MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DISTRIBUTED COMPUTER SYSTEMS
FOR
THE REPUBLIC OF TURKISH NAVY

by

Haldun Pelit

December 1985

Thesis Advisor: Barry Frew

Approved for public release; distribution is unlimited

86   2   28   021

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A164722	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Distributed Computer Systems for The Republic of Turkish Navy		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1985
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Haldun Pelit		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100		12. REPORT DATE December 1985
		13. NUMBER OF PAGES 86
14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution is unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

The Republic of Turkish Navy (RTN), Distributed Computer Systems (DCS), Network Operating System (NOS), Distributed Data Base (DDB) Distributed Operating System (DOS)

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The aim of this thesis is to introduce the concept of distributed computer systems to the Republic of Turkish Navy for its new computer systems. The new computer system is planned to provide data processing facilities to the commands spread over a 20km area at Golcuk. The main elements (networks, operating systems, file servers) of distributed computer systems are explained as well as the concept. The recommended system is a collection of super minis to which a number of personal (Continued)

DD $_{1}^{FORM}{}_{JAN\ 73}$ 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

ABSTRACT (Continued)

computers are connected. A pilot system and a design methodology are described to define and test the system and the user requirements for the new computer system.

Accession For

NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	

By
Distribution/
Availability Codes
Avail and/or
Dist   Special

A-1

Distributed Computer Systems
for
The Republic of Turkish Navy

by

Haldun Pelit
Lieutenant J.G. Turkish Navy
B.S., Turkish Naval Academy, 1978

Submitted in partial fulfillment of the
requirements for the degree of
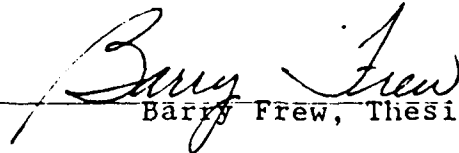
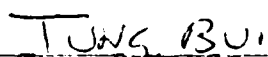MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
December 1985
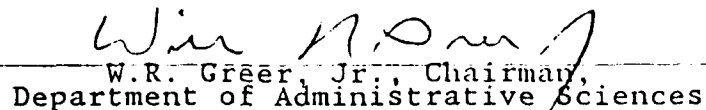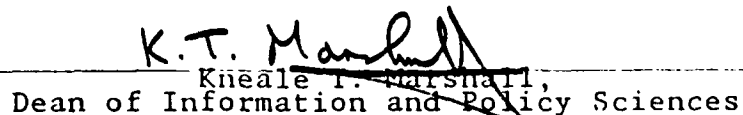
Author: _____
Haldun Pelit

Approved by: _____
Barry Frew, Thesis Advisor

_____
Tung Bui, Second Reader

_____
W.R. Greer, Jr., Chairman,
Department of Administrative Sciences

_____
Kneale T. Marshall,
Dean of Information and Policy Sciences

3

ABSTRACT

The aim of this thesis is to introduce the concept of distributed computer systems to the Republic of Turkish Navy for its new computer system. The new computer system is planned to provide data processing facilities to the commands spread over a 20km area at Golcuk, The main elements (networks, operating systems, file servers) of distributed computer systems are explained as well as the concept. The recommended system is a collection of super minis to which a number of personal computers are connected. A design methodology and a pilot system are described. Also, this thesis defines the user requirements for the new computer system.

4

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

Distributed computer systems have been widely discussed in the context of computer architecture. The advent of wide area networks, such as ARPANET allowed trials involving large computers, distributed databases and distributed resources. The combination of local area networks and micro computer advancements are allowing the concept of distributed computer systems to be explored in a practical manner. The goal of this thesis is to explain the concepts of distributed computer systems based on mini and micro (personal) computers, as a design alternative for computer systems for Republic of Turkey Navy (RTN).

Considering distributed computer systems, will be a new era for the RTN, because in the RTN, distributed computer systems or networked computer systems have never been implemented. All RTN data processing work is done by two separate multi user computer systems. Those computers are used to support the supply system of the RTN at two different sites. One is at the capitol of Turkey (Ankara), another is at the main navy base (Golcuk).

Increasing data processing needs of the RTN force the Chief of Navy to consider building a third computer system at Golcuk which will provide data processing facilities to three fleets, a shipyard, and several ground facilities spread over a 20km area. Traditionally, The Chief of Navy considers new multi user systems because of existing knowledge and experience level. Difficulties including cost of connections for multi-user computer systems in a dispersed organization may be reasons to consider, distributed computer systems as a replacement. This thesis introduces distributed computer systems and proposes a design methodology to the Chief of Turkish Navy.

While presenting the idea of the distributed systems to the Chief of Turkish Navy, this thesis will concentrate on the concept of distributed computer systems and how mini and micro computers can be used.

Several fundamental driving forces support the use of mini and micro computers in a distributed computer system architecture for the RTN's new computer system. For example, the rapidly improving architecture of mini and micro computers, the changing communication technology, the increasing capacity of secondary storage devices, and the software technologies state of art; have made mini and micro computers easily usable, powerful and cheap in distributed computer systems.

Chapter II describes the main parts of the distributed computer systems including networks, network operating system, distributed operating systems and file servers. Some applications of distributed systems are also described.

Chapter III explains general concepts and issues of distributed systems which are common for each different size computer from personal computers to mainframe computers. To understand the concept of distributed computer systems is an important step to solving the design problems of the system.

Chapter IV concentrates on the selection and design methodology for distributed computer systems. The design methodology based on user needs leads to conceptionalization of a pilot (prototype) model before building the real system.

II. ELEMENTS OF DISTRIBUTED COMPUTER SYSTEMS

A Distributed Computer System (DCS) is a collection of processor-memory pairs connected by a communications subnet, logically integrated in varying degrees. The communication subnet may be a geographically dispersed collection of communication processors or a Local Area Network (LAN).

DCS covers many areas including the management of communications, operating systems, distributed database systems, concurrency, fault toleration, ultra-reliable systems, real-time systems, artificial intelligence, and cooperative problem solving techniques

This chapter will concentrate on showing the relationships between DCS and mini and micro (personal) computers.

In general, DCS consist of four parts: LAN or Wide Area Network (WAN), Network Operating System (NOS), Distributed Operating system (DOS), and Distributed file servers. Applications such as Distributed Real-time systems and Distributed Databases (DDB) are explained below. They actually refer to a particular aspect of a DCS and not the entire DCS.

A. LOCAL AREA NETWORK

According to Stalling [Ref. 1] "a local area network is a communication network that provides interconnection of a variety of data communicating devices within a small area". A small area generally refers to a single building or possibly several buildings. A network with a radius of 20km would border between LAN and WAN. LAN's are sometimes classified into three types: a LAN, a high Speed Local Network (HSLN) which is typically found in a large computer center

11

and a Digital Switch/Computerized Branch Exchange (CBX), (HSLN and CBX will not be considered in the thesis).

LAN's have been designed within the frame work of the international organization for standardization's (ISO) reference model for open system interconnection (OSI) to describe the functionablity of a computer network in a structured and layered manner. The ISO model contains seven layers [Ref. 2]. The functions of all layers are explained in Appendix A. The general design criteria provided by OSI layers is illustrated in Table I.

Main considerations of LAN include network topology, network protocols and network performance evaluation.

1. Network Topology

The principal technology alternatives that determine the nature of a local network are topology and communication medium (cable), which refer to the way in which the end points or nodes of the network are interconnected. A topology is defined by the layout of communication links and switching elements and it determines the path that may be used between any pair of node.

There are four main topologies: Bus, tree, ring, and star. In the star topology, each node is connected by a point-to-point link to a common control switch. Communication between any two nodes is via circuit switching. This topology exhibits a centralized communication control strategy. In the ring topology, the local network consists of a set of repeaters joined by a point-to-point uni-directional link in a closed loop. Control is needed to determine at what time each node may insert packets. In the bus topology, there are no switches and no repeaters. All nodes are attached through appropriate hardware, interfacing directly to a linear communication medium. The tree topology is a generalization of the bus topology.

12

TABLE I
GENERAL DESIGN CRITERIA PROVIDED BY OSI LAYERS

LAYERS	7 APPLICATION	6 PRESENTATION	5 SESSION	4 TRANSPORT	3 NETWORK	2 DATA LINK	1 PHYSICAL CONTROL
DESIGN ISSUES							
Connection establishment	X	X	X	X			X
Rules of info transfer				X	X		
Order of messages				X	X	(X)	
Integrity of info transfer			X		X		
Handling of long messages			X		(X)		
Adaptation of fast sender to slow receiver			X			X	
Security	X	X					
Routing Addressing				X	X	X	
Path selection				(X)	X		
Load leveling	X			(X)			
Priorities	X			X			
Recovery						X	
Error control				X	X	X	X
Logging			X	(X)	(X)		
Sharing connections			X	X	X	X	
Network management			X		(X)		

13

communication medium is a branching cable with no closed
loops.

The medium is the physical path between transmitter
and receiver in a communication network. The relationship
between the medium and the topology is shown in Figure 2.1.

Medium	TOPOLOGY			
	Bus	Tree	Ring	Star
Twisted pair	X		X	X
Baseband coaxial cable	X		X	
Broadband coaxial cable	X	X		
Optical fiber			X	

Figure 2.1   Relationship Between Medium and Topology.

A baseband network refers to transmission of signals
without modulation and the entire medium spectrum is
consumed by the signal. Baseband LAN's are typically
accessed via a carrier sensed multi-access collision detect
(CSMA/DC) protocol commonly referred to as Ethernet
protocol.

A broadband network uses frequency division multi-
plexing (FDM) and divides the spectrum of the medium into
channels, each of which carries analog signals or modulated
digital data. For example, some channels may be used for a
point-to-point data communication, at the same time that

14

other channels can utilize a contention protocol such as used in Ethernet, while still other channels may be assigned video traffic and still other available channels, can be dedicated to voice traffic.

Selection of the best network topology and medium for a particular DCS may consider the bus topology versus the ring topology and broadband versus baseband. Figure 2.2 shows the medium comparison for mini and micro computer networks.

A basic characteristic of a bus is the physical ability to provide broadcast communication. The maximum delay is the negligible signal propagation time over the length of the bus. A ring, on the other hand, does not possess this physical capability because a message must be serially transmitted around the ring. However, it must be noted that logically the equivalent of a broadcast transmission can be achieved by a sender using an address code which will cause every node to read the message as it circulates in the ring. Whether a bus or ring topology does this faster or with greater throughput depends significantly on load. With light load there is less delay on a contention bus relative to a token ring. Because there is a high probability that a node will be able to transmit its messages immediately on the bus, there is no waiting for receipt of tokens, as there is on a token ring. Conversely at high loads, contention is so great that the probability is high that a node attempting to transmit will encounter congestion. Indeed, there is no upper bound on delay time on a congestion bus, whereas delay time is bounded for a token ring because a node will be guaranteed to receive the right to transmit, via receipt of the token within a finite time [Ref. 56].

15

```
Advantages                         Disadvantages
- - - - - - - - - -                - - - - - - - - - - - - -

                       Broadband

High capacity                      Modem cost
Multiple traffic types             Installation and
                                   maintenance complexity

More flexible configuration        Doubled propagation delay
Large area coverage
Mature CATV technology


                       Baseband
Cheaper no modem                   Single channel
Simpler technology                 Limited capacity
Easy to install                    Limited distance
                                   Grounding concerns
```

Figure 2.2   Broadband versus Baseband.

For broadband versus baseband, there is a great tendency to use the broadband medium in the LAN because the broadband allows various communication traffic to be applied on the network.

Bus, ring, and star, topologies have advantages shown in Figure 2.3. Choosing one depends on criteria such as , throughput, network flexibility and expandability, device connectivity, cost and reliability. (comparison of more complex network topology will be given in the network performance).

	Bus	Ring	Star
Information throughput	Decreases with each added node	Decreases with each added node	Depend on capacity of central node
Flexibility	High	Moderate	Low
Expandability	High	Moderate	Dependent on capacity of central node
Connectivity cost	Moderate	Low	Initially high, incrementally low
Reliability	High	Low	Moderate

Figure 2.3   Alternative Performance of Topologies.

2. Network Protocols

As explained in the previous section, LANs mostly utilize the ring, the broadband bus, and baseband bus. Common protocols for accessing ring LAN's are the Newhall (token) protocol [Refs. 4,5], the IEEE 802.4 token ring protocol, the pierce protocol [Ref. 6], and the delay insertion protocol [Ref. 7]. Prime and Apollo [Ref. 8] are examples of token ring protocols, the Cambridge ring [Ref. 9] and spider [Ref. 10] are examples of slotted ring protocols, and DDLCN [Ref. 11] is a delay insertion ring.

There are many network protocols implemented on different topologies. To create network compatibility it is necessary to use standard interface boards and protocols to connect to different hosts on the network. For example,

17

Excelan 803.2 compatible board, allows different micros to be connected to a bus network and takes care of all network protocol details. That is why, LAN standards have been developed within the national and international standard communities and The National Bureau of Standards (NBS) supports those standards. Those standards cover address layers 1 and 2 of the OSI layered model. They define the access methods, the physical interfaces and link control procedures for LAN. It is important to recognize that these standards do not provide the ability to communicate on a computer-to-computer basis. It is necessary to develop higher level protocols, whether in a local area networking environment or in an environment comprised of communication technologies.

The lower layers of the ISO reference model {Figure 2.4} that must be used are technology dependent. When dealing with public data networks, the X.25 specification is used but in a local area networking environment there are three separate specifications. The middle layer protocols of the reference model have been developed to work effectively over different communication technologies in support of many different applications. The upper layers involve multiple protocols corresponding to multiple applications.

Those IEEE 802 specifications shown in Figure 2.5 which are concerned with local technologies are:

- IEEE standard 802.3 a bus utilizing CSMA/CD (Ethernet) as the access method

- IEEE standard 802.4 a bus utilizing token passing (GMC token passing) as the access method.

- IEEE standard 802.5 a ring utilizing token passing (IBM token-ring) as access method

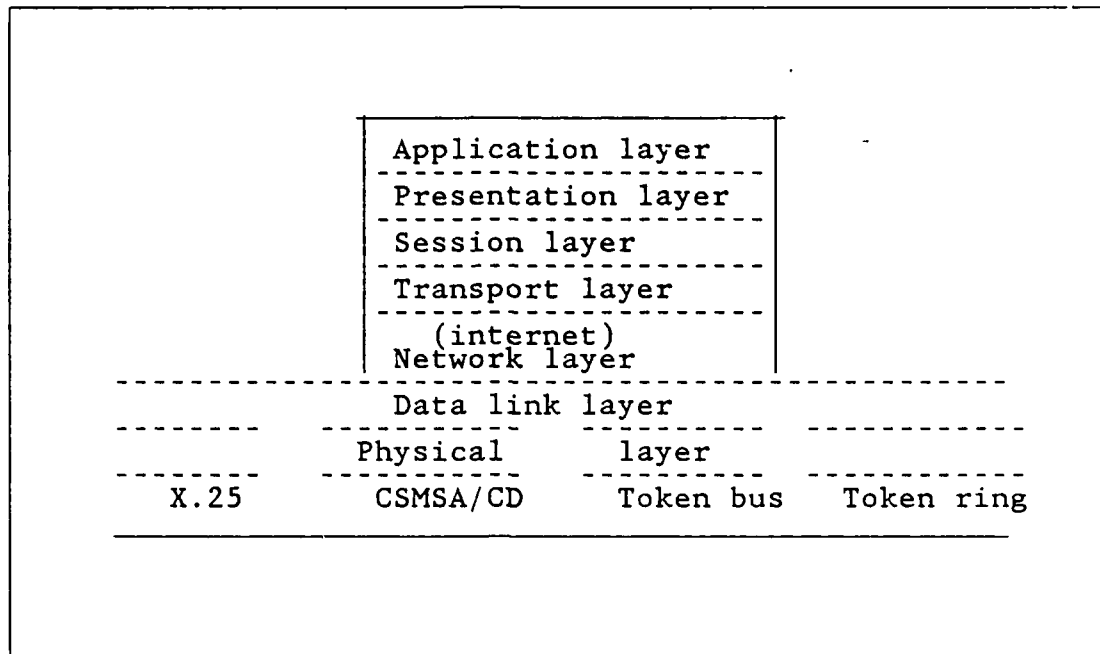- IEEE standard 802.6 for metropolitan area network

```
                 ┌─────────────────────────┐
                 │    Application layer     │
                 │ ----------------------- │
                 │    Presentation layer    │
                 │ ----------------------- │
                 │      Session layer       │
                 │ ----------------------- │
                 │     Transport layer      │
                 │ ----------------------- │
                 │       (internet)         │
                 │     Network layer        │
     ─ ─ ─ ─ ─ ─ ┴ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┴ ─ ─ ─ ─ ─ ─
                      Data link layer
     ─ ─ ─ ─ ─     ─ ─ ─ ─ ─ ─     ─ ─ ─ ─ ─ ─     ─ ─ ─ ─ ─ ─
                   Physical      layer
     ─ ─ ─ ─ ─     ─ ─ ─ ─ ─ ─     ─ ─ ─ ─ ─ ─     ─ ─ ─ ─ ─ ─
       X.25         CSMSA/CD       Token bus       Token ring
```

Figure 2.4   OSI Model for Different Communication Standards.

```
      ┌───────────────────────────────────┐
      │            ( 802.2 )              │
      │              LCC                  │
      │       Logical link layer          │         DATA LINK LAYER
      │   ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─           │
      │              MAC                  │
      │     Multiple access control       │
      │  (802.3) (802.4) (802.5) (802.6)  │
      │   ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─   │
      │                                   │
      │                                   │         PHYSICAL LAYER
      │                                   │
      └───────────────────────────────────┘
```
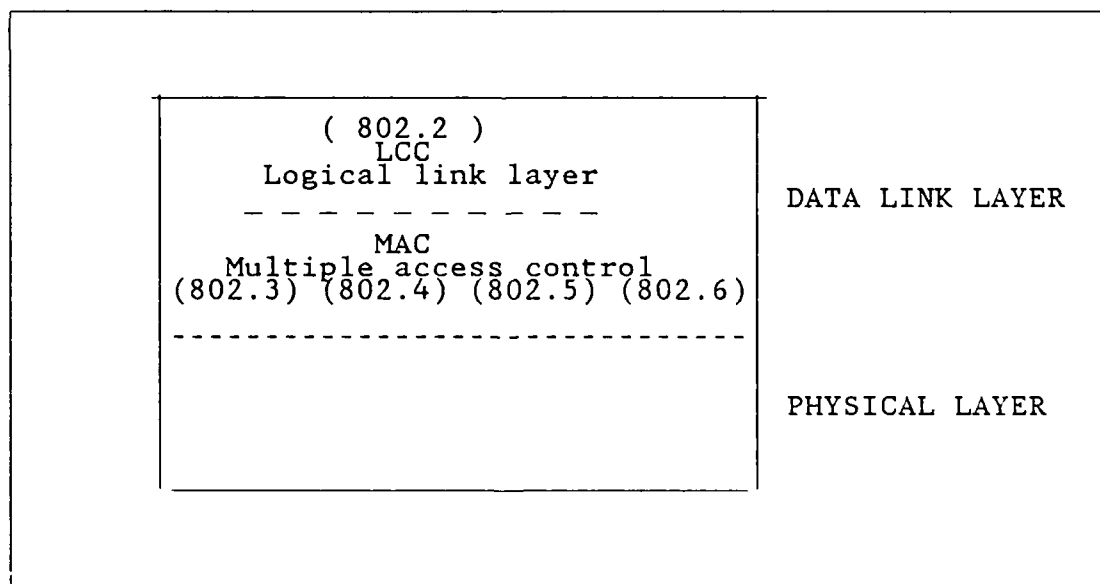
Figure 2.5   IEEE 802 Family.

19

The status of higher layer protocols in ISO reflect consensus at technical and policy levels. The characteristics of higher level protocols are given in Appendix A. In addition, the internet protocol explained in IEEE standards is of particular interest to the local area networking environment. It fits into the upper one-third of layer 3 and provides for internetwork communications on a connectionless basis. This protocol is essential for communications across a combined LAN.

The implementation of those protocols in personal computer networks was achieved by using a combination of hardware and software. The first two layers are implemented in ROM on the communication card plugged into the computer. The higher layer protocols appear as network control software located within the disk server. Sometimes there are exceptions. For example the first four or five layers could be implemented as ROM on a communication card (e.g. IBM PC network). The size of network control software mostly depends on the complexity of the network. For distributed computer systems, the size of communication software would be more than 100 kbytes of RAM.

3. Performance Evaluation of LAN

With proper subnet and distributed operating system (DOS) It is possible to share hardware and software resources in a cost effective manner while increasing productivity and lowering costs.

Some metrics are needed to evaluate the performance of a LAN based on multi-micro computers to help decide which topology is the best for cost/performance and for message-traffic/performance [Refs. 12,13].

The performance metrics which appear below are defined in terms of communication among nodes. This is the most meaningful interpretation of inter-object communication (e.g. micro computers) in a performance context, since a message must first be addressed to a node before the message can be received by higher level objects (functional modules and processes). Nodes in the network do not share any memory; all communication is performed by message passing. Each node is assumed to consist of a processor (CPU) with local memory, a communication processor capable of routing messages without delaying the processor, and a number of connections to communication links connecting the node to other nodes.

Selected LAN metrics defined below compare simple bus networks to complex networks such as the chordal ring, and hypercube. The parameters of networks {Figure 2.6}. were given to make the metrics understandable.

The metrics are:
1. Accessibility: The number of nodes which can be reacted directly with a single message transmitted from a given node,
2. Connectivity: The number of nodes which a given node is connected to
3. Average nodes traversed: The sum of interviewing nodes traversed by messages transmitted by given nodes in order to reach each of the other N-1 nodes.

Network topology, metrics based on cost/performance and message traffic/performance can be used to compare topologies.

b	Branching factor for asymmetric structures
c	Chord length
D	Dimension of mesh or hypercube
K	Number of network nodes
N	Network message population
N^*	Message population where queueing must occur
Ro	Total amaount of service required by a message
L	Mean message path length
Lmax	Maximum source-destination distance
n	Number of levels in an asymmetric structure
w	Lattice width of mesh or hypercube
pe	Processing element
lc	Communication link connection
cl	Communication link
Spe	Mean processing element service time
Scl	Mean communication link service time
Vpe	Processing element visit ratio
Vcl	Communication link visit ratio
Vi	Visit ratio for device i
Si	Mean device i service time
Xo	System message completion rate
LocSize	Size of locality
LV	Average number of links traversed in a (symmetric) symmetric structure with uniform routing
LV	Average number of links traversed in a (asymmetric) asymmetric structure
NumLinks	Number of communication links in network (K.Net-type) of size K
Reach	Number of nodes reachable by traversing 1 (1,Net-type) links

Figure 2.6   Parameters of Network.

22

a. Cost/Performance Evaluation of LAN.

In the cost/performance algorithm, The cost of each structure is defined as a function of the number of network nodes and the unit cost of communications. Cost is significant, only because it permits a comparison of cost performance ratios for various connection networks.

For the cost algorithm, each node of the system is assumed to consist of a processing element (pe), a communication processor (cp), and a fixed number of link connections (lc) joining the node to a bidirectional communication link (cl). The simply defined algorithm is:

$$COST(Net\text{-}type,Net\text{-}size,Cpe,Ccl,Clc)=$$

$$COST=Cpe*(Net\text{-}size)$$
$$+Clc*(Net\text{-}size)*(Num\ of\ conn.\ per\ node)$$
$$+Ccl*(Number\ of\ links)$$

Net-type	type of interconnection structures
Cpe	unit cost of pe-cp pair
Clc	unit cost of a link connection
Ccl	unit cost of communication link

The unit cost of communication links can be of two types: dedicated links between two nodes, and busses shared by two or more nodes. In the first case,   Ccl is simply the cost of each link. In the second case, Ccl is assumed to be the cost of the bus divided by the number of connections to it. The metrics of the networks' topologies are shown for the cost algorithm in Table II.

23

TABLE II
METRICS OF EACH TOPOLOGY

System	Node	Connections	Links
Single bus	K	K	1
Single ring	K	2	$\dfrac{K}{2} - 1$
Double ring	K	$4K$	$2K$
Complete connection	K	$K(K-1)$	$\dfrac{K(K-1)}{2}$
Chordal ring	K	$3K$	$\dfrac{3K}{2}$
Cube-connected cycles	$D2^D$	$3D2^D$	$3D2^{D-1}$
Snowflake	b^n	$2b^n$	$\dfrac{b^n - 1}{b - 1}$
Spannig bus hypercube	w^D	Dw^D	Dw^{D-1}
Tree	$\dfrac{b^n - 1}{b - 1}$	$\dfrac{(b^n + 1)(b - 1)}{b - 1}$	$\dfrac{b^n - b}{b - 1}$

In the cost/performance algorithm the network structures are described in two ways: symmetric structures, or asymmetric structures. Message sent by each node of a symmetric interconnection structure can reach the same number of nodes by traversing (L) communication link for all (L) (e.g. bidirectional ring network). In an asymmetric interconnection structure the number of nodes reachable in (L) links from a given node depend on the location of the source node in the network (e.g. trees, snowflake).

It is possible to compare various interconnection networks under a wide variety of conditions and assumptions. One can examine throughput bounds, cost and

24

cost/throughput bounds for various nodes and link service times. For example, Figure 2.7 shows cost/system-size, and cost/performance relations for various networks [Ref. 13].

b. Message-traffic/Performance of LAN

For the second case, the same assumptions and the same metrics for the cost/performance case are used for message-traffic/performance. In the formulation process, a network can service messages, an absolute upper bound on Xo is given by

$$Xo < \frac{1}{Vb * Sb}$$

where

$$Vb * Sb = max(Si * Vi)$$

for any closed queueing network.

To simplify analysis, the additional assumptions are given that all nodes require the same mean time Spe to perform a computation and all links require Scl to transmit a message.

In addition, I will give the mean message path length

$$L = \sum_{links} Vi$$

and processing element visit ratio
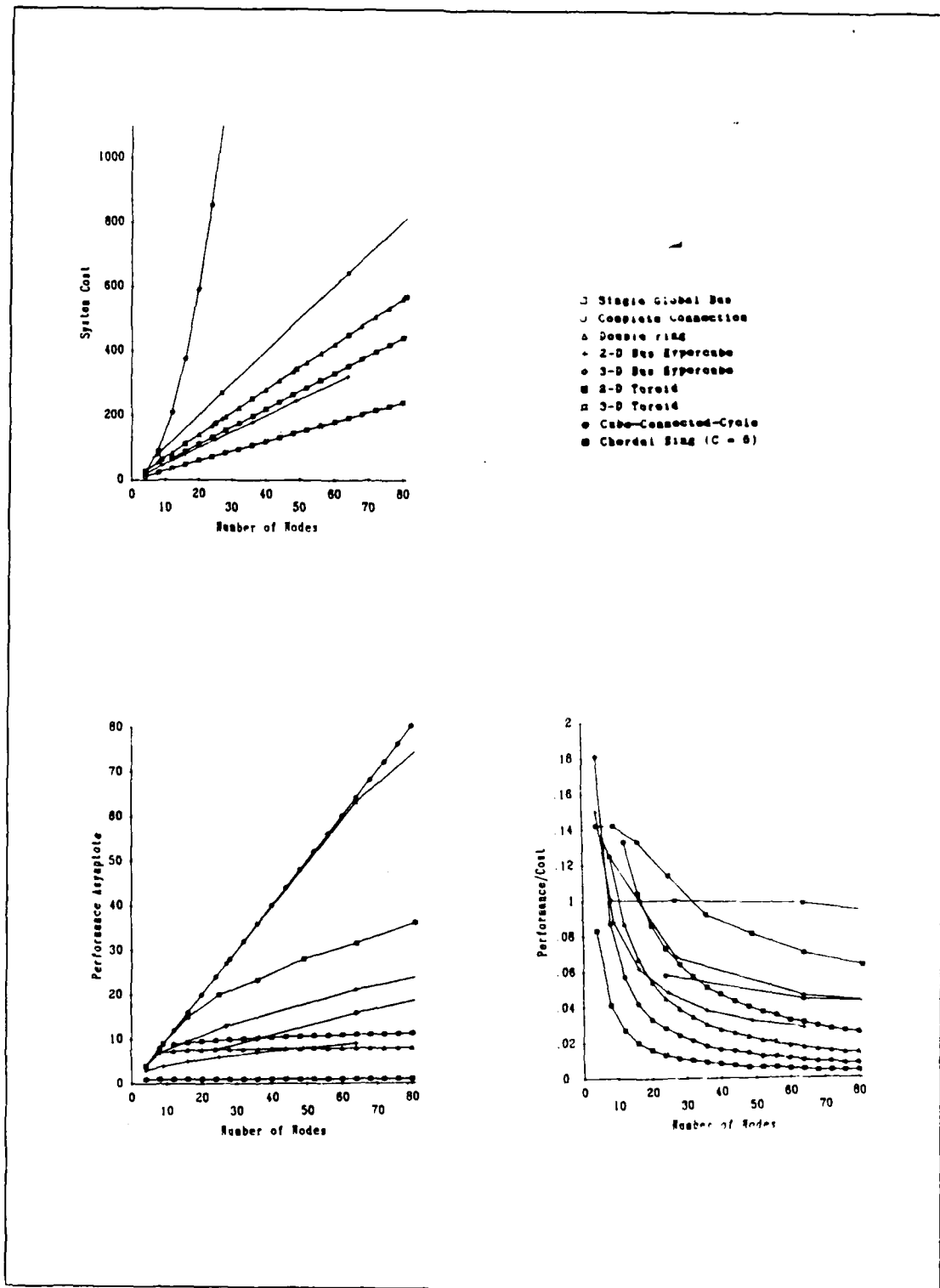
$$Vpe = \frac{1}{K}$$

Figure 2.7   Cost/Performance for Networks.

26

Briefly, Figure 2.8 shows various message-traffic/performance ratios on completion rates (Xo) for different networks.

I have described algorithms for determining cost and performance bounds for distributed message passing systems under some assumptions, more complete detail can be found in Dennig-Bunzen's article [Ref. 14]. In fact, message-traffic, cost and performance are the only figures of merit in distributed systems. A weighted function of cost, performance, reliability, broadcast delay and expansion increments should provide a more precise algorithm of selections.

B. WIDE AREA NETWORKS

A wide area network (WAN) is a geographically dispersed collection of hosts and communication processors where the distances involved are large. Another common name for WAN is long haul networks. Typical WAN's are ARPANET, [Ref. 15], CYCLADES [Ref. 16], TYMNET and telenet. WAN is outside of the scope of this thesis.

C. NETWORK OPERATING SYSTEM

Each host in a computer network has a local operating system that is independent of the network. The sum total of all the operating system software added to each host for communication and sharing of resources is called a Network Operating System (NOS). The added software often includes modifications to the local operating system. NOS's are characterized as being built on top of existing operating systems and they attempt to hide the differences between underlying systems. Tables III and IV are examples of how the OSI layers, explained in the previous section, support
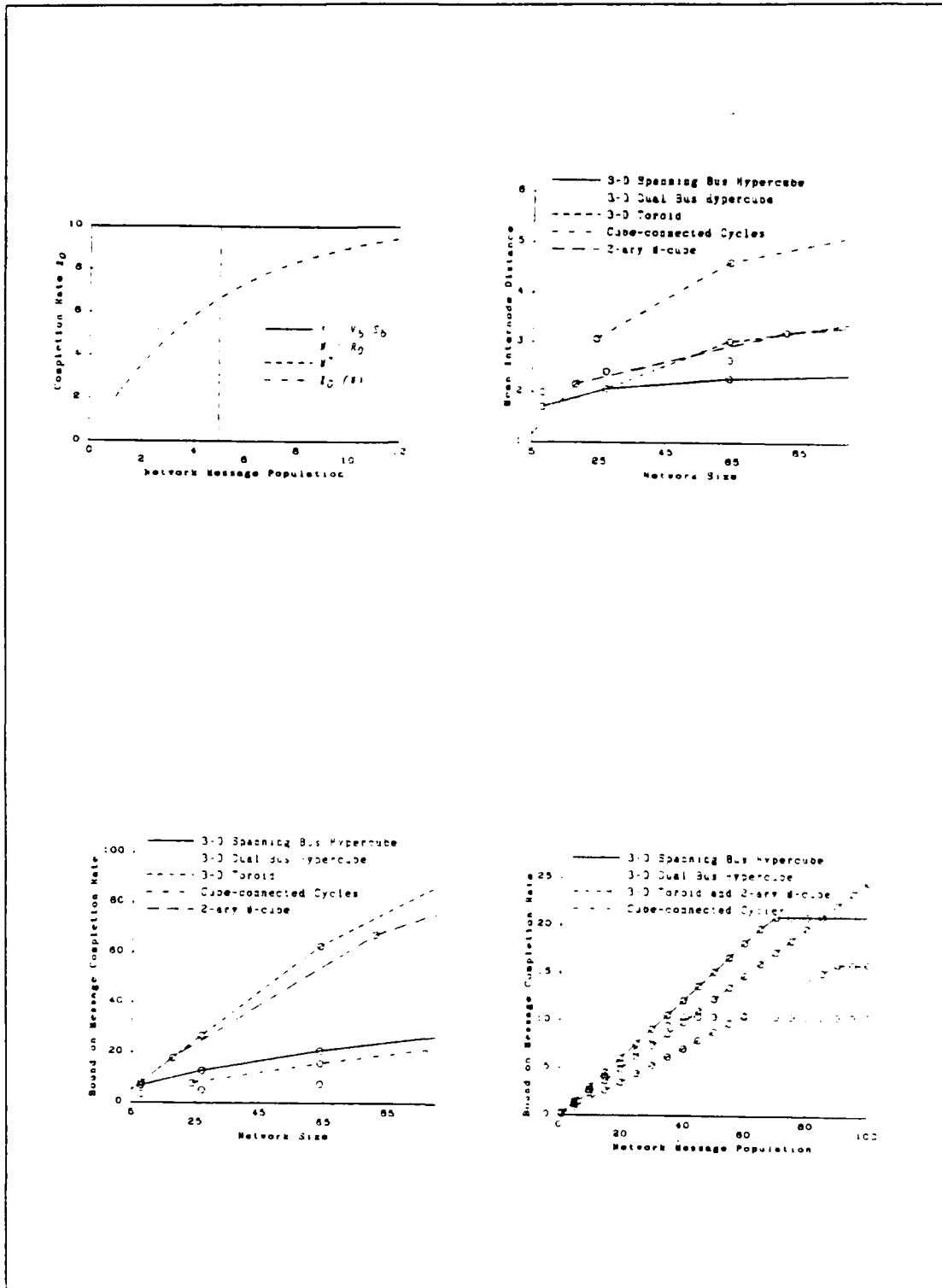
27

Figure 2.8   Message-traffic/Performance for Networks.

the connection control status and network administration responsibilities of NOS.

D. DISTRIBUTED OPERATING SYSTEM

There is one native operating system DOS for all distributed objects in an integrated computer network. A DOS should be designed with the network requirements in mind, so that it manages the resources of the network in a global fashion. Figure 2.11 [Refs. 55,67,68] lists a number of DOS's. Note that the boundary between the network operating system (NOS) and the distributed operating system (DOS) is not clearly distinguishable.

DOS have generally been used in two different LANs, according to the type of communication medium: shared memory systems (tightly-coupled systems) and local computer systems (loosely-coupled systems).

Shared memory systems such as C.mmp [Ref. 17], and Intel 432 [Ref. 18] consist of several processors connected by a bus or cross bar switch to the memory. A global scheduler determines an assignment of ready tasks to the processors. The message can be passed from one process to another at unit cost; they can be copied at the saturation rate of the memory.

Local computer systems such as Cm* [Refs. 19,20,21], Xerox Star [Ref. 22], and Cambridge ring consist of several processors connected by baseband serial communication lines. A global scheduler or a small set of cooperating schedulers assign ready tasks to processors. Some processors may have been preassigned functions such as file server, printer server, mail server or workstation. Message passing cost is proportional to message size. Transmission speeds vary from 4 Mbits/sec up to memory speed.

TABLE III
OSI SUPPORT OF NETWORK CONTROL RESPONSIBILITIES

LAYERS	7 APPLICATION	6 PRESENTATION	5 SESSION	4 TRANSPORT	3 NETWORK	2 DATA LINK	1 PHYSICAL CONTROL
NETWORK OPERATIONAL CONTROL ACTIVITIES							
Control software generation	X						
Switching on/off Network Components							
Data collection	X					X	X
Status Info Network Components					(X)		
Problem determination				(X)	(X)	(X)	
Alternate routing				X			
Test	X	X	X	X	X	X	X
Restoration/Recovery						X	
Service level monitoring	X					X	X
Availability					X		
Response time							
Accuracy				X	(X)	X	
Patching							
Info distribution	X		(X)	X	(X)		

30

TABLE IV
OSI SUPPORT OF NETWORK ADMINISTRATION

LAYERS	7 APPLICATION	6 PRESENTATION	5 SESSION	4 TRANSPORT	3 NETWORK	2 DATA LINK	1 PHYSICAL CONTROL
NETWORK ADMINISTRATION ACTIVITIES							
Inventory control							
Data collection methodology							
Service level agreements	(X)				(X)		
Maintaining vendor files	X						

Trouble reporting facility
  Procedures
  Evaluation
	(X)						

Network accounting	X				X		
Security control		X					
Transfer of files							
Determining span of control	X						
Setting up command sequences	X						
Change management	X						
Data storage media preparation							
Program distribution							
Software generation							

31

```
ACCENT              HXDP
ADCOS               LOCUS
AEGIS               Medusa
ArchOS              MICROS
CAP                 MIKE
Cm*                 MultiNET
CHORUS              RIG
CLOUDS              ROSCOE
DCS                 STAROS
DEMOS MP            TRIX
Domain structure    UNIX
EDEN                Xerox Star
Fully DP            WEB
Intel ITPS          Molecular
```

Figure 2.9   List of Distributed Operating Systems.

Most DOS for loosely-coupling systems are adaptations of shared memory operating systems such as StarOS, Medusa, and Cambridge ring. The shared memory operating system has been adapted to support communication over the local network. This is accomplished by implementing important systems functions as dedicated processes invoked by sending messages and waiting for responses. In StarOS, processes are assigned to specific processors by a small set of coordinating schedulers. In Cambridge ring, processes such as file server, printer server and login server are preassigned to specific machines, user's processes are assigned to the idle mini and micro computers by the login server. In Xerox star, all processes are assigned to machines. Adapting the

architecture of a shared memory operating system for local computer networks expedites development because it requires few new concepts. It produces a uniform user environment that hides the system from user view, resulting in a high transparency characteristic.

The traditional DOS approach is for one copy of the operating system to be available to all user processes and for the user processes to invoke the operating system via a trap on supervisor call instructions (e.g. tightly coupled system). In a bus oriented multi-microprocessor system, the processors, which do not contain the single copy of the operating system, execute much slower than the one processor that does contain the copy of the operating system in its local storage. This occurs because of bus contention incurred by processors which must make remote memory accesses to the operating system code via the bus. One way to prevent performance degradation is to provide a copy of the entire operating system in every processor's local memory (e.g., Xerox star system), but this may be infeasible due to the size of the operating system and required memory size. The solution used in Medusa and Cambridge ring operating systems is to distribute the various operating functions (e.g., process management memory management and file management) to different processors. Each processor is dedicated to performing a single function. Operating system functions are invoked by messages containing parameters which specify the desired service.

Generally, operating systems perform two classes of functions: allocation of resources among computing tasks and extending primitive hardware by implementing a powerful virtual machine. Table V displays the design hierarchy of the new generation single machine operating system. Each level in Table V manages a set of objects of a given type, it does this by providing operations for creating and deleting objects and changing their states.

33

TABLE V
DESIGN HIERARCHY OF OPERATING SYSTEM

Level	Name	Objects	Example operations
1	electronic circuits	registers,gates, buses	clear,transfer, complement, activate
2	instruction set	interpreter, microprogram	load,store, branch,index
3	procedures	procedure segments call stack,display	mark,call, return
4	interrupts	interrupt handler	invoke,mask, unmask
5	primitive process	primitive process semaphores	suspend,resume, wait,signal
6	capabilities	capabilities, domains	make,copy,enter, verify parameters
7	secondary storage	blocks of data, disk drives	read,write, allocate,free
8	virtual memory	segments	read,write,fetch
9	directories	directories	create,attach, search,list,detach
10	file system	files	open,close, read,write
11	pipes	interprocess pipes	open,close, read,write
12	devices	peripherals	open,close, read,write
13	user processes	user process	quit,kill, fork,resume
14	extended types	extended objects	create,delete, invoke operation, verify parameters
15	shell	user programming environment	statement in languages

34

Hiding maps from names to objects is a central principle in many new generation operating systems. In multi-level systems the map for a given class of objects is maintained by the layer for that class. To carry this principle over to multi-processor systems, the design must hide the location of all shared objects (e.g. processors, printers). This requires the solution to two problems: the reliable exchange of information between processes on different processors and the global naming of objects. These problems are solved in DOS by creating a communication layer for the first problem and by arranging common directory tree hierarchies for the second.

1. Communication Layer

The communication layer provides a single mechanism for exchanging information between two processes, independent of whether they are on the same processor Figure 2.10, can be inserted in the middle of the design hierarchy between the virtual memory level and the directory level Figure 2.11. If the communication layer were higher than the directory level, the directory manager would have no access to interprocessor communications ; the task of managing files and other objects across several processors would then fall to the user. If the communication layer were lower than the virtual memory, level segments could not be used to send or receive information across information channels. In other words, placing the communication layer at the lower level of primitive process can be optimal only in a very tightly coupled system in which the main memory units of each processor are part of the same address space. Placing the communication layer at the high user level can be optimal only for loosely coupled systems whose operating systems use incompatible schemes for naming, sharing and communications. The placement of the communication layer at

an intermediate level is a compromise permitted by the
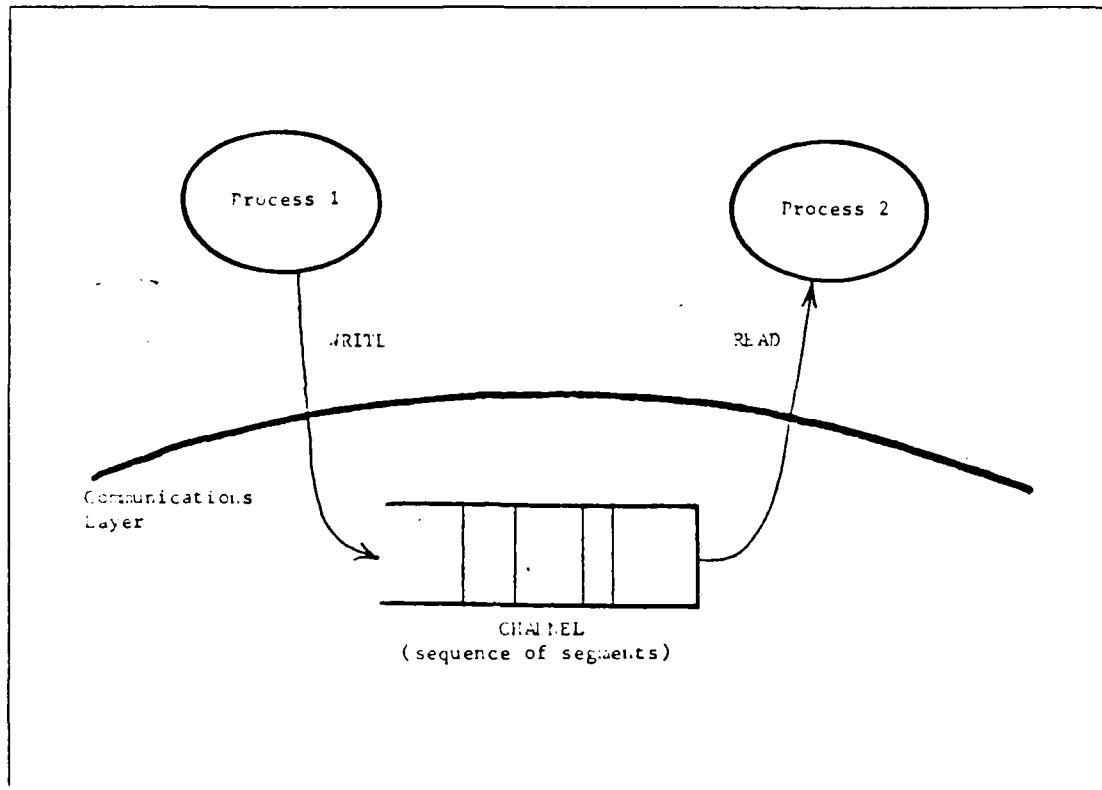moderate degree of coupling among processors on a local
network.

Figure 2.10   User View of Communication Layer.

The communication layer has to implement the network
protocols (Figure 2.12).   Those protocols are able to
recover from errors such as lost messages, duplicated
messages and out of messages.

2. Directory Layer

The directory layer implements a system wide direc-
tory structure that permits tree path names to be used as
global names for any permanent objects in Table V. Figure

36

```
           Electronic circuits
           Instruction sets
           Procedures
           Interrupts
           Primitive processes
           Secondary storage
           Capabilities
           Virtual memory
           COMMUNICATIONS
           Directories
           Files
```
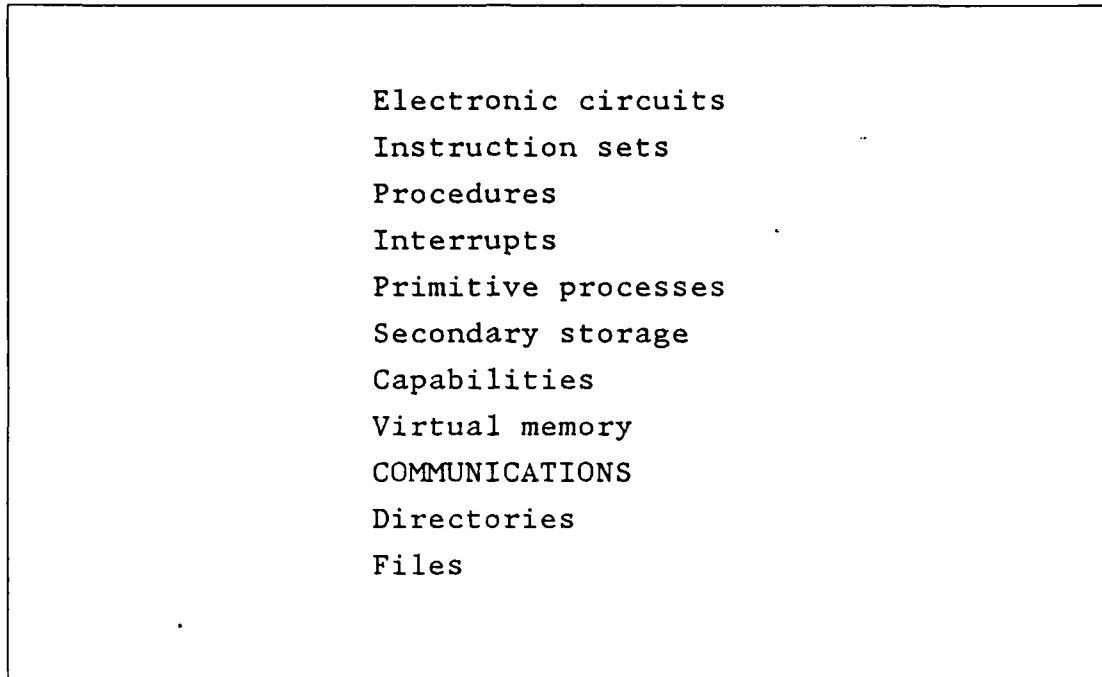
Figure 2.11   Insertion of Communication Layer.

2.13 illustrates that each entry of a directory contain
name, access and capability fields for each object listed. A
directory containing only self and parent entries is consid-
ered empty. The directory level simply stores global capa-
bilities but does not attempt to interpret them.

It also has the responsibility for ensuring that the
directory hierarchy is consistent across all processors.
This can be accomplished by replication in a distributed
database system. Any operation that modifies a nonlocal
directory must broadcast the change to update processes in
directory levels of the other machines. To control the
number of update messages in a large system, the full direc-
tory tree showed be kept on a small set of machines imple-
menting "stable store" copies of the portions of the
directory structure being accessed. User loggin-update
information need be sent only to the stable store machines
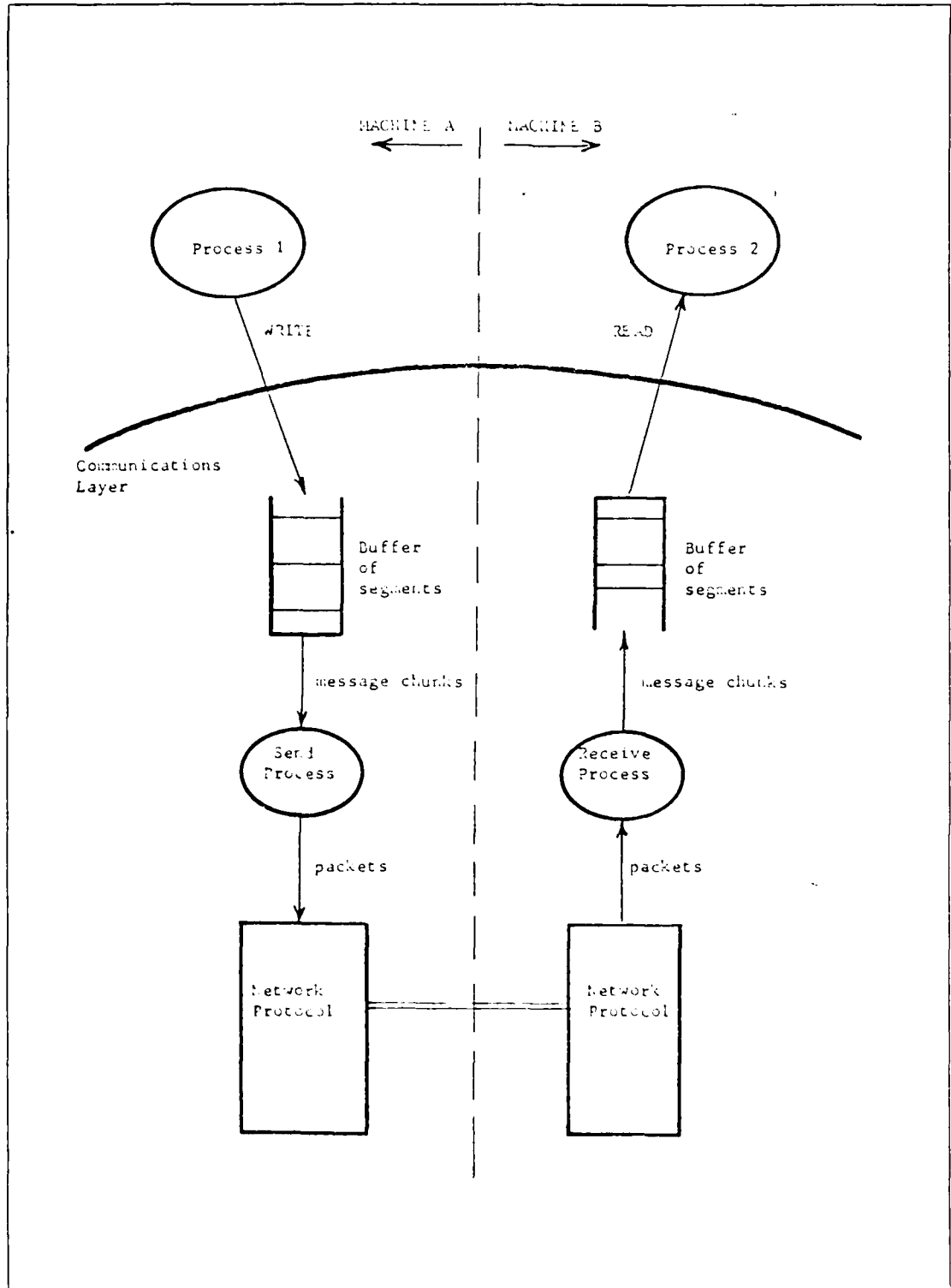and then to affected workstations.
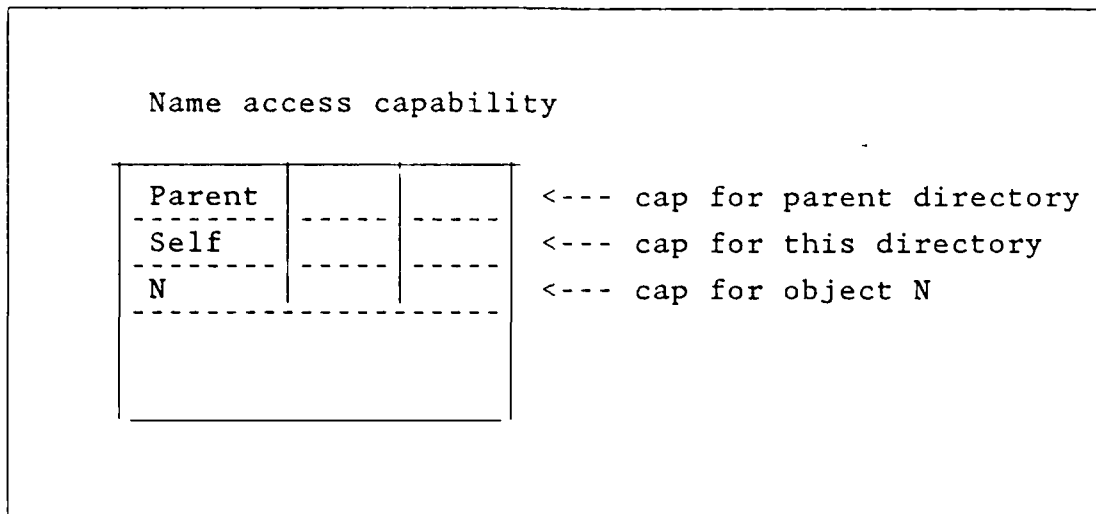
37

Figure 2.12   Internal View of Communication Layer.

```
Name access capability

 _____
| Parent |_____|_____|       <--- cap for parent directory
|--------|-----|-----|
| Self   |_____|_____|       <--- cap for this directory
|--------|-----|-----|
| N      |                   <--- cap for object N
|------------------------|
|                        |
|                        |
|_____|
```

Figure 2.13   Format of Directory.

A major goal of DOS is to make the system trans-
parent and act as a single processor for each user.

E. DISTRIBUTED FILE SERVERS

A file system is an integral part of a NOS, a DOS and a
distributed database system. In most distributed systems the
file system is considered a server which fields requests
from users as well as from the rest of the operating system.
File servers support the vision that there is a single
logical file system. In fact, there may be many different
file systems depending on the level of sophistication . The
file server may support replication, movements of files and
reliable updates to files in addition to the common file
commands. Examples of file servers are : Cambridge ring file
server [Ref. 23], DFS [Ref. 24], Felix file server
[Ref. 25], ROE [Ref. 26], Violet [Ref. 27], WFS and SEE
[Refs. 28,18].

F. DISTRIBUTED COMPUTER SYSTEM APPLICATIONS

1. Distributed Real-Time Systems

Nuclear power plants, process control applications and weapon fire control systems are inherently distributed and have severe real-time constraints and reliability requirements. Airline reservation and banking applications are also distributed, but have less severe real-time and reliability constraints and are easier to build. Examples of some real-time systems are ESS [Ref. 30], REBUS [Ref. 31], SIFT [Ref. 32], and AEGIS [Ref. 33].

2. Distributed Database Systems

An important application of DCS's is to allow data-bases (in contrast to file systems) to be geographically distributed across the network while at the same time being logically integrated. Increased reliability and performance can also be attained with a distributed database. Perhaps the best known distributed database systems are: Distributed Ingres (relational database) [Ref. 34], R* [Ref. 35], and SDD-1 [Ref. 36] which are implemented mainly on mini computer networks. Although there are some databases written for micro computers such as REBU, there are no known practical ways to distribute the database among networked personal computers.

Although the list of actual DCS's explained above can't achieve the full potential benefits of distributed computing. These systems would provide valuable input to formulate solutions for DCS.

III. CONCEPT OF DISTRIBUTED COMPUTER SYSTEMS

This chapter provides a perspective on six interrelated DCS issues: the object model, access control, distributed control, reliability, heterogeneity, and efficiency. These fundamental issues are described, problems associated with these issues are viewed as topics which span the network, operating system and database on mini-micro computers.

A. THE OBJECT MODEL

The object model is a collection of information and a set of operations defined for that information. An object is an instantiation of a data abstraction. One definition of an object is an incarnation of a resource. All hardware and software resources of a DCS can be regarded as objects. All objects required by a process at some instance of time is called a domain. This includes code and environment objects such as process control blocks and file control blocks.

Typical distributed systems functions such as (static and dynamic) allocation of objects, sharing objects across the network and providing interface between separate objects are all "conceptually" simple. The object concept is powerful and most DOS's listed in Figure 2.9 are based on objects., Consider a centralized name server that is causing a performance bottleneck and reliability problems. Reprogramming the name server in a distributed fashion might relieve these problems without effecting users of the name server object because to users of the name server the inter- - face has remained the same.

The major problems with object based systems has been poor execution time performance resulting from inefficient

41

implementation of access to objects (note: the execution time would be slower for personal computers). If the granularity of an object is too large, then the benefits of an object based system is lost. The needed are better architectures, better performance analyses, matching current organization structure to object based system, to choose good object granulatity and the ability to reduce the losts and execution times. As examples of curing problems of object module, consider network topology, bus or ring topology is very general and flexible that provide a convenient design model for representing objects in LAN type of DCS. The bus topology provides a direct path (logical) from each objects and allow (N) nodes to be connected in mesh topology in order of $N(N-1)/2$. The ring network is faster than bus network and delay time in ring network is shorter than in bus network.(the information on various networks were already given in the Chapter II).

The object model is considered fundamental to DCS research because it is a convenient primitive for DCS's which simplifies design, implementation and extensibility.

B. ACCESS CONTROL

Accessing the resources [Ref. 37] can be controlled in two ways:serial access, and use of an access control list.

1. Serial Access

Serial sharing of a single resource can be forced by locking. But there are other interesting techniques:

(1) Serial access [Ref. 38] in a ring can be forced by using a token protocol.

(2) Time division multiplexing and frequency division multiplexing is a static serial sharing of the communication medium.

(3) Polling is a technique where a central host queries stations one at a time.

(4) Reservation schemes [Ref. 40] require stations

42

to reserve future time slots on the communication medium to completely avoid conflict.

(5) Limited contention protocols allow contention for serial use of the bus under light loads to reduce delays, but generally becomes a collision free protocol under heavy loads.

(6) Virtual token scheme [Ref. 41] has been developed for concurrency control in a distributed database system. The token circulates on a virtual ring and carries with it a sequencer that delivers sequential and unique integer values called tickets.

2. Access Control List

Access to a resource must be restricted to the set of allowable users. This is usually done by an access control list or an access control matrix. Dynamic changes to access rights cause some difficulties. But solutions are available [Ref. 39].

In addition to sharing resources serially, DCS's contain resources which can be shared simultaneously. Resources that can be shared simultaneously pose no difficulty if accessed individually. However, it is some times necessary to access a group of resources at the same time. A transaction is an abstraction which allows programmers to group a sequence of actions into a logical unit. Protocols for resolving data access conflicts between transactions are called concurrency protocols. There are three major classes of concurrency control protocols:

(1) Locking is a well known technique that is already used at all levels of a system.

(2) Time stamp ordering [Ref. 42]: Time stamps all accesses to data and then some common rule is followed by all transactions in such a way as to ensure serial access. This technique is also used at all levels of system.

(3) Validation [Ref. 42] is a technique which permits unrestricted access to data items but then checks for potential conflicts at the commit point. The commit point is the time at which a transaction is sure to complete. This technique is useful when few conflicts are expected.

43

The concurrency control protocols as well as the previously mentioned access control techniques of the network are implemented across the entire spectrum from centralized to distributed control. Hence, access control is very closely related to the discussion in the next section on distributed control. It is difficult to choose the right access control technique for a given set of objects because access control has direct impact on efficiency and reliability.

C. DISTRIBUTED CONTROL

The various forms of distributed control depend on the application and requirements for DCS. The majority of work in distributed control is based on extensions to the centralized state-space model and can be more accurately described as decomposition techniques [Ref. 43] rather than distributed control. Large scale problems are partitioned into small problems each smaller problem being solved. It requires extensive computing power, making them more suitable for application programs rather than system functions.

In this case, the distributed control of more demanding types will be considered across three levels: network, DOS, and DDB levels.

1. Network

Functions in the network such as routing [Ref. 44] and congestion control are good candidates for distributed control.

a. Routing

Routing is the decision process which determines the path a message follows from its source to its destination. Some routing schemes are completely fixed, others

44

contain fixed alternate paths where the alternative is chosen only for failures. This kind of routing is called adaptive routing. Adaptive routing schemes may be centralized where a routing control center calculates good paths and then distributes these paths to the individual processors of the network in the same period. But it is not a distributed control.

Routing algorithms which exhibit distributed control typically contain (N) copies of the algorithm (one at each communication processor). Information is exchanged between communication processors periodically or asynchronously depending on traffic load. Such algorithms have the potential for good performance and reliability because distributed control can operate in the presence of failures and quickly adapt to changing traffic patterns. Several problems arise in such algorithms including the phenomena known as ping ponging (message looping) and poor reaction.

b. Congestion

Congestion can, during high traffic, cause completely performance collapses. Solutions for congestion include preallocation of buffers and message discarding. Congestion control for one algorithm set of permits circulates around the network and a set of permits is fixed at each processor. Whenever a communication processor wants to transmit a message it requires a permit, either one assigned to that site or a circulating permit. When a destination communication processor removes a message from the network it regenerates the permit. This algorithm limits the number of messages in the network depending on the number of permits in the system (e.g. token ring and token bus systems).

2. Distributed Database

Distributed control for DDS involves data integrity, consistency and concurrency controls. One well known concurrency control software procuct is INRIA [Ref. 41]. It maintains integrity of the database in the presence of concurrent users. The distributed controllers must cooperate to achieve system wide objectives for good performance subject to data integrity constrains. Another algorithm is based on two phase locking and atomic actions. Removing the data integrity constrains from algorithms will improve performance of DDB but control problems become much more difficult to manage.

3. Distributed Operating System

In the operating system area, functions like scheduling, deadlock detection, access control and file servers are candidates for being implemented with distributed control. Most operating system functions must run in real-time (synchronization) with minimum overhead (time sensitive).

In summary there are many forms of distributed control. Deciding which form is appropriate for each function in a DCS is difficult. Deciding how distributed control algorithms of different forms will interact with each other under one system is even more complex. The advantages of designing proper algorithms in the right combination will be improved performance, reliability and extensibility.

D. RELIABILITY

Reliability is a fundamental issue for any system. But it is perhaps, even more important to DCS. Reliability includes the following entities:

-Error: an item of information which when processed by

46

the normal algorithms of the system will produce a failure.

-Failure: an event at which a system violates its specifications.

-Fault: a mechanical or algorithmic defect which may generate an error (a fault may be permanent, transient or intermittent).

-Fault avoidance: ability of a system to avoid entering a fault state.

-Fault tolerance: employs error detection and redundancy to allow the circumvention of faults without operator intervention.

Reliability can be defined as the degree of tolerance against errors and faults. Increased reliability comes from fault avoidance and fault tolerance. Reliability is a multi-dimensional activity that must simultaneously address some or all of the following: fault detection, fault masking, retries, recover, restart, repair configuration and reintegration. Instead of explaining each activity, all these issues are treated at four levels: network, DOS, programming language and DDB.

1. Network

In the network, data link protocols use handshaking techniques with positive feed back; frames contain error detection codes such as CRC; timers are used to access the last message, last token or network partioning. Some networks create an abstraction called a virtual circuit that guarantees reliable transmission of messages. Flow control protocols attempt to avoid congestion, lost message due to buffer overruns and possible deadlock due to heavy traffic and not enough buffer space. Alarm and other high priority

47

messages are used to identify dangerous situations needing immediate attentions. Routing algorithms can contain multiple routes to each destination when failures occur.

2. Distributed Operating System

All techniques used in the network can also be used in DOS. Reliable DOS should also support replicated files, exception handlers, interprocess communication, testing procedures run from remote processors and avoid single points of failure by combination of replications, back up facilities and distributed control. Distributed control is used for file servers, name servers, scheduling algorithms. Reliable interprocess communication would enforce "at least one" or "exactly once" semantics depending on the type of interprocess communication being invoked.

3. Programming Languages

ARGUS [Ref. 45] is an example of a distributed programming language. A distributed program written in ARGUS may, however, experience deadlock. It supports atomic objects, transactions, nested actions, reliable remote procedure calls, stable variables, periodic and background testing procedures.

4. Distributed Databases

Distributed databases make use of many reliability futures such as stable storage, stable transactions, nested transactions [Ref. 46], commit and recovery protocols [Ref. 47], nonblocking commit protocols [Ref. 48], termination protocols, check-pointing, replication, primary backups and timeouts to detect failures. Termination and recovery protocol impact database reliability.

48

a. Termination Protocols

Termination protocols [Ref. 50] are used in conjuction with nonblocking commit protocols and are invoked at failure detection time to guarantee transaction. It attempts to terminate (commit or abort) all effected transactions at all participating host processors without waiting for recovery. This is an extremely important feature when it is necessary to allow as much continuation as possible.

b. Recovery Protocol

The major functions of recovery protocol are to restart the system processes and to reestablish consistent transaction states for all transactions affected by the failure, if this has not been already accomplished by the termination protocol.

It is obvious that a termination (clean up) and recovery protocol is required at all levels in the system. For example, termination and recovery protocols may themselves be distributed to enhance reliability. However the distributed termination protocol typically require $N(N-1)$ messages during a round of communication where N is the number of participating objects. This is too costly for slow networks but it may be acceptable on fast local networks or within a uniprocessor. The benefits would be greater reliability and better availability.

Reliability is strongly related to object oriented DCS. For example, object oriented systems confine errors to a large degree and define a consistent system state to support rollback and restart. The goal of reliable system design is to create "reliable" objects out of unreliable objects. For many systems it is too costly to incorperate an extensive number of reliability mechanisms. The major challenge is to integrate solutions to all these

49

issues in a cost effective manner and produce a reliable system.

E. HETEROGENEITY

Incompatibility problems arise in heterogeneous DCS in a number of ways and at all levels. First, incompatibility is due to the different internal formatting schemes that exist in a collection of different communication and host processors. Second, incompatibility arises from the differences in communication protocols and topology when networks are connected to other networks via gate ways. Third major incompatibilities arise due to different operating systems, file servers and database systems.

The easiest solution to the problem for simple DCS is to avoid the issue by using a homogeneous collection of machines and software. If it is not practical, then some form of translation is necessary. Better solutions are applicable under certain situation, including an intermediate translator or an intermediate standard data format.

Intermediate translators accept data from the resource and produce an acceptable format for destination. This is effective when the number of different types of necessary conversions is small. For example, a gateway links two different networks and acts as an intermediate translator.

In the case of intermediate data format, if the number of different types to be dealt with grows very large, then a single intermediate translator becomes unmanageable. In this case an intermediate standard data format (interface) is restored, hosts convert to the standard data and another conversion takes place at the destination by choosing the standard most common format in the system, the number of conversions can be reduced. Two methods of interfacing standard data formats for networked micro computers are program

abstract machine and combining different operating systems in the same processor. An abstract machine which provides interface between application software and different processors is complex. The second approach iš used in some office automation systems. It collects different CPU chips on a single board. For example, Strict co-processor card combines 8086 and 68000 chips and allows the use of XENIX/UNIX OP and the powerful distributed databases that run under UNIX. In addition Quadran developed an Apple emulater card for IBM PC to run CP/M 86 and CP/M OP. The abstract machine approach is based on complex software which covers much memory space and has been used in mini and large computers, but is not available for micro computers. To solve heterogeneous problems by using abstract machines for micro computers requires special purpose computers to work as file servers, stable stores. As defined above, the abstract machine approach would become much expensive than hardware which contains more than one different operating system chip for a small size DCS. In addition, hardware would be easy to. install into the computer and would be cheap.

In general, the problem of providing translation for the movement of data and programs between heterogeneous hosts and networks has not been solved. The main problem is ensuring that such programs and data is interpreted correctly at the destination host. It is inevitable that imcompatibilities will exist in DCS because it is natural to extend such systems by interconnecting network by adding new hosts and communication processors and by increasing functionablity with new software. Furthermore the main function of NOS, DOS and file servers is to present uniform logical interfaces (views) to the end user from a collection of different environments.

F. EFFICIENCY

DCS are meant to be efficient in a multitude of ways. Efficiencies of DCS are:

(1) Resources developed at one host can be shared by other hosts by limiting duplicate efforts.

(2) To share expensive hardware resources minimizes the cost.

(3) To improve response time and throughput of user process.

Once the system is operational, improving response time and throughput of the user process is the responsibility of scheduling and resource-process management algorithms [Refs. 51,52,53,54,55].

1. Resource and Process Management

Typically the resource and process management require synchronization techniques which provide structure and order to interactions and thus give rise to a stable and consistent system resource and process management.

Synchronization techniques are well suited for regulating access to shared resources are classified as "access synchronization" techniques (all access control techniques mentioned in the object model are of this type). Synchronization techniques are well suited for maintaining specified as " coordinating synchronization" techniques (these techniques are used in distributed control techniques). Finally, techniques used to change the logical relations which are maintained by synchronization tecniques are classified "meta-synchronization" techniques.

a. Access Synchronization

Access synchronization is a mapping from a randomly generated order of requests into an organized temporal order so as to maintain the consistency of the

52

shared resource. The temporal accessing order is meant to achieve consistency requirements.

The parameters of access synchronization are:

(1) The type of request, such as read or write.

(2) Sharing policy, such as priority (e.g.priority of incoming message used for ready process in CHORUS operating system).

(3) The state of shared resource, such as the shared buffer being empty or full.

(4) In DCS there is also the logical dependency between distributed resources. When members of a set of resources are update dependent such a set is called "update set". For example, distributed and replicated files from an update set (transaction). The logical dependence between distributed and shared resource is at the heart of distributed access synchronization.

b. Coordinating Synchronization

Coordinating synchronization is one of the central tasks of a DCS. It provides a state exchange mechanism between processes. The parameters of coordinating synchronization are:

(1) On the sender's side, a list of receivers of a given event.

(2) On the sender's side an exception handling routine when the expected response from the receiver doesn't occur.

(3) On the receiver's side , a list of authorized senders for each type of event in order to guard against unauthorized sending and a routine to help the operating system to find out why unauthorized sending happens.

(4) On the receiver's side, a set of procedures to operate on the received events.

(5) An interprocess communication protocol to handle the communications between the event senders and event receivers.

c. Meta Synchronization

Meta synchronization is used to exchange the existing logical relations without stopping the system (otherwise it would be too expensive to stop the system as a whole). This could be the result of a system object failure, so there is no way, but to alter the logical relations, to adapt to remaining hardware. It could also be the result of system expansion.

2. Scheduling

The scheduling algorithm is related to the resource allocation because a process will not be scheduled for the CPU if it is waiting for a resource. Generally, it would be better to handle system characteristics either of two ways:

(1) System characteristics: include the number, type and speed of processors, the allocation of data and programs, the amount and location of replicated data and programs, how data are partioned, partitioned functionality in the form of dedicated processors, any special purpose hardware, characteristic of the communication protocols and clocks used in the system.

(2) Scheduling algorithm characteristics: A good scheduling algorithm would take system characteristics into account. Scheduling algorithm characteristics include: the type and amount of state information used, how and when that information is transmitted, how that information is used, when the algorithm is invoked, adaptability of algorithm and stability of algorithm.

In this chapter, the global issues of the DCS were already described. The degree of complexity of the DCSs extended from networked computers to reliable DCSs depends on what degree those issues have been used in the system. For lower complexity the system would be a networked computers like today's PC networks. A well designed PC network can achieve file transfer, print pooling, remote access and record locking between hosts using 7 layers of ISO protocol in the network. The implementation of 7 layers

are provided by writing the first two or three layers into ROM on the communication card and by writing software programs for the higher levels. The communication software covers more than 100 Kbyte RAM in memory. During communication between nodes, networks take the CPU away from executing application programs (e.g. database, spreadsheet). When micro computers are considered as part of DCS, the architecture of micro computers will not be adequate to handle all the issues of DCS. It requires more memory space, and more co-processors. For example, PC vendors are willing to build communication cards with powerful communication processors and ROM without using the CPU and software. But this new product will not be able to help micro computers provide all issues of DCS.

IV. SELECTION AN DESIGN OF DISTRIBUTED COMPUTER SYSTEMS

In this chapter, I will discuss the selection of DCS based on mini and personal computers and show a design methodology of a computer system for the organization.

A. SELECTION OF COMPUTER SYSTEMS

Considering distributed computer systems' advantages of cost/performance, cost/communication, reliability , and user satisfaction play a main role in building DCSs instead of multi-user systems.   But today's improving VLSI technology has made the power of super mini computers similar or more powerful than the main frames of the 1970's. For example, the cost per MIPS (million instruction per second) for super mini computers ranges from $70,000 to $120,000 compared with main frame (IBM 3080)computing at $210,000 per MIPS [Ref. 58]. Figure 4.1 illustrates the classes of today's computers [Ref. 59]. The result of producing powerful mini computers has been changing the idea about connecting micro computers to network and replaced it with an idea about to connecting micro computers and dumb terminals to a central super mini computer via ring or bus network (departmental approach to computer networks) to reduce the workload and to establish the local control mechanism on the main network.

There are many reasons which support the idea of central minis in network and in DCS. If the multi-user systems based on super minis versus personal computer networks are compared to the system cost Figure 4.2 and on the cost per user basis Figure 4.3 the personal computer network would be more efficient for small networks (less than 6 users) but for more than six users, multi-user system will be more

56

```
Class              Description                        Cost
--------------------------------------------------------------
Supercomputers     >Gbyte physical memory             >$10M
                   >Gbyte virtual address space
                   <1 nsec cycle time

Mainframes         >10Mbyte physical memory           >$1M
                   >Gbyte virtual address space
                   <50 nsec cycle time

Super mini comp.   >1Mbyte physical memory            <$250K
                   >Gbyte virtual address space
                   <100 nsec cycle time

mini computers     >1Mbyte physical memory            <$100K
                   >Gbyte virtual address space
                   <150 nsec cycle time

micro computers    <1Mbyte physical memory            <$5k
                      virtual support generally
                      not available
```

Figure 4.1   Classes of Computers.

efficient [Ref. 60]. The extension cost of multi-user
systems would be cheaper than adding a new PC to the
network. But if the multi user system is at its performance
limit, it will cost a great deal for the next user.

The advantages of using central super mini computers
would solve the lack of using personal computers
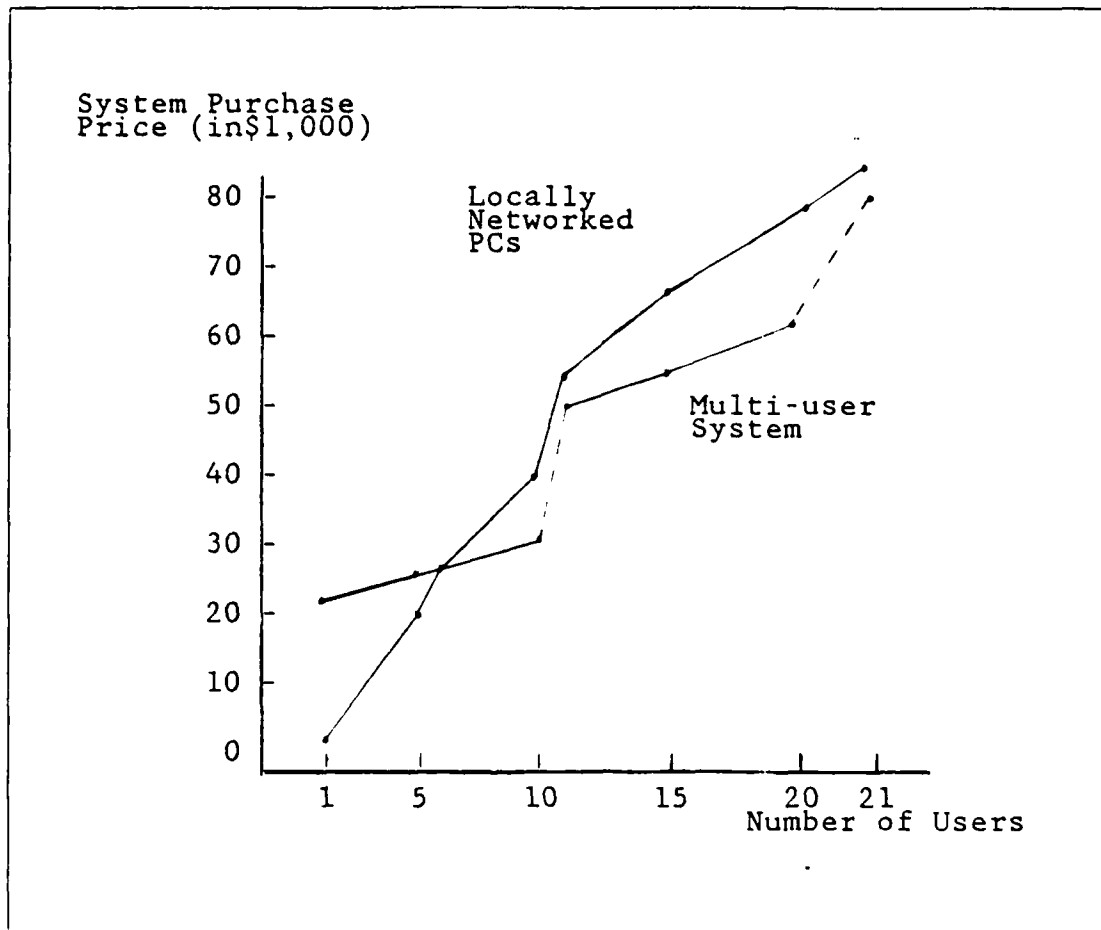(lower-level machines) in DCS [Ref. 69] A central mini in

57

Figure 4.2   System Cost Comparison.

the star topology connected other central minis via a bus or a ring network.   (split data systems) Figure 4.4 configures a departmental CPU and executes functions such as local database management and process control, networking and communications (access control functions), workstation supports, downloading the application and OS software into PCs without using their disk drives, and more advance services such as document image capture and processing.   In addition, departmental minis on the network achieves centralized control.
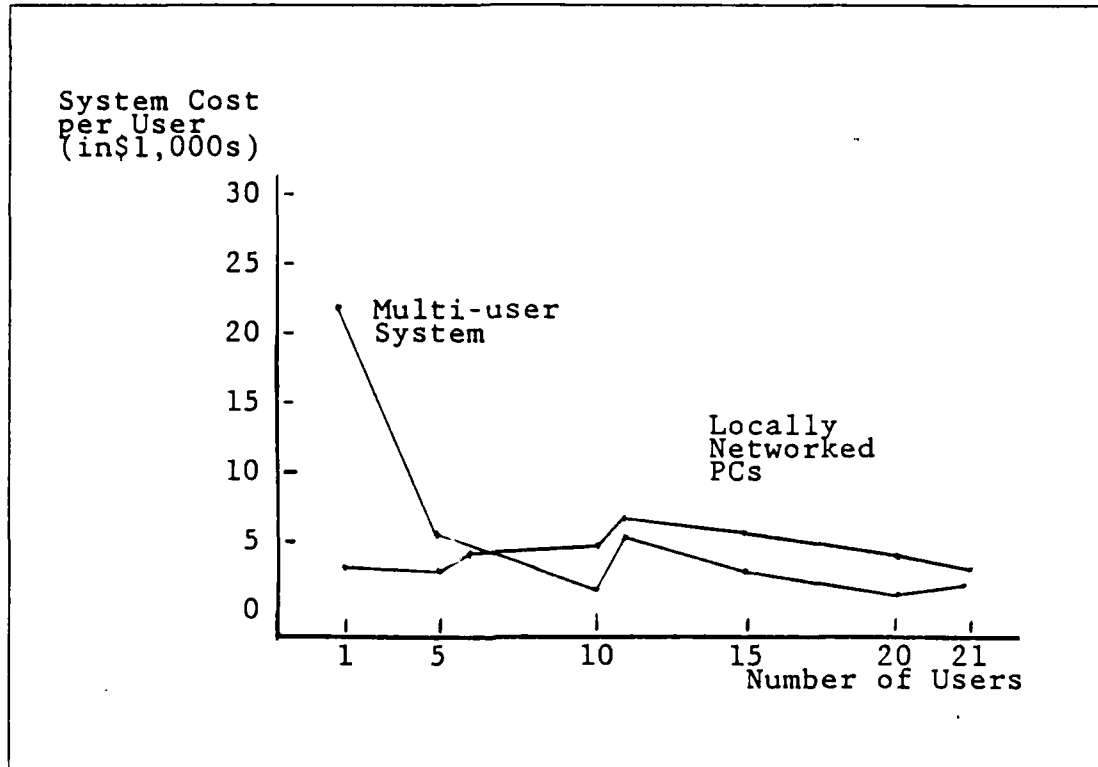
58

System Cost
per User
(in$1,000s)

Figure 4.3   Cost per User Comparison.

The central mini approach should also solve the problems
of heterogeneity. For example, heterogeneity has been
solved by writing emulating software (abstract machines)
used by DEC to connect IBM PCs to VAX 11/750, 11/785 and
8600 mini computers. As a server, a central mini provides
file server facilities for downloading programs to PCs, a
global name server for system and a local name server for
the department.

Another advantage of this approach is that it achieves
not only connection of micro computers but also connection
of dumb terminals to the central mini which act like a
server for a group of PCs and acts like a main frame for
dumb terminals to communicate to other network objects and
other networks.

59

Ring Network
(Split Data System)

Mini computer   Hierarchical Data System

PCs

Bus Network
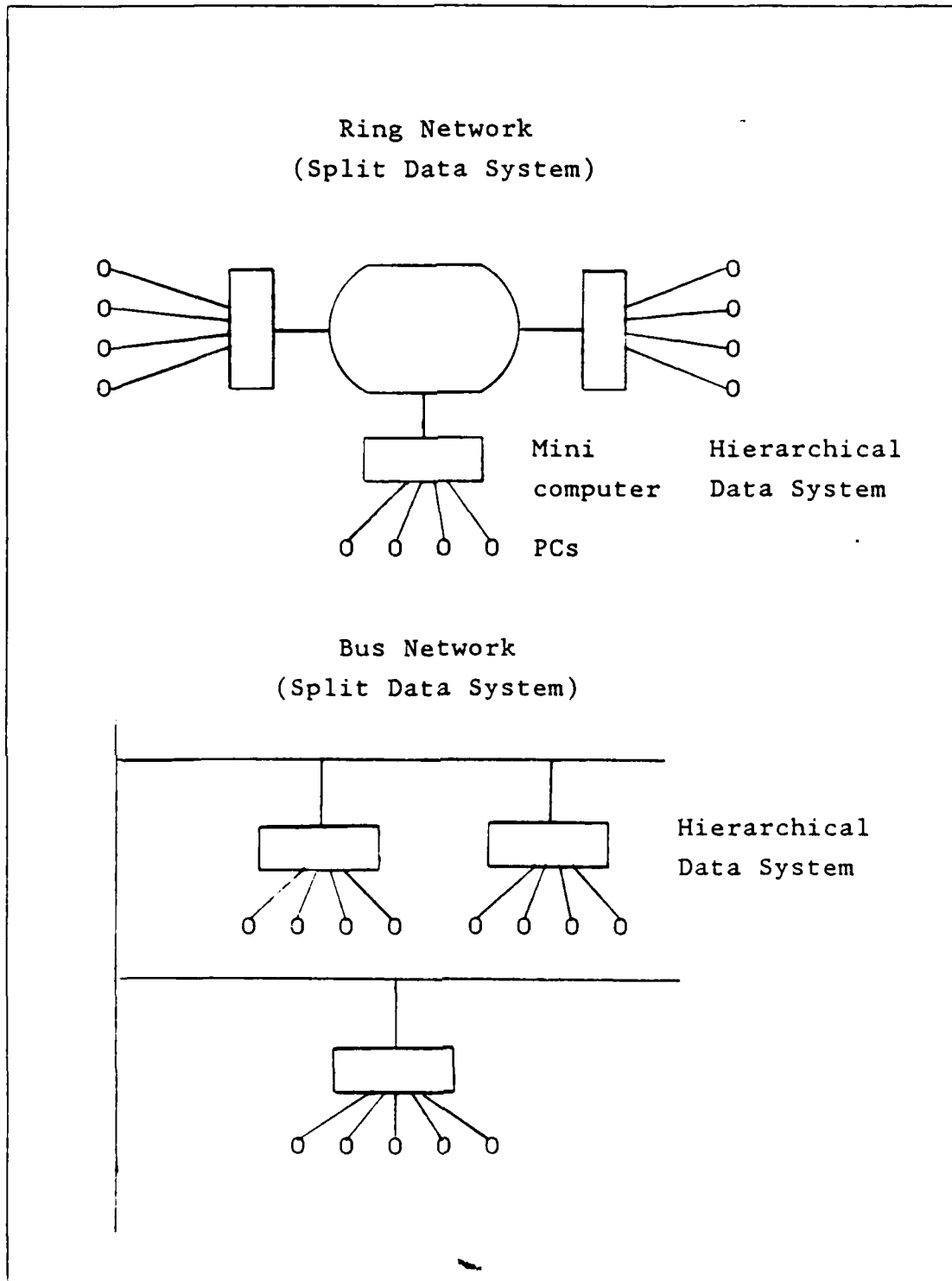(Split Data System)

Hierarchical Data System

Figure 4.4   Networks for Departmental Approach.

Centralized minis appear to be a better solution for the RTN's new computer system than the multi-user system. The new computer system is thought to provide various data processing facilities to the commands spread over a 20km area. A multi-user computer system as a third computer center would be more costly because of the higher cost of the extensions, the telecommunication, and the repairof long communication lines, although the economics of storage devices for large computers are cheaper. The RTN wants a reliable computer system. Multi-user computer systems can not assure a reliable system for the RTN.

Centralized minis recommended to the RTN have some advantages and match the geographically separate commands. With respect to cost, the installation of mini and micro computers to the commands do not require much money. The software cost is lower for mini and micro computers. Although multi-us systems have some advantages such as data security, lower data redundancy, and higher data standardization, the high data traffic on long communication lines would cause congestions, deadlocks, and slow response time. The standard data structures should be achieved by one design group (explained in the next section) for each centralized mini.

The structure of the data distribution on centralized minis for the new system is planned to provide each command's needs. It is made up of connected number of hierarchical data systems. A hierarchical data system consists of a central super mini computer and a number of PCs. The central mini obtains the master copy of data, controls data flows (e.g. reports, supply orders), and provides an update process between itself and the PCs. During update processing, a change is made to the data in one PC. This change should be passed to the central mini either immediately, or depending on its update priority.

61

The split data system consists of a number of central minis connected via a bus or a ring network. Each hierarchical data system has its own local data or a copy of global data in the same data structure. To use local data or global data in a central mini depends on the the RTN's policy for the new system. Global data transactions should be processed between the higher-level machines.

While deciding the network topology for an organization, the research may find support a topology and a data distribution which matches the organization's structure and its budget. There is no way to say which is the better opology. But the central super mini computers may be better than other other topologies because it possesses both the advantages of centralized and decentralized DCS.

B. DESIGN OF DISTRIBUTED COMPUTER SYSTEM

There are many interesting and challenging problems associated with the design of DCS with the matching of a DCS to the structure of an organization. These problems arise primarily because a DCS has many objects which must fit the organization structure and policy.

In designing DCS process, the responsibility of solving problems will be divided into two groups: The first is the research group which considers user's decisions, and the definitions of the organization requirements; the second is a technical group which considers the solutions provided by the research group to build a DCS for the organization.

1. Research Group

The most important stage in systems design is the definition of system requirements. It is important for the following reasons. First it is essential for both the technical designers and users of the system to be absolutely

clear about what they want the system to do. Only if system requirements are clearly defined will technical specialists have a set of required outputs against which they can match existing hardware and software. Similarly, only if system requirements are clearly defined will users have a set of precise expectations of what they want the new system to help them achieve. These expectations should be specified as a set of objectives directed at improving personnel and department efficiency, effectiveness, job satisfaction and quality of working life. Poor definition of system require- ments occurs when users are inadequately consulted about their needs or not motivated to think seriously about them. Such problems can be avoided if a research group consists of user representatives, and technical specialists [Refs. 63,70]. It is created to undertake the definition of system requirements and the current organizational system.

The research group, before developing techniques for measuring and collecting data concerning the system require- ments, could be used to assess the effect of distributed systems on their users. They must observe the DCSs on the market to assess their capabilities and their drawbacks. Observation can be applied to four types: laboratory experi- ments of DCs; true experiments of DCS in the organizations; quasi-experiments in the organization; non-experimental designs.

Laboratory experiments examine the effects of DCS. Although these experiments have produced interesting results, at present they are of limited value in DCS. They cannot recreate important paths of the system environment and their results often cannot be generalized.

Quasi-experimental designs include some experimental controls but lack a random assignment of subjects to the studies. Research that uses a quasi-experimental design can eliminate many threats to the validity of a study . It may

63

may possibly show how post study differences between test and control groups in the research group were caused. For example, post test changes in effectiveness of the test group might be due to events other than the introduction and use of system.

Nonexperimental designs can be applied to case studies which can aid hypothesis formulation, but can rarely test hypotheses adequately because of their limited scope.

True experiments of DCS in the organization help to make cost estimations, to obtain users' ideas on use of system and to decide new systems decisions. But imitating a system completely for another organization often does not work. The best way is for a pilot system (prototype system) [Ref. 70] design for new the desired system. Observations from four or more of them shpuld be used.

A pilot system which is a simulation of the actual DCS could be used to apply collected data from observations. The pilot group members would consist of section mangers and their secretaries from the management service functions, internal and external technical consultants, and the head of the department. During the pilot process, some measurements are applied to the data. Table VI shows the measurement instruments. The measurement instruments developed for the pilot include a prestudy questionnaire on organizational effectiveness, an extensive survey questionnaire form for determining which respondents communicated during a typical specified time period, an activities/communication log for static, a method of monitoring the system, and structured interviews.

TABLE VI

MEASUREMENT INSTRUMENTS

PRETEST   POST-TEST

1.Questionnaire 1 examined   1.Questionnaire 1, plus
  -System requirements   items on attitudes
  -Communication in the organization   toward the pilot
  -Information use, access and problems
  -Attitudes to technology
  -Job design
  -Quality of working life
  -Demographic data

2.Network interaction analysis   2.Same
  -Examined the communication network

3.Activities communication log   3.Same, plus data compared
  -A detailed account of time use   no. 4 below
   and communication patterns

65

Table VI

MEASUREMENT INSTRUMENTS (cont'd)

4. Data generated from the

system, e.g.

- Kinds of applications

used

- Sytem use over days,

weeks,months

- Learning curve

- Commands used

- Computer communication

patterns

5. Post-test interviews

of experimental group

- examined problems with

system trainig,

documentation and

futures desired

66

In general, the pilot system imposes the user communication, time use, access to information, attitude toward DCS technology and quality of working life.

Using the results of measured collected data in the pilot system, the research group is able to present a business case to management to extend the project to an operational system covering most of the custom systems division. Managers, project leaders, administrative personnel and other employees are being placed on the system.

The operational system has added several functions to the ones provided on the pilot system. User profiles tailor the system's user interface to the skill levels, requirements, and preference of each user. For example, some users prefer a menu-based interface [Ref. 66] (e.g. Zog/USS Carl Vinson DCS). The user calls the menu, which appears on the screen, then he chooses the function he needs by typing in the function number listed in the menu. Users who run into trouble may be assisted by a computer-aided instruction facility. The menu-driven approach doesn't need a high degree of training for the user in the system and can be used in systems (e.g. government, military) which display a high degree of personel turnover.

In summary, the methodology from the prepilot system to the operational system should be divided into three phases (Table VII).

TABLE VII

DESIGN METHODOLOGY

	PREPILOT PHASE	PILOT SYSTEM PHASE	OPERATIONAL SYSTEM PHASE
DATA FROM USER	Organizational scan Diagnosis Pilot systems analysis	System monitoring data User feedback	System monitoring data Pilot system pretest Operational system analysis Operational system cost justification
DESIGN STAGE		--->Pilot system design	--->Operational system design

During the prepilot phase, data are collected to design the pilot system. During the test phase, information is drawn from the users and from the existing system to help evaluate and define it. During the operational distributed system phase, post-test and system monitoring data are combined with system analysis data to enable the specification and cost justification of a fully operational system.

C. TECHNICAL GROUP

The technical group consists of most members from the research group and begin work on the operational system phase and continue to work throughout the system life cycle. The technical group should design the system according to technical options found by the research group. In fact, the technical options found after defining the system requirements are related with the job satisfaction objectives, explains the available range of equipment and tools which fit best with the achievement of the objectives.

Generally, the technical group consist of two smaller groups: a network design group and a system design group.

1. Network Design Group

Network design group is responsible to write the communication software and construct the network topology. They must build the desired network system in the design and/or enhancement phase according to the criteria listed in Appendix B.

a) connection establishment

b) rules of information transfer

c) order of messages

d) integrity and security of information transfer

e) handling long messages

f) adaptation of fast sender to slow receiver

g) routing

69

h) recovery

i) sharing connection

j) logging

k) network management.

2. System Design Group

The system designer group should concentrate on writing DOS and application softwares (e.g. database programs) and creating functional modules, building the operating system under the following issues which were already described in Chapters II and III. Note that it is difficult to place a boundary between the network and the system design group because some issues given are common.

a) naming and addressing objects

b) allocating functions to modules and modules to processors

c) distributing data across nodes of the network

d) specifiying appropriate levels of performance

e) determining methods for deadlock prevention, avoidance, detection and recovery

f) determining procedures for recovery from the system malfunction

The important thing is that the complexity of the system depends on the amount of spending money which top manager thinks. This cost effectiveness problem should be solved by the technical options that the research group and the technical group prepared. Sometimes the options will not support the idea of designing a reliable system because of budget constraints.

V. CONCLUSION

Distributed computer systems have been presented as a candidate system that be appeared for the RTN's new computer system. The reason for showing the distributed computer systems as an alternative system is that a distributed computer system potentially provides significant advantages including: better performance, reliability, resource sharing, and extensibility. The extensibility and reliability advantages of a distributed computer system meet the RTN's needs for the new system. The new computer system is planned to provide data processing to a number of dispersed commands at Golcuk-Istanbul.

The applications of distributed computer systems are a new era for the RTN. The aim of chapter II was to introduce the main elements of distributed computer systems. The local area network with its topology, standards, and cost/performance and message-traffic/performance for various networks. While the RTN is determining its requirements for the new system, it should decide on a topology based on reliability, and cost and message-traffic/performance ratios and select one of the three well known network standards (CSMA/CD, token bus, token ring) for the new system.

Distributed operating systems were also explained to show the differences between an uni-processor operating system and a distributed operating system and between tightly-coupled and loosely-coupled systems. For the RTN, combination of the tightly and the loosely-coupled operating systems should be examined. The central minis in a split data system could use a loosely-coupled operating system to handle their own local data. The combination of two distributed operating systems should be selected by the

71

Chief of Turkish Navy based on several research projects. A number of operating systems was given in Figure 2.11.

In chapter III the concept of the distributed computer systems was described. If all those issues are understood by a system design group, the designers easily mix the system requirements and the technical options to build an efficient distributed computer systems for the organization.

A centralized mini approach was recommended to the RTN as an application of distributed computer systems with a hierarchical data system because of the dispersed commands located at Golcuk. The central minis are connected to each other via a bus or a ring network in the system. The advantage of this split data system gives the commands autonomous control of their local data and allows them to update global data and critical data between their central minis without local users.

The most important thing required to design an efficient distributed system is that the Chief of Turkish Navy select the members of the design group carefully. The design group should consist of a high ranking officer, wide representation of users, technical consultants (internal and external consultants), computer system vendors, and external system design consultants. Definitions of the system requirements and the technical options should be applied on a pilot system first. A pilot system helps the designer group to measure the system's inputs and outputs by using defined technical options and also provides a training opportunity to key personnel of the new system.

APPENDIX A
CHARACTERISTICS OF ISO MODEL

Layer 1 :   Raw bit transfer

Allocation of pins

Establishing connection

Bit error control

Layer 2 :   Data link connection activation and deactiva-
tion (these functions include the use of
physical multi point facilities to support
connections)

Mapping data units provided from the network
layer into data link protocol units for
transmission

Multiplexing of one data link connection onto
several physical connections

Delimiting of data link protocol units for
transmission

Error detection recovery and notification

Identification and parameter exchange with
peer data link parties

Adaption of speed between sender and receiver

Layer 3 :   Network addressing and end point identifica-
tion

73

Multiplexing Network connections onto data link connections provided by the next lower level

Segmenting and/or blocking to facilitate data transfer

Service selection when different services are available

Selection of service quality based on parameters such as ressidual errors, service availability, reliability, throughput, transit delay, and connection establishment delay

Error detection and recovery to support desired quality of service

Error notification to higher levels when service quality cannot be maintained

Sequenenced delivery of data if available from a particular implementation

Flow control

Expedited data transfer as an optional service

Connection reset with loss of enroute data and notification to using parties

Termination services when requested by a using party

Congestion control

Providing billing information

Layer 4 : 1. Establishment phase

Selection of network service as a function of parameters such as throughput transit delay, set-up delay and error characteristics

Management of transport connection to lower level connections

Establishment of appropriate data unit size

Selection of functions used during data transfer

Transport of data from higher levels

2. Data transfer phase
   Blocking

   Concetenation

   Segmenting

   Multiplexing of connections provided by lower levels

   Flow control in a session-oriented, end-to-end sense

   Maintenance of the identity between the two transport functions acting on behalf of the parties of the conversation

   Error detection for lost, damaged, duplicated, misordered or misdelivered data units

   Error recovery to address problems detected in this layer or signalled from lower levels

   Transport of expedited data which flows out side normal flow control mechanism

3. Termination Phase

Notification of termination reason

Identification of connection terminated

Optional information as required

Layer 5 :   Session-connection establishment

Session-connection release

Normal data exchange

Expedited data exchange

Guarantine service (ability of a sender to
control delivery at a receiver)

Interaction management (two-way alternate and
two-way simultaneous transmission control)

Exception reporting

Mechanism for session connection synchroniza-
tion

Management of the session

Address translation

Layer 6 :   Session initiation and termination requests

Negotiation and renegotiation of presentation
image

Data transformation

Data formatting

Syntax selection

Library routine provision

Encryption to ensure security

Layer 7 :   Identification of intended communications partners and their availablity and authenticity

Establishment of authority to communicate

Agreement on required privacy mechanism

Determination of cost allocation methodology

Determination of resource adequacy to provide an acceptable quality of service

Synchronization of cooperating applications

Establishment of error recovery responsability

Agreement on data validity commitment

Identification of data syntax constrains

Information transfer

APPENDIX B
DESIGN CRITERIA OF NETWORK ARCHITECTURES

Connection establishment: A network usually has many nodes; some of them have multiple processes. Thus, techniques are required for specifying who mants to talk to whom. Inany layer where multiple designations are present, addressing is needed should be solved as well

Rules of information transfer: Should transmission be simplex, half duplex, or full dublex?

Order of messages: How does the protocol direct the receiver to allow the seperate message parts to be sequenced properly

Integrity of information transfer: Information must be transmitted accurately. The logic used in a function of type of node, linking used, and protocols implemented. In addition, the receiver should have ways of telling the sender which messages have been received in error.

Security of information transfer: The security level of information is maintained within the communications function only. It presents a defence mechanishm against external, unauthorized penetration of the network. Security can include multiple levels. But the higher the security level provided, the more overhead checking and logging requirements have to be included

Handling long messages: The mechanism of disassembling, transmitting and reassemling messages, in order not to lose long messages, must still guarantee econmy of short messages.

Adaptation of fast sender to slow receiver: Some kind of buffering must be available to avoid receiver's overload. the most promosing method is to continously gather information on the actual reciever status, whether the receiver is able to receive, and if so what quantity.

Routing : It provides for the most efficient transmission of information from source to destination. The transmission should consider all requirements in terms of service level of efficiency. The following activities are included addressing, path selection, load leveling, priority structures.

Logging : The ability to retain copies of traffic that have been transferred through the network nodes is a key control element. Major benefits include the ability to retrieve and retransmit additional traffic copies and a real-time 'database' for network recovery.

Sharing connections: These are critical for enhancing the overall economy of networking. The sharing problems can be addressed in many parts of networks, such as processing and link utilization.

Network mangement: The architecture must ultimately be capable of maintaining the communication systems service level with optimum capacity at reasonable costs.

LIST OF REFERENCES

1. Stalling, N., _Local Networks_, pp. 35-55, Newyork:McMillan, 1984.

2. Tanenbaum, A.S., _Computer Networks_, Prentice-Hall, 1981.

3. Stalling, N., _Local Network_, McMillan, 1984.

4. Farber, D.J., "The distributed computer system," _Proceeding 7th Annual Proceeding 7th Annual Proceeding 7th Annual IEEE Computer Soc. Int. Conf._, February 1973.

5. Newhall, E.E., "An Experimental Distributed Switching System to Handle Bursty Computer Traffic," _Proc. ACM Symp. Probl. Opt. Data Communication Systems_, October 1983.

6. Pierce, J., "How Far can Data Loops Go," _IEEE Transaction on Communication_, v. COM-20, June 1972.

7. Liu, M.T. and Li, C., "Design of the distributed double loop computer network (DDLCN)," _Journal of Digital Systems_, v. 5, n. 2, 1981.

8. _Apollo Domain Architecture_, Apollo Computer Inc., February 1981.

9. Needham, R.M. and Herbert, A.J., _The Cambridge Distributed ComputingSytem_, London:Addison-Wesley, 1982.

10. IBM Research Report RJ2914, _Object Naming and Catalogue Managementfor a Distributed Database Manager_, by Lindsay, B., August 1980.

11. Liu, M.T. and Li, C., "Design of The Distributed Double Loop Computer Network (DDLCN)," _Journal of Digital Systems_, v. 5, n. 2, 1981.

12. Reed, D.A. and Schwetman, H.D., "Cost Performance Bounds for Multi Microcomputer Networks," _IEEE Transaction on Computers_, v. C-32, n. 2, June 1983.

13. Reed, D.A., "Estimating The Performance of Multi Microcomputer Network," _IEEE 1983 International Workshop on Computer System Organization_, pp. 72-89, 1983.

14. Denning, P.J. and Bunzen, J.P.,"The Operational Analysis of Queueing Network Models," Computing Survey, v. 10, n. 3, pp. 225-2261, September 1978.

15. McQuaillan, J.M. and Walden, D.C., "The ARPA Network Design Decisions," Computer Networks, v. 1, August 1977.

16. Pouzin, L., "Presentation and Major Design Aspects of The Cyclades Computer Network," Proceeding 3rd ACM Data Communication Conf., November 1973.

17. Wulf, W.A., Levin, R. and Harbison, S.B., HYDRA/C.mmp, An Experimental Computer System, McGraw-Hill, 1981.

18. Intel Corporation, Introduction to the IAPX 432 Aarchitecture, 1979.

19. Swan,R.J., Fuller, S.H. and Siewiorek, D.P., "Cm* a modular multiprocessor," Proceeding of The National Computer Conf., pp. 636-644, June 1977.

20. Jones, A.K.and Durham, I., "StarOS, a Multiprocessor Operating System for the Support of Task Forces," Proceeding of The 7th Symp. on Operating Systems Principles, pp. 117-127, December 1979.

21. Ousterhout, J.K. and Scelza,D.A. "Medusa: An Experiment in Distributed Operating System Software," Communications of ACM, v. 23, pp. 92-105, February 1980.

22. Smith, D.C., "The star User Interface: An Overview," Proceeding of the AFIPS National Computer Conf., pp. 515-528, 1982.

23. Dion, J.,"The Cambridge File Server," ACM Operating System Review, October 1981.

24. Mitchell, J. and Isreal, J., "issues in Design and Use of Distributed File System," ACM Operating System Review, June 1982.

25. Older, W. and Fridrich, M., "The FELIX File Server," Proceeding 8th Symp. Operating System Principles (SIGOPS), PP. 37-44, December 1981.

26. Ellis, C.S. and Floyd, R.A., "The ROE File System," Proceeding 3rd Symp. Reliability Distributed Software Database System, December 1983.

27. Ford, A., "Violet: an experimental decentralized system," Operating SystemReview, v. 13, n. 5, December 1979.

28. Swinchart, D. and Boggs, G., "WFS: A Simple Shared File System for a Distributed Environment," Proceeding 7th Symp. Operating System Principles, December 1979.

29. Mitchel, J.G. and Dion, J., "A comporison of Two Network Based File Servers," Communication on ACM, v. 25, pp. 233-245, April 1982.

30. Bell System Technology Report A Real Time Database Management System for ESS, by Barclay, D.K. and Byrne, E.R., November 1982.

31. Ayache, J.M. and Diaz, M., "REBUS: A Fault Tolerant Distributed System for Industrial Control," IEEE Transaction on Computers, v. C-31, July 1982.

32. Schwartz, R.C. and Mellar, P.M., "Formal Specification and Mechanical Verification of SIFT," IEEE Transaction on Computers, v. C-13, July 1982.

33. Apollo Computer Inc., Apollo Domain Architecture, February 1981.

34. Stone, H.S., "Multiprocessor Sheduling with the Aid of Network Flow Algorithm," IEEE Transaction on Software Engineering, v. SE-3, January 1977.

35. IBM Research Report RJ2914, Object Naming and Catalogue Management for Distributed Data Manager, by Lindsay, B., August 1980.

36. Berstein, P. and Goodman, N., "Concurrency Control in a Sytem for Distributed Database," ACM Computer Surveys, v. 13, n. 2, June 1981.

37. McGraw, J.R. and Andrews, G.R. "Access Control in Parallel Programs," IEEE Transaction on Software Engineering, v. SE-5, January 1979.

38. Thomas, R.H., "A Majority Consensus Approach on Concurrency Control for Multiple Copy Database," ACM Transaction on Database Sytems, v. 4, n. 2, 1978.

39. Jones, A.K., "The Object Model:A Conceptual Tool for Structuring Software," Springer-Verlog:Lecture Notes in Computer Science, v. 60, 1978.

40. Tanenbaum, A.S., Computer Networks, pp. 97-127, Prentice-Hall 1981.

41. Stephen, A., "A Distributed System for Real-time Transaction Processing," IEEE Computer, v. 14, 1981.

42. Kung, H.T. and Robinson, J.T., "On Optimistic Methods for Concurrency Control," ACM Transaction on Database Systems, v. 6, n. 2, June 1981.

43. Tenny, R.R and Sendell, N.R., "Structures for Distributed Decision Making" IEEE Transaction on System Management, v. SMC-11, pp. 517-527, Agust 1981.

44. Gallenger, R., "A Minimum Delay Routing Algorithm Using Distributed Computation," IEEE Transaction on Communication, v. COM-25, June 1977.

45. Liskow, B. and Schifler, R., "Guardians and Actions:Linguistic Support for Rebust, Distributed Programs," Proceeding 9th Symp. Principles for Programming Languages, pp. 7-19, January 1982.

46. Moss,J.E.B., "Nested Transactions and Reliable Distributed Computing," Proceeding 9th Symp. Reliabiltity Distributed Database Systems, July 1982.

47. Skeen, D. and Stonebraker, M., "A Formal Model of Crash Recovery in Distributed Sytems," IEEE Transaction on Software Engineering v. SE-9, n. 3, May 1983.

48. Department of Computer Science Cornell University Ithaca Technical Report, The Inherent Cost of Nonblocking Commitment, by C. Dwark and D. Skeen, May 1983.

49. Gray, J.N., "Notes on Database Operating Systems," Springer-Verlog Operating Systems:An Advance Course 1979.

50. Schoum T.I, "A Decentralized Termination Protocols," IEEE 1st Symp. Reliabilty Distributed Software Database Systems, 1981.

51. Stankovich, D., "Load Balancing in Distributed Systems," IEEE Transaction on Software Engineering, v. SE_8, n. 4, July 1982.

52. Chu, V.W. and Efe, K., "Task allocation in Distributed Data Processing," IEEE Computer, v. 13, pp. 57-69, November 1980.

53. Efe, K., "Heuristic Models Of Task Assignment Sheduling in Distributed Systems," IEEE Computer, v. 15, June 1982.

54. Store, H.S., "Control of Distributed Process," IEEE Computer, V. 11, pp. 97-108, July 1978.

83

55. Rome Development Center RADC-TR-81-203 Decentralized Resource Management in Distributed Computer Systems, by H.L. Applewhite and E.D. Jensen, February 1982.

56. Naval Postgraduate School Final Report, A Distributed Operating System Design and Dictionary/Directory for the Stock Point Logistics Integrated Communication Environment, by N.F. Schneidewind and D.R. Dolk, November 1983.

57. Topscott, D., "Investigating the electronic office," Datamation, v. 6, n. 2, February 1981.

58. Haber, L., "Super Minis Shape up as Departmental CPUs," Mini-Micro Systems V. 5, n. 5, April 1985.

59. Hogan, W. and McCormick, F., "Benchmark Figure Reveals Power of Super Minicomputers," Research and Development, v. 34, April 1985.

60. Serlin, O., "Departmental Computing:A Choise of Strategies," Datamation, v. 10, n. 6, May 1985.

61. Nutt, G.J., "An experimental Distributed Modeling System," ACM Transaction on Office Information Systems, v. 1, n. 2, April 1985.

62. Phillips, S., "Distributed Systems and Their Protocols," Computer Comminications, v. 8, n. 4, April 1984.

63. Saltzer, J.H. and Reed, D.P., "End-to-End Arguments in System Design," ACM Transaction on Computer Systems, v. 2, n. 4, November 1984.

64. Chandy, K.M. and Lampart,L. "Distributed Snapshots:Determining Global States of Distributed Systems," ACM Transactions on Computer Systems, v. 3, n. 1, February 1985.

65. Terplan, K., "Performance Impacts of Network Architectures," Journal of Capacity Management, v. 2, n. 3, 1984.

66. Navy Personnel Research and Development Center NPRDC TR 85-1, The ZOG Technology Demonstration Project:A System Evaluation of USS Carl Vinson (CVN 70), December 1984.

67. Paker, Y., Multi-Microprocessor Systems, pp. 48-77, Academik Press Inc., 1983.

68. Stankovich, J.A., "A Perspective on Distributed Computer Systems," IEEE Transaction on Computer Systems, v. CS-6, n. 2, February 1984.

69. Martin, J., *Design and Strategy for Distributed Data Processing*, pp. 243-270, Prentice-Hall Inc., 1981.

70. Boar, B.H., *Application prototyping*, John Wiley & Sons, 1984.

INITIAL DISTRIBUTION LIST

		No.	Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145.		1
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100.		2
3.	B. Frew, Code 54 Fw Naval Postgraduate School Monterey, California 93943-5100.		1
4.	Professor T. Bui, Code 54 Bd Naval Postgraduate School Monterey, California 93943-5100.		1
5.	Deniz Kuvvetleri Komutanligi Bakanliklar, Ankara/TURKEY.		5
6.	Deniz Harp Akademisi Komutanligi Ayazaga, Istanbul/TURKEY.		1
7.	Deniz Harp Okulu Komutanligi Tuzla, Istanbul/TURKEY.		1
8.	Haldun Pelit Gulluk Pelit Motel Milas, Mugla/TURKEY.		1