301

DATA SECURITY AND DATA INDEPENDENCE IN A MOBILE MILITARY SYSTEM

by

WILLIAM PAUL AKINS

B.A., San Francisco State University, 1972

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas
1980

Approved by:

E. Klenger

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

PAGE

THIS BOOK CONTAINS NUMEROUS PAGES WITH THE ORIGINAL PRINTING ON THE PAGE BEING CROOKED.

THIS IS THE BEST IMAGE AVAILABLE.

Chapter 1 - Introduction

1.0 Background - The introduction of computer hardware components that are small, light and capable of withstanding the battlefield environment has precipitated a remarkable number of potential applications for the military. Systems are in development to manage intelligence collection, support intelligence analysis, support command and control functions, record and disseminate radiation dosages, manage communications links, control weapons systems, account for supplies, control movements, and provide personnel data. Nearly every aspect of a functioning Army has been examined for its automation potential. In addition to the hardware advances there have been several processing techniques that offer even greater potential to the Army. These techniques include networking, distributed processing, data base management systems, and distributed data base management systems. In contrast to the immediate recognition of the hardware advantages several of these techniques have not yet been applied to Army applications. This report concentrates on distributed processing and its applicability to mobile military systems.

1.1 Strategic versus Tactical - This report is concerned with tactical systems. These systems are relatively small and mobile. They support forces that are physically present on the battlefield. Strategic systems support the National Command Structure, the CIA, and other agencies. They are relatively large systems that are located in secure areas in the United States. The problems faced by strategic systems are very similiar to the problems faced by commercial and academic installations. These problems have been widely investigated and discussed. The problems that apply to tactical systems are currently unique to the military. Tactical systems face unusual environmental, reliability, security, and maintainability problems that have not been thoroughly investigated to date. In some cases serious feasibility issues remain before tactical computer systems can be fielded with confidence.

1.2 Structure of the Report - This report addresses distributed processing in a tactical system. The report consists of a general requirements statement which includes the definition of a postulated system; discussion of system resiliency needs; discussion of system security needs; discussion of the need for data independence; and concluding remarks.

1.3 Definitions - The following definitions are used throughout the report:

a. Distributed Processing
There is so much confusion over this term that a "thorough" definition would merit publication. Enslow has identified five components for distributed processing. These are: a multiplicity of general-purpose resource components that can be allocated dynamically; a physical distribution of these resources through a communication network; a high-level operating system to control the distributed components; system transparency; and cooperative autonomy. [ENSL78] This explanation is closer to a checklist than a definition. The real key to distributed processing is the autonomy component. The capability for components to cooperate and yet maintain their own identities is unique to distributed processing. Distributed processing can only be achieved in certain environments. The environments are those identified by Enslow and are based upon the degrees of hardware, data base and control decentralization. Multiple processors or multiple computers, replicated cooperating or multiple fully cooperating control, and some scheme of distributed files, replication or partitioning are the necessary environments to achieve distributed processing. Katzan describes a distributed system as one "in which there are several autonomous but interacting processors and/or data stores at different geographical locations". [KATZ78] Distributed processing then is the capability for a group of dispersed, independent but (cooperating) components in particular environments to provide users with processing resources as required.

b. Distributed Data Base Management System (DDBMS)
The concept of a DDBMS is even less clear than distributed processing. Maryanski implies that a DDBMS is a generalized interface for different DBMS conventional systems. He states that a DDBMS "... would reside on a heterogeneous computer network with different data base systems available at various processors". [MARY77E] This may be an

evolutionary step toward realization of a DDBMS but it is not a fair definition of the DDBMS itself. A DDBMS should provide the data base management services on a system basis rather than nodal basis. These services include the usual data definition and data manipulation capabilities of conventional systems plus the capabilities unique to a distributed system. These unique capabilities include the means to distribute data (partitioned or replicated), distribute directories, manage updating of replicated data, and rollback and recovery. These unique capabilities are the reasons a true DDBMS is not currently practical. When they do become available then a DDBMS will provide its services throughout the distributed system rather than managing the efforts of a collection of DBMS's located at different nodes.

c. Network

Anderson has defined networks as being made up of a combination of computer interconnections. The interconnections are constructed from paths and Switching elements. Paths are the medium by which messages are transferred between the other system elements. Switching elements are entities which may be thought of "intervening intelligence" between the sender and receiver. A network is a combination of paths and switching elements. [ANDE75A]

d. Mobile System

A mobile system is one in which all elements of the system can be moved without external support. User areas, power sources, and all associated hardware are installed in trucks. A mobile system can operate at a variety of locations with a minimum of time required to move the system from one location to another.

e. Tactical System

A tactical system is a mobile system that must have the additional capabilities of cross-country movement, and the ability to operate in a variety of environments. These environments include the various weather conditions that can be encountered and the fact that an enemy force is actively trying to locate and destroy the system.

1.4 Use of Computers in the Military - The following overview traces the use of computers by the military in general terms. A detailed history of military computer use is not part of this report. However, a general framework assists in understanding the need for distributed systems and the external problems that already exist.

1.4.1 Single Machines - Several single machine systems have been in use for some time. The most noteworthy of these is the Field Artillery Data Analysis Center (FADAC). FADAC provides ballistic computational support to artillery units. These single machine systems were self- contained and were not required to interface with any other system.

1.4.2 Multiple Machines - Systems that have been recently fielded or are about to be fielded are characterized by several micro or mini computers that are linked together to form a single system. Examples of this configuration include the Tactical Operating System (TOS), and the All Source Analysis System (ASAS). [HILS79, MAHA79, STUB79] These systems utilize the capabilities associated with networks. Users may be geographically dispersed and may enter and leave the network of their own volition. Each of these systems performs services for a single functional area. TOS provides support for command and control while ASAS provides support to the intelligence activities.

THIS BOOK CONTAINS NUMEROUS PAGE NUMBERS THAT ARE ILLEGIBLE

THIS IS AS RECEIVED FROM THE CUSTOMER

1.4.3 Multiple Functional Areas - The next logical step in the development of military computer systems is the design and fielding of systems that will support more than one functional area. Current research has recognized this need. However, to date resources are being spent in developing patchwork arrangements to tie the existing systems together. This action has not been successful. The existing systems were developed independently. Choices of hardware, software, data standards, and communication standards were all made independently. As a result it has proven very difficult to tie these existing systems together. In addition, each of the functional areas is controlled by a different organization. None of these organizations is inclined to surrender some of its independence in order to achieve a truly integrated system.

1.4.4 Postulated Division System - This report uses a notional system as a basis for investigating the possibility of achieving better data security, data independence, and system resiliency in a tactical system. The system is fully integrated by functional area. It supports command and control, intelligence, and logistics simultaneously. The liklihood of such a system being developed does not depend on research or the state of the art. It does depend on the organizational politics cited above.

Chapter 2 - Postulated Division System

2.0 Description of a Typical Military System - The system used is representative of several developing military computer systems. These systems include the Tactical Operating System (TOS), which provides command and control support; Combat Service Support System (CS3) which provides logistics support; and the All Source Analysis System (ASAS) which will provide intelligence support. Each of these systems provides its support in an independent manner. The interfaces between them are, in most cases, handled manually rather than electronically. This interface policy is partially due to security policies but principally to uncoordinated development of the separate systems by separate elements. The model described in this report incorporates features of all of the existing systems. It has been developed as a paradigm from which to discuss distributed processing. The model is developed through three separate views. While not an actual system it incorporates the general requirements for a military system at the division level.

2.1 Organizational View - The current organization of an

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

Army division has evolved over the last fifteen years. Today's organization is shown in Figure 1.

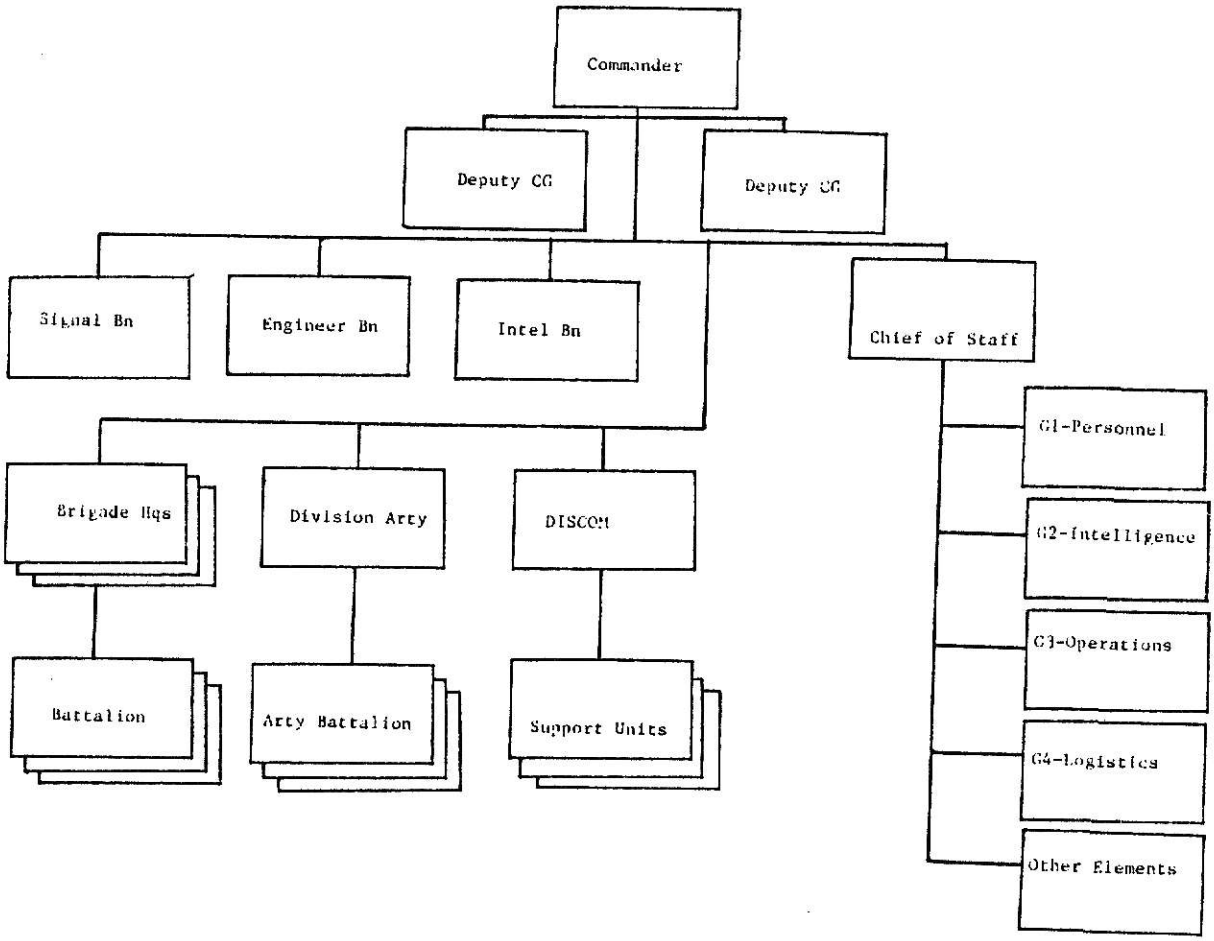


FIGURE 1

The traditional line and staff functions are annotated. The line organizations generally are the "doers" for their respective functional areas. Their information requirements are of a flow (I/O) nature rather than processing. They report information to be analyzed and used to support current and future decisions and receive information based on their current or expected mission. The staff organizations are characterized as "planners". They receive and evaluate reports from internal and external sources and apply updates to the data bases as appropriate. They also do extensive analysis in preparing reports and orders for future operations. An explanation of each element s functions can be found in Appendix A. This view of the organization is the official copy of the organization of reference. When changes to the organization itself are considered or applied they are usually done based upon this view. In many cases there are overlaps between the staff and line organizations and functions. These overlaps occur in performance of a variety of functions. For example, elements from the G2, G3 Division Artillery, and Intelligence Battalion combine to form the Tactical Operations Center (TOC) and the Tactical Command Post (TAC CP). [STAR79] This combination renders the organizational view nearly useless when examining information flows.

2.2 User View - Figure 2 is a user view of the organization within the framework of the user functions.

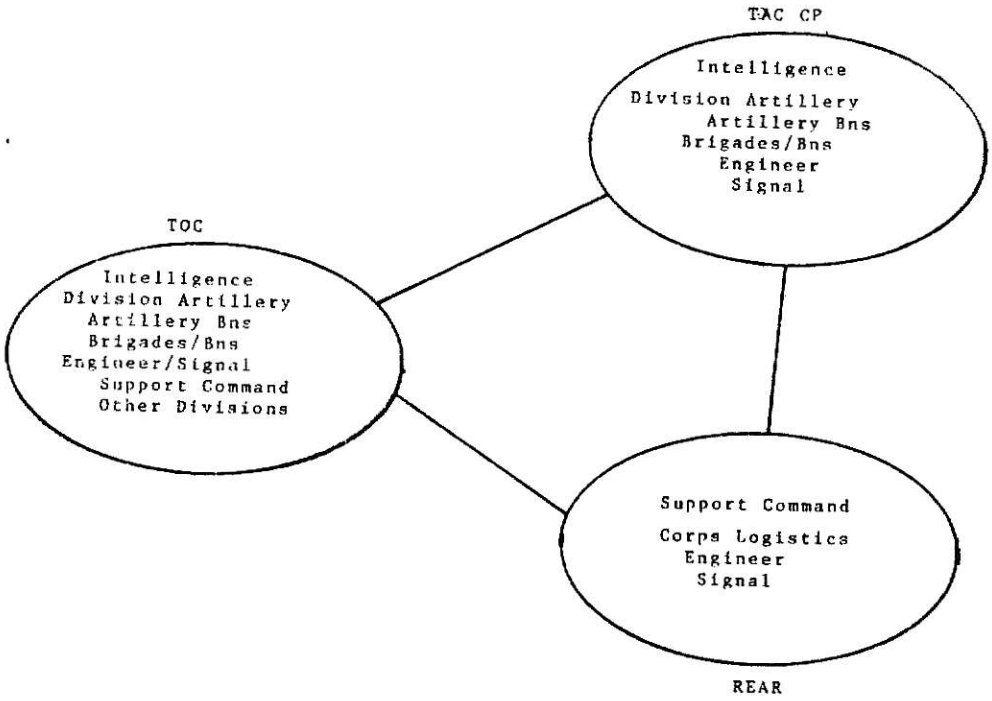


FIGURE 2

A mapping of the current systems onto the user view is shown at figure 3.

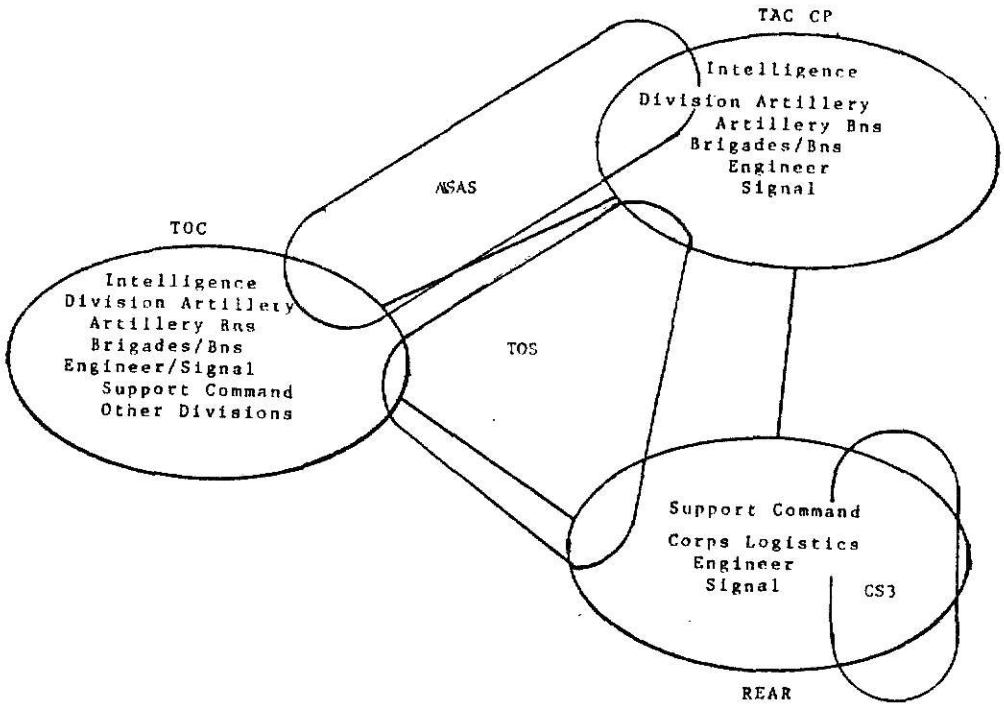
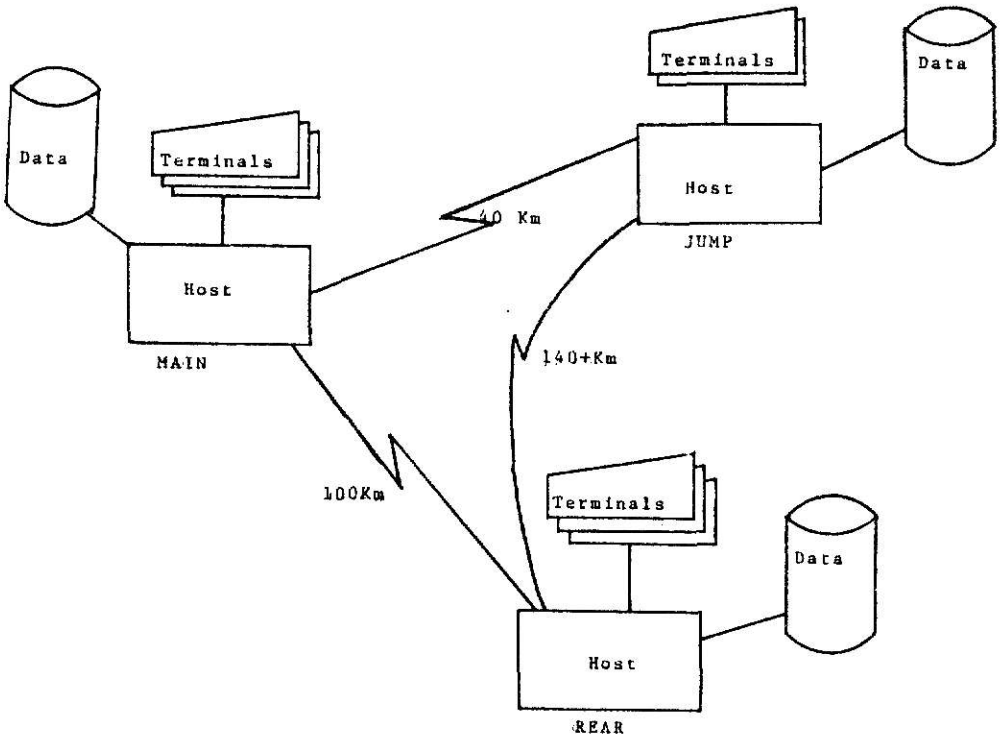


FIGURE 3

The logistics system (CS3) is separate functionally and geographically. The command and control and intelligence systems share locations but their functions operate independently. This has caused or will cause a proliferation of manual interfaces. A more appropriate system is suggested below. It is called the Postulated Division System (PODS). It incorporates the command and control, intelligence, and logistics functions into a single distributed system. The PODS consists of three nodes. The JUMP node is a relatively small configuration used by the commander to control his forces. It is usually located close to the front lines. It is that portion of the PODS that provides support to the Tactical Command Post (TAC CP). Its geographic location dictates a small size and high mobility. Yet the commander requires access to any information in the distributed system. The MAIN node provides service to the division staff and other elements of the Tactical Operations Center (TOC). Its capabilities and size are both larger than the JUMP. The REAR node provides the logistic and administrative services as a part of the distributed system. Its size and function dictate a safer location and less frequent movement. Other entities provide information to or receive information from the division system. For purposes of discussion all elements connected

to the division system may be viewed as terminals from at least one of the nodes in the system. In actuality, several of these elements are networks themselves but the simplistic view is more manageable for this paper. One very interesting aspect of this view is the restructuring or reordering of relationships when compared to the hierarchical view. Brigade headquarters and their subordinate battalions are treated as like elements by the division system. This speeds information flow but can disrupt traditional command relationships.

2.3 System View - A view of the system as seen by a system designer is shown in Figure 4.



SYSTEM VIEW

FIGURE 4

The nodes have different functions and capacity and therefore are likely to be heterogeneous. The network structure is the DDC(direct, dedicated, complete) architecture of Anderson s taxonomy scheme.[ANDE/5A]. Communication in all cases is by radio link rather than cable. Approximate geographic dispersion is indicated in Figure 4. These distances are very dynamic as will be seen below. Note, that while there is some reflection of the user view here there is little that resembles the hierarchical view.

2.4 General Need for System Resiliency - The division system is entirely mobile. However, this means the system is easily moved, not that the system operates while moving. In order to maintain continuous service a modified leapfrog technique is used. One node (usually the JUMP) leaves the network and moves to a new location. After the JUMP returns to operation other nodes leave the network, move and return to the network. An estimated frequency of movement table is shown below.

NODE	FREQUENCY
JUMP	2 per Day
MAIN	1 per Day
REAR	2 per Week

TABLE 1: Frequency of Node Movement

While total movement time varies, an average "away from the network" time is four hours. In addition to the normal component failures experienced by commercial and academic networks, a military network must anticipate the loss of a node due to enemy action. In this case, there is little hope that the data resident on the affected node can ever be recovered. The network must be capable of sustained operation without one or more nodes if necessary. Further, maintenance of the communications is difficult on a battlefield. Any node may find itself operational but unable to communicate with the rest of the network. In those situations the node must be able to operate in some independent or autonomous manner until a connection with the network can be restored.

Chapter 3 - System Resiliency

3.0 System resiliency is the ability of a distributed system to continue operations despite component failures This general definition raises more questions than it answers. The ability to continue operations despite component failures can affect system design, software design, hardware configuration, hardware selection and database design. If a system is to be truly resilient then this capability must be recognized as the system is initially described and designed. In most existing systems the need for system resiliency was recognized after design was well underway or even after the system was already in operation. The methods used to achieve varying degrees of system resiliency have been patches and as such have usually provided the additional capability at great cost, if at all.

3.1 The central difficulty in achieving a high degree of system resiliency is the need to insure that database consistency is not violated as a result of component failures. [ROTH77B] That is, the user must be able to access the required information regardless of component failures.

Components in this context may be individual pieces of equipment but are usually construed to be a complete node in the network.

3.2 In order to preserve this database consistency Rothnie identifies five broad areas that must be considered. These are reliable broadcast, operation with missing nodes, restarting a node, failure detection, and partitioning. [ROTH7/B]

3.2.1 Reliable Broadcast - Reliable broadcast is the capability to positively insure that a message sent to more than one node will in fact reach all addressed nodes. The sending node must be aware that the message has reached all destinations. Most communications systems available today do not support reliable broadcasting.

3.2.2 Operation with Missing Nodes - The system must continue to function despite the absence of one or more nodes; the critical condition becomes what to do with updates for a node that is missing. Most systems available today treat the missing node as if it does not exist. This can create unacceptable delays if one node must have access to the missing node. Some mechanism must be available to continue operation of the remaining nodes while accumulating the essential transactions for later use by the missing node.

3.2.3 Restarting a Node - Once a failed node has been restarted its data bases must be brought to a current state. Transactions that would have been directed to this node must now be applied. The principal technique in use is called "Persistent Communication" by Rothnie. This implies that any message sent will be received at some time, regardless of the condition of the destination node at the time of transmission or the sending node at the time of reception. Another alternative worth investigating is the "recuperating node". A recuperating node is defined as:

> Recuperating Node
> After a failed node has returned to operation it is placed in a special status within the network. While in this status all accumulated transactions are applied without interruption from current (and perhaps higher priority) tasks.

When the recuperating node is up to date its status changes from recuperating to fully participating. The feasibility of the recuperating node has not been investigated. It has some obvious advantages as will be seen later in this paper.

3.2.4 Failure Detection - In order to operate with missing nodes, or to know that a node has failed, a means must be available to quickly and efficiently detect a node failure. Most provisions today are based on a time-out condition. If a node does not respond in a predetermined time then the node is deemed to be non-operational. While this concept

serves adequately from a system operator's view, it may be poor or even unacceptable from the user or organizational point of view. For critical realtime functions the user may need a failure detection algorithm that will both recognize failures and reroute transactions to other nodes as appropriate. There is little evidence of a link between failure detection and operation with missing nodes to insure a smooth transition. If a distributed system is to provide reliable service to the user system turbulence must be held to a minimum when a node is failing or has failed.

3.2.5 Partitioning - This area overlaps restarting a node and operating with a missing node. However, the point of view has changed from a node to the data. The weak and strong consistencies of SDD-1 also are applicable to this area. [ROTH//B] How can consistency (all nodes have current data) be maintained in an environment of failed, recuperating, operating nodes? The design decisions on what data elements will be stored at which nodes will have a great impact on the operation of the network. Data that is strictly partitioned will be relatively easy to restore after node failure but will be totally unavailable if its node fails. Conversely replicated data will be more likely to be available in a network with missing nodes but updating recuperating nodes will be more complex. The current

philosophy given by Rothnie states:

> In general, the correct action to take when a partition is repaired depends on the semantics of the database, the topology of the partition, and the transactions which have been run during the partition. The integration process is likely to require human decision making assisted by automatic conflict detection and some limited automatic conflict resolution. [ROTH77B]

If in fact human decision making is required during the restoration process then in a turbulent net there is a real possibility that users will spend a significant amount of time restoring nodes.

3.3 Specific System Resiliency Requirements - Although the division system used here is only theoretical, some of the specific requirements for system resiliency are readily identified. The requirements are based upon the user view but must be implemented in the system view.

3.3.1 Operation with a Missing Node - The system's "normal" state will be one in which nodes are missing. Individual node availabilities based on Table 1 are:

```
JUMP = 1-(8/24)   =  .6667
MAIN = 1-(8/24)   =  .6667
REAR = 1-(16/168) = .9047
```

In addition there will be nodes out of the network due to communication failures, hardware component failures, and destruction. No firm estimates are available for these factors. A conservative estimate is that 10 per cent of the

time a node will not be available for one of the three reasons noted above. This figure could approach 20 per cent in an actual violent battle situation. This estimate is based on the time a node is available (not moving) therefore the total probability that a node is available is the sum of the probabilities for availability based on movement requirements and based on the other factors. Thus the total availability for the Jump node becomes:

P(JUMP) = 1-((.3333)+(.10)) = .5667

The probability that the entire network is available can be estimated if the availabilities of individual nodes are independent events. Network availability then is the product of the probabilities for each of the nodes in the network.

P(NETWORK) = P(JUMP) * P(MAIN) * P(REAR)

By using the probabilities already developed the total network availability then becomes:

P(NETWORK) = (.5667)*(.5667)*(.8048) = .258

Thus the division system will operate with missing nodes at least 75% of the time. A means of achieving system resiliency in this area is essential.

3.3.2 Reliable Broadcast - If the total network will only be available 25% of the time then some means to either save all pending updates until a node is recuperating or to

direct all changes to a designated replication for later transfer to the affected node must be developed.

3.3.3 Failure Detection - In most cases a node failure will be directed. That is, a node will fail or leave the network in order to relocate. In those cases the network can be warned of the failure and a failure detection mechanism is unnecessary. When a failure is due to a communications loss, hardware failure or destruction then some means must be available to detect failures. This action should include failure detection, automatic rerouting of retrievals and updates, and informing the other nodes (and the other nodes users) of the outage. While this action is not trivial it is essential. An automatic failure detection may be the first indication that the node from which the commander is directing the battle is not available. This situation must be recognized quickly so that a deputy commander can assume command (even for short peroiods of time). This rapid succession of command keeps battlefield confusion at a miniimum level.

3.3.4 Design Considerations - Based upon the node and network availabilities already developed it is obvious that some portions of the data base must be replicated. The general replication requirements are shown in Table 2.

Type of Information	Node JUMP	MAIN	REAR
Friendly Unit Status	30/100/0	70/30/70	60/60/40
Enemy Unit Status	40/100/0	100/40/60	20/100/0
Intelligence Analysis	30/100/0	100/40/60	10/100/0
Logistics Data	20/100/0	40/100/0	100/40/60
Planning Information	35/100/0	65/30/70	35/30/70

A/B/C

A - Per cent of total data base held by node
B - Per cent of node's data that is replicated
C - Per cent of node's data that is unique

TABLE 2: General Replication Requirements

Design considerations will also include identification of applications software that will require replication. This will be necessary to permit an individual node to continue to operate independently if necessary.

3.4.0 Existing Methods for Managing System Resiliency - Most efforts to date to achieve system resiliency are based on exception processing. That is, the normal state of the network will be one in which all nodes are operational. In addition, the algorithms in use are communications oriented rather than function and system oriented. As a result, the algorithms tend to concentrate on the technical neatness of a particular situation rather than concentrating on how best to continue functional service to the user. In order to compare existing methods with the needs of the division system two such methods will be used as examples. These are the algorithms used in the ARPA and MERIT networks.

3.4.1 MERIT Network - The approach taken in the MERIT network is based on the assumption that in most networks it will be impractical to have direct connections between each possible node pair. [TAJI77] Tajibnapis then develops a method to maintain network topology information at each node. Thus, at any given time a node that originates a message can route the message along the shortest path to the destination. The topology information is maintained in Distance and Route tables. When a node notes the absence or presence of a neighboring node the Distance and Route tables are updated and a special class of message (NETCHANGE) is sent throughout the network. This technique is strictly

designed to manage the topology of the network. Within that view there is no apparent provision for positively identifying the "new" neighbor. While Tajibnapis indicates further research is required in the areas of flow control and congestion control there is no mention of the capability for the network (or individual nodes) to continue operation in a partial network. [TAJI77] This can probably be attributed to the nature of the MERIT network. Each node is an "associated member" with the MERIT network while each node in the PODS is a "voting member". The difference in degree of participation is significant. For this reason the MERIT scheme does not provide an existing solution for the division system.

3.4.2 ARPA Network - The methods used to manage the ARPA network are, in most cases, more elegant than those used by MERIT. The network has member nodes with different capabilities and roles in the network. Some members are classed as Terminal Interface Message Processors (TIP) while most are Interface Message Processors (IMP) having one or more hosts resident. [ORNS72] While this is closer to the topology of the division system the methods used to manage the network are centered on the same concerns as the MERIT scheme. Topology, message routing and the like receive the greatest attention. This is likely due to the same reasons

as the MERIT network. Both of these networks were built around existing centers. These centers became member nodes but retained their original roles at their respective locations. When a network such as the division system is postulated as a single cohesive distributed network then the means required to manage the network tend to be expressed in terms of the network as a whole rather than individual nodes.

Chapter 4 - Data Security

4.0 DATA SECURITY - In the context of this report data security is a measure of how the system, network, software, data base, and individual data items are protected from unauthorized access. Unauthorized access includes both accidental and intentional access occurrences. Each of the components must be afforded protection if the system itself is to achieve the required security. A current scheme of the different types of security is shown in Figure 5.

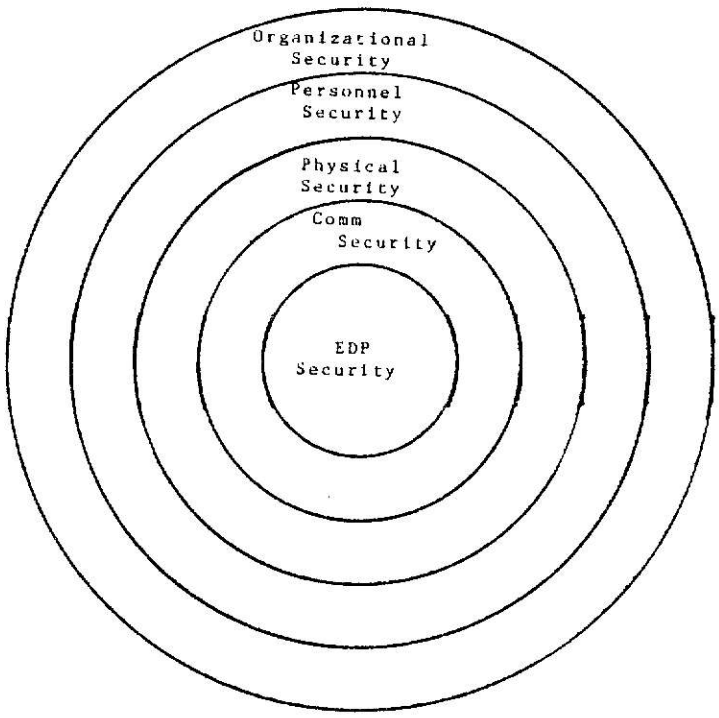


FIGURE 5

This hierarchy of security levels serves for military and non-military systems alike. However, tactical military systems face a greater likelihood of experiencing an attempted intentional unauthorized access than most other systems. Enemy forces will certainly try to access the system while it operates on the battlefield. Sophisticated physical security devices are not practical on a battlefield. As a result, the level of security protection in other areas becomes even more important. Some categories of system vulnerability are defined in Appendix B. This examination of data security will include protection of sensitive information, access and access control, surveillance, system integrity, and the notion of a network security center. The Network Security Center is one possible solution to the battlefield security problem.

4.1 Protection of Sensitive Information - The PODS will store and process sensitive information concerning current orders, planning, logistics and intelligence. Generally this information is incorporated into a classification scheme; however some information is further protected and separated from all other classified information. Thereore the different levels of sensitive information found in the PODS are shown in Table 3.

LEVEL	ACCESS
UNCLASSIFIED	UNLIMITED
CONFIDENTIAL	SLIGHT LIMITATIONS
SECRET	SOME LIMITATIONS
TOP SECRET	MORE LIMITATIONS
OTHER	GREATEST LIMITATIONS

Table 3: Classification Levels

The security mechanisms used to protect these levels of classified information must assure the required level of safety while still providing ease of access. The irony of this situation is that the information requiring the greatest degree of protection is the same information that generally requires the fastest response time by users. That is, an access to unclassified information requires little protection and therefore little overhead, but that same access is usually of low importance and can easily be delayed. Conversely, an access to the OTHER CATEGORY requires the greatest degree of protection but the accesses are usually of great importance and demand the shortest response time available. In addition to the protection-response time dichotomy, current legal restrictions prohibit storing all data on the same system. This is caused by the void in software systems certified to provide support to users with different access rights.

4.2 Access Request and Authorization - Identification and

authentication usually precede a request for access to a network resource since some knowledge of the requestor is required to determine if an access right exists. The information about the rights of the requestor is indicated in the Lampson/ Denning [COLE78] three dimensional authorization matrix (Figure 6).

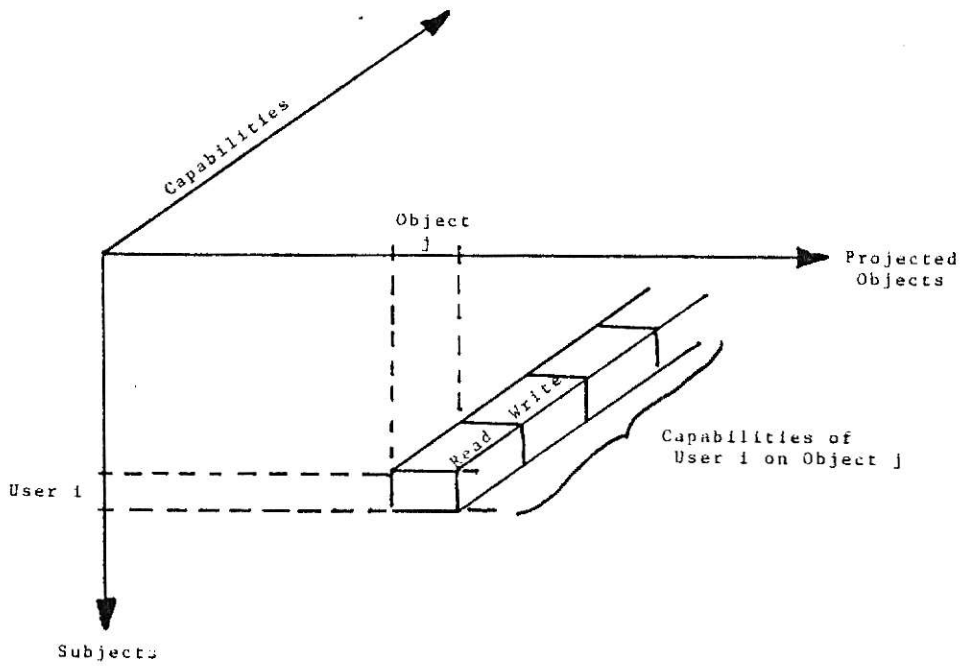


FIGURE 6

Specific techniques of implementing this information may vary but the information required seems to be fairly consistent from system to system. Using the matrix each requestor and each object have an access profile. An access is permitted when the profiles for the requestor and the object match. Three general access issues will be discussed. These are 1) access authorization design principles, 2) authorization checking at local and remote nodes and 3) component authorizations. [COLE78]

4.2.1 Access Authorization Design principles - These principles are widely accepted and will only be noted here.

1. Least Privilege - No requestor is granted access rights beyond that required to perform his function. This equates to the military term of "need to know". In addition, access to resources should be separated whenever the separation adds to security. This is the current solution for separating the OTHER category of information in the PODS. [JONE73]

2. Least Common Mechanism - Shared or common mechanisms must be kept to a minimum unless they are common mechanisms used expressly for security purposes. This further emphasizes the need to keep users separated from each other, and separated from resources unless they are required. [POPE74]

3. Reference Monitor Approach - The mechanisms used to control access must be: 1) always invoked, 2) isolated from unauthorized alteration, 3) proven to be reliable. The concept of the Network Security Center developed later is somewhat analogous to the reference monitor. [ANDE75]

4. Object versus Path Protection - In single systems the access mechanisms used generally

center on the user and the object. In networks there is a third area requiring protection - the path from the user to the object. In the PODS path protection is not as significant as in the current situation of attempting to interface several existing systems. [COLE78]

4.2.2 Authorization Checking at Local and Remote Nodes - When a requested resource is at the same node as the requestor the access authorization check can be made at that node by comparing the requestor's profile with the resource's profile. If the resource is not local to the user's node then two additional operations are needed: 1) locating the resource in the network, and 2) sending the user's profile to the node owning the resource or vice-versa. [COLE78]

4.2.3 Finding the resource can usually be resolved by one of two methods: 1) an explicit declaration of the location by the user - this is the easiest to implement but places an unnecessary burden on the user. In the case of PODS the user is already concerned with the battle, personnal safety, hunger, etc. and where possible the system should provide maximum services to him rather than requiring him to remember extra extraneous facts. 2) the resource is found by a table look-up in a network directory - this technique provides better service to the user. It also permits the system to manage resources during expansion and contraction of the network on the battlefield. The directory technique

however, does introduce additional overhead that slows response, complicates security mechanisms and creates a system that is nearly impossible to certify. [COLE78]

4.2.4 After the resource is located in the network a determination must be made as to where the access profile match will be done - at the user's location or at the resource's location. If the check is done at the resource's location then in essence the key has been taken to the lock. The same match takes place in either case however. Another possibility is to require the profile matches at both locations. It would then be possible to assure a double check on all remote requests. The overhead introduced with this technique would be prohibitive for the additional protection offered. Therefore a single match at the resource's node will be used in subsequent discussions.

4.2.5 Composite Authorizations - In any given transactions several resources may be required (a node, a program, data, etc). The user's profile must be matched to all of the resources required by the transactions. The lowest level of access for the set of resources used will then become the access for the transaction. For example if a user is authorized to write or update a data element but the program available has read only access (due to node availability) then the user is advised that at the present

time, read only is permitted but update is not. The user can then decide what action he will or will not take. The special case of a user s access being lower than any of the resources is discussed in the surveillience section.

4.3 Surveillance - The objective of any surveillance mechanism is to assure that a security threat can be detected and an appropriate reaction triggered. The mechanism provides a means of personal accountability on the system. It also can provide defenses against detected intrusions. Some of the requirements for a surveillance mechanism include:

a. Recognition of predesignated events
b. Invocation of predesignated reactions to specified events
c. Collection of predesignated information (journalling)
d. Provision for ready access to surveillance data. [OCRE78]

4.3.1 - Predesignated Events - These events include any transaction requiring a user profile/resource profile match. The match itself is not the trigger to invoke the surveillance mechanism. The two are independent. This independence is necessary to detect attempts to bypass the profile match. The predesignated events may be defined such that both the match and the surveillance mechanisms are invoked or the surveillance mechanism is invoked. The definition of these events must be done through a protected

interactive means that can be used quickly and easily. This capability will be essential if an intrusion is suspected or confirmed. Additional events may need to be added quickly to bound the possible compromise.

4.3.2 Reactions to Specified Events - A variety of reactions should be available depending on the event that has triggered the reaction. An unauthorized access that has resulted in no success may require a different reaction than one that has actually retrieved or changed data. The reactions that could be used include 1) normal journalling and execption listing, 2) real-time alert to security and command personnel, 3) termination of the offending user, 4) automatic invocation of a "special" process to allow the offender to continue and yet receive no real data.

4.3.3 Event Journal - The journal is used to collect specified information whenever the surveillance mechanism is invoked. The journal record provides the commander and his security representative with the history necessary to account for the security of the system and to refine the definition of predesignated events as necessary. The journal should contain at a minimum:

a. Identifiers of all involved elements (users, resources, etc.)
b. Nature of the event
c. Indication of success of the event, if known
d. Event data such as date, time,

classification, etc. [OCRE78]

The software that performs the journalling function will be critical to the entire system. If it is not properly designed the overhead incurred will be reflected in overall system performance most of the time. The nature of the journalling function dictates that it be totally transparent to the user, require little if any delay in a user's requirement, and be totally reliable. This is impossible to achieve when existing systems are bound into a new network. If a network is designed as a network then the overhead will not be noticed as the initial response time to the user will already include the journal overhead. This is one reason that users generally are not satisfied with networks that have been created from existing systems and the security mechanisms are overlaid on the new network.

4.3.4 - Command Inspection/Review - The data in the journal must be available to command and security personnel in two forms. Routine batch reports are necessary to provide continuing routine management of the surveillance function. More importantly, the journal must be available on an interactive basis to permit rapid reaction to a detected event. The interactive basis must be supported by a query language that will permit rapid pattern analysis to isolate a potential offender quickly. This capability should be

present in the intelligence applications of PODS.

4.3.5 Automatic Invocation of "Special" process - Where possible the automatic invocation of a "special" process would serve to keep an offender on the system until appropriate action can be taken. This provides the real capability to not only detect but neutralize an offender and yet protects the security mechanisms themselves. If an offender is immediately detected and forced off the system he has gained a certain amount of knowledge about the security mechanisms protecting the system. If however, he is permitted to operate (with valid or false data) he then does not recognize that he has been detected and gains no knowledge of the security mechanism he has invoked.

4.3.6 Other Uses - The journal also provides an excellent data base of management information on the overall network itself. Refinements on data allocation, routing strategies, and priorities may all be accomplished based on the data collected through the journal process.

4.4 System Integrity - An essential part of system security is system integrity, the condition of correct and predictable functioning of the total system. [OCRE78] This includes hardware, software, and operating procedures and policies. It also specifies that data integrity is maintained under normal and abnormal operating conditions.

The notion of system integrity is a link between the needs for security, resiliency and independence in the PODS. Integrity of procedures and policies is a well recognized but rarely achieved requirement. This report will note potential areas where some degree of hardware and software integrity can be achieved.

4.4.1 Hardware Integrity - Specific areas of hardware integrity include:

a. Error detection and correction capabilities such that no single element failure can go undetected.
b. Power failure detection so that alternate power sources are immediately accessed.
c. Positive unique device identification to support the necessary resource profiles discussed in access rights.
d. A positive clear capability to permit compartmented data to be moved and the device to be readily reused.
e. A line-break detection capability to permit the rerouting necessary to be accomplished readily. This capability was discussed more fully in the examination of system resiliency requirements. [OCRE78]

4.4.2 Software Integrity - Specific areas of software integrity include:

a. Formal proofs of program correctness while not currently feasible are still a primary desire to assure software integrity.
b. Operating system kernel structures that permit exclusive access and proven control of processes and resources.
c. Degradation due to invocation of a surveillance mechanism should be isolated to the process that created the need for the invocation. [OCRE78]

4.5 The Network Security Center - A possible solution to the security problems previously discussed is the concept of a Network Security Center. Usage of one or more Network Security Centers (NSCs) permits global access control and journalling as well as a "security interface" between each node and the network. This interface includes a cryptographic device that guarantees end-to-end security rather than link-to-link security. This capability offers the following additional capabilities:

a. Information is protected in intermediate switches as well as on the communication links. This minimizes the verification concerns for the individual switches.

b. Any misdelivered messages are unintelligible to the receipient. This is a good defense against an enemy force attempting to enter the network and gain intelligence. An audit of misdelivered messages provides a security manager with a powerful tool to trace possible compromises.

c. Access control by user and device is facilitated. If the proper key is not used the user does not gain access to the device. [HEIN78]

The role of the NSC is displayed in Figure 7.

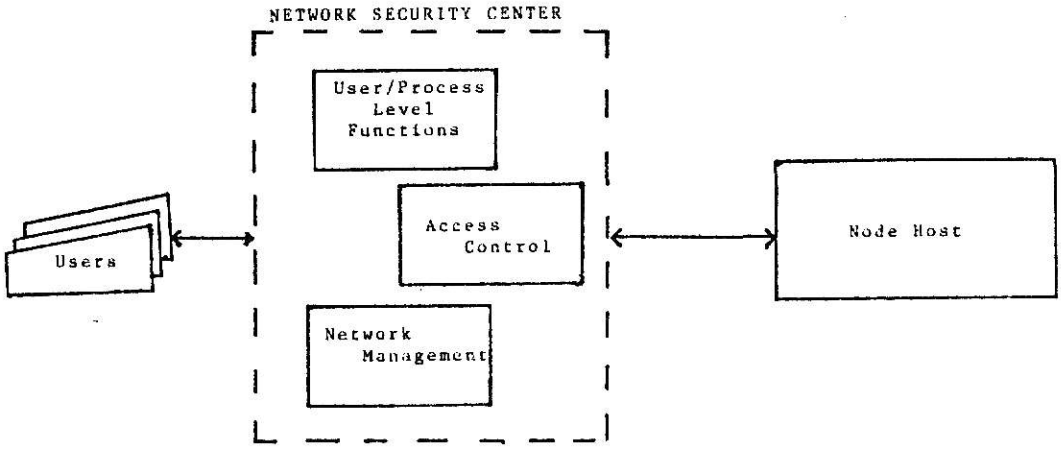


FIGURE 7

A user must always initially connect to a NSC to acquire access to the network itself. [HEIN78] Subsequent operations will perform access and audit functions for the individual processes used by a user, but the initial access verification of a user is only done periodically during a 24 hour period. An access may originate from either a human user or from a computer system that is trying to satisfy a request from a user. In the former case, the NSC carries on an interactive dialogue with the user, requesting the necessary identifiers and/or authenticators. For computer requests these items are rigidly formatted and are sent as a single request after all required connections have been made. Some of the principal benefits of the NSC are outlined below:

4.5.1 Multilevel Security - The NSC provides the capability for logical subnets to share the same physical communications network. The different user subnets can be kept isolated and hence secure from one another, since the NSC controls who can communicate with whom. The NSC also controls the access to different physical devices. This permits the storage of different classes of data on different devices and insures that access to the devices is based on a positive match through the NSC. In practice there is the potential for more than one NSC at a given

node. Each NSC would control access to a given subset of devices and/or users. All requested accesses would initially go to the node NSC and then to the appropriate NSC for the class of the request. Figure 8 is mapping of this capability onto PODS.

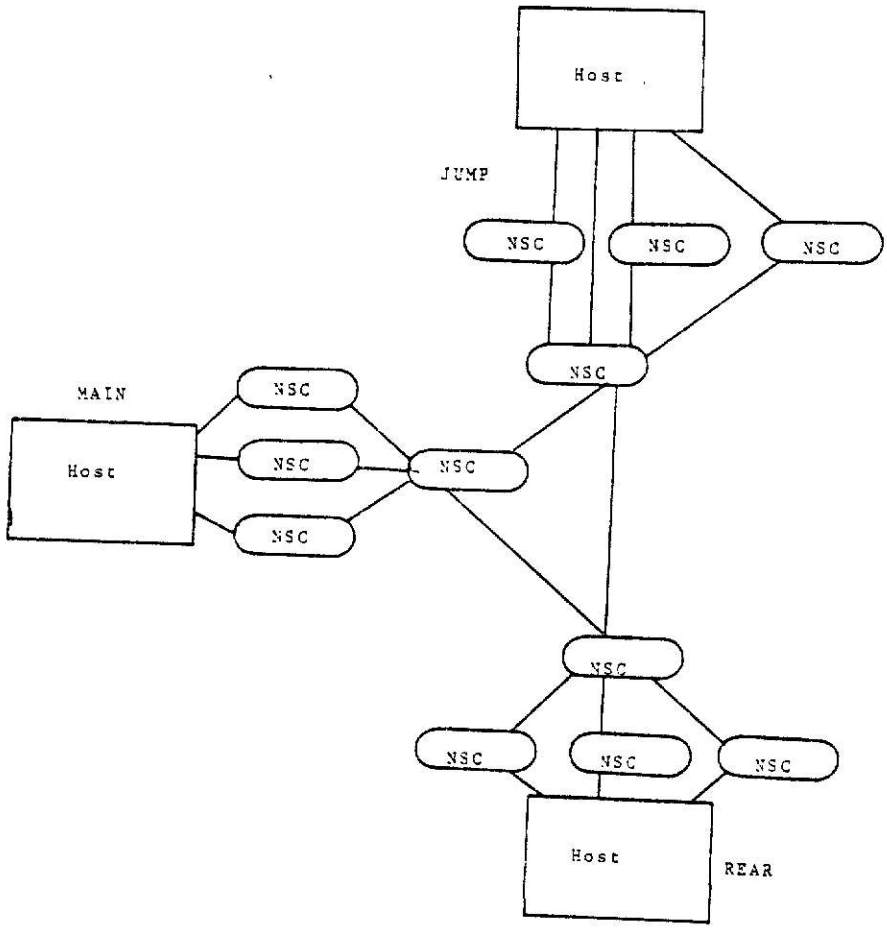


Figure 8

4.5.2 Granularity - The NSC offers the capability to define granularity of access rights as finely as is needed. Since the NSC is separate from the hosts the degree of granularity used does not add additional overhead to the host itself.

4.5.3 NSC and System Resiliency - It is obvious that the concept of the NSC also offers a real potential to achieve the desired capability for system resiliency discussed in Chapter 3. Since the NSC is separate from the host it could easily accomodate reliable broadcast, failure detection, routing strategies, and control of the recuperating node. The operations for these functions would then be localized in a frontend machine. This would permit the host machines to meet the size and weight restrictions necessary for mobile tactical systems.

CHAPTER 5 - DATA INDEPENDENCE

5.0 As a concept data independence is fundamental to distributed processing. The ability to attain data independence is one of the primary reasons distributed processing is feasible today. One definition of data independence is:

> Data independence is the capability of a program to be isolated from its data environment, the data as stored, as it is shared by other programs, and as it is manipulated by the data base (system) manager to enhance performance.

The parentheses are mine. This isolation permits systems, nodes, programs, data, and users to be added to or deleted from a network without disrupting the network. This flexibility is obviously essential to a tactical system such as PODS. With nodes entering and leaving the network or being destroyed the system must attain a high degree of independence in order to function. This independence can be separated into classes based upon a given view of the system.

This chapter will address the following classes of independence: 1)Node Independence; 2)Machine Independence; 3)DBMS Independence; 4)Storage Device Independence; 5)Data

Structure Independence;

5.1 Node Independence - Complete node independence may not be possible for the PODS. Some information may not be permitted on certain nodes due to the classification of the information. If a node can only store up to SECRET information then TOP SECRET information cannot be stored on that node. The degree of node information that can be achieved will be determined by the presence or absence of a Network Security Center for that category of data. Some degree of node independence is then possible. Data belonging to a certain category can be resident on any node that possesses an NSC for that category. This independence will permit a system designer to incorporate requirements for replication and partitioning while maintaining the necessary resiliency and security needs.

5.2 Machine Independence - This property is also addressed as portability. It is a measure of the ease or difficulty of transferring data from one vendor's hardware to another vendor s hardware. In the case of PODS, machine independence may not be required since it is designed as a single integrated system and thus all associated hardware will be compatible. In the case of systems already in the field machine independence is critical. Little machine independence has been achieved to date because each system

has been developed independently. As a result no two systems use the same vendor hardware. This has created significant problems when it becomes necessary to interface two or more existing systems. The translation involved has already proven to be time consuming and expensive. PODS potential to circumvent the machine independence problems is significant.

5.3 DBMS Independence - This property is currently even more elusive than machine independence. Most DBMSs available today are supplied by vendors. The software and, in some cases, the data structures are proprietary. Therefore the capability to move from one DBMS to another without disrupting applications programs is not feasible. DBMSs are now being used in some tactical systems. The problems associated with a lack of DBMS independence have not yet been recognized within the military. In the case of PODS a single Distributed DBMS was envisioned. This will eliminate the need for DBMS independence within the PODS. However, the need for this independence external to the PODS remains and requires further work.

5.4 Storage Device Independence - This degree of independence is readily achievable today. System catalogs permit data to be stored on a variety of devices without affecting application programs. This does not include a

change in access method (sequential, random, ISAM, etc.). There is currently no method to achieve this independence as the change usually entails significant changes to the application programs.

5.5 Data Structure Independence - Changes to the data structure used that add elements generally can be accomplished without affecting current software. This is one of the strong properties of current DBMSs. However wholesale changes of the structure may still require modification of existing application programs. This degree of independence is still in the research area.

CHAPTER 6 - CONCLUDING REMARKS

6.0 The potential for distributed processing in tactical systems is real. Historically new technological advances have required years of testing before they were actually placed on a battlefield. The testing period was usually required to determine what impact the advance would have on the organization's structure as well as hardware reliability. Distributed processing should not require the same testing. Its inherent flexibility indicates that the system can be designed to match the organization rather than modifying the organization to fit the system. The flexibility comes from the strong potential to achieve higher degrees of data security and data independence in a distributed system.

6.1 Further Work - There is however, additional research and testing necesary before a system such as PODS can be fielded. The specific areas yet to be addressed are summarized below.

a. The concept of a Network Security Center requires testing. Both simulations and prototypes will be required to determine the feasibility of such a concept. The testing must address the NSC s capability to maintain the different subnets separately. The potential for solving the

multi-level security problem is particularly important for tactical systems.

b. The concept of a recuperating node also requires testing, ;both by simulation and prototype. A determination is required as to the overhead required by a recuperating node while it is in the recuperation process. Can the process be accomplished in a timely manner or is more research required?

c. Once the NSC and recuperating node concepts have been tested then the PODS itself should be simulated. It offers a commander flexibility that is not available today.

REFERENCES

ADIB78 Adiba, M., et. al., "Issues in Distributed Data Base Management Systems: A Technical Overview," Proc. Fourth International Conference on Very Large Data Bases, October 1978, pp. 89-110.

ANDE78 Anders, F., et. al., "Intelligence Security Subsystem--Final Technical Report," Harris Corporation, Electronic System Division, March 1978.

ANDE75 Anderson, D.R., "Data Base Processor Technology," Proc. AFIPS NCC 1975 Vol. 44, pp. 389-395.

ANDE75A Anderson, George A. and Jensen, E. Douglas, "Computer Interconnection Structures: Taxonomy, Characteristics, and Examples," Computing Surveys, Vol. 7, No. 4, December 1975, pp. 197-213.

ASCH75 Aschim, F.,"Data Base Networks--An Overview," Management Informatics, Vol. 3, No. 1, February 1974, pp. 12-28.

BANE79 Banerjee, Jayanta and Hsiao, David K., "DBC--A Database Computer for Very Large Databases," IEEE Transactions on Computers, Vol. C-28, No. 3, March 1979.

BECK78 Becker, Hal B., "Let's Put Information Networks Into Perspective," DATAMATION, Vol. 24, No. 3, March 1978, pp. 81-86.

BELF75 Belford, Geneva G., et.al., "Initial Mathematical Model Report-- Research in Network Data Management and Resource Sharing," Center for Advanced Computation (CAC) Document No. 169, University of Illinois at Urbana-Champaign, Urbana, Illinois.

BELF75A Belford, Geneva G., "Network File Allocation--Research in Network Data Management and Resource Sharing," Center for Advanced Computation (CAC) Document No. 203, University of Illinois at Urbana-Champaign, Urbana, Illinois.

BELF75B Belford, Geneva G., "Technology Summary: Research in Network Data Management and Resource Sharing," Center for Advanced Computation, University of Illinois at Urbana-Champaign, Urbana, Illinois, May 1975.

BERK78 Berkowitz, B., "A Distributed Database Management System for Command and Control Applications--Semi-annual Technical Report III," Computer Corporation of America, July 30, 1978.

BERN77 Bernstein, Philip A., et. al., "A Distributed Database Management System for Command and Control Applications--Semi-annual Technical Report I," Computer Corporation of America, July 30, 1977.

BERN78 Bernstein, Philip A., et. al., "The Concurrency Control Mechanism of SDD-1: A System for Distributed Databases (The Fully Redundant Case)," IEEE Transactions on Software Engineering, Vol. SE-4, No. 3, May 1978, pp. 154-168.

BERN78A Bernstein, Philip A. and Shipman, David W., "A Formal Model of a Concurrency Control Mechanism for Data base Systems," Proc. Berkley Workshop on Distributed Data Bases and Computer Networks, September 1978.

BERR77 Berra, P. Bruce, "Data Base Machines," SIGIR Newsletter, Winter 1977, pp. 4-23.

BOOT72 Booth, G. M., "The Use of Distributed Data Bases in Information Networks," Proc. First International Conference on Computer Communication: Impacts and Implications, October 1972, pp. 371-376.

BOOT76 Booth, G. M., "Distributed Information Systems," Proc. AFIPS NCC 1976, Vol. 45, pp.789-794.

BURR77 Burr, William E. and Gordon, Robert, "Selecting a Military Computer Architecture," Computer, Vol. 10, No. 10, October 1977, pp. 16-23.

CANA74 Canaday, R. E., et. al., "A Backend Computer for Data Base Management," Communications of the ACM, Vol. 17, No. 10, October 1974, pp. 575-582.

CATE78 Cate, Paul E. (Major), "Large-Unit Operational Doctrine," Military Review, Vol. LVIII, No. 12, December 1978, pp. 40-47.

CHAM77 Champine, George A., "Six Approaches to Distributed Data Bases," DATAMATION, Vol. 23, No. 5, May 1977, pp. 45-48.

CHAM78 Champine, George A., "Four Approaches to a Data Base Computer," DATAMATION, Vol. 24, No. 13, December 1978, pp. 101-106.

CODE78 Codespoti, D. J. and Maryanski, F. J., "A Microprocessor Distributed Operating System for a Minicomputer," Computer Science Department, Kansas State University, TR CS 78-09, March 1978.

COLE77 Coleman, Aaron H. and Smith, William R., "The Military Computer Family: A New, Joint-Service Approach to Military Computer Acquisition," Computer, Vol. 10, No. 10, October 1977, pp. 12-15.

COLE78 Cole,Gerald D., "Design Alternatives for Computer Network Security", NBS Special Publication 500-21, Vol 1, January 1978.

COTT77 Cotton, Ira W., "Computer Network Interconnection: Problems and Prospects," NBS Special Publication 500-6, April 1977.

DEPP76 Deppe, M. E. and Fry, J. P., "Distributed Data Bases: A Summary of Research," Computer Networks, Vol. 1, No. 2, February 1976.

ELOV74 Elovitz, Honey S. and Heitmeyer, Constance L., "What Is a Computer Network," IEEE 1974 NTC Record, pp. 149-156.

ENSL78 Enslow, Philip H., Jr., "What is a Distributed Data Processing System?", Computer, Vol. 11, No. 1, January 1978, pp. 13-21.

FISH76 Fisher, Paul S., Hankley, William J., and Maryanski, Fred J., "porting Software to Multiple Minis: A DBMS Case Study," Computer Science Department, Kansas State University, TR CS 77-12, December 1976.

FISH77 Fisher, Paul S. and Maryanski, Fred J., "Design Considerations in Distributed Data Base Management Systems," Computer Science Department, Kansas State University, TR CS 77-08, April 1977.

FISH78 Fisher, Paul S. and Maryanski, Fred J., "Concepts and Problems in Distributed Data Base Management Systems," Computer Science Department, Kansas State University, TR CS 78-17, February 1978.

GREE77 Greene, William (Major) and Pooch, Udo W., "A Review of Classification Schemes for Computer Communications Networks," Computer, Vol. 10, No. 11, November 1977, pp. 12-21.

HARD78 Hardgrave, W. T., "Distributed Database Technology: An Assessment," Information & Management 1978, pp. 157-167.

HEIN78 Heinrich, Frank, "The Network Security Center: A System Level Approach to Computer Network Security", NBS Special Publication 500-21, Vol 2, January 1978.

HENS78 Henson, Carle C., et. al., "Preliminary Design Study for VTS Processing Display Subsystem (Final Report)," International Computing Company, June 1978.

HILS79 Hilsman, William J. (Major General), "C3I Communications Vital in Integration of the Force-Multipliers," ARMY, Vol. 29., No. 3, March 1979, pp. 31-33.

HSIA77 Hsiao, D. K., Kanan, K., and Kerr, D. S., "Structure Memory Design for a DataBase Computer," Proc. ACM 77, December 1977, pp. 343-350.

HSIA78 Hsiao, David K., Kerr, Douglas S., and Madnich, Stuart E., "Privacy and Security of Data Communications and Data Bases," Proc. Fourth International Conference on Very Large Data Bases, October 1978, pp. 55-67.

HSIA79 Hsiao, David K., "Database Machines Are Coming, Database Machines Are Coming," Computer, Vol. 12, No. 3, March 1979, pp. 7-9.

JONE73 Jones, A. K., "Protection Structures", PhD Thesis, Carnegie-Mellon University, 1973.

KATZ78 Katzan, Harry Jr., An Introduction to Distributed Data Processing, New York: Petrocelli Books, Inc., 1978.

KENT78 Kent, Stephen T., "Network Security: A Top-Down View Shows Problem," DataCommunications, Vol. 7, No. 6, June 1978, pp. 57-75.

KIRK78 Kirkley, J. L., "Happiness Is Distributed Processing," DATAMATION, Vol. 24, No. 3, March 1978, p. 79.

KLEI76 Kleinrock, Leonard, "On Communications and Networks," IDEEE Transactions on Computers, Vol. C-25, No. 12, December 1976, pp. 1326-1335.

KUNI77 Kunii, T. L. and Kunii, H. S., "Design Criteria for Distributed Database Systems," Proc. Third International Conference on Very Large Data Bases, October 1977, pp. 93-104.

MAHA79 Mahaffey, Fred K. (Major General), "C3I for Automated Control of Tommorrow's Battlefield," ARMY, Vol. 29, No. 3, March 1979, pp. 26-30.

MARY76A Maryanski, Fred J., Fisher, Paul S., and Wallentine, Virgil E., "Evaluation of Conversion to a Back-end Data Base Management System," Computer Science Department, Kansas State University, TR CS 76-08, March 1976.

MARY76B Maryanski, Fred J. and Wallentine, Virgil E., "A Simulation Model of a Back-end Data Base Management System, Proc. Pittsburgh Conference on Modeling and Simulation, April 1976, pp. 243-248.

MARY76C Maryanski, Fred J., et. al., "A Minicomputer Based Distributed Data Base Management System," Proc. IEEE NBS Trends and Applications Symposium: Micro and Mini Systems, May 1976, pp. 113-117.

MARY76D Maryanski, Fred J., "Language Specification for a Distributed Data Base Management System," Computer Science Department, Kansas State University, TR CS 76-13, May 1976.

MARY76E Maryanski, Fred J., "Memory Management in Distributed Data Base Systems," Computer Science Department, Kansas State University, TR CS 76-14, October 1976.

MARY77A Maryanski, Fred J., "A Survey of Developments In Distributed Data Base Management Systems," Computer Science Department, Kansas State University, TR CS 77-01, January 1977. (Another version of this TR can be found in Computer, Vol. 11, No. 2, pp. 28-38.)

MARY77B Maryanski, Fred J., "A Deadlock Prevention Algorithm for Distributed Data Base Management Systems," Computer Science Department, Kansas State University, TR CS 77-02, February 1977.

MARY77C Maryanski, Fred J. and Fisher, Paul S., "Rollback and Recovery in Distributed Data Base Management Systems," Computer Science Department, Kansas State University, TR CS 77-05, February 1977.

MARY77D Maryanski, Fred J., "Performance of Multi-Processor Back-end Data Base Systems," Computer Science Department, Kansas State University, TR CS 77-07, April 1977.

MARY77E Maryanski, Fred J., et. al., "Distributed Data Base Management Using Minicomputers," Computer Science Department, Kansas State University, TR CS 77-22, October 1977.

MARY77F Maryanski, Fred J. and Kreimer, Daniel E., "Effects of Distributed Processing in a Data Processing Environment," Computer Science Department, Kansas State University, TR CS 77-23, November 1977.

MARY78A Maryanski, Fred J., Norsworthy, Kevin E., and Norsworthy, Kirk S., " A System Architecture for Distributed Data Base Management," Computer Science Department, Kansas State University, TR CS 78-15,

April 1978

MARY78B Maryanski, Fred J., "Distributed Data Base Management Systems," Computer Science Department, Kansas State University, TR CS 78-19, July 1978.

MARY78C Maryanski, Fred J. and Nikravon, Nasrin, "Simulation of a Functionally Distributed Computing Facility: Central System Model," Computer Science Department, Kansas State University, TR CS 78-29, July 1978.

MARY78D Maryanski, Fred J., "The Management of Redundant Data in a Distributed Data Base," Computer Science Department, Kansas State University, TR CS 78-21, September 1978.

MARY78E Maryanski, Fred J., Fisher, Paul S., and Wallentine, Virgil E., "Data Access in Distributed Data base Management Systems," Computer Science Department, Kansas State University, TR CS 78-14, December 1978.

MARY78F Maryanski, Fred J. et. al., "Distributed Data Base Management Using Minicomputers," INFOTECH State of the Art Report Minis Verses Mainframes, 1978, pp. 141-157.

MARY79 Maryanski, Fred J., et. al., "A Prototype Distributed DBMS," Computer Science Department, Kansas State University, TR CS 78-08, January 1979.

MILL78 Miller, Myron, "A Survey of Distributed Data Base Management," Information & Management 1978, pp. 243-264.

MORG77B Morgan, D. E., Taylor, D. J., and Custeau, G., "A Survey of Methods for Imporoving Computer Network Reliability and Availability," Computer, Vol. 10, No. 11, November 1977, pp. 42-50.

MORG77A Morgan, Howard L. and Levin, K. Dan, "Optimal Program and Data Locations in Computer Networks," Communications of the ACM, Vol. 20, No. 5, May 1977, pp. 315-322.

ORCE78 Orceyre, Michel j., and Courtney, Robert H., "Considerations in the Selection of Security Measures for Automatic Data Processing Systems", NBS Special Publication 500-33, June 1978.

ORNS72 Ornstein, S. M., et al, "The Terminal IMP for the ARPA Computer Network", AFIPS Conference Proceedings, Vol 49, 1972.

POPE74 Popek, G. J., and Kline, C. S,., "Verifiable Secure Operating System Software", NCC, 1974.

RAMA77 Ramamoorthy, G. S., Ho, T. K., and Wah, B. W., "Architectural Issues in Distributed Data Base Systems," Proc. Third International Conference on Very Large Data Bases, October 1977, pp. 121-126.

ROTH77 Rothnie, James B. and Goodman, Nathan, "An Overview of the Preliminary Design of SDD-1: A System for Distributed DataBases," Proc. 2d Berkley Workshop on Distributed Data Management and Computer Networks, May 1977, pp. 38-57.

ROTH77A Rothnie, James B. and Goodman, Nathan, "A Survey of Research and Development in Distributed Database Management," Proc. Third International Conference on Very Large Data Bases, October 1977, pp. 48-61.

ROTH77B Rothnie, James B., Goodman, Nathan, and Bernstein, Philip A., "The Redundant Update Algorithm of SDD-1: A System for Distributed Databases (The Fully Redundant Case)," First International Conference on Computer Software and Applications, November 1977.

SAND78 Sanders, Ray W., "Comparing Networking Technologies," DATAMATION, Vol. 24, No. 7, July 1978, pp. 88-94.

SCHU79 Schultz, Brad, editor, "Special Report--The Move to Distributed Processing," COMPUTERWORLD, July 1979.

SHEP78 Shepherd, Alan J., "A British Example of Distributed Computing," DATAMATION, Vol. 24, No. 3, March 1978, pp. 87-91.

SHEP77 Shepherd, Mark Jr., "Distributed Computing Power: A Key to Productivity," Computer, Vol. 10, No. 11, November 1977, pp. 66-74.

SLON77 Slonim, Jacob, Unger, Elizabeth A., and Fisher, Paul S., "Data Base Management System Environments Present and Future," Computer Science Department, Kansas State University, May 1977.

SLON78 Slonim, Jacob, Schmidt, David, and Fisher, Paul, "Considerations for Determining the Degree of Centralization or Decentralization in the Computing

Environment," Computer Science Department, Kansas State University, May 1978.

STRA79 Straut, Robert P., "Fighting Outnumbered and Winning," Military Review, Vol. LIX, No. 5, May 1979.

STAR79 Starry, Donn A. (General), "A Tactical Evolution--FM 100-5," Military Review, Vol. LVIII, No. 8, pp. 2-11.

STUB79 Stubblebine, Albert N.,III (Brigidier General), "C3I for Automated Focus on Intelligence Picture," ARMY Vol. 29, No. 3, pp. 34-36.

TAJI77 Tajibnapis, William D., "A Correctness Proof of a Topology Information Maintenance Protocol for a Distributed Computer Network," Communications of the ACM, Vol. 20, No. 7, July 1977, pp. 477-485.

UNGE79 Unger, Elizabeth A., et. al., "Integration of a DBMS into a Network Environment," Computer Science Department, Kansas State University, April 1979.

VAN 79 Van Rensselaer, Cort, "Centralize? Decentralize? Distribute?", DATAMATION, Vol. 25, No. 4, April 1979, pp. 88-97.

WALD77 Wald, Bruce and Salisbury, Alan, "The Computer Family Architecture Project: Service Perspectives and Overview," Computer, Vol. 10, No. 10, October 1977, pp. 8-11.

WALL77 Wallentine, Virgil E., "Project Report for Functionally Distributed Computer Systems Development: Software and Systems Structure, Part I," Kansas State University, October 1977.

WALL76 Wallentine, Virgil E., et. al., "MIMICS Design Overview Functionally Distributed Computer Systems Development: Software and System Structure," Computer Science Department, Kansas State University, TR CS 77-4, December 1976.

WALL75 Wallentine, Virgil E. and Maryanski, Fred J., "Implementation of a Distributed Data Base System," Computer Science Department, Kansas State University, TR CS 75-11, November 1975.

WHIT76 White, James E., "A High-Level Framework for Network-Based Resource Sharing," Proc. AFIPS NCC 1976, Vol. 45, pp. 561-570.

YAMA74 Yamaguchi, K., and Merten, A.G., "Methodology for Transfering Programs and Data," ACM-SIGMOD Workshop on Data Description, Access and Control, May 1974, pp. 141-155.

APPENDIX A

Comander - The individual who controls and is responsible for the 16,000 man division.

Deputy CG - Two general officers that are used in various roles chosen by the commander. They may perform line or staff roles.

Chief of Staff - The manager of the staff units. Controls and coordinates the various staff units.

Staff Elements - Perform planning tasks in their respective areas The "other" block includes such specialized staff officers as the Division Surgeon and the Division Chaplain.

Signal Battalion - Provides long range communications to the division on an area basis. This includes data, record traffic and messenger service.

Engineer Battalion - Provides construction and barrier support to all units of the division.

Intelligence Battalion - Provides systems for collection and processing of intelligence information for the division.

Brigade Headquarters - An intermediate line entity that controls 2-5 battalions depending on the situation.

Division Artillery - The command and control unit that coordinates all fire support. This includes the artillery battalions, air strikes and naval gunfire.

DISCOM - Division Support Command controls the supply, maintenance and medical activities available to support the division as a whole.

Battalion - The basic line building block of the division. Usually contains 600-800 men and associated equipment.

Arty Battalion - A line unit that provides artillery fire support to a brigade.

Support Units - A collection of maintenance, supply and medical battalions that perform support roles for the division as a whole.

APPENDIX B

VULNERABILITY CATEGORY DEFINITIONS

1. Unauthorized Modification of Data Internal to the System (DMI) Vulnerabilities include unauthorized modification of data residing within the system. This includes insertion of new data as well as modification or deletion of existing data.

2. Unauthorized Destruction of Data Internal to the System (DDeI) Vulnerabilities include unauthorized destruction of data stored on the system. The destruction in this case is intentional as opposed to accidental.

3. Unauthorized Disclosure of Data Stored Internal to the System (DDiI) Vulnerabilities include unauthorized disclosure of existing data, using existing software and/or system facilities.

4. Unauthorized Modification of Programs Internal to the System (PMI) Vulnerabilities include insertion of new program modules and modification or deletion of existing programs by using an application system, system programs, or system facilities.

5. Unauthorized Modification of Data External tothe System (DME) Vulnerabilities include unauthorized physical modification of data stored or handled externally. Operations such as data preparation, insertion and modification of new data, and deletion of existing data are included.

6. Unauthorized Modification of Programs External to the System (PME) Vulnerabilities include any unaithorized modification of a program stored outside the system proper. Insertion os cards into existing decks, replacing tapes or disks are examples.

7. Unauthorized Modification of Computer Equipment or Supplies (CE&SM) Vulnerabilities include insertion of new elements, substitution of one element for another and modification of existing elements.

DATA SECURITY AND DATA INDEPENDENCE IN A MOBILE MILITARY SYSTEM

by

WILLIAM PAUL AKINS

B.A., San Francisco State University, 1972

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas
1980

This report focuses on a theoretical basis for achieving data security and data independence in networks currently being developed by the U. S. Army. While the report is centered on military systems, the discussions are equally applicapable to any mobile network. Use of a network is assummed.

A model, the Postulated Division System (PODS), is established as a paradigm in order to examine the needs for system resiliency, data security and data independence in such a system. PODS incorporates the capabilities of several systems that are currently under development by the Army. It provides support for command and control, intelligence and logistics functions.

A definition for system resiliency is developed. This includes an explanation of the components that make up system resiliency. Military requirements for system resiliency are high. The dynamics of a modern battlefield dictate frequent movement. This in turn indicates that the entire network will be available a small percentage of the total time. A concept of a "Recuperating Node" is developed to provide a rapid but controlled means to restore a node to a network.

Discussions of data security include access rights, and methods to provide positive control of access to both data and devices by various classes of users. The use of multiple Network Security Centers is developed. These centers provide a means to manage security without introducing an unnecessary burden on the host computers. The Network Security Center also provides a potential for managing separate compartments of information on the same physical set of hardware.

Discussions on data independence are directed to the potential to achieve such independence if the needs for system resiliency and data security are already met.

The report concludes with specific suggestions for further research and testing. Specific areas are identified that will require testing by simulation and by construction of prototype systems.