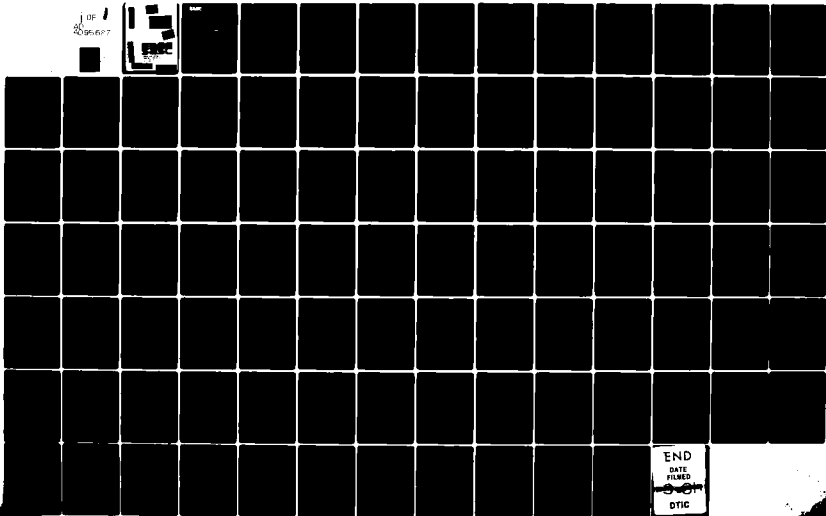
AD-A095 687 SYSTEMS AND APPLIED SCIENCES CORP RIVERDALE MD F/G 9/2
STAR NETWORK DISTRIBUTED COMPUTER SYSTEMS EVALUATION RESULTS.(U)
FEB 81 N DEMESA, I SHOLA DASG60-80-C-0010
UNCLASSIFIED SASC-ADPSI-CD-013 NL

1 OF 1
AD
A095687



SYSTEMS AND APPLIED
SCIENCES CORPORATION
6811 KENILWORTH AVENUE
RIVERDALE, MARYLAND 20840

SYSTEMS AND APPLIED SCIENCES CORPORATION

12 Feb 81

FINAL REPORT.

2

STAR NETWORK
DISTRIBUTED COMPUTER SYSTEM
EVALUATION RESULTS.

SASC-ADPSI-CD-013

Contract DASG 60-80-C-0010

[redacted]

12 Feb 81

The views, opinions, and/or findings contained in this report are those of the author(s), and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

12 93

Prepared by:
N. DeMesa
~~Sr. Data~~ Systems Engineer

Prepared by:
I. Shola
Sr. Data Systems Analyst

Approved by:
J. Danaher
Project Manager

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

15 16

PREFACE

This technical report is provided to the U.S. Army Ballistic Missile Defense (BMD) Systems Technology Program Office in fulfillment of CDRL Item A02 of the DDP Star Network Investigation, contract number DASG60-80-C-0010. This contract is one of element of the advanced Data Processing Subsystem Investigations (ADPSI).

This is the final of two technical reports addressing hierarchical network design concepts for the BMD Underlay System. The purpose of this report is to describe two candidate hierarchical network architectures for the BMD Underlay System. The report also contains the detailed network evaluation technique which was used in the synthesis of the final candidate architectures. Further, this report contains results and measurements used to validate the candidate architectures.

This report contains a number of inventions which were a result of this study effort. They include a conceptualization of a ZONE defense mechanism which reduces processing requirements for BMD Underlay, and the development of quantitative implementation risk functions.

By providing technical information and establishing a friendly atmosphere the seed for the above inventions was provided by Messrs. S. Liu and R. W. Parker of McDonnell Douglas Astronautics Company. Their patience is greatfully acknowledged.

Comments and suggestions are welcomed by Systems and Applied Sciences Corporation for consideration and incorporation as revisions to the report. Please submit your comments to Messrs. J. Danaher or N. P. DeMesa III, Systems and Applied Sciences Corporation, 1100 South Claudina Place, Anaheim, California 92805, or phone (714) 999-1177.

RECORD OF CHANGES

REV.	DATE	REVISION DESCRIPTION
Orig.	6-24-80	Draft Version

CONTENTS

Appendices

FIGURES

TABLES

1.0 Introduction

This report is a continuation of the Preliminary Hierarchical Network Design Report (SASC-SED-LA-107). In the initial report the requirements of the BMD Underlay System were depicted using data flow analyses, which included data flow diagrams and narrative descriptions of the processes and the data flows between processes. Also, the characteristics of cannonical hierarchical networks were described and a candidate network design was proposed based on the data flow analyses.

This report refines and expands on this design and presents an alternate design. A method of quantitative design risk assessment of large, high throughput DPS functions like the BMD Underlay System is presented. Finally, a model of executive overhead is presented based on a new queue model, which reduces model error associated with high processor utilization.

In reviewing design literature, it has been determined that a set of design criteria must be established. Without a succinct quantitative set of design criteria, designers focus on a few issues at the exclusion of all others. Consequently, the first goal of this report is to establish a set of design guidlines which are to be used in the network evaluation and design. These guidelines represent an expansion of the MDAC guidelines.

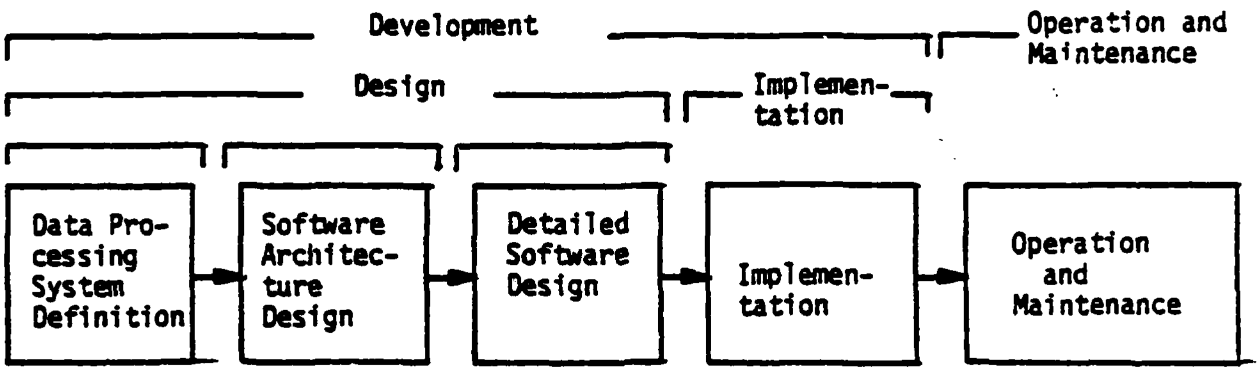
The network evaluation technique is structured to create a design which conforms to the design criteria with minimum effort. In the normal software development cycle described by Figure 1-1, the systems design is not restricted to a particular topology. Traditional designs and evaluation techniques present topological alternatives. It is not within the scope of this report to present these tradeoff's. However, criteria and guidelines which are not topologically dependent will be identified. The hierarchical network characteristics included herein as Appendix C will prove useful in evaluating topological tradeoff's.

This report separates functional partitioning from the network evaluation technique. This separation clarifies two new concepts which are the result of this study:

- The creation of quantitative design risk functions

- The creation of the zone defense concept

These two concepts are applicable to any distributed implementation of the BMD Underlay System.



Software System Life Cycle

FIGURE 1-1

2.0 Summary

The remaining sections of this report are described as follows:

Network Evaluation Technique (Section 3.0) - This section summarizes the analysis performed on the candidate networks.

Functional Partitioning (Section 4.0) - This section summarizes the rationale for the new design of the TAP tasks.

Control & Communications (Section 5.0) - This section summarizes the development of a hypothetical executive used in the network evaluation.

Candidate Networks (Section 6.0) - This section summarizes the important characteristics of the candidate networks.

Conclusion (Section 7.0) - This section presents the conclusions obtained from the various sections of this design study.

3.0 Network Evaluation Technique

The Network Evaluation Technique is based on meeting the minimum network design criteria and minimizing risk functions. Both are described in Appendix A. Due to the complex interaction of the design goals, simultaneous consideration of all system goals and constraints is not practical. Three distinct phases of the design evolve to satisfy the goals and constraints. These three phases are used iteratively to arrive at the final two candidate networks.

First, the network guidelines are applied to the data flow diagrams to create candidate network designs. Since this phase does not involve quantitative assessment, compliance is determined by inspection (desk analysis).

The second phase of analysis involves computation of the various risk functions to determine how efficiently the network design performs the critical thread functions. Further, the design is analyzed to determine that the rules of hierarchical decomposition are being folliwed. The critical threads are then evaluated assuming overhead and delays based on a hypothetical executive.

The third and final phase of the network evaluation technique includes the evaluation of queue lengths and supplying various probablistic functions to determine the messages per enablement of various critical threads. At this point, the overhead and the risk functions are recomputed. The goal is to substantiate overhead and throughput predictions caused by executive overhead, which depend on queue length and messages per enablement.

In general, the network design/evaluation cycle would continue until designs with acceptable risk levels meet minimum design criteria. However, the development of risk functions clearly indicate that major risk reduction is only possible if the critical threads are restructured to require less processing or if the processing capacity (MIPS) of the nodes (processors) is increased.

4.0 Functional Partitioning

Previous research has not shown any clearly superior approach to the functional partitioning of tasks to nodes in a distributed processing system. Previous MDAC studies have developed a great deal of data in this area. The initial hierarchical network analysis report summarized these results by the use of annotated data flow diagrams. However, a simple method of task partitioning by inspection of the data flow diagrams proved difficult. The lack of a simple methodology forced a reevaluation of the definition of functional partitioning.

Functional partitioning is divided into two distinct areas. First, the characterization of units of work, (task definition) is created to conform to the design criteria. Second, to minimize design risk (providing an evolutionary plan for system expansion, etc.).

Data flow diagrams should indicate a preference network topology by inspection. The existing data flow diagrams indicated no topology preference. The first phase of functional partitioning must be structured toward a distributed architecture so the data flow diagrams will exhibit a distributed topology by inspection.

Reevaluation indicated that a violation of one of the first rules of structured design had been commited. A detailed design had been created from another detailed design not from a conceptual design. (See Figure 1-1 for correct methodology).

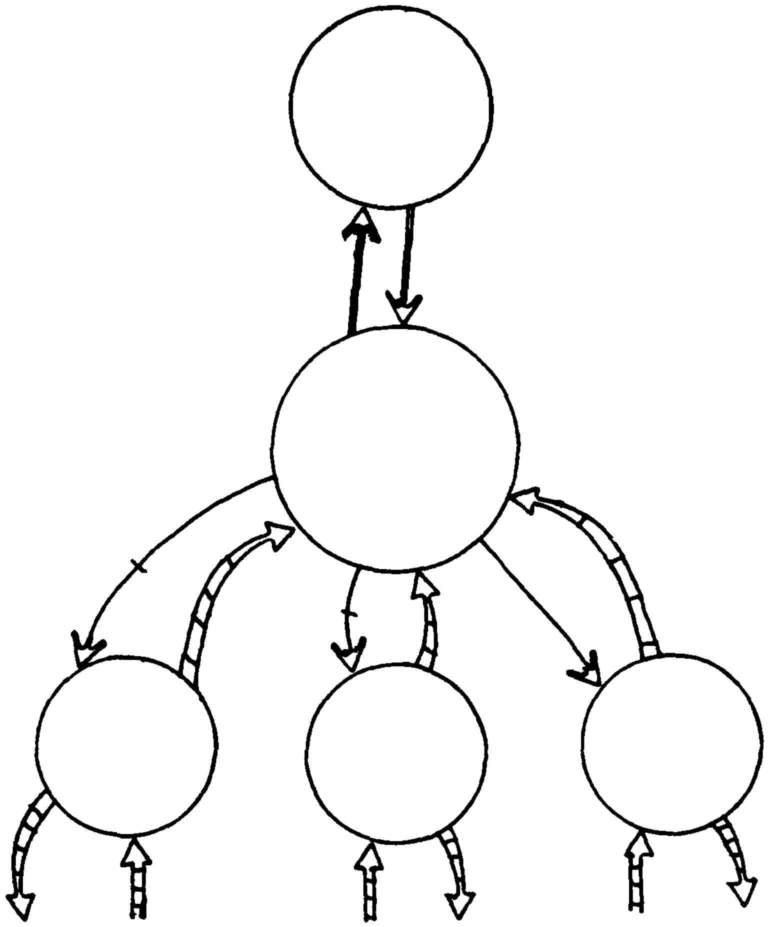
As a result, an entirely new tap task organization has been created and described in detail in Appendix B. Tasks which perform similar functions have the existing tap task name with the prefix H. Their respective processing requirements have been determined from extrapolation from the existing tap tasks. The major emphasis in the conceptualization of these new tasks has been distributed control. This emphasis is chosen because the hierarchical architecture is particularly well suited for handling distributed control.

Three abstract concepts must be defined and explained to continue any discussion on functional partitioning. Threads, tasks, and subroutines are used throughout documents of the BMD Underlay studies and are subject to various interpretations. For the purposes of this report, a subroutine is defined as an isolated unit of work based on the parnes structure design criteria. A task is defined as a unit of work based on the parnes criteria and on the identification of concurrent functions. A task is composed of subroutines. A thread is defined as a unit of work based on a distributed processing functional requirement or identification of concurrent functions. A thread is composed of tasks.

The task allocation method chosen is hierarchical process decomposition. This hierarchical structure can be described by subordinates performing decisions on input data and passing their conclusions and limited data to superiors for further decisions and allocation of additional resources. This structure is described in Figure 4-1. Additional hierarchical constraints include relaxed time constraints on the analysis by superior nodes and a general reduction in the amount of communications traffic as one traverses up the hierarchy.

The system, as previously implemented, is depicted in Figure 4-2. The basic hierarchical structure is present, described by reduced communications at higher levels. An alternate structure described in Figure 4-3 reduces the timing requirements at the higher levels. The major drawback of this approach is that the average total system MIPS requirements is increased, these are described in more detail in Appendix B. This is typical of the tradeoff's between distributed and central control.

The conceptual designs which result from functional partitioning are used in the network evaluation by combining task processing requirements with the executive processing requirements. The processing requirements of the executive are extrapolated from a hypothetical executive design.



HIERARCHICAL PROCESS STRUCTURE

Figure 4-1

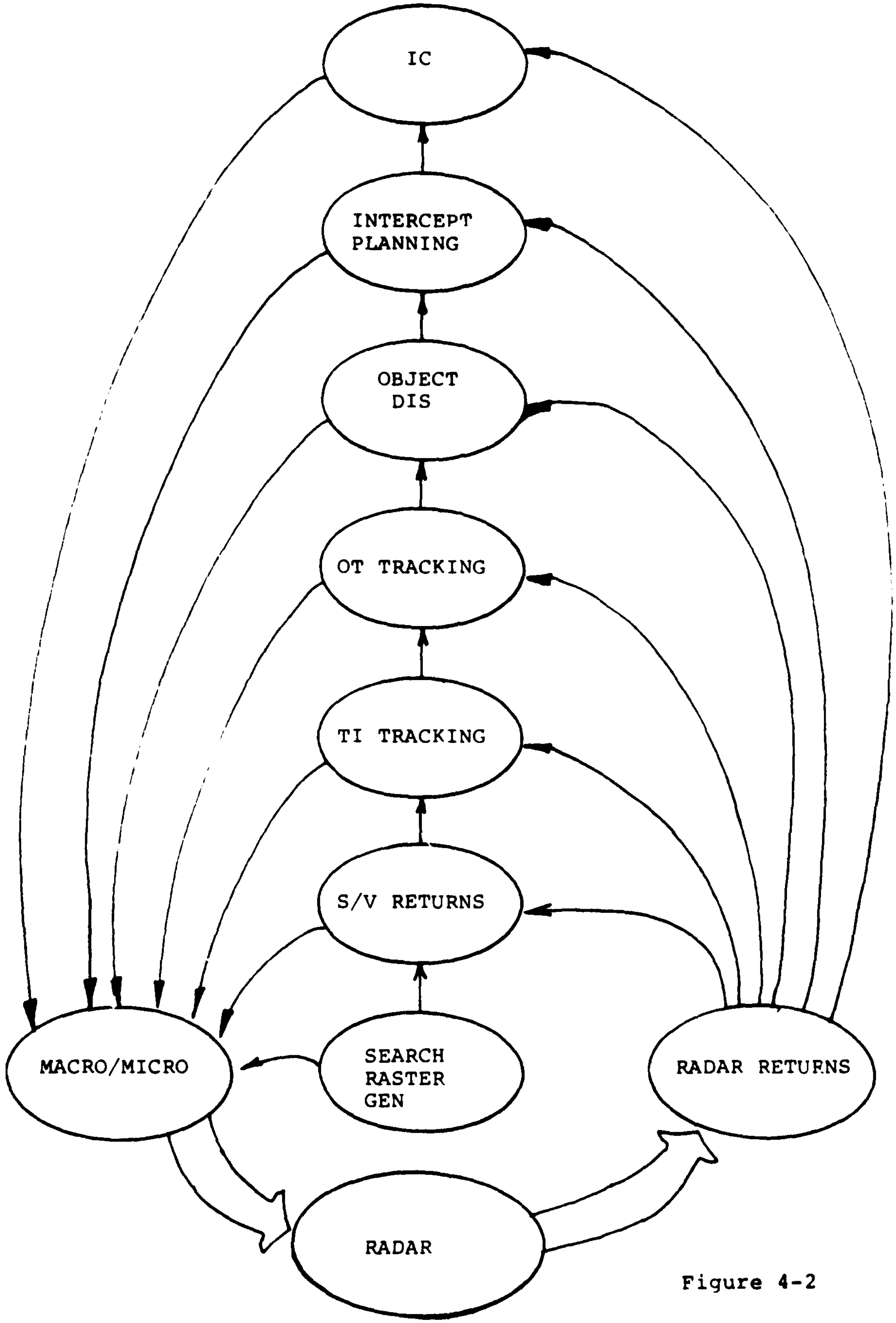

IC
INTERCEPT PLANNING
OBJECT DIS
OT TRACKING
TI TRACKING
S/V RETURNS
MACRO/MICRO
SEARCH RASTER GEN
RADAR RETURNS
RADAR

Figure 4-2

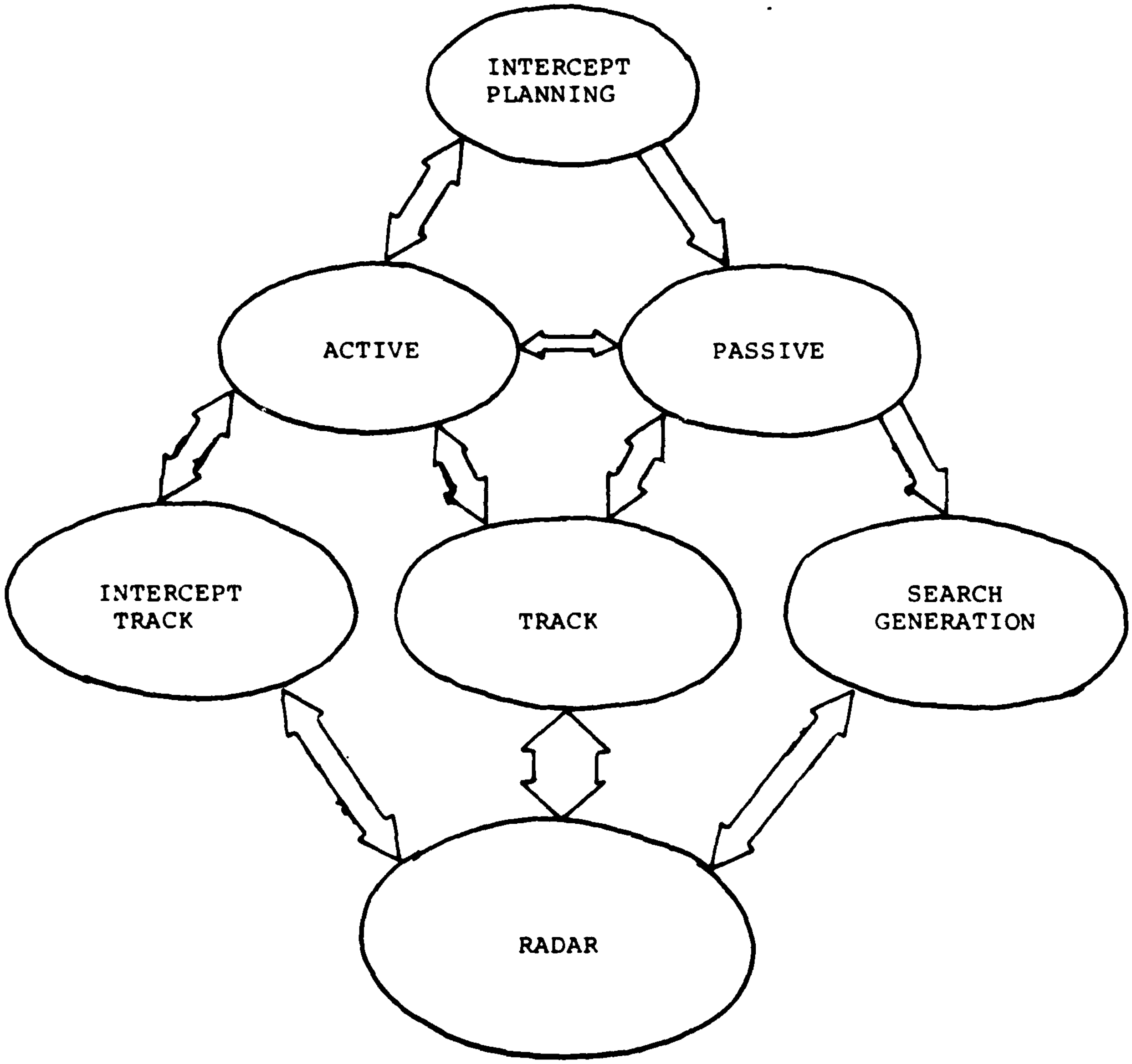

INTERCEPT
PLANNING
ACTIVE
PASSIVE
INTERCEPT
TRACK
TRACK
SEARCH
GENERATION
RADAR

Figure 4-3

5.0 Control & Communications Overhead and Delays

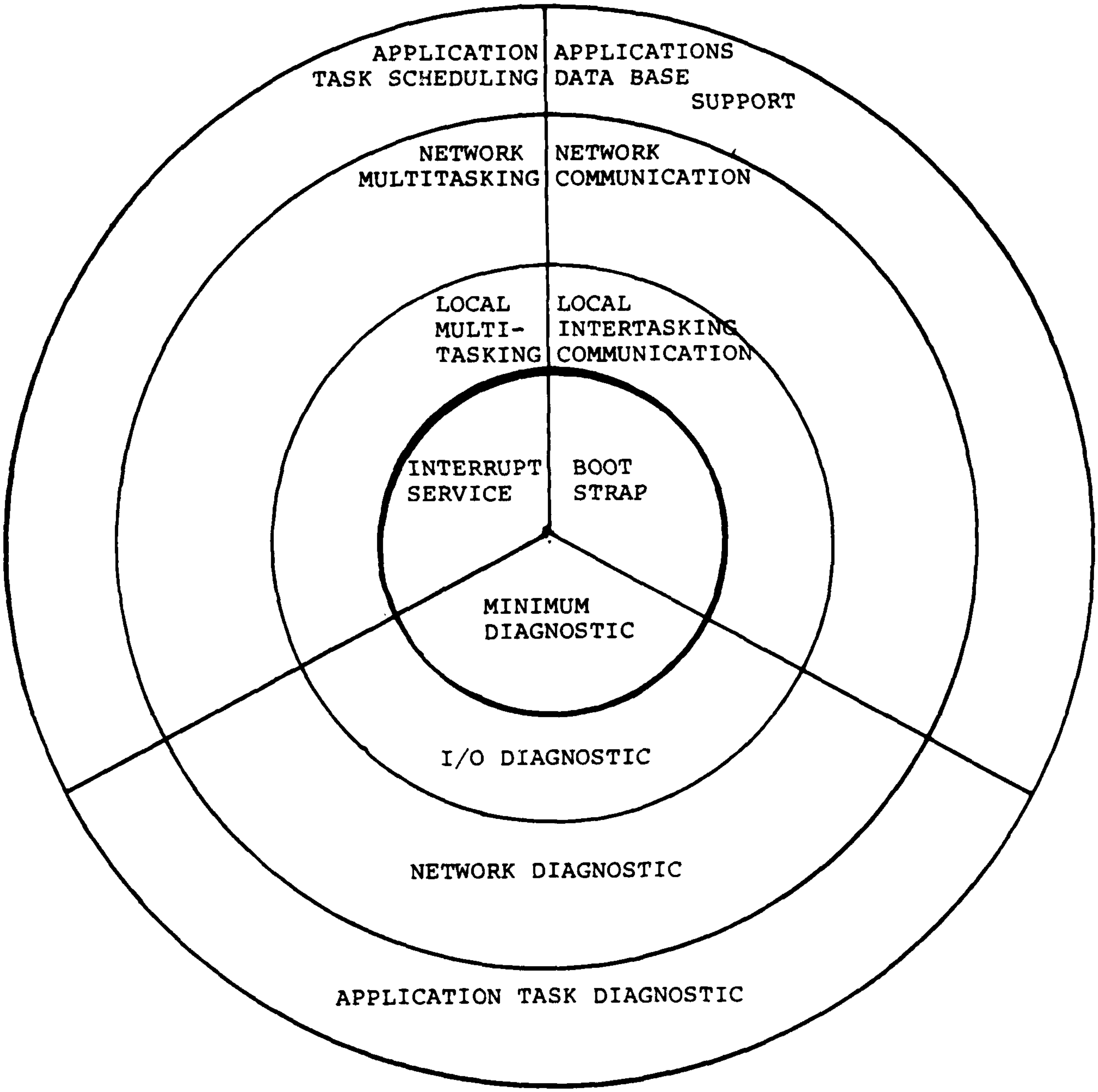
The determination of the executive processing requirements and the executive delays requires the creation of a hypothetical executive and the determination of the communications overhead and delays. The DPS executive has been hypothesized with a number of design goals which are in Appendix G. In summary, these design goals include support of fault isolation, use of HOL's (High Order Languages), and support of distributed data flow control.

The executive is a layered executive described in Figure 5-1. The executive initiates the search and verify task. All other task invocation is done by data requests. Since the system is a data flow system, the task invocation can be modeled via petri net diagrams (see Figure 5-2) which enhance isolation of logical errors and provides information about network stability.

The estimation of the executive overhead and delays is performed from a model of the executive which is created from queue models of each layer of the executive. The structured decomposition of the executive provides a hierarchical queue model with expandable accuracy.

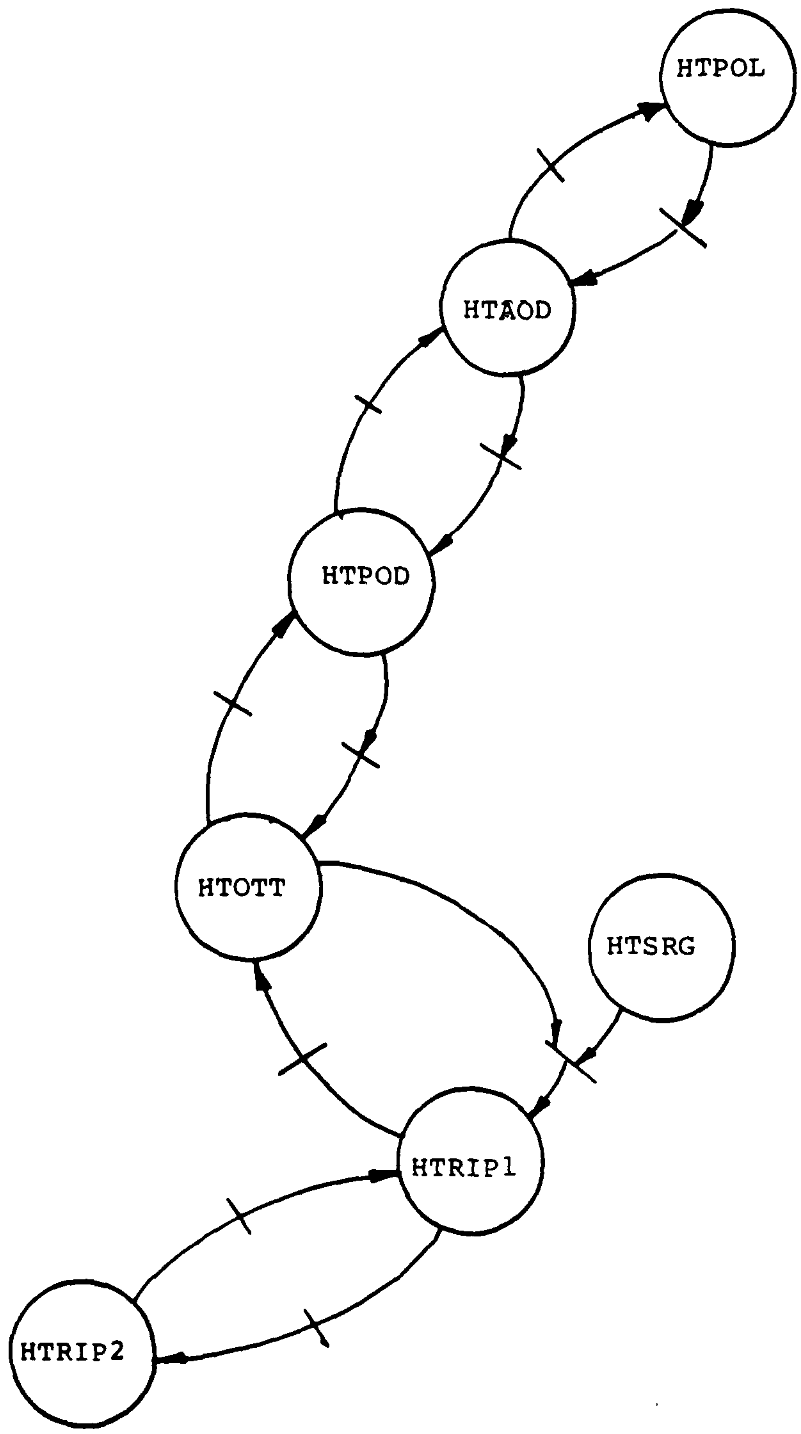
To evaluate communications overhead requires the creation of a hypothetical communications protocol. This protocol must support communications between any two nodes in the system without violating distributed control rules. (See creation of a central routing table, Appendix G). Further, the protocol should distribute the existing overhead to the sending and the receiving nodes and not at intermediate routing nodes. As a result, the system data flow diagrams can be used for a direct computation of throughput requirements with minimization of delays and hidden bottlenecks.

The communications protocol chosen provides a tree address scheme in the header of each message in the system. Routing information is appended to the header of a message as the message transverses levels in the hierarchical network. Communications overhead is modeled as the additional number of bytes appended to a message relative to the message size in bytes.



EXECUTIVE HIERARCHY

Figure 5-1



PETRINET DESCRIPTION OF NEW TAP TASK STRUCTURE

Figure 5-2

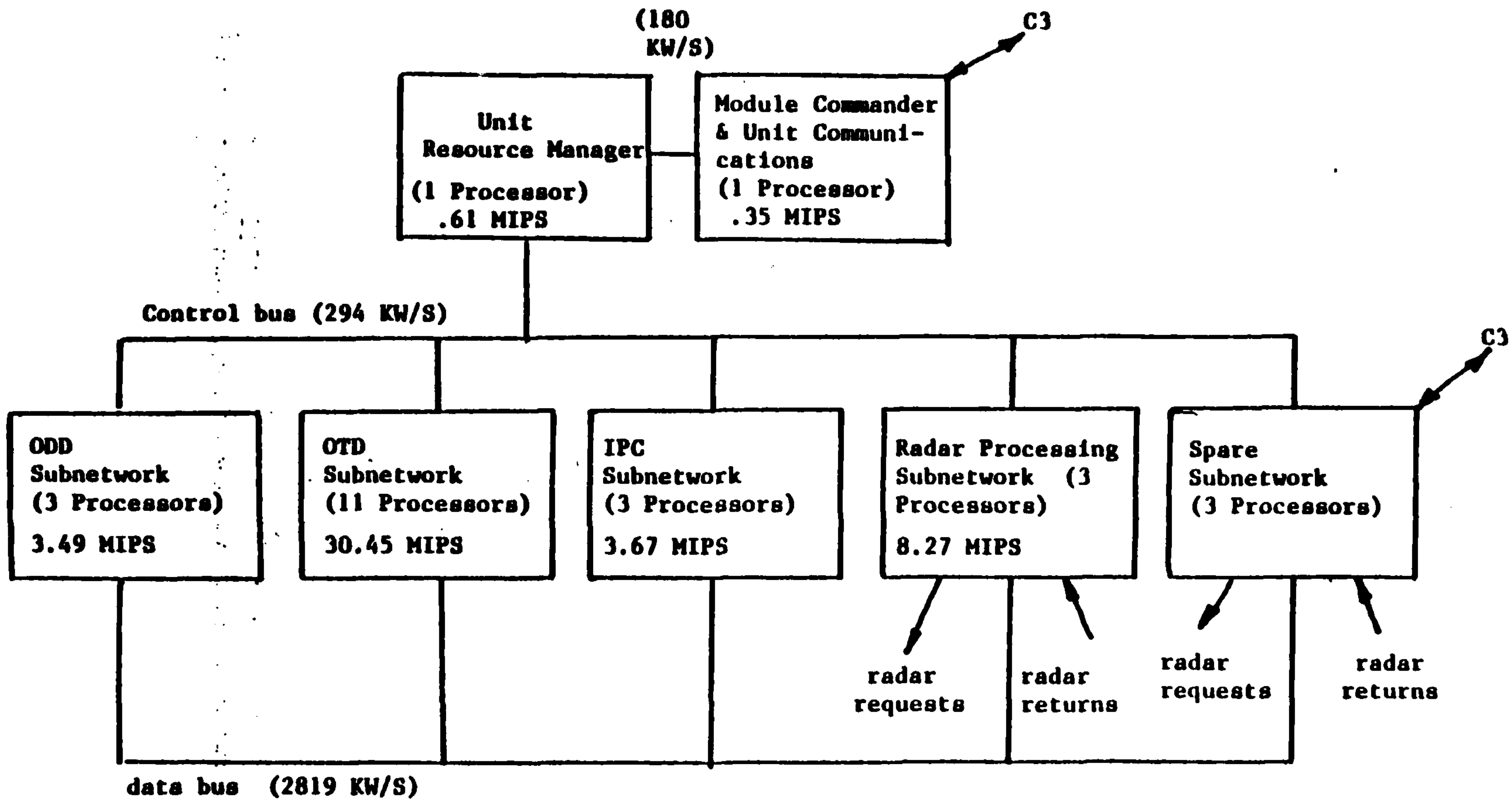
6.0 Candidate Networks

Two types of hierarchical topologies are considered for use in the BMD Underlay DPS. The first network is constructed of 4 MIPS minicomputers. The second network is constructed of 1 MIPS microprocessors.

The initial candidate network of 4 MIPS minicomputers has been substantially redesigned from the design described in the Preliminary Report. The initial configuration is described in Figure 6-1. The major rationale for its redesign is its sensitivity to workload (See Figure 6-2). Further, too much information about its design limitations are available from inspection. The new design is capable of supporting the existing tap task structure and the task structure proposed in this report. It can be described as a processing pyramid with task functional assignments based on hierarchies within the pyramid (See Figure 6-3).

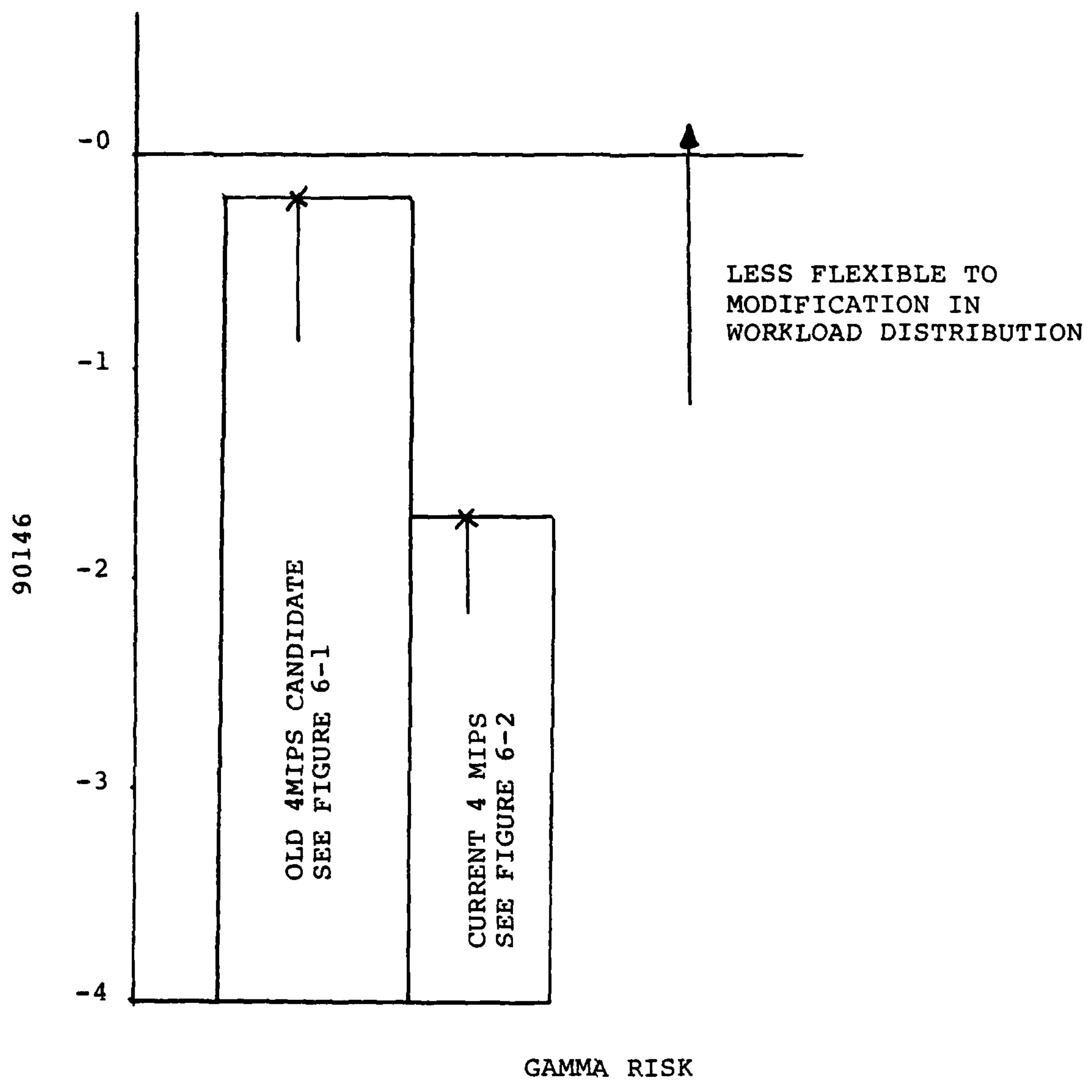
The second candidate network is composed of 1 MIPS microprocessors (See Figure 6-4). The configuration is an expanded pyramid and represents an expansion of the topology used for the 4 MIPS processor configuration. Due to the design symmetry the 1 MIPS topology can be described by Figure 6-5 which is similar to the 4 MIPS configuration with different MIPS per node.

The detailed configurations are described in Appendix E and Appendix H. The GE FFP was chosen for the 4 MIPS topology and the Intel IAPX432 was chosen for the 1 MIPS topology. The rationale for the choice of these processors is included in these appendices.



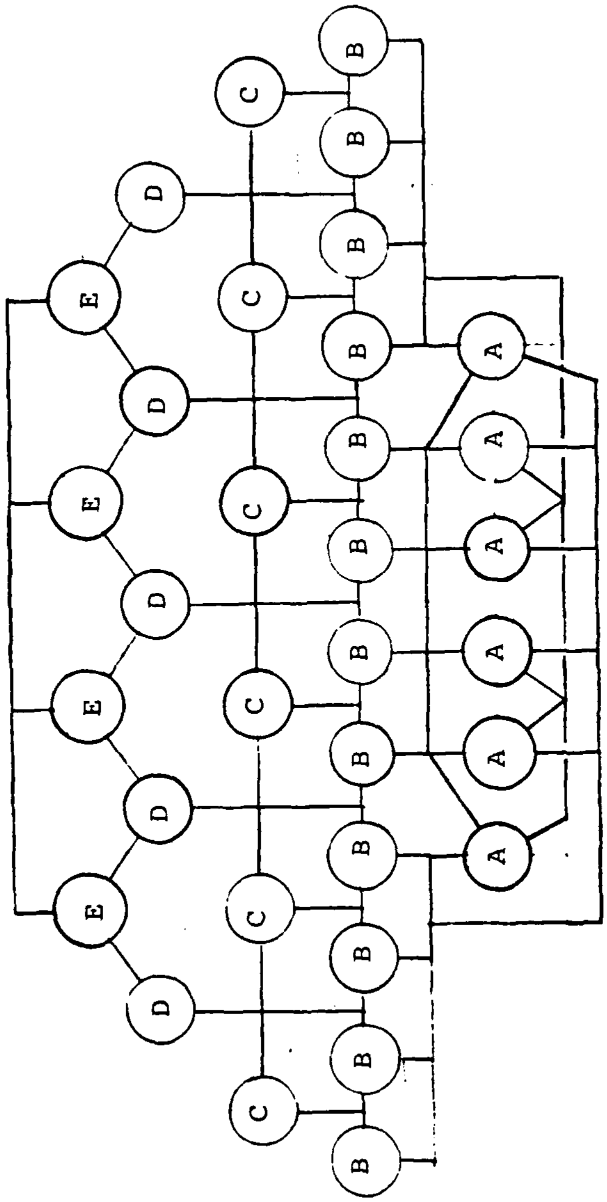
PRELIMINARY BMD UNDERLAY NETWORK DESIGN

Figure 6-1

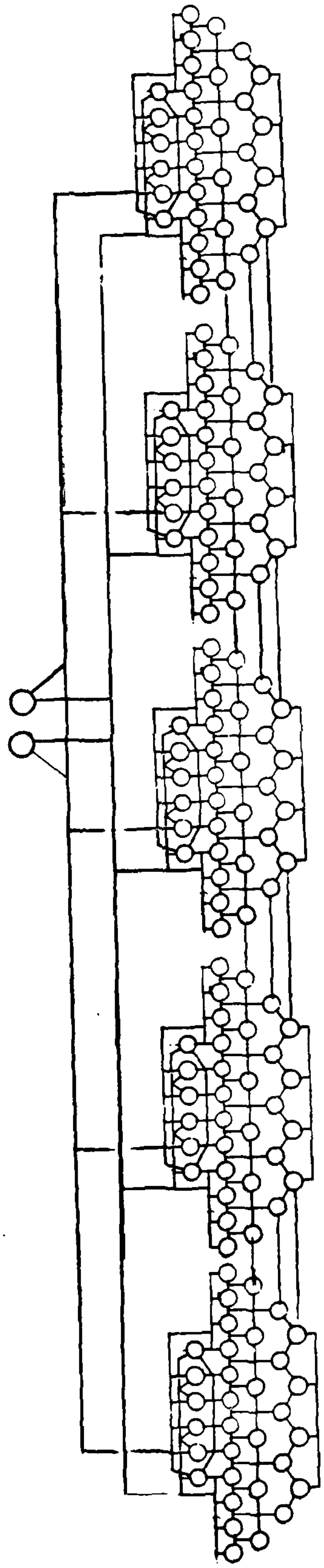


LONGEST TIME CRITICAL THREAD

Figure 6-2

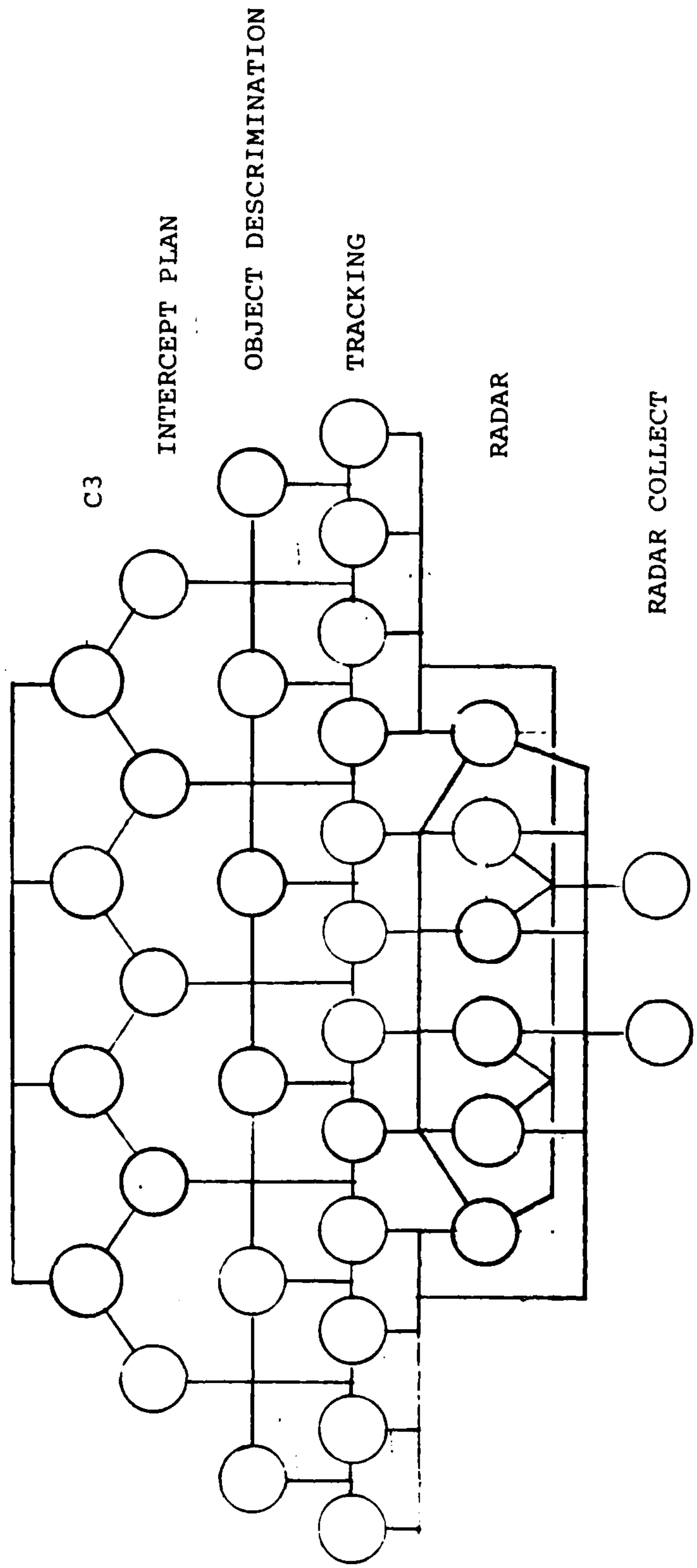


4MIPS TOPOLOGY FIGURE 6-3



1 MIPS TOPOLOGY

FIGURE 6-4



1 MIPS EQUIVALENT
FIGURE 6-5

7.0 Conclusions

A software model for executive overhead has been developed for distributed processing with emphasis on ease of software development by isolation of synchronization and other topological details from the TAP tasks. Further, restrictive interfaces have been developed to assist in software validation and fault isolation.

Six risk functions have been created for design evaluation. Quantitatively, these functions have identified risks associated with distributed processing and central processing designs. The alpha, beta, gamma, and delta functions are inventions which are a direct result of this study. The epsilon and the phi functions are a result of research by Finkel et al[1].

Two types of hierarchical topologies have been presented as possible candidates for the BMD Underlay DPS. One topology is constructed from 4 MIPS minicomputers (GE FFP). The second topology is constructed from 1 MIPS microprocessors (IAPX432). The merits of the various topologies are analyzed by evaluating their corresponding risk functions.

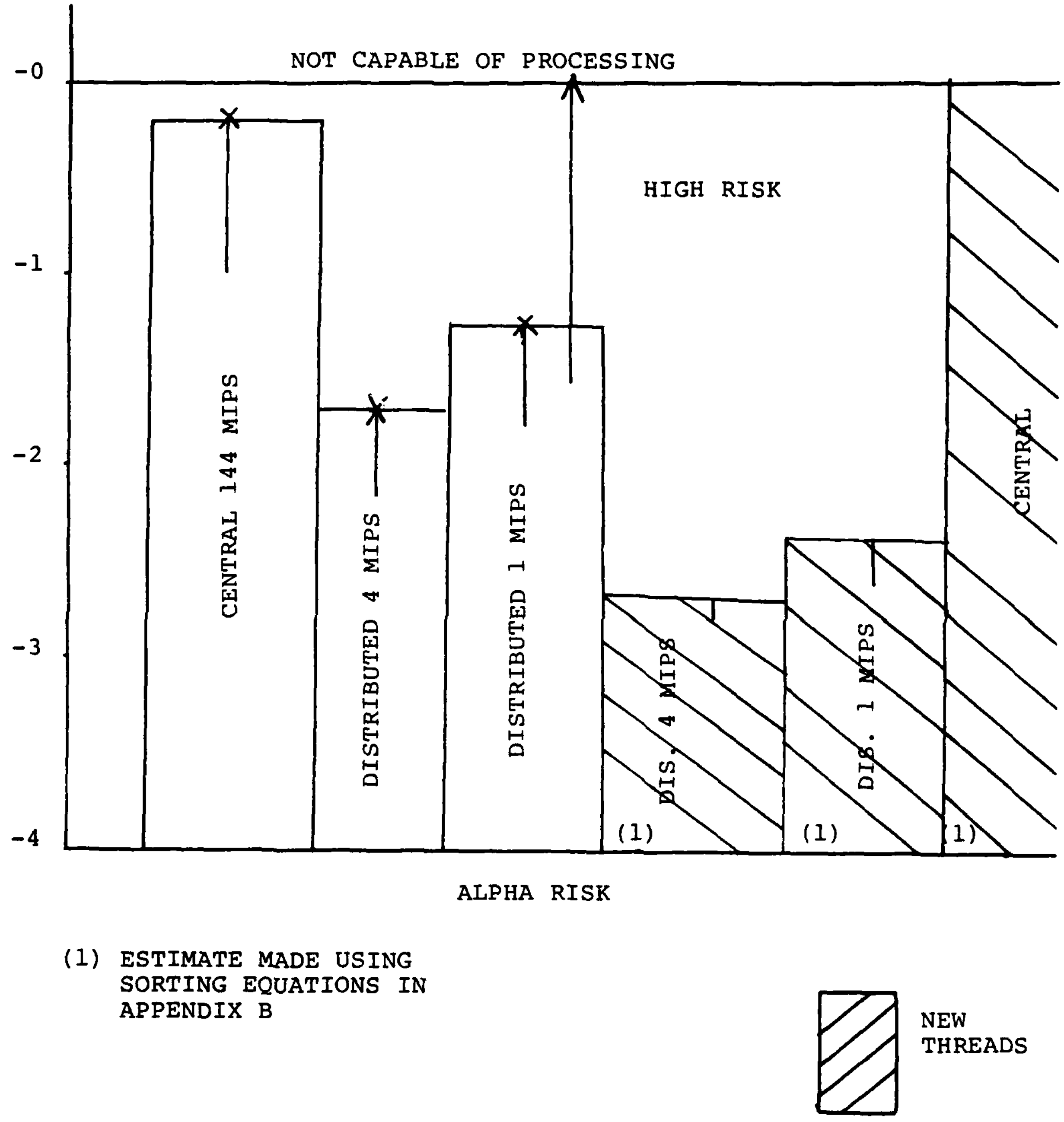
The 1 MIPS topology using microprocessors has appeal because it can be used as the basis for the LOAD system and tactical image processing systems. Further, because of the small data requirements a segment of TOTT is a good candidate for VLSI. It can attain the necessary parts volume because it can be used in a variety of guidance systems, in the LOAD system, and in tactical image processing systems.

The candidate networks exhibit the following characteristics because they are hierarchical: (1) support of the use of concurrent HOL's[2], (2) reduction of the conceptual complexity of the system due to the structured hiding of information, and (3) support of levels of conceptualization in the understanding of the system.

The candidate networks exhibit the following characteristics because they are symetrical: (1) reduction in the volume of system software due to conceptual recursion, (2) reduction in the system complexity due to a reduction in the number of details in the system, and (3) reduction in counter measure effectiveness.

In summation, the BMD Underlay System has been identified as a conceptual hierarchical control and a physical hierarchical resource system. The major characteristics of these hierarchies have been summarized. The negative characteristics of a resource hierarchy are inescapable with the current definition of the BMD Underlay System. However, implementation of the positive characteristics of hierarchical control require creation of a hierarchical topology.

The ability of the hierarchical network to perform a number of other tactical applications attests to its versatility. Its ability to support levels of comprehension expands its supportability. In the final analysis it will be the supportability of a tactical system which will determine its true deterrent capability.



LONGEST TIME CRITICAL THREAD

Figure 7-1

APPENDIX A - A Network Design Guideline

A.1 Design Guidelines

The network design guidelines are a quantitative assessment of the DPS's ability to meet: (1) system throughput requirements, (2) system availability requirements, (3) risk requirements, and (4) system cost constraints.

System throughput and system availability are boolean requirements for the scope of this design effort, that is, the candidate design must meet a minimum set of criteria for both of these requirements. However, how easily it meets these requirements and how much reserve capacity the design includes is subject to a quantitative assessment.

The system implementation risk has not been quantitatively considered in previous BMD studies. Concern for meeting throughput requirements has been paramount. The use of HOL's and new data structure techniques to reduce implementation cost and risk have been considered a luxury. Including them within a design does not preclude implementation which is devoid of these techniques. As one considers the evolving technology and delays in the implementation of the BMD Underlay System, ignoring these techniques does not seem prudent.

Consideration of system implementation costs is not within the scope of this study effort. However, due to various technologies considered, the recurring system costs can be affected by orders of magnitude. As a result they will be identified in general terms.

The four general categories described above are checked by the use of these detailed guidelines:

1) Does the design exceed node processing throughput?
2) Does the design exceed total processing throughput?
3) Does the design exceed communications throughput?
4) Is the design expandable?
5) Is the design easily modifiable?
6) Do the critical threads meet availability requirements?
7) Does the design promote easy validation?
8) Is the system stable?
9) Is the design easy to understand?

The above design criteria are common to most distributed system designs. However, strategic systems have additional contraints on their availability. Two criteria are added to the list:

1) Succeptability to unintentional sabotage

2) Succeptability to intentional sabotage

A.2 Risk Functions

The alpha, phi, epsilon, beta, gamma, and delta risk functions have been developed so that designs can be evaluated quantitatively. These risk functions provide a method to compare some of the desirable qualities which have been identified in the guidelines. The alpha, beta, gamma, and delta functions have been developed so that they can be used in both the conceptual and detailed design phases of the system. Further, they can be applied in selected subsections of the design with effectiveness.

A.2.1 Alpha Risk Function

In attempting to develop meaningful design criteria for construction of distributed processing architectures certain large system implementation problems should be noted. Most data processing systems with the size and complexity of 1000 man years or more experience difficulty in meeting system throughput requirements. The degree of miscalculation of the processing requirements to achieve the desired throughput is usually orders of magnitude. This is one of the major reasons for the long software support cycle. It is therefore reasonable to consider a design measurement (alpha) which evaluates system risk using a logic of the utilized processing potential.

In central processing systems, this design measurement (alpha) can be expressed as the logarithmic ratio of the total MIPS required to perform the system function to the total MIPS available.

ALPHA=LOG10(MIPS$REQUIRED/MIPS$AVAILABLE)

With distributed processing systems it is not readily apparent how one determines an equivalent design metric, because expansion is possible by the addition of processors. With the central processing approach additional processors cannot be added so alpha represents the potential risk of the implementation. Analysis of the metric alpha indicates it can be expressed as: (1) the total number of MIPS requires to perform the system function divided by the total number of MIPS available, or (2) the serial activity with the highest throughput requirement divided by the highest rate at which that serial activity can be performed.

In a Central Processor System:

TOTAL$MIPS$REQUIRED = LARGEST$MIPS$SERIAL$ACTIVITY
TOTAL$MIPS$AVAILABLE= LARGEST$THROUGHPUT$OFAPROCESSOR

This definition of alpha is equivalent for both the central processing system and the distributed processing system and represents a primary measure in determining the risk associated with both design methodologies.

A.2.2 Phi & Epsilon Risk Functions

Distributed processing systems exhibit limitations in expansion for two reasons. First, the hardware can limit expansion due to size, power, and communications fan out. Second, overhead and delays associated with control and communications can limit expansion.

Finkel et al[1] have derived a method of identification of internode distances and a method of identification of bus loading with the expansion of a particular topology. Internode distances can be used to identify topological expansion limitations due to bus communications fanout. Bus loading can be used to identify projected throughput requirements on communications lines with network expansion. These will be identified as the phi and epsilon functions in this report, respectively.

The phi function (internode distance) is determined by comparing the Hamming addresses of the various nodes. With the development of an equation for the expansion of a topology an equation for the average distance is developed. For more details, see the article[1].

The epsilon function (bus loading) is determined from the probability that a message between a random pair of nodes will traverse a particular communications bus. The epsilon function identifies the bus with the most traffic.

Finkel et al have identified the phi and epsilon function for a number of topologies: the star, the p-cube, the sparse flake, and the dense flake. If the topology of the proposed hierarchical networks are redrawn they resemble the dense flake. As a result, the equations developed by Finkel et al are used for the candidate topology.

A.2.3 Beta Risk Function

It is desirable to identify a risk function associated with network stability. The development of the risk function, beta, is based on a detailed analysis of various executive functions and specific applications functions. Specifically, all tasks whose processing time is a function of queue length are included in this function. The specific example considered are the TRIP execution and the executive overhead times developed in Appendix D.

Applications tasks and the executive perform a fixed amount of instructions and a variable amount of instructions per request. Significant variations in the variable number of instructions are associated with queue lengths. This tendency to expand units of work with the increase in workload is responsible for the traditional exponential utilization curve. Adjusting operation of the system so it functions in a less saturated region expands its workload capacity and enhances its stability.

Enhancement of stability is a direct result of designing a system so it operates in an environment where fixed instructions per functions are larger than the variable instructions per function. Although this problem has been developed macroscopically, the equivalent occurs microscopically at all levels of design detail. It is the emergence of slowdowns due to microscopic detail within the system which cause hidden bottlenecks.

The beta function has been defined as the logarithmic ratio of the number of variable instructions to the number of fixed instructions.

BETA=LOG10(VARIABLE$INSTRUCTIONS/FIXED$INSTRUCTIONS)

A.2.4 Gamma Risk Function

The assessment of topology specificity is important from countermeasure considerations and from identification of the software life cycle. Specifically, if the network topology provides enough information about the expected processing loads, appropriate counter measures can be developed. Identification of the software life cycle shows that workload and the characteristics of the processing will evolve and change while the system matures. The ability for the topology to support system maturation is another important risk function. This idea is best depicted in Figure A-1. Note that the area under the histogram represents the total processing requirement. Figure A-2 shows another histogram with the same area but with different distribution workload. The gamma function is developed to identify design sensitivity to a change in work distribution in the network.

The gamma function is defined as the logarithmic ratio of the maximum throughput of the smallest dedicated processing center to the maximum throughput of the system.

GAMMA=LOG10(MIPS$SMALLEST$CENTER/TOTAL$MIPS)

A.2.5 Delta Risk Function

The determination of control stability has been neglected in most DPS systems. As the complexity and the size of the systems has increased (with increases in throughput) this has become a problem. One of the major requirements for distributed control is that the control dynamics be quicker than the system dynamics. (See Appendix H for more details). Specific examples in BMD Underlay Systems include the constraints on intercept control. The delta function is developed to identify risk in not meeting control requirements.

The delta function is defined as the logarithmic ratio of the maximum throughput requirement of the largest control thread to the minimum throughput requirement of the smallest non control critical thread.

DELTA=LOG10(LARGEST$CONTROL$MIPS/SMALLEST$THREAD$MIPS)

A.2.6 Summary of the Risk Functions

The development of the alpha, beta, gamma and delta functions, like the development of the epsilon and phi functions of Finkel (6.1), is an attempt to identify quantitative risk criteria for the development of networks. Additional work must be done to associate these risk functions in various designs. Unlike the epsilon and phi functions, the alpha, beta, gamma, and delta functions can be applied to a hierarchical networks or a subset of the network. Their creation was justified based on the requirement for a macroscopic set of design criteria. However, they can be the basis for construction of microscopic analysis of an existing system (analysis of a detailed design).

Some interesting observations can be made about the identification of risk in a large distributed processing system. First, the alpha function clearly shows that conceptual design has significant impact in the risk associated with the implementation of a system.

The beta function impacts network simulation and queue modeling. Specifically, arrival assumptions made for analytical queue models accurately describe systems with small queue lengths. Further, when the queue lengths are long, the arrival distributions are unimportant (A.1). Quantitization of low and high beta metrics remain to be investigated.

The gamma function has provided conceptual insight into the importance of both hierarchical and symetrical design topologies.

The delta function can be used to identify risk of maintaining stable control in the system. The assessment of control stability is of significant importance in DPS systems with strategic deterent capabilities for tactical systems.

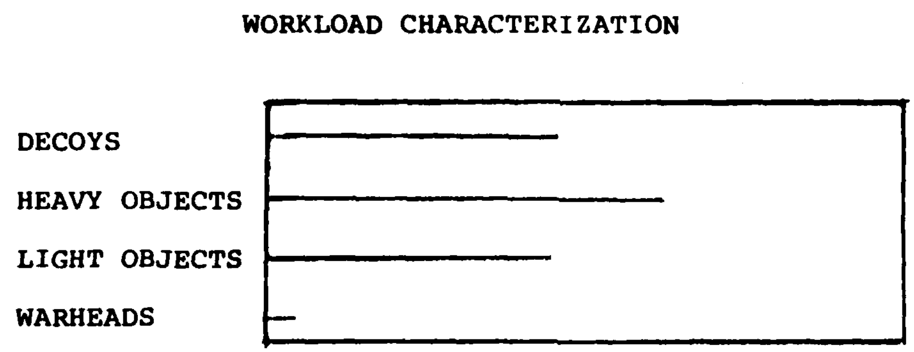


FIGURE A-1

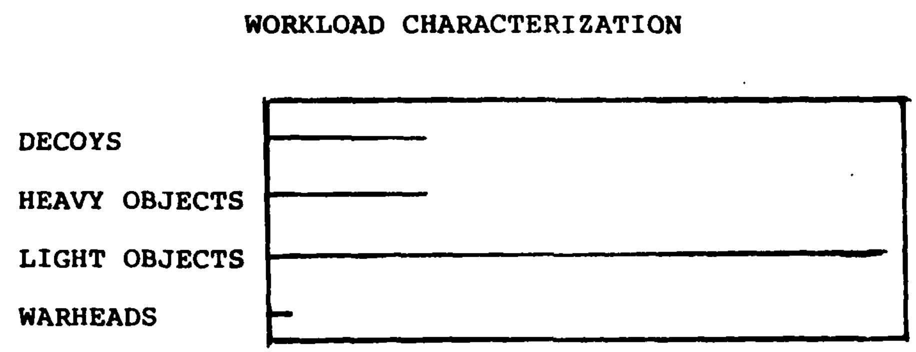


FIGURE A-2

APPENDIX B - Functional Partitioning

The task and the thread reorganizations, resulted by identifying the design risk and the design methodology which evolved from the development of large software systems. Specifically, this represents a conceptual redesign of the various TAP tasks.

Distributed control has been emphasized as providing a macroscopic reason for the redesign of the TAP tasks. This redesign will optimize a distributed hierarchical topology. The identification of a TAP task for detailed analysis will provide a microscopic example that will clarify techniques used to create distributed control.

A number of criteria have been developed to identify tasks which provide control. They include tasks which preceed and end threads, tasks which appear in multiple threads, tasks which are owner's of data bases which are shared by a number of tasks, and tasks which invoke other tasks. From inspection of the threads, TRIP and TC31 have been identified as providing control.

TRIP has been selected for detailed evaluation because: (1) it is critical to the operation of the system, (2) it has the most number of enablements per second, (3) it has a high beta, (4) it was studied in detail by GE, and (5) is important in the zone defense concept.

The major goal in the evaluation of TRIP is to identify TRIP processing activities that are best distributed. This identification is performed at three levels. First, those activities which can easily be distributed with no logical reorganization are identified. Second, those activities which can be performed utilizing a physical aspect of the topology are identified. Third, those activities whose distribution would require a major redesign with major performance improvement are identified.

The proposed new structure of the TRIP subroutines is described in Figure B-1. It includes a central HTRIP1 and a distributed HTRIP2. Using the first criteria, the tasks have the representative subroutines as shown in Figure B-1. This subroutine assignment does not reduce the total work performed by TRIP. However, this organization has created more parallel activity and allows radar usage to increase.

A new clerical C TRIP task has been created which contains trade and parts of TMISH utilizing the broadcast capabilities of the distributed architecture to isolate the overhead of accounting and enhacing of fault isolation (see Figure B-1).

The final new structure proposed for TRIP and described in Figure B-2 indicates the creation of two new functions which can be performed concurrently. TEMP01 and TEMP02 have been conceived to reduce the total processing elapse time of TRIP under large loading at the expense of increasing the processing elapse time under small loading. Further, the number of MIPS for TRIP has increased.

TEMP0 appears to have processing times which are proportional to the number of requests pending. This apparent dependency has severe problems as the radar becomes busy. The distributed goal is to collect radar requests in a number of separate regions so the processing time will not be so variable. Specifically, the request arrays in local regions will be kept small. TEMP0 maintains large sorted arrays. The TEMP0 processing can be equated to a sorting function.

It has long been established that the total time to sort items is the square of the number of items being sorted. If a two stage sorting function is established where the secondary sorting is done in aggregates of sorted items, then total sorting time is reduced. The equations for a single stage sort, a two stage sort in a central processor, and a two stage sort in a distributed processing system have been tabulated and included herein as Table B-3.

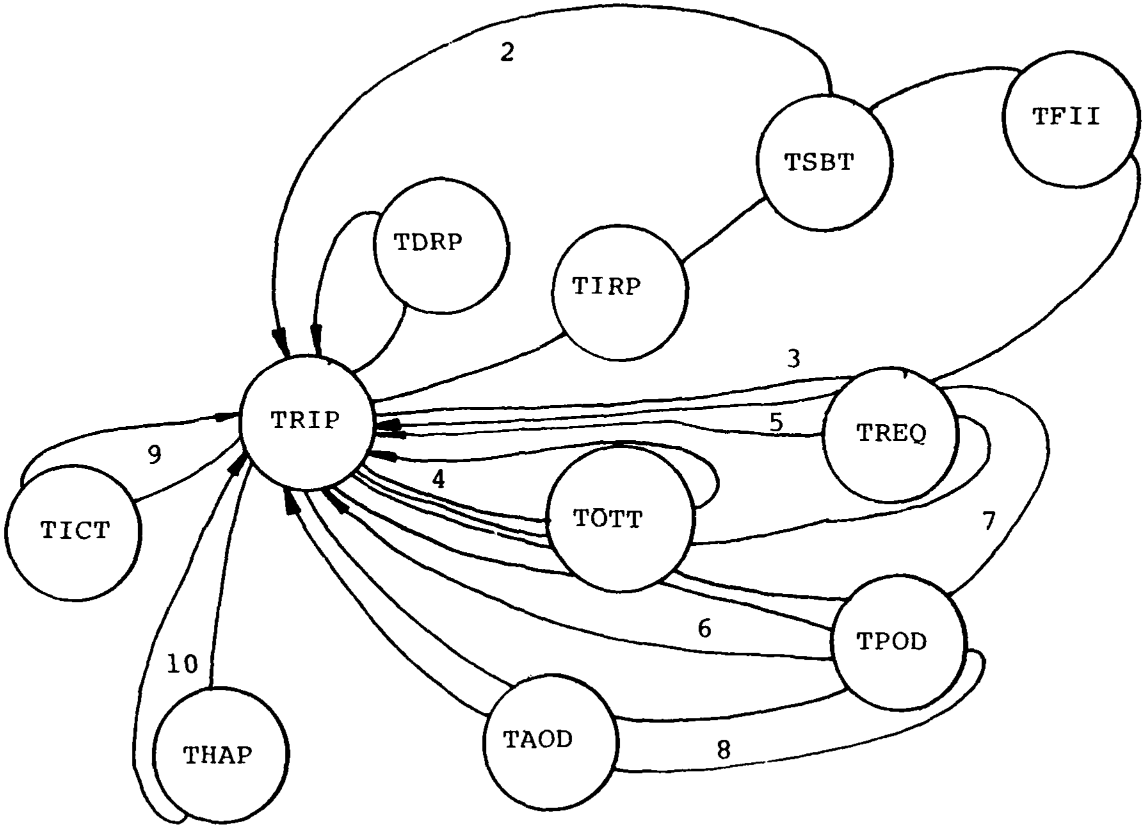
The microscopic effects of sorting on processing requirements have been identified. Further, the microscopic description of multistage sorting has been developed. The macroscopic conceptualization of the multistage sort for the BMD Underlay System is the creation of a zone defense for various hierarchies of processors (see Figure B-4). With this construct, presorting of radar requests is possible on a zone basis. If a uniform distribution of targets were present at each zone Table B-5 gives the theoretical reduction in processing time which could be achieved.

The zone defense concept creates and distributes functional autonomy within the system. This further implies that systems expansion can be easily achieved along with providing an excellent means for fault isolation.

At the next higher level of design abstraction it is desirable to restructure threads to reduce the number of time critical threads. If a lower level of the hierarchical system continued to maintain tracking, object discrimination could proceed at a slower rate. It does not appear unreasonable for object discrimination to take as long as 1 second with a maximum of 8 radar requests processed during that period.

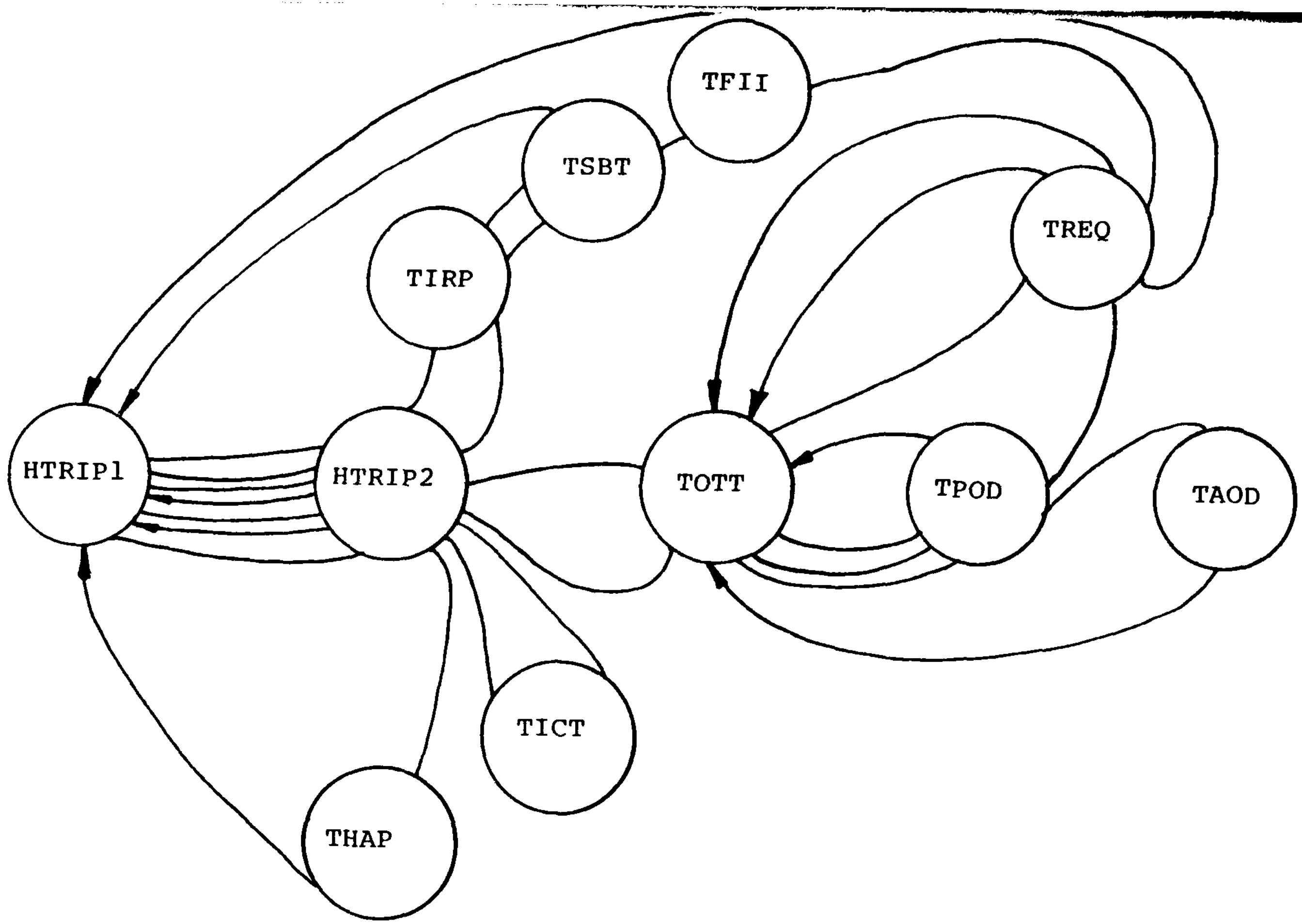
Figure B-6 describes the new thread organizations which have been conceived for the radar return threads. Table B-7 clearly shows that this restructure has increased the total MIPS requirement for performing equivalent functions. However, the number of time critical threads and the processing of time critical threads have been reduced.

The methods and concepts proposed are designed to optimize operation in a distributed processing environment and to optimize hierarchical topologies. The zone defense of the BMD Underlay System identifies new methods of hierarchical control which are desirable from a battle management point of view.



OLD THREAD STRUCTURE

B-1



NEW THREAD STRUCTURE

B-2

Single Stage	N^2X
Two Stage Central	$2S^2X+S\ (\frac{N}{S})^2X$
Two Stage Distributed	$2S^2X+(\frac{N}{S})^2X$

$$\frac{N}{S} = M$$

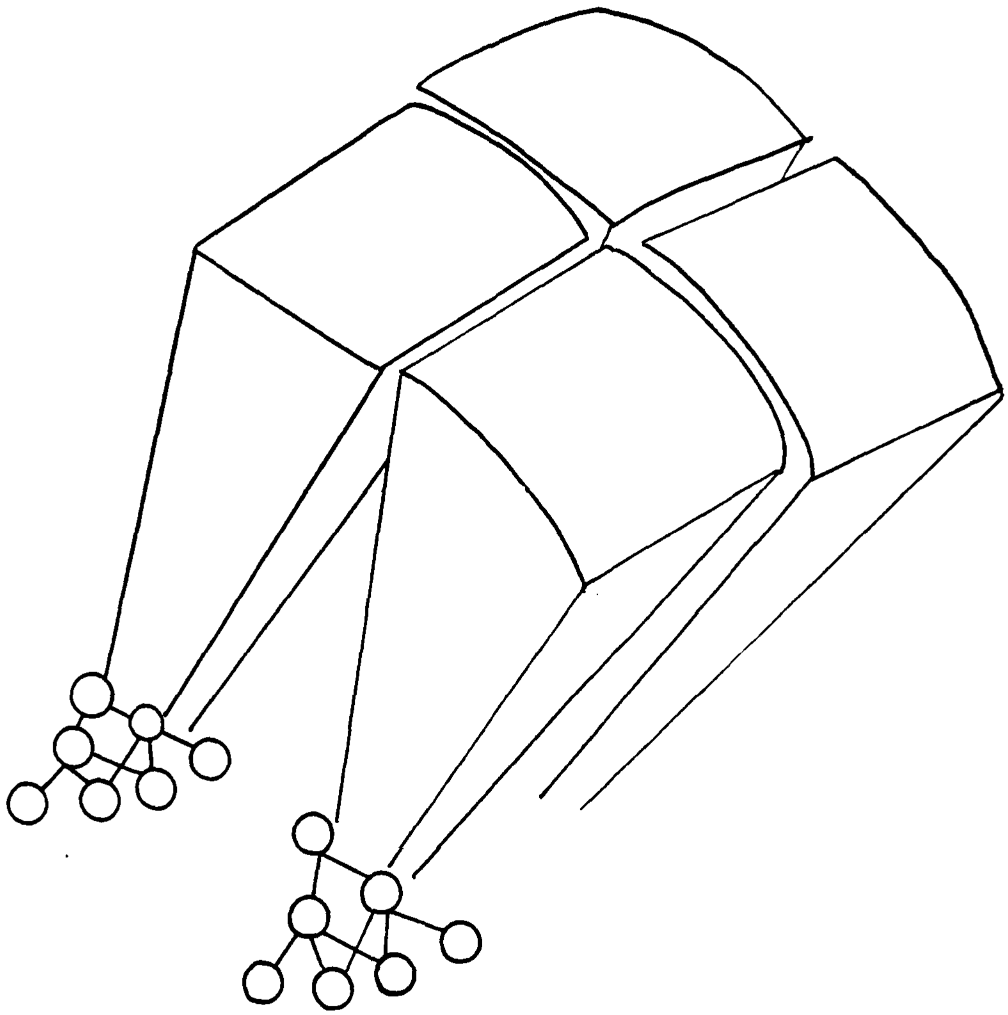
N = number of items

M = number of items in subgroup

S = number of subgroups

WORST CASE SORTING STRATEGY

B-3



ZONE DEFENSE

B-4

2ØØØ	Pulses
5ØØ	Pulses per Zone
4	Zones

$4.0 \cdot 10^6 X$	Single Stage	$4.10^6 X$
$1.0 \cdot 10^6 X$	Two Stage Central	$32X+1 \cdot 10^6 X$
$2.5 \cdot 10^5 X$	Two Stage Distributed	$32X+25 \cdot 10^4 X$

PROCESSING TIMES FOR WORST CASE EVENLY DISTRIBUTED ZONES

B-5

SUBSET CRITICAL PORT-TO-PORT THREAD DEFINITION (B-6)

1. From Radar Subsystem to Radar Subsystem[4]:

Verify Return	TRIP-TDRP-TRIP[1]
Track Initiate	TRIP-TIRP-TSBT-TRIP[1]
Last TI	TRIP-TIRP-TSBT-TFII-TREQ-TRIP[1]
Normal and Post-Commit Track/ Maintenance Track	TRIP-TOTT-TRIP[1]
Drop Track/Track Rate Change	TRIP-TOTT-TREQ-TRIP[1]
Passive Discrimination/Track to Discrimination Turnaround	TRIP-TOTT-TPOD-TRIP[1]
Active Discrimination Request/ Drop Track	TRIP-TOTT-TPOD-TREQ-TRIP[2]
Active Discrimination	TRIP-TAOD-TPOD-TRIP[1]
Interceptor Guidance and Track	TRIP-TICT-TRIP[2]
Pulse Replacement/Reschedule	TRIP-TRIP[1]
Object Reacquisition	TRIP-THAP-TRIP[3]

Footnotes:

(1) ESW Spec Part II, Volume I (1 Jul 78)

(2) ESW Spec Part II, Volume I (5 Jun 74)

(3) ESW PDR (28-30 Nov 73)

(4) Only these threads are applicable to the unit level. The rest are applicable to the module level.

SUBSET NEW CRITICAL PORT-TO-PORT THREADS B-7

DROP TRACK	HTRIP1-HTRIP2-HTOTT-HTREQ-HTRIP2-HTRIP1
LAST TI	HTRIP1-HTRIP2-HTSBT-HTFII-HTIRP-HTRIP2-HTRIP1
TRACK INITIATE	HTRIP1-HTRIP2-HTSBT-HTIRP-HTRIP2-HTRIP1
VERIFY RETURN	HTRIP1-HTRIP2-HTDRP-HTRIP2-HTRIP1
ACTIVE DISCRIMINATION (CRITICAL THREAD)	HTRIP1-HTRIP2-HTAOD-HTRIP2-HTRIP1
TRACK	HTRIP1-HTRIP2-HTOTT-HTRIP2-HTRIP1
INTERCEPTOR GUIDANCE	HTRIP1-HTRIP2-HTICT1-HTRIP2-HTRIP1
ACTIVE DISCRIMINATION /DROP TRACK	HTRIP1-HTRIP2-HTOTT-HTRE Q-HTRIP2-HTRIP1

CRITICAL TAP TASKS (TASKS IN CRITICAL THREADS) B-8

	MAXIMUM EXECUTION FREQUENCY	MESSAGES PER ENABLEMENT	AVERAGE MESSAGE SIZE	AVERAGE NUMBER OF INSTRUCTION
TRIP	1500	2	60	6145
TDRP	@% 25	2	60	13631
TIRP	180	2	200	9073
TSBT	180	2	92/2	7827
TREQ	75	2	55/2	3566
TPOD	215	5	120	34705
TAOD	27	5	100	36135
TFII	43	3	600	27316
TOTT	801	3	370	25875
TICT	156	3	360	221,05
THAP	10	2	208	2506

TRIP WORKSHEET - B9

	AVERAGE	MAX	
TTRIP	90	90	
TARSN	2000	3500	
TSORT	250	300	
TMOVE	300	360	
TWONG	45	90	
TMISH	600	1500	
TEMPO	1200	2400	
TSWIG	780	1950	
TRADE	135	135	
TACIT	240	400	
TUPK1	50	50	
TWIST	75	120	
TITAN	200	200	
THREAD	180	240	
TRIP	6145	11335	INSTRUCTIONS PER ENABLEMENT

APPENDIX C - Hierarchical Network Characteristics

To discuss hierarchical network characteristics the BMD Underlay System will be viewed as a hierarchical network made of of levels. The BMD Underlay System is characterized as being made up of two types of hierarchical network architectures: (1) a hierarchical resource system (HR system), and (2) a hierarchical control system (HC system).

The BMD Underlay System can be described as a HR System due to the shared, structured access to the radar and interceptor farm. As a result, the BMD Underlay System possesses three characteristics which are associated with HR systems.

First, there is the potential for a bottleneck at the superior node of the network. This limitation expresses itself as a bottleneck in the radar unit at the data processing level and at the hardware level.

Second, there is the catastrophic effect of a failure of the principle resource. Specifically, failure of the radar or interceptor farm is lethal to the system.

Third, the HR architecture is unable to expand without new technology. Specifically, expansion of the radar units maximum number of pulses per second requires hardware innovation.

In summary, the characteristics of the BMD Underlay System associated with the HR architecture exemplifies some of its principal weaknesses. The use of a single radar unit and a single interceptor farm create severe limitations to expansion of the system.

HC is described by subordinates processing requests and passing their conclusions up to their superiors for further commital of resources. The module commander, skymap, and nuclear enablement exemplify HC. Four characteristics can be associated with HC systems.

First, there is superior fault isolation and recovery. Specifically, the superior nodes in the system are capable of redistributing work when inferior nodes fail. Additionally, with superior node failure subordinates continue to perform their function so repair or removal of the defective superior node would allow work to continue as normal. Consequently, if subordinates are assigned or report to alternate superiors, failure of superior nodes are correctable.

Second, the ability to understand an HC system is enhanced due to information hiding and an ability to support levels of system comprehension. Specifically, the hierarchical (top down) design removes consideration of unimportant implementation details, hiding information. Further, HC systems are distributed control systems with various abstractions created to allow distributed control. These abstractions provide a method for the description of the system with different levels of comprehension.

Third, HC systems are designed for a range of expansion. Expansion can continue until there is a systems requirement to add additional levels in the hierarchical network because the effective usage of an additional level within the hierarchy requires redefinition of all tasks within the system

Fourth, HC systems respond slowly and unpredictably to new uncharacterized threats. Specifically, lower levels of the hierarchy are required to pass raw data to their superiors because they are incapable of analyzing the threat. Consequently, bottlenecks can develop which impede the normal operation of the system.

In summation, there are three positive characteristics associated with hierarchical control and one negative characteristic. The construction of a network based on HC has more positive attributes than a network constructed with a HR architecture.

APPENDIX D - Network Executive

A distributed layered executive has been conceived for the BMD Underlay DPS. The DPS executive is designed to maximize the software portability of the system, optimize the characteristics of a hierarchical network, support expandability, support a low MTTR for easy maintenance, and maximize isolation of the DPS executive and the DPS algorithms.

Portability is not an academic issue. With the growth in available microprocessors and their capabilities, the final system may be targeted for entirely different microprocessor hardware. Two methods are used to enhance portability. The executive is constructed in layers. The local operating system which is a layer of the executive is divided into a microprocessor specific kernel and a general local executive which contains the I/O drivers.

Optimizing the characteristics of a hierarchical network consist of utilizing the networks superior fault tolerance characteristics, while minimizing its potential for a bottleneck at the superior node level. Distributed control and asymmetric communications overhead are used to support both goals. Distributed control distributes control overhead throughout the system. Asymmetric communications overhead places most of the overhead for communications at the senders and receivers node. Both techniques maximize isolation of processing centers within the network and allow the use of data flow graphs for the computation of throughput requirements.

System expandability is a principal argument for the use of the distributed network. Diminishing returns on throughput expansion with the addition of processors has been recognized as the result of communications and control overhead. The use of distributed control and adhering to certain principals of hierarchical problem decomposition allow a hierarchical network to continue to be expandable.

Support of the maintenance goal is accomplished by providing layered maintenance software abstracted to the layers of the executive. At the lowest level, individual processors are capable of fault isolation with only ROM and CPU functioning. At the highest level, the executive is capable of providing runtime reconfiguration in the event of a hardware error. The development of the maintenance software in layers supports the development of levels of comprehension with the ability to support intuitive repair.

The goal in maximizing the isolation of the executive with the specific DPS algorithms is desirable due to the evolution that the system will experience and for security purposes. Compartmentilization of information is fundamental to structured design and will reduce software development risks.

The creation of a model for the executive and communications processing requirements is accomplished by identification of shared resources and internal data bases within the layered structure of the executive. The identification of shared resources identifies the queues which the model will contain. Identification of the data bases characterizes the processing associated with these queues.

Analysis of the layers of the executive outlined in Figure D-1 shows three basic queues. First, there is the scheduling queue which represents the queue associated with task invocation. Second, there is the I/O resource timeout queue which represents the queue associated with timeout requests. Third, the communications queue describes the queue associated with waiting for access to the communications bus for transmission of data.

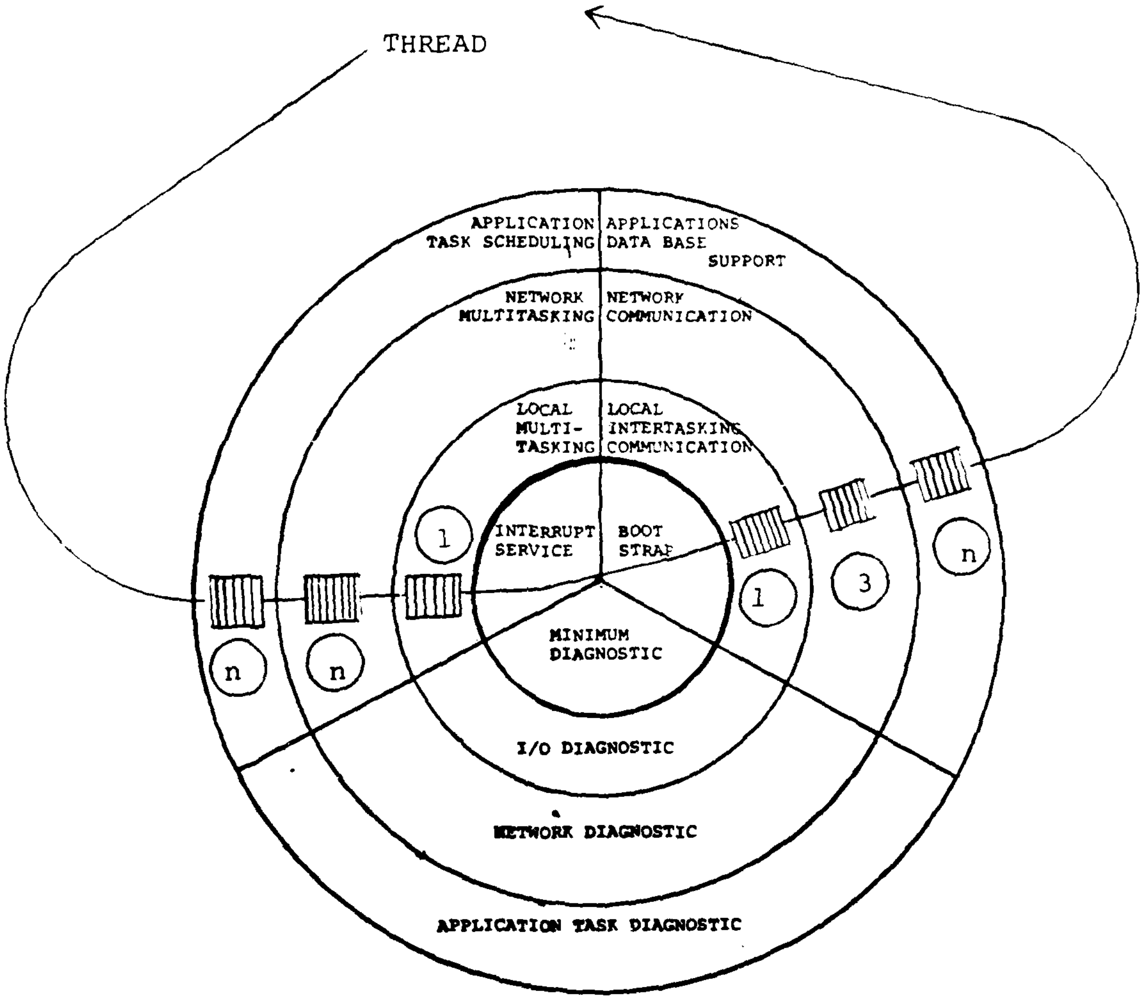
These three queues exhibit different service functions and number of servers. The scheduling queue has as many servers as processors which can perform that task. The I/O resource queue is a single server queue. Finally, the communications queue has the number of servers equal to the number of alternate communications paths between communicating nodes.

The service functions of all three queues is proportional to the average size of the queue. Specifically, the task, the I/O resource, and the communications queues require sorting of requests or replies. This sorting function is proportional to the square of the number of items being sorted. Service time per request can be expressed as:

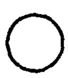
```
TOTAL$INSTRUCTIONS$PER$REQUEST=FIXED$INSTRUCTIONS +
                                VARIABLE$INSTRUCTIONS
WHERE:

VARIABLE$INSTRUCTIONS = SORT$INSTRUCTIONS*(AVERAGE$QUEUE$SIZE)**2
```

EXECUTIVE HIERARCHY




QUEUE


NUMBER OF SERVERS

n=NUMBER OF PROCESSORS PER LEVEL OF HIERARCHY

Figure D-1

APPENDIX E - Hierarchical Network Based on GE FFP

The GE FFP and the 2-AU-80 are the two processors considered for the development of a hierarchical network within the 4 MIPS throughput range. Both processors are microprogrammed and both processors have awkward instruction sets when compared to more general architectures. The development capabilities of both processors are adequate for software development. The GE FFP is capable of emulation on the PDP-11/70 or VAX machines. The 2-AU-80 has a Pascal compiler developed by MDAC. The GE FFP has available a Fortran compiler and a multitasking executive.

Both processors contained the same type of architecture depicted in Figure E-1. Both processors contain global memory and both have a high speed parallel communication bus with a bandwidth of 4 megawords. However, the GE FFP provides Hamming correction and error checking on its global memory.

The GE FFP was chosen based on its error checking capabilities over the 2-AU-80 as a candidate for the 4 MIPS hierarchical network. Specifically, the global memory had error correction. Further, the extra ALU options available in the GE FFP (though they could not be used to enhance throughput) could support runtime computation verification with little extra cost and overhead.

The risk functions and the projected critical thread port-to-port times are depicted in Figures E-2 and E-3. The remainder of this section contains the work sheets and the computer output used to determine the critical threads.

RISK FUNCTIONS

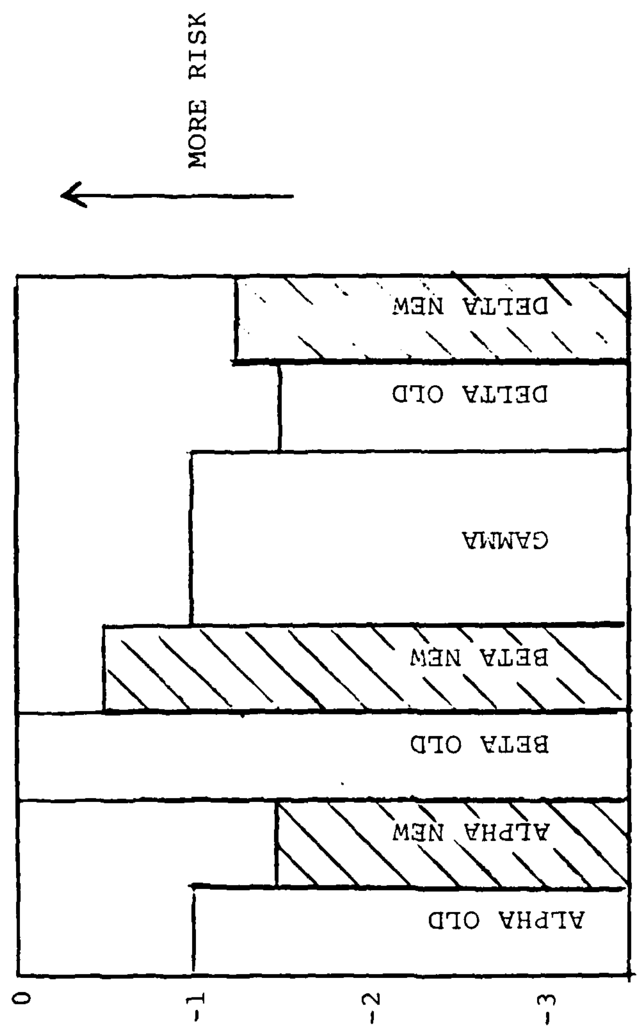


TABLE E-2

FIGURE E-3A

SUBSET CRITICAL THREADS FOR 4.00 MIPS

TASK SCHEDULING QUEUES (A,B,C,D,) = .00 .81 2.02 .00
COMMUNICATIONS QUEUES (0,1,2,3) = .07 .10 .75 .00

HTRIP1-HTRIP2-HTDRP-HTRIP2-HTRIP1 33.14 MILS

HTRIP1-HTRIP2-HTSBT-HTIRP-HTRIP2-HTRIP1 33.89 MILS

HTRIP1-HTRIP2-HTSBT-HTFII-HTIRP-HTRIP2-HTRIP1 41.64 MILS

HTRIP1-HTRIP2-HTOTT-HTRIP2-HTRIP1 36.14 MILS

HTRIP1=HTRIP2=HTOTT-HTREQ-HTRIP2-HTRIP1 39.26 MILS

HTRIP1-HTRIP2-HTAOD-HTRIP2-HTRIP1 66.64 MILS

HTRIP1-HTRIP2-HTICT1=HTRIP2-HTRIP1 32.99 MILS

SUBSET CRITICAL THREADS FOR 4.00 MIPS (FIGURE E-3-B)

TASK SCHEDULING QUEUES (A,B,C,D) =	.00	.81	2.02	.00
COMMUNICATIONS QUEUES (0,1,2,3) =	.07	.10	.75	.00

TRIP-TDRP-TRIP 33.14 MILS

TRIP-TIRP-TSBT-TRIP 3.89 MILS

TRIP-TIRP-TSBT-TFII-TREQ-TRIP 41.64 MILS

TRIP-TOTT-TRIP 36.14 MILS

TRIP-TOTT-TREQ-TRIP 36.89 MILS

TRIP-TOTT-TPOD-TRIP 44.64 MILS

TRIP-TOTT-TPOD-TREQ-TRIP 47.51 MILS

TRIP-TAOD-TPOD-TRIP 75.39 MILS

TRIP-TICT-TRIP 35.89 MILS

TRIP-TRIP 29.08 MILS

APPENDIX F - Hierarchical Network Based on APX432

The principle microprocessors considered for the 1 MIPS topology are the 68000, the microflame, the Z8000, and the APX432. The major goal in the consideration of a 1 MIPS topology was to take advantage of commercially available software, development systems, and modular hardware.

The Z8000 and the microflame were eliminated from consideration for use in the 1 MIPS topology because they multiplexed their address and data busses which would limit throughput expansion. The IAPX432 was chosen over the 68000 because of available software and its speed.

The IAPX432 as depicted in Figure F-1 with 8 megahertz clock is capable of 1 MIPS throughput. Further, it contains a ROM'ed operating system. Figure F-2 shows various critical threads with projected completion times. The remainder of this section contains work sheets used for the computation of these times and the computer printout from the queue model.

iAPX 432 SINGLE BUS MULTIPROCESSOR SYSTEM

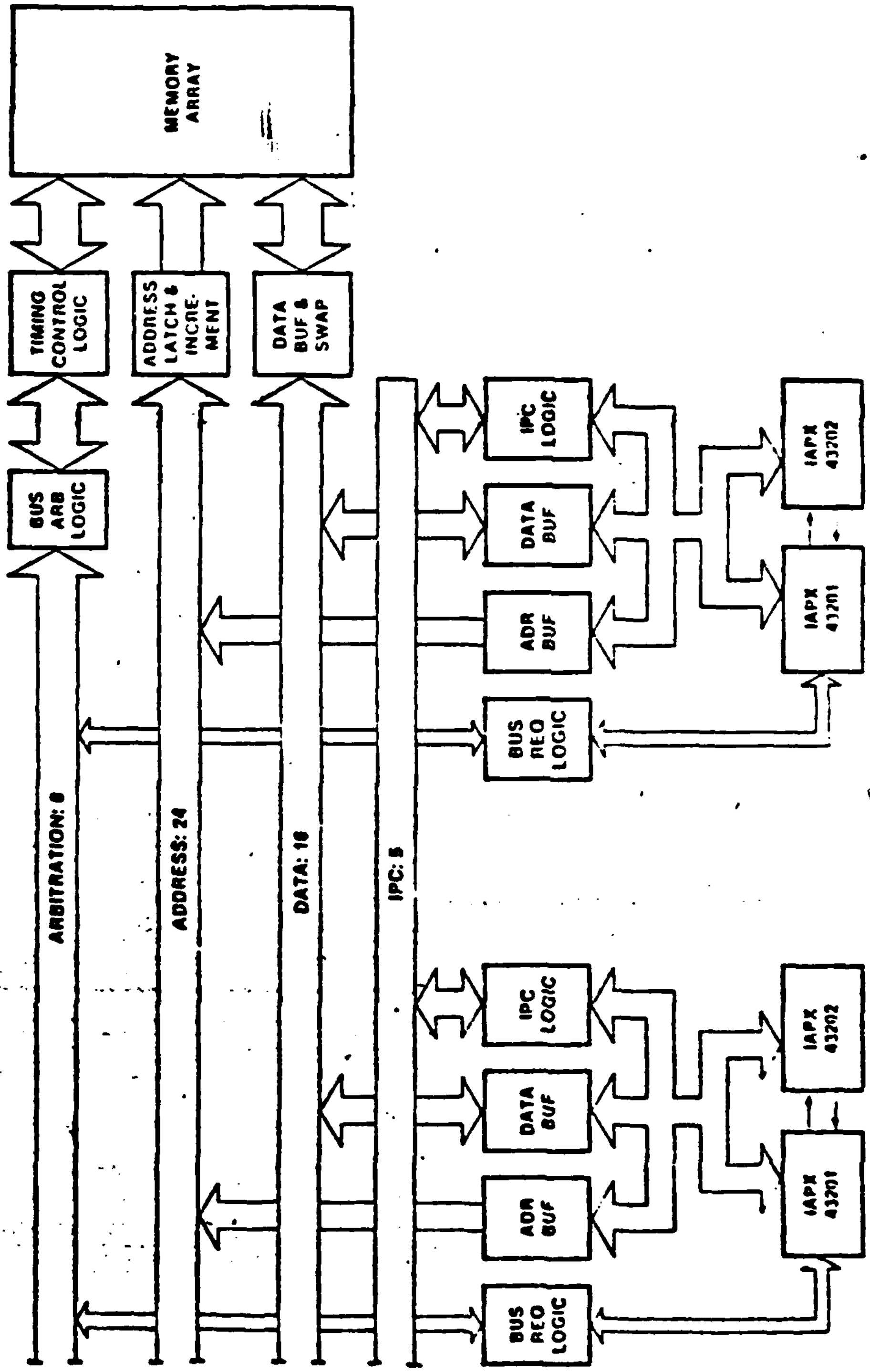


FIGURE F-1

SUBSET CRITICAL THREADS FOR 4.00 MIPS (FIGURE F-2-A)

TASK SCHEDULING QUEUES (A,B,C,D) =	.00	.81	2.02	.00
COMMUNICATIONS QUEUES (0,1,2,3) =	.07	.10	.75	.00

TRIP-TDRP-TRIP 33.14 MILS

TRIP-TIRP-TSBT-TRIP 33.89 MILS

TRIP-TIRP-TSBT-TFII-TREQ-TRIP 41.64 MILS

TRIP-TOTT-TRIP 36.14 MILS

TRIP-TOTT-TREQ-TRIP 36.89 MILS

TRIP-TOTT-TPOD-TRIP 44.64 MILS

TRIP-TOTT-TPOD-TREQ-TRIP 47.51 MILS

TRIP-TAOD-TPOD-TRIP 75.39 MILS

TRIP-TICT-TRIP 35.89 MILS

TRIP-TRIP 29.08 MILS

SUBSET CRITICAL THREADS FOR 4.00 MILS (FIGURE F-2-B)

TASK SCHEDULING QUEUES (A,B,C,D) =	.00	.81	2.02	.00
COMMUNICATIONS QUEUES (0,1,2,3) =	.07	.10	.75	.00

HTRIP1-HTRIP2-HTDRP2-HTRIP1 33.14 MILS

HTRIP1-HTRIP2-HTSBT-HTIRP-HTRIP2-HTRIP1 33.89 MILS

HTRIP1-HTRIP2-HTSBT-HTFII-HTIRP-HTRIP2-HTRIP1 41.64 MILS

HTRIP1-HTRIP2-HTOTT-HTRIP2-HTRIP1 39.26 MILS

HTRIP1-HTRIP2-HTOTT-HTREQ-HTRIP2-HTRIP1 39.26 MILS

HTRIP1-HTRIP2-HTAOD-HTRIP2-HTRIP1 66.64 MILS

HTRIP1-HTRIP2-HTICT1-HTRIP2-HTRIP1 32.99 MILS

IAPX432

VLSI HARDWARE FOR MAINFRAME FUNCTION

FUNCTION	MICROCODE ALLOCATION	
	BITS	PERCENTAGE
BASIC INSTRUCTION SET	3680	6%
FLOATING POINT ARITHMETIC	11680	18%
RUN-TIME ENVIRONMENT	6400	10%
VIRTUAL ADDRESSING	4800	7%
FAULT HANDLING	2640	4%
SILICON OS	26400	40%
MULTIPROCESSOR CONTROL	8640	13%
DEBUG SERVICES	1280	2%
	64K BITS	100%

FIGURE F-3

APPENDIX G - Stability

DPS network stability analysis is in its infant stages. Present signals and systems control theories are applicable to process control applications, however, existing theories have not been abstracted to a level where they can be applied to DPS functions.

To put stability in proper perspective three areas of analyses are identified: (1) the application algorithms of a system, (2) system saturation, and (3) system control.

It is not within the scope of this study to review the application algorithms. This area has been the subject of exhaustive analyses and numerous techniques exist for determining the stability of algorithms.

System saturation is a new area which has evolved with the advent of distributed processing. In central processors, system saturation is proportional to the total processing capability of the system. With the advent of distributed processing and dedicated subsystems, subsystem saturation can occur long before total system saturation.

System control stability is determined by analysis of the control algorithms and by the analysis of control and communications saturation.

Determination of the stability of the candidate designs is limited to verifying that there is no throughput saturation, and identifying those control structures which appear stable.

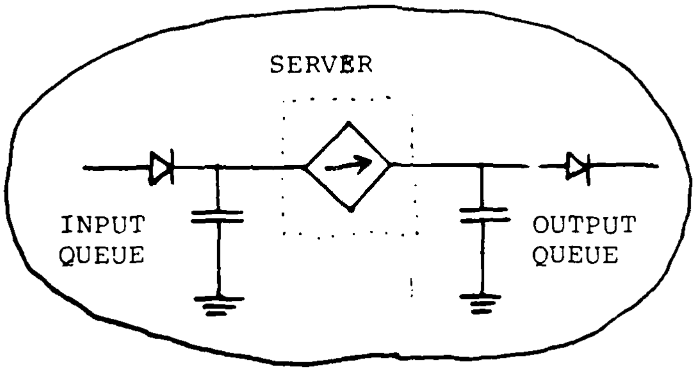
In an effort to identify those control structures which are stable, circuit theory analogues of communications lines were developed. Also, the primary requirements for distributed control were identified.

The circuit theory model provided the mechanism to isolate communications throughput performance (see Figure G-1). It represents a communications system where messages are stored in a common incoming queue of finite size (see Figure G-2). This construct provides a method of verifying stability under various loads using circuit theory modeling and simulation techniques.

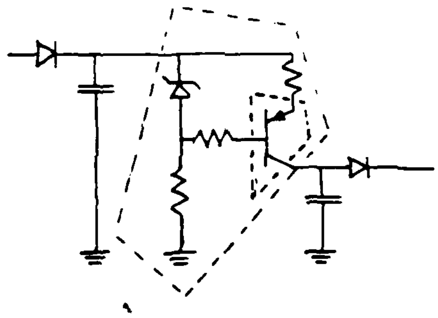
An additional distributed control requirement for stability is that the system control response be faster than the system dynamics. The delta function detailed in Section A addresses this facet of control stability.

QUEUE / CIRCUIT MODEL

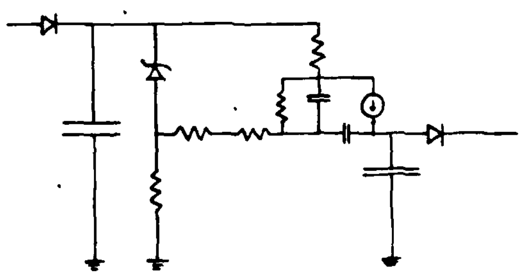
FIGURE G-1



NODE PROCESSOR IN QUEUE MODEL



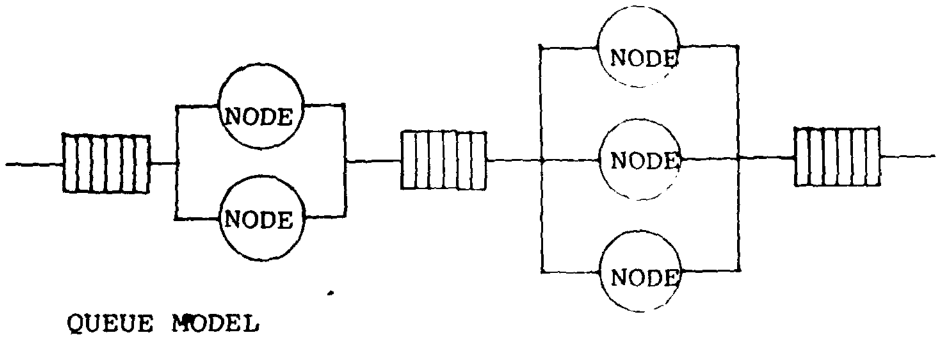
TRANSISTOR MODEL OF SERVER
(PROCESSOR IN QUEUE MODEL)
(CURRENT SOURCE OR SINK)



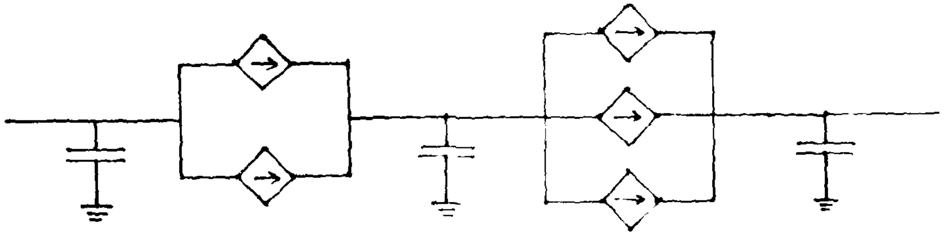
HYBRID PI MODEL OF TRANSISTOR
(MODELS DELAYS)

QUEUE/CIRCUIT MODEL

FIGURE G-2

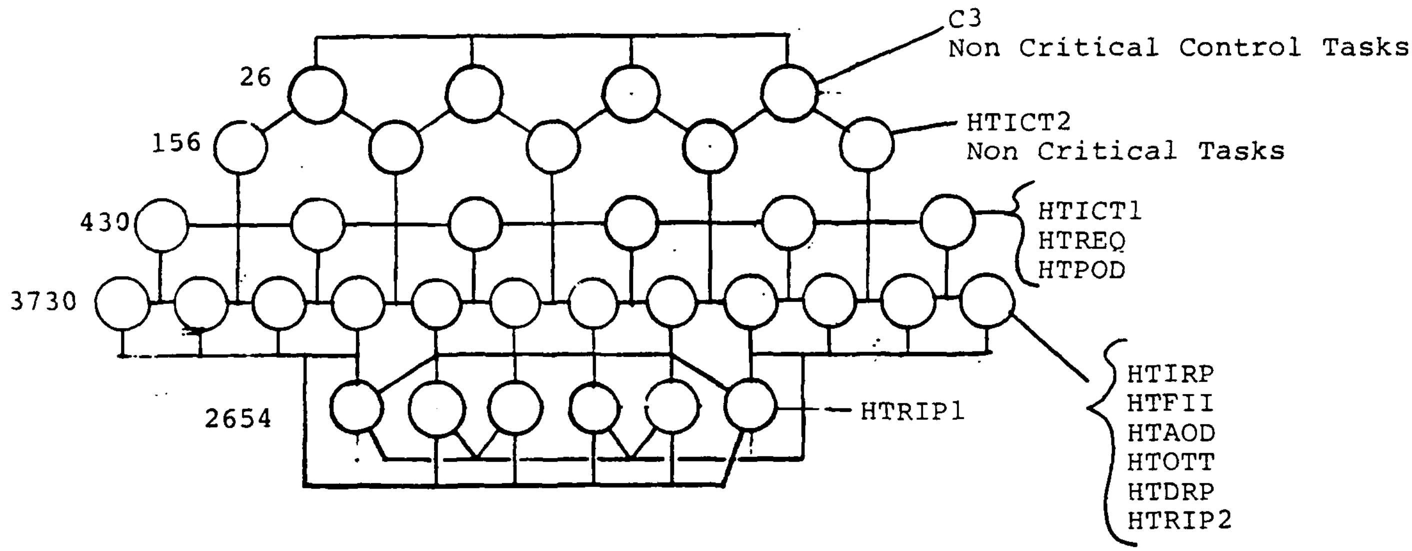


QUEUE MODEL

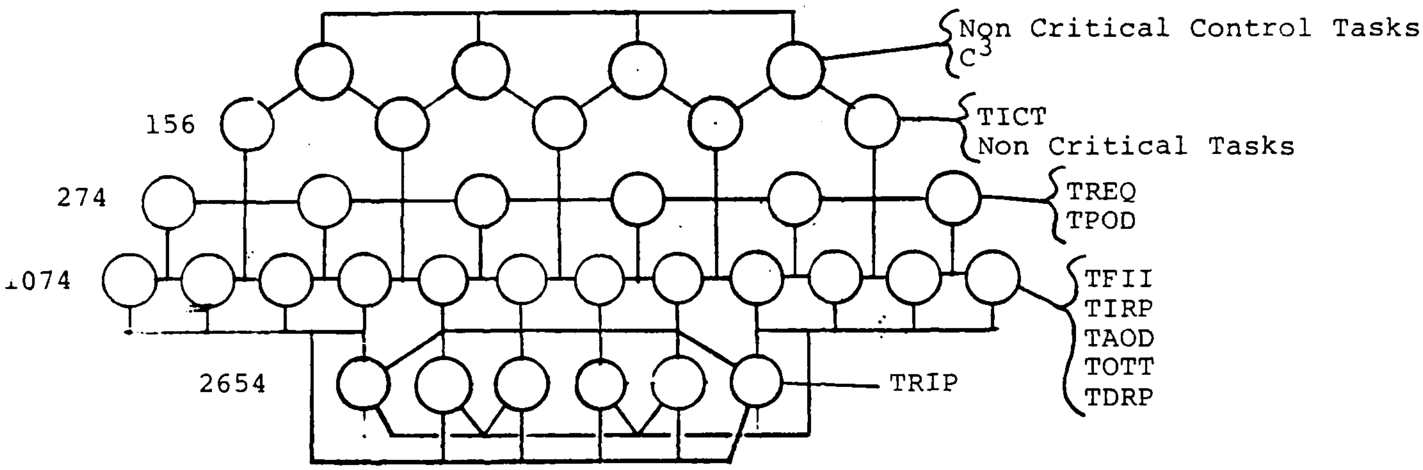


EQUIVALENT CIRCUIT MODEL

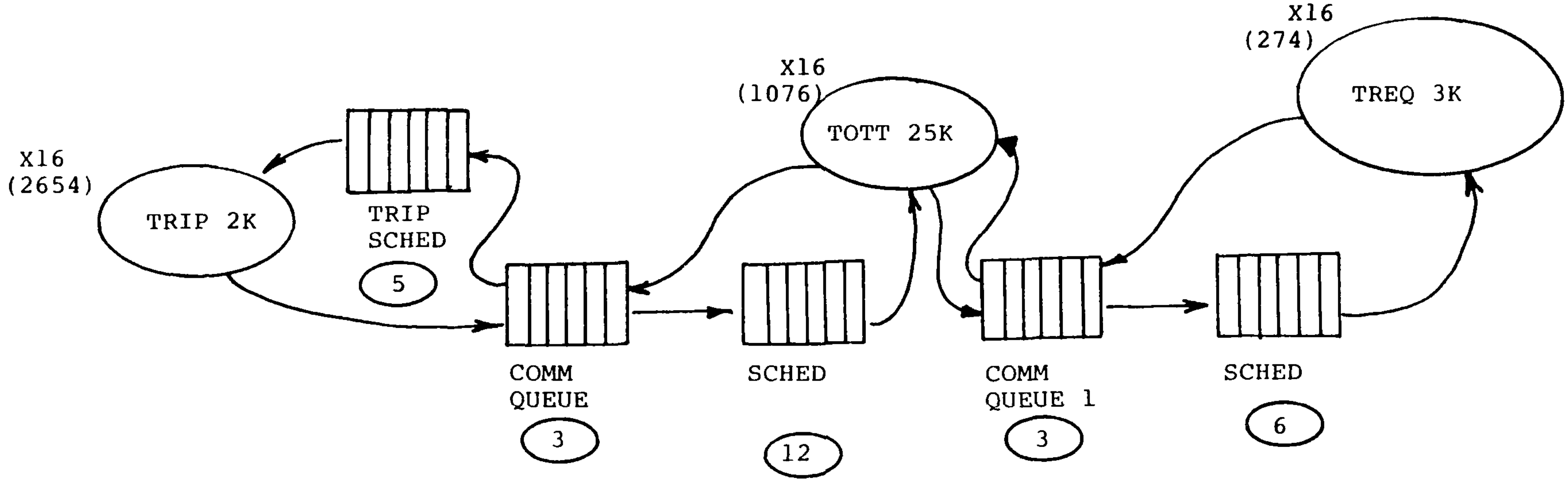
APPENDIX H



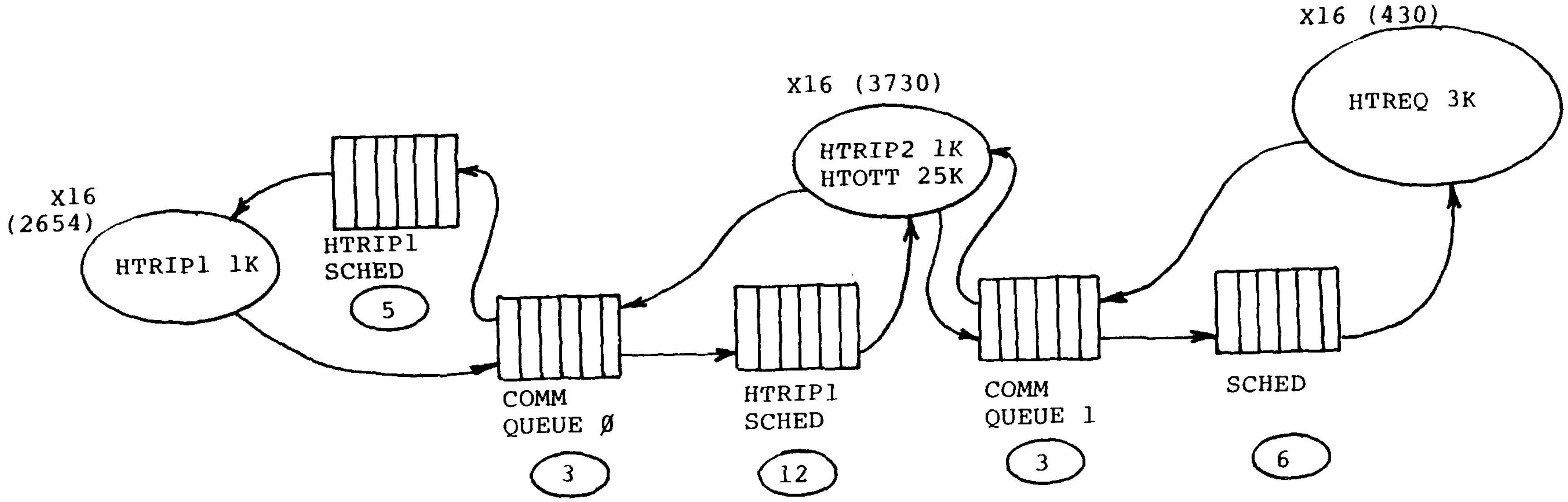
NEW THREAD
TASK ASSIGNMENT



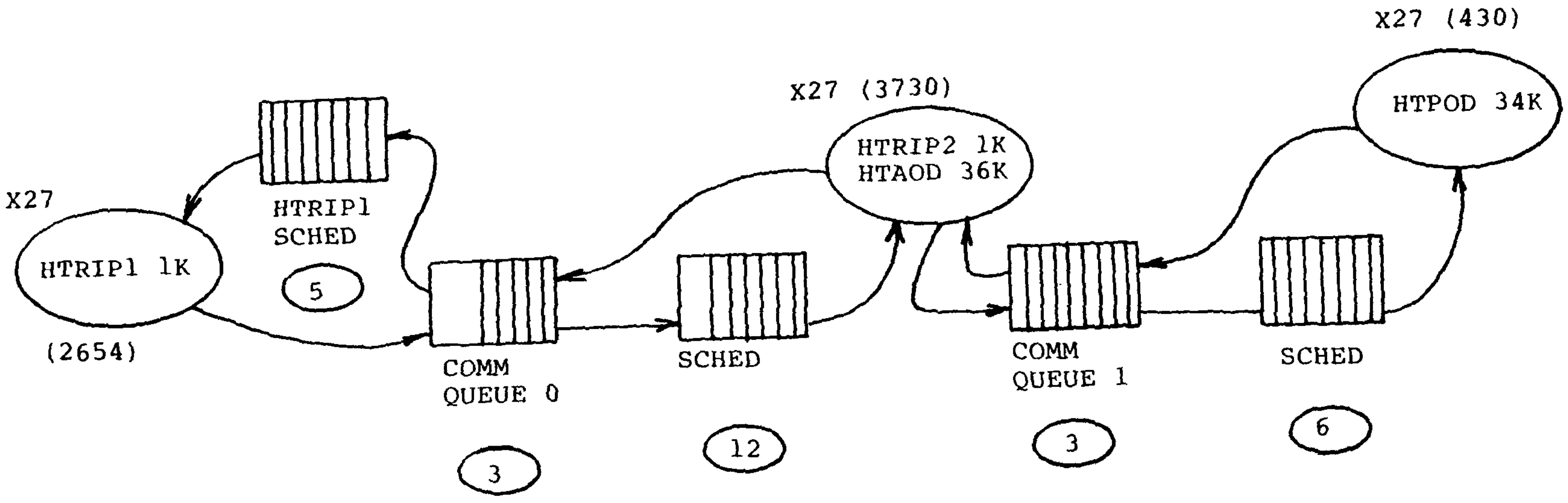
OLD THREAD TASK ASSIGNMENT



DROP TRACK
THREAD TRIP-TOTT-TREQ-TRIP



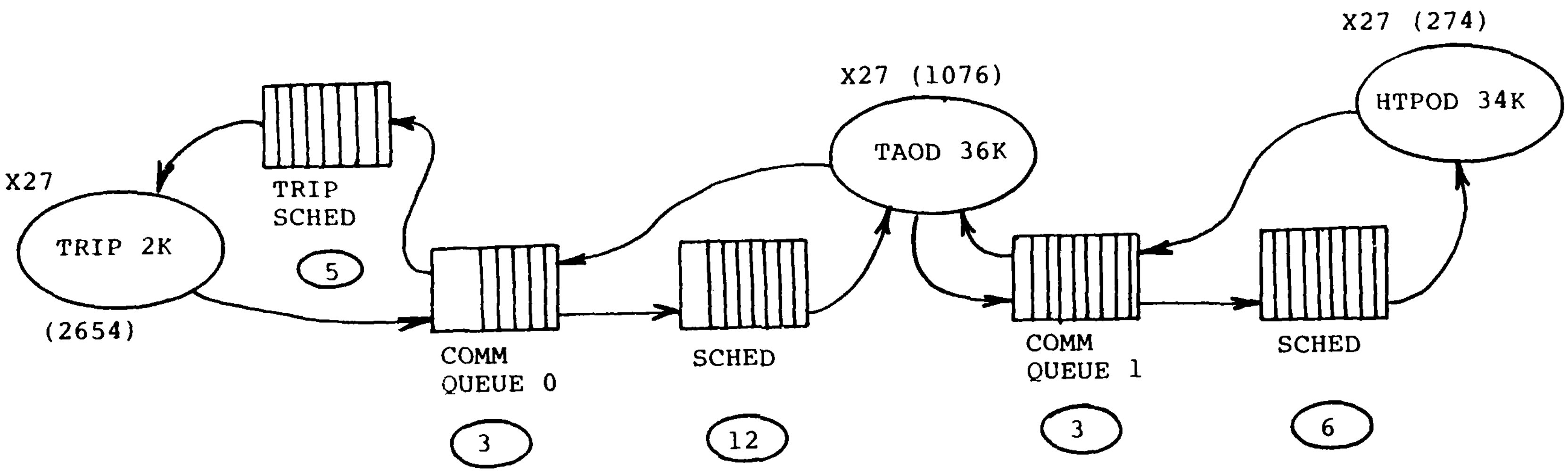
DROP TRACK THREAD: HTRIP1-HTRIP2-HTOTT-HTREQ-HTRIP2-HTRIP1



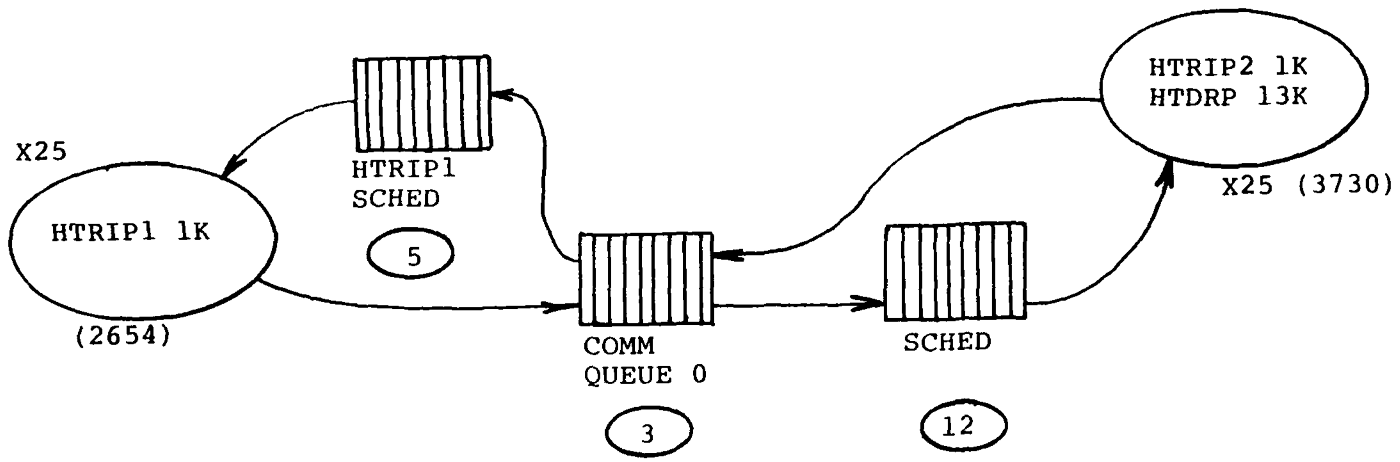
ACTIVE DISCRIMINATION (NOT CRITICAL!)

THREAD - HTRIP1-HTRIP2-HTAOD-HTPOD-HTRIP2-HTRIP1

NEW CRITICAL - HTRIP1-HTRIP2-HTAOD-HTRIP2-HTRIP1

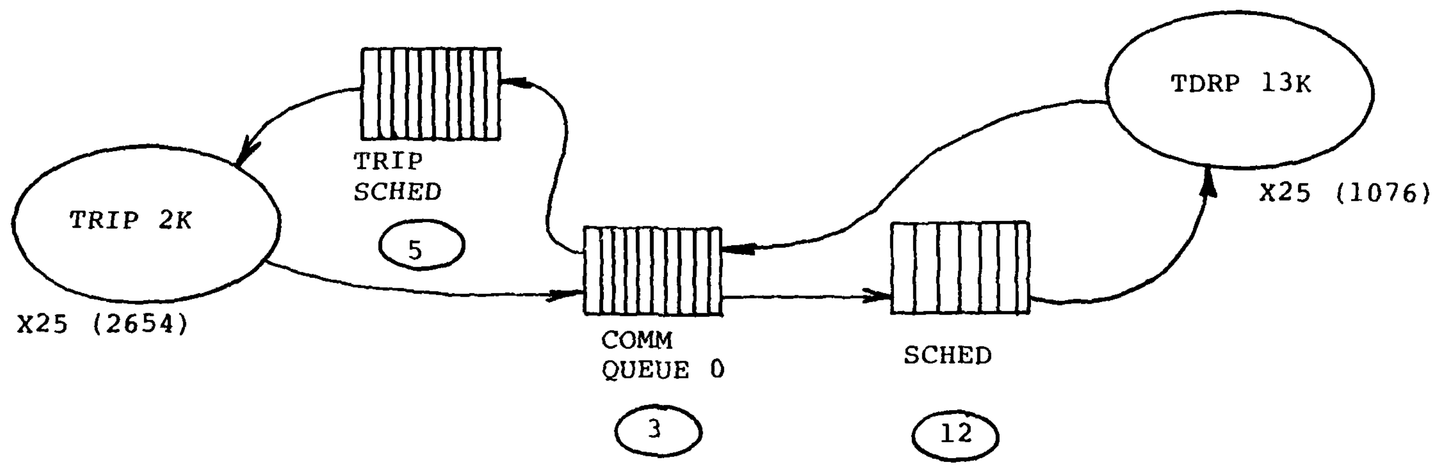


ACTIVE DISCRIMINTATION
THREAD: TRIP-TAOD-TPOD-TRIP



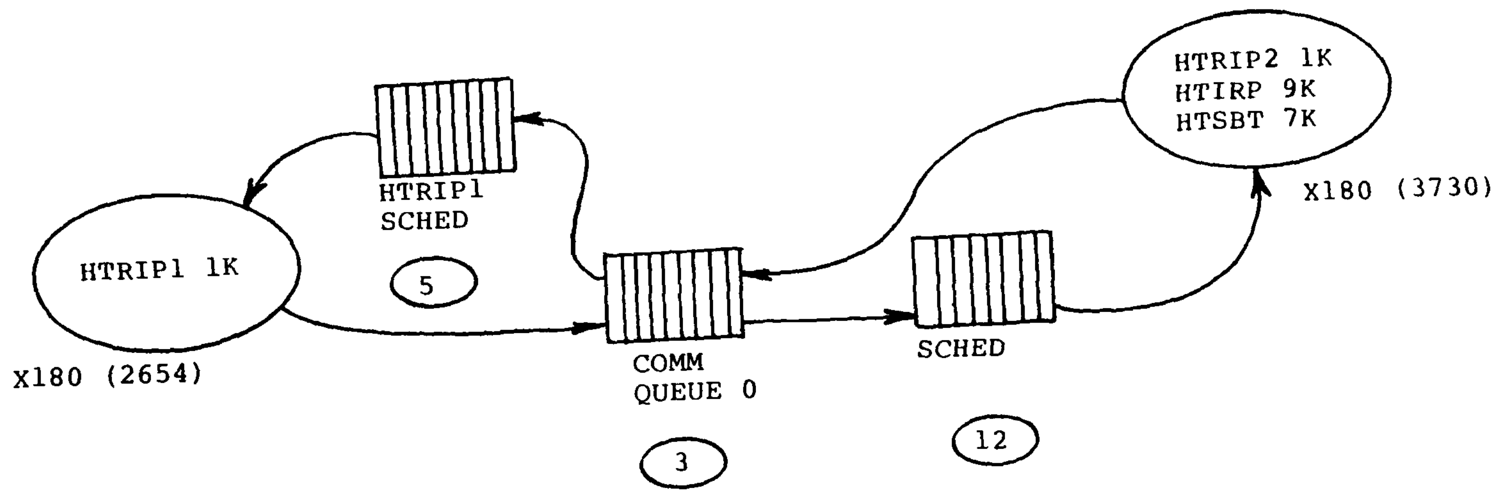
VERIFY RETURN

THREAD: HTRIP1-HTRIP2-HTDRP-HTRIP2-HTRIP1



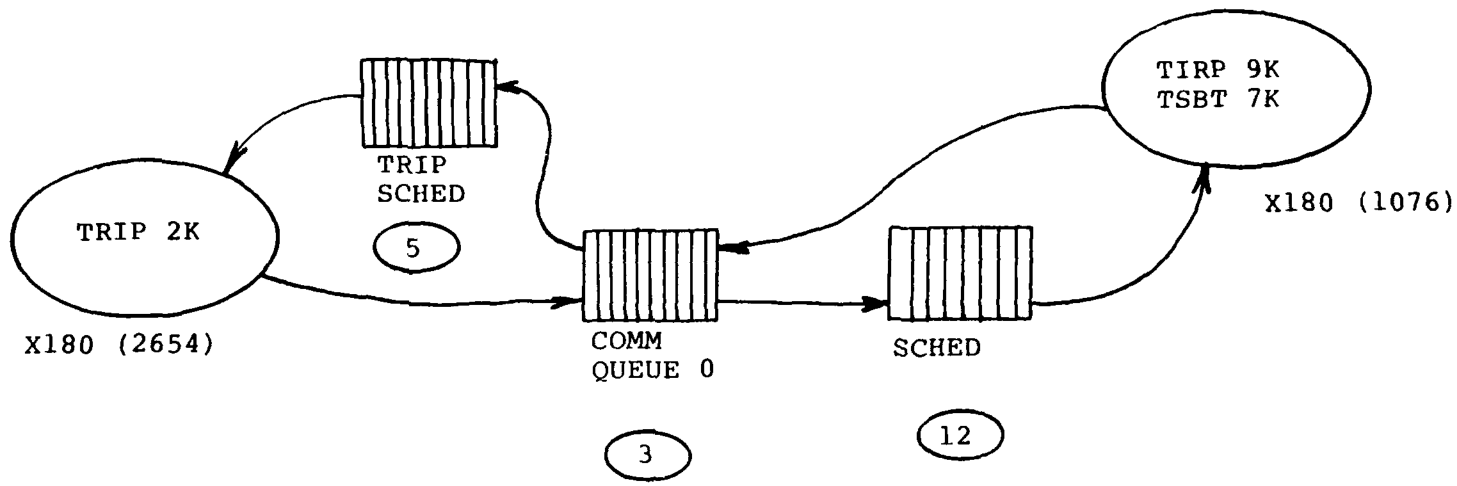
VERIFY RETURN

THREAD: TRIP-TDRP-TRIP



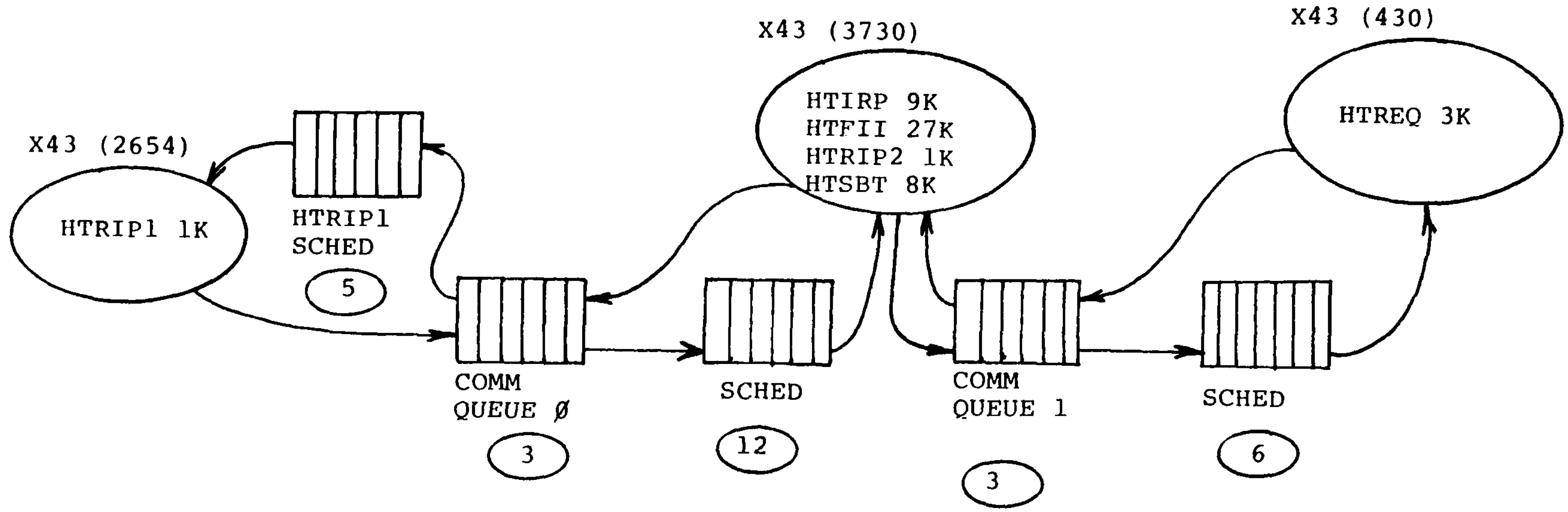
TRACK INITIATE

THREAD: HTRIP1-HTRIP2-HTIRP-HTSBT-HTRIP2-HTRIP1



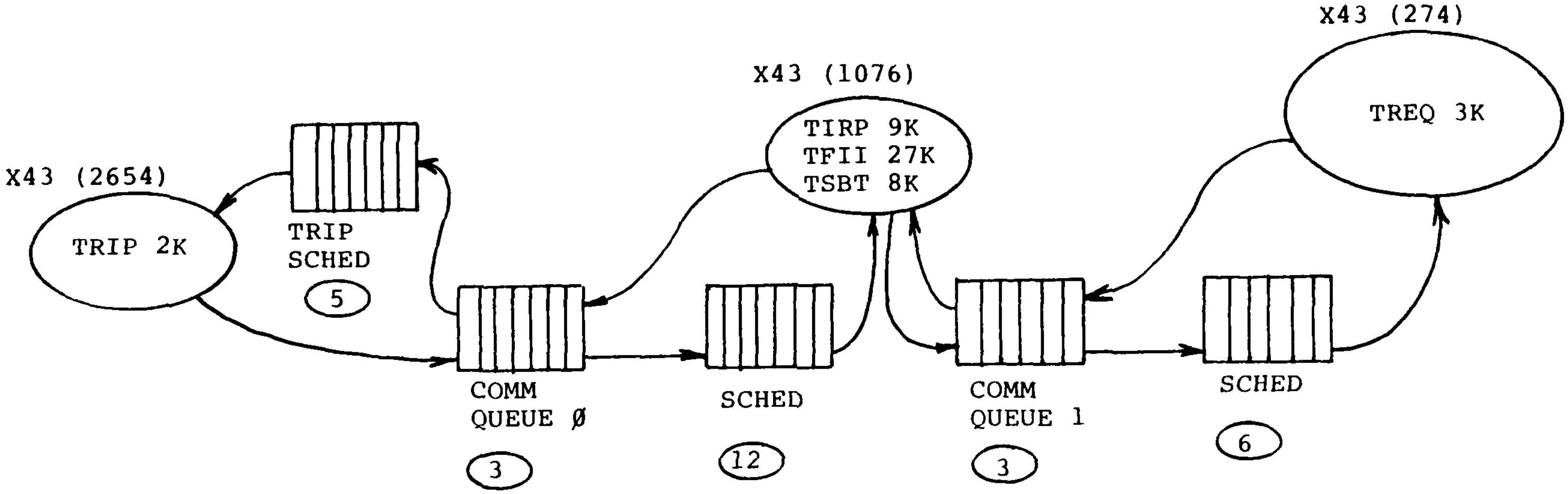
TRACK INITIATE

THREAD: TRIP-TIRP-TSBT-TRIP



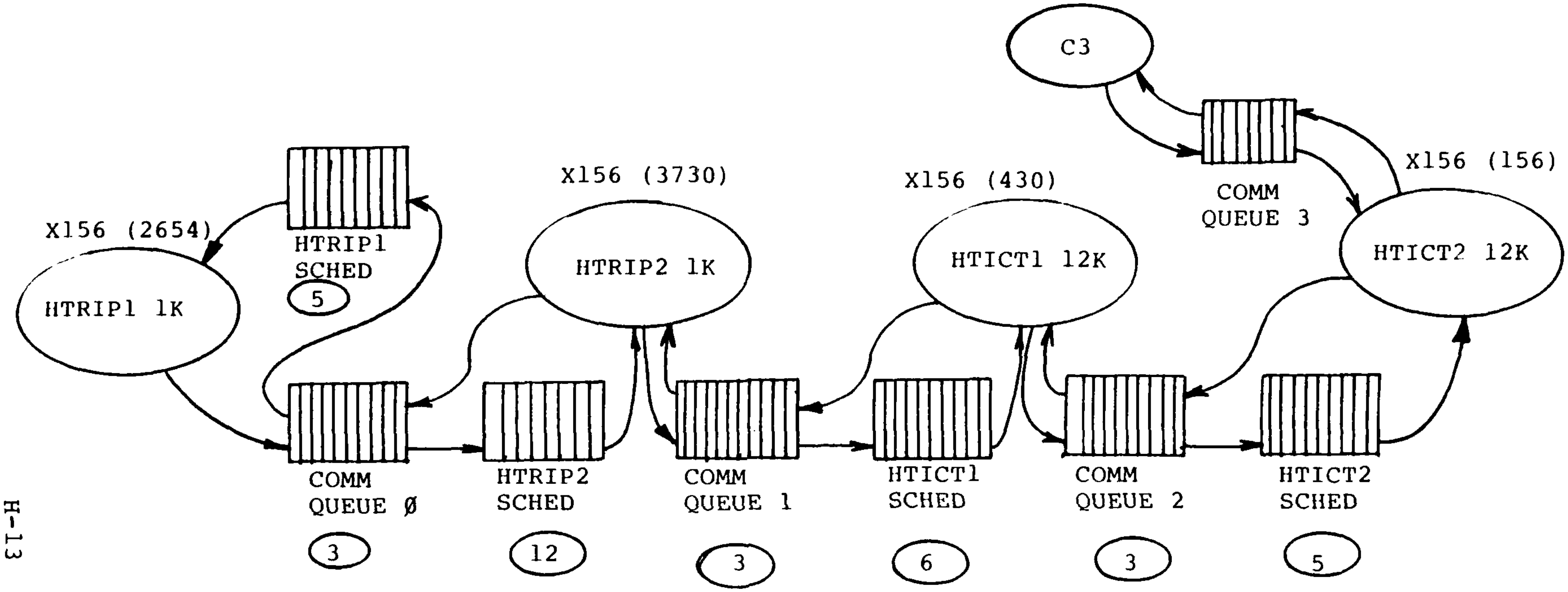
LAST TI

THREAD: HTRIP1-HTRIP2-HTIRP-HTSBT-HTFII-HTRIP2-HTRIP1



LAST TI

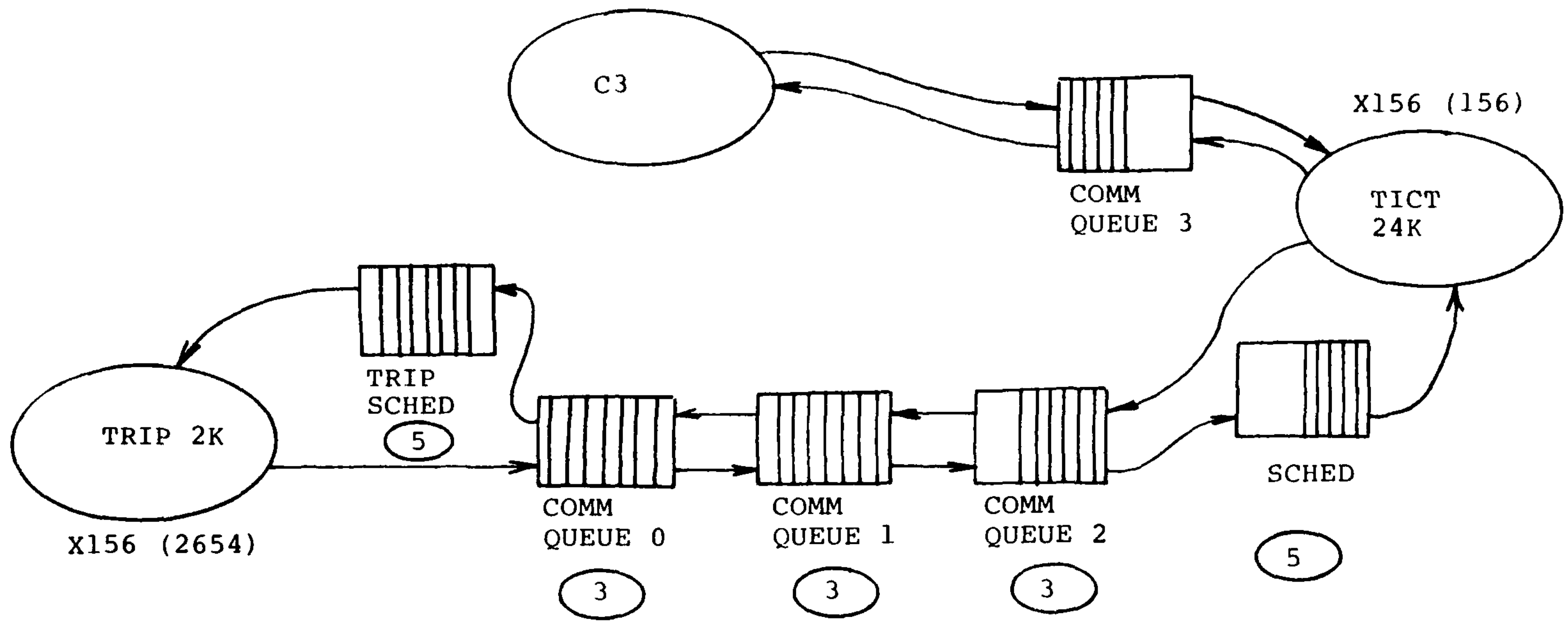
THREAD: TRIP-TIRP-TSBT-TFFI-TREQ-TRIP



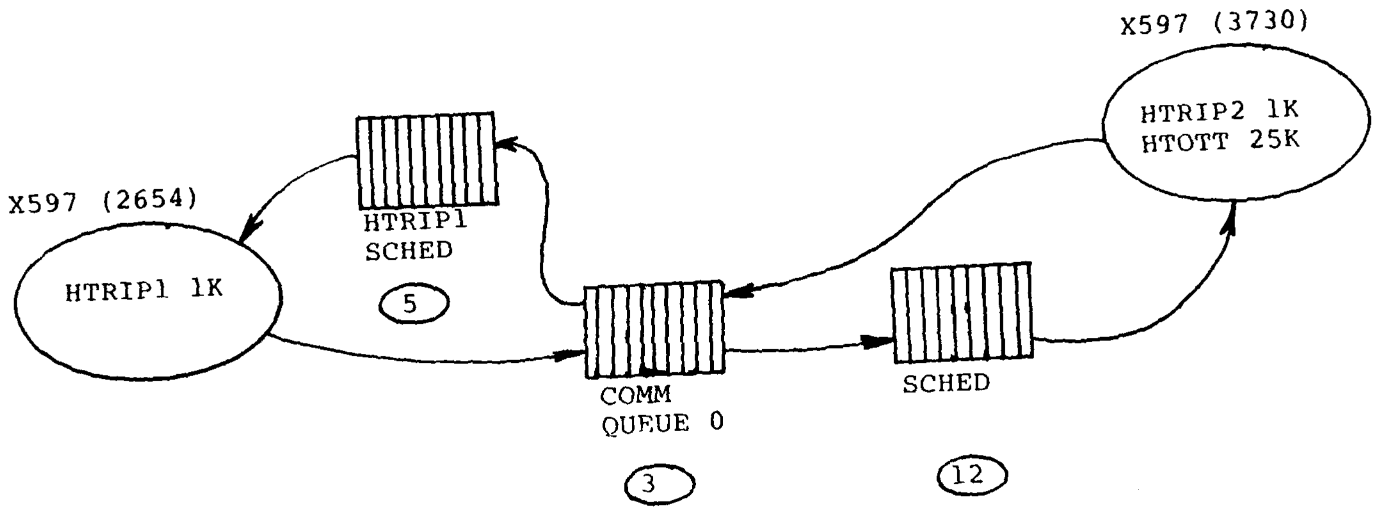
INTERCEPTOR GUIDANCE (NOT CRITICAL)

THREAD: HTRIP1-HTRIP2-HTICT1-HTICT2-HTRIP2-HTRIP1

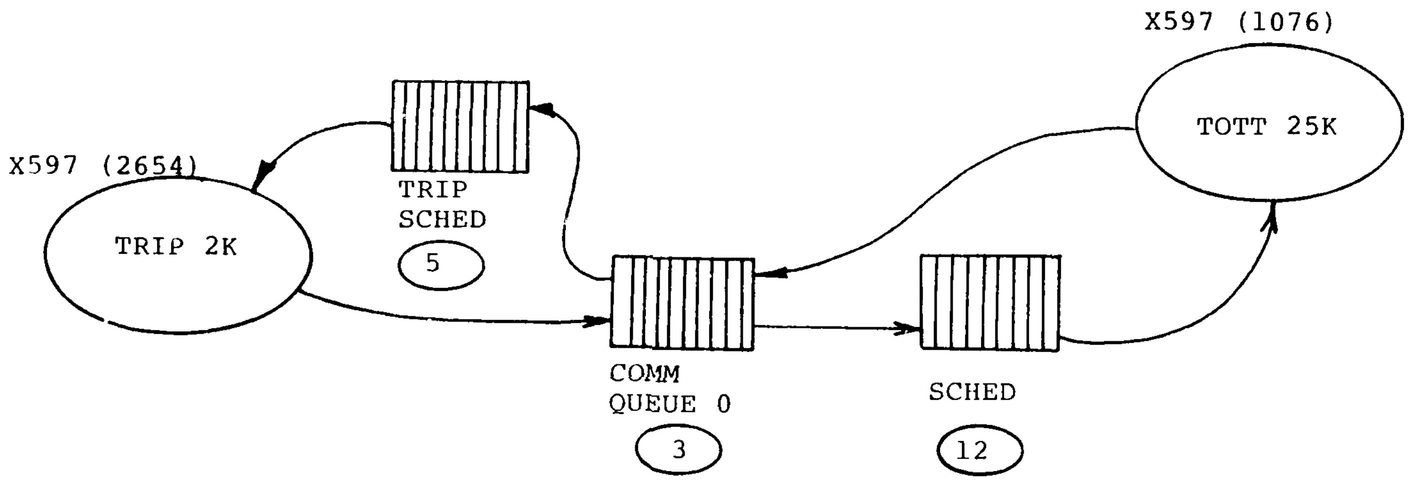
NEW CRITICAL: HTRIP1-HTRIP2-HTICT1-HTRIP2-HTRIP1



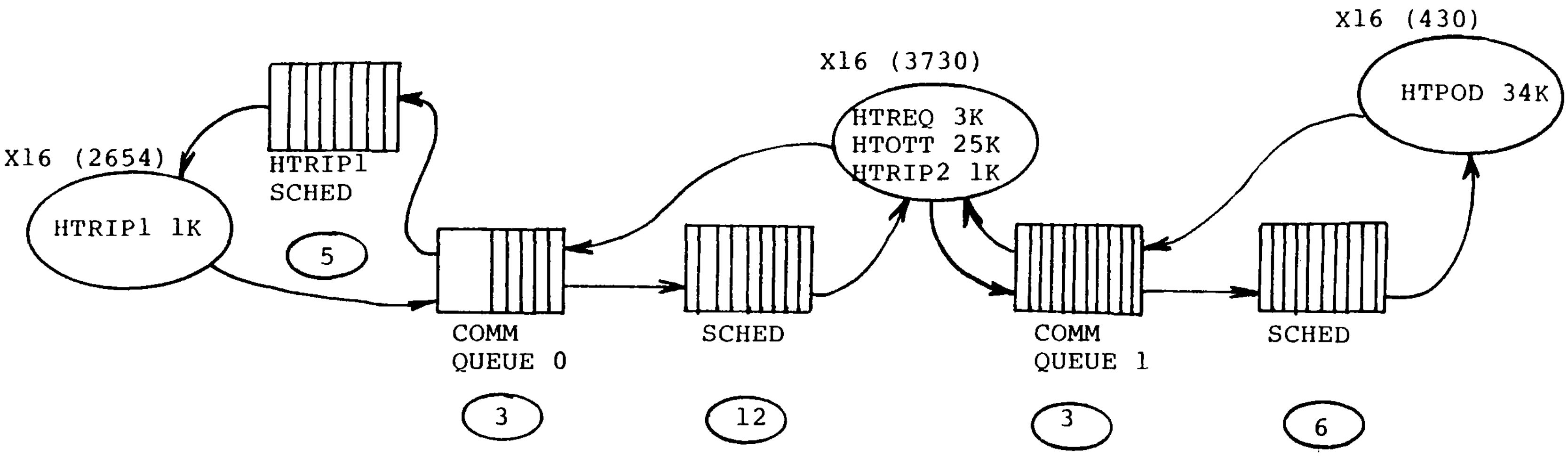
INTERCEPTOR GUIDANCE AND TRACK
THREAD: TRIP-TICT-TRIP



TRACK
THREAD: HTRIP1-HTRIP2-HTOTT-HTRIP2-HTRIP1



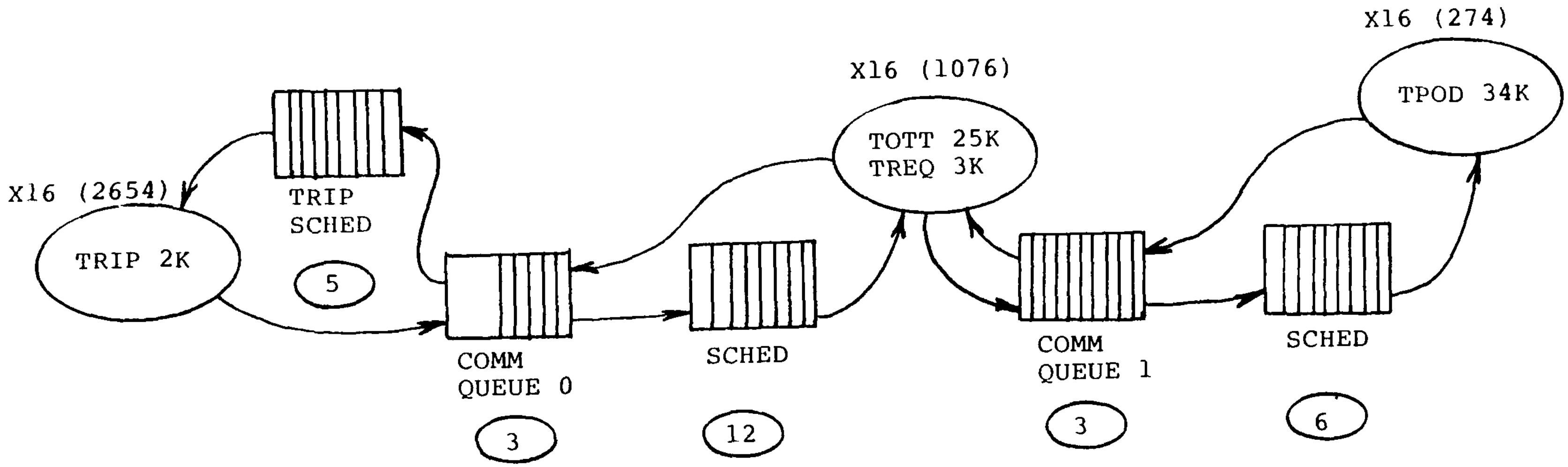
TRACK
THREAD: TRIP-TOTT-TRIP



ACTIVE DISCRIMINATION/DROP TRACK (NOT CRITICAL)

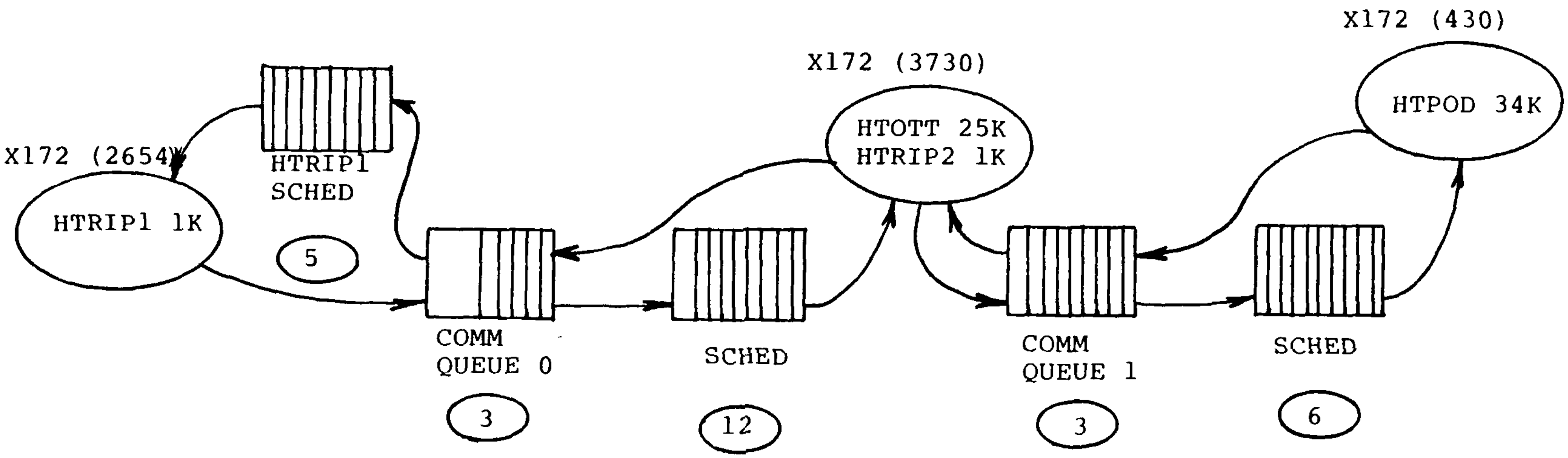
THREAD: HTRIP1-HTRIP2-HTOTT-HTPOD-HTREQ-HTRIP2-HTRIP1

NEW CRITICAL IS: HTRIP1-HTRIP2-HTOTT-HTREQ-HTRIP2-HTRIP1



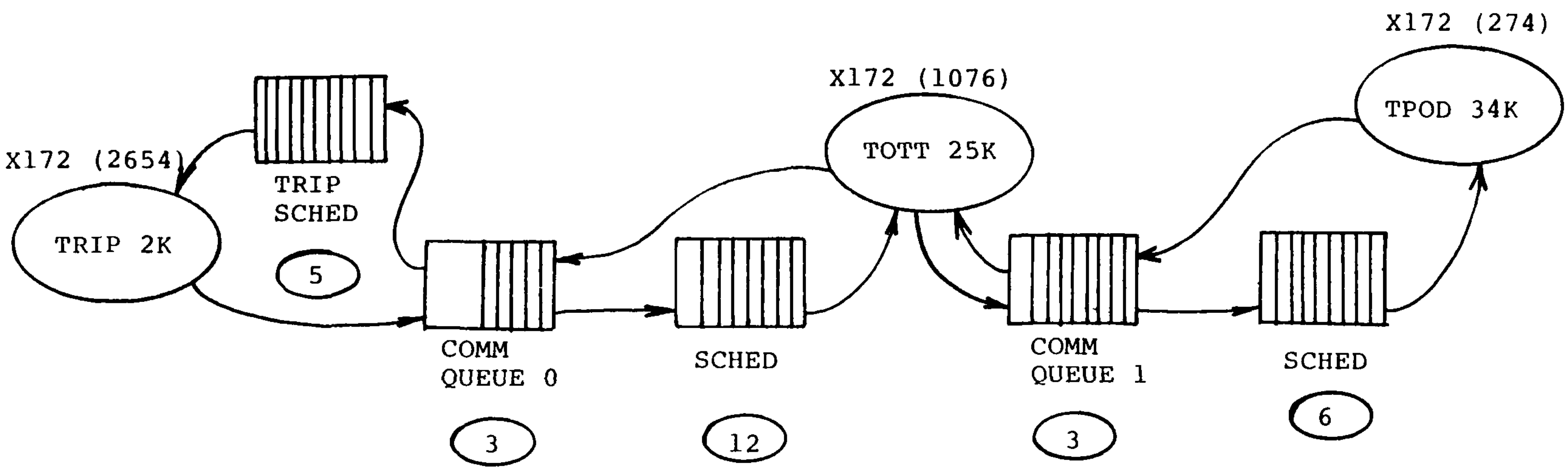
ACTIVE DISCRIMINATION/DROP TRACK

THREAD: TRIP-TOTT-TPOD-TREQ-TRIP



PASSIVE DISCRIMINATION (NOT CRITICAL)

THREAD: HTRIP1-HTRIP2-HTOTT-HTPOD-HTRIP2-HTRIP1



PASSIVE DISCRIMINATION

THREAD: TRIP-TOTT-TPOD-TRIP

APPENDIX I - COST SUMMARY

The following are cost data through October 1980. Final cost figures will be provided with the final version of this report.

The majority of the contract costs were for direct labor. There was one major travel expenditure to the Washington, D.C. area and the remaining travel costs were for local trips to the McDonnell Douglas facility and university libraries for research material. The Other Direct Charges were reproduction expenses for the ADPSI Report and research articles used as the basis for the final reports.

Following is a breakdown of these costs as of October 31, 1980.

PRELIMINARY COST SUMMARY AS OF 10/31/80

Direct Labor	
2501 Hours at an average Hourly Rate of $16.512/hour	$41296.29
Fringe Benefits @ 28%	11562.96
Subtotal	52859.25
Overhead @ 32.1%	16967.82
Subtotal	69827.07
Travel	1015.57
Other Direct Charges	253.70
Subtotal	71096.34
G&A @ 10.7%	7607.31
Subtotal	78703.65
Fee @ 9.73%	7657.86
Total Costs and Fee	86361.51

APPENDIX J

D. Tavangarian, "Signal Flow Graphs for Network Simulation", SIMULATION, Mar. 1980, pp. 79-92.

R. Gardner, "Multi-Level Modeling In SARA", Computer Science Department Internal Memo #165, UCLA.

D. B. Wortman, "An Introduction to Simulation, Simulation Languages, and Applications of Simulation", Pritsker & Associates, Inc., West Lafayette, Indiana.

M. J. Gonzales, "Deterministic Processor Scheduling", Computing Surveys, Vol. 9, No. 3, Sept. 1977, pp. 194-203.

W. Mayeda, et al, "A Boolean Theory of Network Flows and Metrics and Its' Applications to Particle Transmission and Clustering", Networks, Vol. 9, 1979, pp. 249-281.

O. Dahl, et al, "Some Features of the Simula 67 Language", Digest of the Second Conference on Application of Simulation, 1968, pp. 29-31.

G. Estin, "Evaluation Methods in SARA - The Graph Model Simulator", 1979 Conference on Simulation, Measurement and Modeling of Computer Systems, pp. 189-203.

J. Carroll, "A Closed Queueing Loop with one Non-Expotential Server", 1979 Conference on Computer Systems, pp. 99-106.

S. Shapiro, "A Stochastic Petri Net with Applications to Modeling Occupancy Times for Concurrent Task Systems", Network, Vol. 9, 1979, pp. 375-379.

E. D. Katz, et al, "The Minuteman Communication Network Simulator", Digest of the Second Conference on the Applications of Simulation", 1968, pp. 202-204.

J. Zahovjan, "An Exact Solution Method for the General Class of Closed Separable Queueing Networks", 1979 Conference on Simulation, Measurement and Modeling of Computer Systems, pp. 107-112.

F. W. Molina, "A Survey of Resource Defection Decomposition in Mathematical Programming", Computing Surveys, Vol. II, No. 2, June 1979, pp. 95-103.

J. M. Bertrand, et al, "Some Comments on Cybernetics and Control", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-8, No. 4, Nov. 1978, pp. 805-809.

Y. Haimes, "Overlapping Coordination of Hierarchical Structures", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-8, No. 12, Dec. 1978, pp. 745-747.

J. Dennis, "A Computer Simulation Facility for Packet Communication Architecture", The 3rd Annual Symposium on Computer Architecture, Vol. 4, No. 4, Jan. 1976, pp. 58-63.

W. R. Moore, "Some Fault Tolerance Design Principles for Modular Redundant Structures", FTC-5, June 75, pp. 119-123.

J. T. Lawson, et al, "Distributed Data Processing System Design - A Look at the Partitioning Problem", Tutorial Distributed System Design, Oct. 79, pp. 225-230.

J. A. Edwards, et al, "Optimal Partitioning of Workload of Distributed Systems", Tutorial Distributed System Design, Oct. 1979, pp. 231-236.

B. P. Bushels, et al, "Partitioning and Allocation of Logical Resources in a Distributed Computing Environment", Tutorial Distributed System Design, Oct. 1979, pp. 247-276.

B. Bussel, et al, "Partition and Allocation of Logical Resources in a Distributed Computing Environment", Tutorial Distributed System Design, Oct. 1979, pp. 291-298.

A. F. Gardenas, "Analysis and Performance of Inverted Data Base Structures", Tutorial: Centralized & Distributed Data Base Systems, Oct. 1979, pp. 235-245.

M. Schkolmick, "A Clustering Algorithm for Hierarchical Structures", Tutorial: Centralized & Distributed Data Base Systems, Oct. 1979, pp. 260-278.

S. Mahmound, et al, "Optimal Allocation of Resources in Distributed Information Networks", Tutorial: Centralized & Distributed Data Base Systems, Oct. 1979, pp. 419-431.

K. D. Levin, "Optimal Program and Data Locations in Computer Networks", Tutorial: Centralized & Distributed Data Base System, Oct. 1979, pp. 432-441.

Gerard Le Lann, "An Analysis of Different Approaches to Distributed Computing", Proceedings, 1st International Conference on Distributed Computing Systems, Oct. 1979, pp. 22-232.

R. Y. Kain", CHIMPNET: A Network Testbed", 4th Conference on Local Computer Networks, pp. 27-36.

M. Melman, "A Generic Program for the Modeling and Simulation of Distributed System Networks", 4th Conference on Local Computer Networks, pp. 97-103.

D. R. Mott, "A Distributed Computing Architecture for Real-Time System Control and Information Processing", 1st International Conference on Distributed Computing Systems.

M. S. Mahmoud, "Multilevel Systems Control and Applications, A Survey", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-7, No. 3, March 1977, pp. 125-143.

W. J. Graybeal, N. W. Pooch, Simulation: Principles and Methods, Winthrop Pub, Cambridge, Mass., 1980.

L. M. Benningfield, G. V. Logo, Circuit and System Theory, John Wiley & Sons, New York, 1979.

F. J. Ricci, Analog/Logic Computer Programming and Simulation, Spartan Books, New York, 1972.

H. Kobayashi, Modeling and Analysis: An Introduction to System Performance 'Evaluation Methodology, Addison-Wesley Pub Co., Reading, Mass., 1978.

T. B. Smith, III, "A Damage-and-Fault-Tolerant Input/Output Network", FTC-4, 1974, pp. 4-7, 4-11.

K. N. Levitt, et al, "A Comparative Study of Architectures for Fault-Tolerance", FTC-4, 1974, pp. 4-16, 4-21.

A. Avizienis, et al, "A Model for Transient and Permanent Fault Recovery in Closed Fault Tolerant Systems, FTC-6, June 1976, pp. 182-190.

ATE

LMED

8