
<td>1,8, 5,7</td>
<td>4,8</td>
</tr>
<tr>
<td>6</td>
<td>2,3,5</td>
<td>3, 2,5</td>
<td>6,7, 2,3,5</td>
<td>2,5</td>
</tr>
<tr>
<td>7</td>
<td>4,5,8</td>
<td>5,8, 4</td>
<td>5, 4,8</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>2,3,6</td>
<td>2,6, 3</td>
<td>1, 2,3,6</td>
<td>2,6</td>
</tr>
<tr>
<td>9</td>
<td>1,7,8</td>
<td>7, 1,8</td>
<td>2, 1,7,8</td>
<td>1,7</td>
</tr>
</tbody>
</table>

The inclusion of data distribution in the formulation of $S$ is the main area of departure of this research from previous work on network survivability. The importance of including the data distribution within a DDP in enumerating survivability is demonstrated in the numerical results presented in the previous section.

The main purpose of this preliminary activity in deriving a DDP survivability criteria was to establish the validity of the index $S$. Future plans call for deriving better mathematical representations and improving the computational algorithms for $S$. Assuming the availability of better representations for DDP survivability, it becomes possible to evaluate DDP architectures, data distributions, and to develop algorithms to achieve optimal DDP survivability in terms of these parameters.

The authors want to take this opportunity to express our appreciation for the generous support of this research activity by Mr. Clifford McLain, and Dean Ray, Defense Civil Preparedness Agency; Center for Academic and Administrative Computing; and helpful suggestions by Professor T.

TABLE II.—Value of survivability index $S$ for various network structures and file distributions

<table>
<thead>
<tr>
<th>DDP Architecture</th>
<th>Survivability Index $S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure</td>
<td>Structure Name</td>
</tr>
<tr>
<td>3a</td>
<td>Grid</td>
</tr>
<tr>
<td>3b</td>
<td>Linear Z</td>
</tr>
<tr>
<td>3c</td>
<td>Ring with Tail</td>
</tr>
<tr>
<td>3d</td>
<td>Swastika</td>
</tr>
<tr>
<td>3e</td>
<td>Star</td>
</tr>
<tr>
<td>3f</td>
<td>Ring</td>
</tr>
<tr>
<td>3g</td>
<td>Ring with 3 Tails</td>
</tr>
</tbody>
</table>
Lee, and other members of the Electrical Engineering and Computer Science Faculty at The George Washington University.

REFERENCES
