N90-19417

1989

NASA/ASEE SUMMER FACULTY RESEARCH FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

MARSHALL AVIONICS TESTBED SYSTEM (MAST)

Prepared By:	Wayne D. Smith
Academic Rank:	Associate Professor
University and Department:	Mississippi State University Computer Science Department
NASA/MSFC:	
Laboratory:	Information and Electronic Systems
Division:	Computers and Communications
Branch:	Communications Systems
NASA Colleagues	Frank H. Emens Delisa Young
Date:	August 11, 1989
Contract No.:	The University of Alabama in Huntsville NGT-01-008-021

ABSTRACT

This report contains a summary of the work accomplished in the summer of 1989 in association with the NASA/ASEE Summer Faculty Research Fellowship Program at Marshall Space Flight Center. The project was aimed at developing detailed specifications for the Marshall Avionics System Testbed (MAST). This activity was to include the definition of the testbed requirements and the development of specifications for a set of standard network nodes for connecting the testbed to a variety of networks. The project was also to include developing a timetable for the design, implementation, programming and testing of the testbed. Specifications of both hardware and software components for the system were to be included.

ACKNOWLEDGEMENTS

I would like to thank my NASA colleagues on the MAST project, Frank Emens and Delisa Young, who did so much to make my summer research effort at NASA productive. They have consistently offered information, direction and encouragement during my stay at Marshall Space Flight Center. Without their help, little would have been accomplished.

I also need to thank Porter Clark and David Trice who, while not directly connected with the MAST project, nevertheless offered invaluable assistance during the summer. They generously provided me with lots of much needed assistance throughout the project. Leann Thomas also provided much needed encouragement and support. Thanks also to Leslie Murphy for help with the VME bus.

Finally, many thanks go to Charlene Breazeale for her assistance with the Xerox machine and related office equipment devices and procedures. Without her help, the paperwork would never have gotten finished.

TABLE OF CONTENTS

LIST OF FIGURES

1 Introduction

This report deals with the design and structure of the proposed Marshall Avionics System Testbed. This system is intended to provide a universal testbed for testing data communications networks of any type. This will include ground based networks as well as networks placed aboard space vehicles.

The testbed will provide facilities for network simulation experiments, as well as actual hardware interfaces for various types of flight and ground based hardware. NASA engineers and/or contractors can utilize the MAST facility to test proposed hardware or software components of a network. The MAST facility will provide the capacity to passively monitor network activity or to actively participate in network data transfer activities in order to stress the network to any desired level.

2 MAST System Overview

The MAST system consists of a network within a network that uses the facilities of one network to test the performance of the other network. Figure 1 indicates the general concept of the MAST facility. In addition to the two networks, there are a series of nodes that interconnect the networks in various ways.

2.1 Network Test System

The inner network is the network that is undergoing testing (TN). Due to the design of the MAST facility, only data pertaining to the network that is being tested is ever routed over the TN. Because of the generalized nature of MAST, there are few restrictions as to the topology or configuration of the TN. Bus, tree, token ring and even star type networks can be monitored and tested with MAST. The system can be reconfigured rapidly so that different types of networks can be tested in succession with minimum reconfigurration time.

The outer network is the control network (CN), and is associated with the testing and monitoring of the TN. This network is a 80 Mbps fibre optic network, and is used for all traffic pertaining to the initialization and monitoring of tests on the TN. Because the outer network must monitor and control the TN that is transporting data at rates up to 10 Mbps, it is designed to be significantly faster (≈ X10) than the network that is being tested.

2.2 Provisions for Hardware and Software Testing

The nodes that connect the two networks are of several types. The most significant node is the MAST Controller Node (MCN). This node provides facilities for MAST system software development, as well as providing the capabilities to initiate and monitor network tests for purposes of system software development. This node can be accessed through the system console, ten RS-232 ports or through an IEEE 802.3 (Ethernet) interface. In order to preserve the integrity of test results, system test operations must be initiated and controlled from the system console. Whenever a system network test is not in progress, the software development facilities are available.

A number of network interface controllers (NIC) comprise the remainder of the nodes that connect the network. These NIC nodes fall into three general classes which are:

Hardware Interface Nodes
Node Emulation Units
Network Monitor Nodes

The hardware interface nodes are used to interconnect various types of non-network hardware to both the TN and the CN. There are nodes with provisions for both digital and analog input signals. These NIC's are programmable, and can format the input data into the appropriate protocol for the network under test.

The node emulation units provide the capability for emulating any type of unit that might be connected to the TN. By using appropriate emulation units, the equivalent of a full network system can be provided in order to establish a test environment for a new node without the expense of providing a full hardware mock-up of the TN. In addition the node emulators can be used to run tests on network parameters without any actual network hardware in place.

The network monitor modes provide the capability to monitor the TN. These can be operated in an entirely passive (spy) mode that will induce no traffic on the TN, since all the monitor traffic will be routed over the CN. One monitor can also be operated in the active (probe) mode. In this mode, the monitor is used to induce traffic onto the network so that the performance of the network in response to this traffic can be monitored. The active monitor node can also be used to generate traffic volumes that will stress the network toward its upper limit capabilities. The active monitor can generate intentional errors and emulate a malfunctioning node by losing tokens, generating collisions, etc.

2.3 Future Expansion Capabilities

All the components of the mast system are designed so as to permit easy and quick modification of the system. Different types of TN interfaces can be configured from the MCN by down-loading software from the MCN to the appropriate NIC's. The NIC's can also be added to or removed from the TN under software control. Additional hardware interfaces can be connected to the TN by attaching these units to hardware interface nodes that are in place on the network and by loading these NIC's with appropriate software from the MCN via the CN.

It is, in fact, possible to have several hardware networks physically present in the MAST system at one time and to operate these separate networks under completely independent test conditions. Of course, only one network can be tested at a time without affecting the traffic flow, but since the system can be reconfigured by simply down-loading different software, this restriction should not constitute a major problem.

Due to the modularity of the MAST design, any of the components of the system can be replaced without having an adverse effect on the system. Nodes can be added or deleted as desired. The network under test can be changed as needed. If future needs dictate, the CN could be replaced with a faster model, such as FDDI. Even the MCN can be replaced if that were to be considered desirable.

3 MAST System Hardware Components

The MAST system hardware consists of several microcomputers that are connected between two or more physical networks. One of the networks is the TN, while the other is the CN. The microcomputers serve as nodes interfaces and node emulators on the TN.

3.1 Network Capabilities

As already mentioned, the network capabilities of the MAST system are considerable. The CN is used to initialize and monitor test experiments on any of a number of TNs. The dual network capability relieves the TN from the responsibility for transporting test configuration and metrics information, and avoids inducing monitoring artifacts onto the TN.

3.1.1 Control Network

The CN is the basis of the MAST system. It is currently configured as a 80 Mbps, bi-directional, counter-rotating, fiberoptic Proteon ProNet 80 network. The speed of this network is roughly an order of magnitude faster than any network likely to be tested by MAST in the near future. This high speed allows for the collection and transmission of network metrics from the TN in real time.

The CN consists of a hardwired configuration that is likely to remain static over relatively long periods of time. Nodes may be added to or deleted from the CN as needed, but this type of modification is relatively un critical with respect to time. Such modifications are not likely to take place while tests on the TN are in progress.

The CN provides the medium that permits the performance of four essential functions of the MAST system:

Initiate the system from a cold start
Configure the network prior to testing
Monitor the network during testing
Reconfigure the test network during testing

System initialization from a cold start is a mundane but necessary function. Power failures and system maintenance will require an occasional cold start. When the MCN is initially powered-up, the MAST system will have to be initialized. This will require configuring the MCN itself, and also configuring the individual NIC's. The software to configure the NIC's will be transmitted from the MCN to the NIC's via the CN.

Configuration of the TN prior to testing consists essentially of down-loading selected software modules to the NIC's, over the CN. The software modules that are down-loaded will determine the configuration of the network under test. For this operation, the high speed of the CN is not essential, but does contribute to the capability to reconfigure the system after one test in preparation for another test. With the appropriate test hardware already in place, reconfigurration of the system for other tests should require minimal time.

During the test phase, the CN can be used to pass TN performance metrics to the MCN. This capability permits a less expensive network monitor than would otherwise be possible, because the monitor node does not require data storage capabilities. Depending on the test, the monitor node can distill the data and forward only descriptive statistics to the MCN. The speed of the CN makes it possible for the NIC to forward raw data directly to the MCN if desired, however.

During an active test run, the MCN can transmit control data to the passive monitor NIC as needed. The MCN can, for example, wait for certain events to happen before instructing the monitor NIC to begin observing the network traffic. The MCN can also discontinue monitoring at any desired point during the test.

In the case of the active (probe) monitor, the MCN can initiate a number of node malfunction operations by down-loading software and/or instructions to the probe monitor NIC. These malfunctions could include a simulated dead node, continuous or an unusually high frequency of transmissions from a node, loss of token, duplicate tokens, intentional collisions, etc. Software to control these operations would be down-loaded to the probe monitor NIC via the CN, and be activated and disabled by the MCN via the CN.

During a test, the CN can also be used to permit the MCN to reconfigure the TN on-the-fly while a test is in progress. This would permit the simulation of stations dropping off the network or being added to the network in real time. This capability would be especially useful for simulating the operation of the network of a multi-stage space vehicle, where stage separation results in a planned, but nonetheless radical changes in the topology of the TN.

3.1.2 Test Network

While the CN is relatively fixed in topology and structure, the system is designed so that the TN can be varied at will. Each NIC serves as a connection between the

CN and one other network. The CN will remain more or less permanently connected to a number of NIC's. Each NIC is then in turn connected to one or more other networks that are to serve as a TN.

The operation of each NIC is controlled by the MCN by means of the software that is down-loaded from the MCN to the NIC. Each NIC can be loaded with software that will activate that node onto the TN as needed. Alternatively, any NIC can be loaded with software that will cause that node to remain inactive during any specific test. By enabling or disabling various NIC's, the operation of the test node can be controlled directly from the MCN.

With this software reconfigurration capability, it is possible for there to be several TNs physically in place at one time. By selectively activating the appropriate nodes, the TN can be changed very rapidly. This not only allows changing the structure of a particular type of network, but even permits changing the TN from Mil-Std-1553 to MAP to Ethernet, etc.

MAST hardware is currently provided to configure the TN into any of several standard networks. These include MIL-STD-1553, MAP and Ethernet. NIC's for other networks can be added to MAST as needed.

The Mil-Std-1553 network is a popular network for use in avionics systems. This network was developed by the UASF, and is a primary/secondary type bus network. The primary station polls secondary units that then have the opportunity to transmit data to the primary or another secondary unit.

The MAST hardware currently supports 15 NIC's that will each connect to two separate Mil-Std-1553 busses. Since there is some common circuitry in the two circuits, both 1553 networks cannot be operational at the same time.

The Manufactures Automation Protocol (MAP or IEEE 802.4) network was developed for use in manufacturing environments. This is a token bus network that has the advantage of insuring that a node with data to transmit will eventually have an opportunity to do so. The network operates in a fashion similar to a token ring network except that the physical interconnection between nodes is a bus topology. The token is passed in a ring fashion, but the ring exists only in the order of token passing determined by software.

The MAST hardware currently has two NIC's that can be configured as MAP network nodes. Additional units can be added as needed.

The Ethernet (IEEE 802.3) is a Carrier Sense Multiple Access/Collision Detection (CSMA/CD) network. It has a bus or tree topology, and is probably the most widely used network in the world today. The network provides each node with an equal opportunity to access the bus. When a unit wants to transmit a message, it waits for an idle bus and then transmits. If two units attempt to transmit at the same time, a collision occurs. In this case, both nodes stop transmitting and try again later.

The MAST system has provisions for two NIC's with an interface to Ethernet. These nodes provide an interface for network testing. There is an additional 802.3 interface on the MCN, but this interface is intended primarily for use during software system development. It could, however, be used to interface to the TN (802.3) if desired.

The generic NIC unit is easily interfaced to any other potential TN. As will be explained below, the cpu and CN interface are standard to all NIC's. To provide an interface to any other network, only the interface from a VME bus to that specific network would have to be added. This would include token ring networks and all types of proprietary vendor specific networks.

There are currently no NIC units configured for networks other than the three discussed above.

3.2 System Computers

The basis of the MAST system is the MAST Control Node (MCN) and a number of Node Interface controllers (NIC). Each of these components consists of a microcomputer system dedicated to the specific task assigned to each node. Each microcomputer is independent and self-contained, including rack, card cage and power supply. In the interest of standardization for software development, all microcomputers are based on Motorola 680xx microprocessors.

3.2.1 The Mast Control Node

The MCN is the heart of the MAST system. The MCN provides the facilities for two essential tasks associated with the MAST system. It serves as a software development environment for producing the software needed for the operation of MAST during system tests. It also provides the software needed to initiate, monitor, terminate and report the results of all network system tests. The software development facilities are provided through the operating system of the MCN computer itself. The test facilities are provided through the a MAST Network Operating System (MNOS).

The MCN is a Heurikon model # HSE/17 digital computer with a Motorola 68030 processor, 4 MBytes of main memory in a VMEBus cage. The computer has a 80 MB hard disk, a 1.2 MB floppy disk drive and a cartridge tape system. There are interfaces for a system console, an Ethernet connection and ten RS-232 connections. Users may interface to the software development system either via the ten RS-232 ports or via the Ethernet interface.

3.2.2 Node Interface Controllers

The Node Interface Controllers (see Figure 2) consist of a several separate interface controllers that are interconnected to the MCN through the CN. All MAST NIC nodes contain some common hardware, with variations only in the interface to the TN associated with that node. The common hardware includes a VMEBus card cage with power supply, a Heurikon Model HK68D/V2F 68020 based microcomputer printed circuit card with a separate system controller board. Limited system software is provided in EPROM for each node to facilitate assembly of the system.

Each node is also equipped with a ProNET interface to the CN. This interface consists of a Proteon 1580 control card, a Host Specific VME interface and the Proteon 3280 fiberoptic interface controller. This results in a total of four VME cards in each NIC card cage.

In addition, each node contains appropriate interface cards for the network to which the node in question is to be interfaced. These network specific cards are one of the following:

For Mil-Std-1553: BUS 65502 card
For MAP : MVME 272 and MVME 371FA cards
For Ethernet : Excellan LC-302 card

It would be possible for a single node to contain an interface to more than one TN, but such a configuration would probably not be cost effective. The interaction between the traffic on two different networks would make the analysis of network performance very difficult.

3.2.3 Hardware Component Interface

For a node that is to function in a node emulation mode, only the modules outlined above will be used, depending on the network under test. When a hardware component is to be interfaced to the network, additional translator cards will be utilized.

For a digital network interface, Data Translation DT 1417 card can be used to interface up to four eight-bit ports onto the VME bus and thence to the network. For analog signals, a similar DT 1414 board provides 16 analog to digital (A/D) channels, and also has 16 programmable digital inputs. In either case, the NIC computer reads the input data as required and packages the data for transmission on the TN.

3.2.4 Network Monitors

A critical function of the MAST system is the capability to monitor network traffic while the network test is in progress. This function is accomplished through special network monitors. A network monitor is a node that has been modified so as to respond to all traffic on the network, not just the traffic addressed to a particular node. Such monitors will observe traffic on the network, and keep records relating to the performance of the network.

A standard NIC node will require hardware modification in order to perform as a monitor. Specialized software will also be required for the monitor nodes. There will have to be one modified network interface monitor for each TN that is to be monitored. Since only one network test will be active at any one time, all these interfaces can be installed in a single NIC that is dedicated to the network monitor function.

The passive monitor will simply observe the TN without itself actively participating in the traffic on the TN. That is, it observes and records traffic, but is neither the source or destination for any network traffic. The monitor function may be started, suspended or stopped by messages sent to the monitor from the MCN over the CN. Data collected by the monitor may be transmitted directly to the MCN as raw data, or the monitor may be programmed to forward only statistical data to the MCN. The monitor can observe all traffic, or only a selected subset of traffic as determined by the needs of the user.

A probe monitor can also be configured from the NIC hardware discussed above. In this case, the monitor is active in the traffic on the network. The probe monitor will introduce traffic onto the network and then observe the effects of this traffic. Some types of data cannot be collected by a passive monitor, and must be collected by a probe monitor. For example, only a probe monitor can determine the average time delay in acquiring network facilities for the transmission of data.

The probe monitor can also be used to stress the network to any desired level. By generating a high volume of spurious traffic, the effects of this traffic on the throughput of traffic from other nodes can be observed. Probe monitors can also generate intentional errors and interference on the network if desired.

3.2.5 Gateways, Bridges and Routers

Since each node in the MAST system has a hardware interface to two networks, the capability is present to program any node to perform as a bridge, gateway or router. By adding the appropriate software, any node could perform any of these functions. This means that a bridge between ProNET 80 and either Mil-Std-1553, MAP, or Ethernet represents would be relatively easy to implement. Such utilization of a node would, however, dedicate that particular node to the bridge, gateway or router function, and would prohibit its use during MAST tests.

A more interesting possibility would be to utilize a node with two different network interfaces along with a ProNET interface. This configuration would provide a bridge between the two networks in question, and permit monitoring the traffic on two different networks simultaneously. With proper programming it would also permit the collection of performance metrics of internet traffic between the two networks.

4 MAST System Software Components

There are two major components of the MAST system software. These components are the Software Development System and the MAST Network Operating System.

4.1 Software Development System

The software development system is a relatively standard UNIX operating system environment that provides the software tools for developing the modules required for the MAST Network Operating System (MNOS)as described below. Facilities provided include a compiler for C that generates code that can be down-loaded to the NIC's. There is also a 68030 assembler, a linker, a debugger and various device drivers. FORTRAN and Pascal compilers can be added later.

All the system development software is accessible through the ten RS-232 ports or the Ethernet interface on the MCN. The development system is available at any time that a network test is not in progress. The advisability of using the software development system while a network test is in progress requires further investigation.

4.2 MAST Network Operating System

The MNOS is the heart of the system testbed. The MNOS will permit the configuration of a network prior to a test by down-loading appropriate software to the NIC's. MNOS will initiate the test by activating the software within the appropriate NIC's. During the test, performance metrics will be collected. After the test, MNOS will make the data relating to network performance available for analysis.

An overall chart of the structure of the MNOS is included in Figure 3. This figure depicts the general interrelationship between the components of the system. To the maximum extent possible, MNOS is a menu driven system with graphical displays of TN configuration, operation and performance.

4.2.1 Test Configuration and Initiation

The test configuration section (Figure 4) of the MNOS allows the user to establish the software environment to configure the network on which the test is to be conducted. Prior to this stage, all hardware components must be in place. Note that all NIC's are permanently attached to both the CN and TN. Only hardware units that are to undergo testing would have to be added to the basic testbed.

The test configuration section permits the user to specify the type of network (Mil-Std-1552, MAP or Ethernet) to be tested. The NIC's that are configured for interface to the TN will then be marked as available for test use. All other NIC's will be disabled by software.

From the available nodes, the user may then choose any combination of hardware interface nodes and software emulation nodes. Appropriate software is down-loaded to insure that the selected nodes perform in the desired manner. Any unneeded nodes are disabled by software.

The user can select the number and types of network monitors that are to be used during the test. The user also specifies the type of metric information that is to be collected during the test, and when the information is to be collected. Probe monitor parameters may also be set during configuration of the network.

When the network is fully configured, the test is initiated (Figure 5). During the initiation, parameters relating to the test, such a length of test, abnormal termination criteria, etc. are established. The MCN then signals all NIC's to begin the test operation.

4.2.2 Test Monitoring

While the network test is in progress, the MCN will present the user with selected real time system operation statistics (Figure 6). The user will be able to select from several graphical displays that contain the network operating characteristics. The user can monitor the overall operation of the TN and terminate or modify the test if that appears desirable.

The user will also be given the opportunity to interact with the network nodes in real time. For example, the user can disconnect nodes from the network or add new nodes to the network. The user can also modify the monitor data collection instructions or command the probe monitor to change operating modes.

4.2.3 Test Analysis

After the test is completed, the user can review the statistics that were collected by the MCN during the test (Figure 7). The user has the choice of access directly to the raw data, or may choose from several levels of distilled data. In order that data from one network test can be analyzed while another test is in progress, the test results will be available in hard copy and also from the test data base via the software development system facilities. Adequate statistical routines will be available to permit analysis of the results.

5 System Operation

The basic purpose of MAST is to provide a generalized testbed that will permit the testing of network components. The most common test would probably consist of testing a single hardware unit that is to be added to an existing network in order to evaluate the effectiveness of the unit and its effect on network performance. In many cases, however, it may be desirable to run a test on the characteristics of the proposed hardware device through the use of a software emulation module that will mimic the predicted performance of the hardware prior to testing the hardware interface itself.

It is also possible to test a completely new network design using either emulation nodes or actual hardware interfaces for all the nodes on the proposed network. MAST provides the capability to take an existing network configuration and modify the network protocols to determine the effects that different protocols would have on network performance. Being completely general, MAST also permits varying the number of elements in a network as well as evaluating different priority schemes for the elements.

5.1 Network Test Overview

The user begins a test from the MCN console. From the initial menu, the user chooses to enter the test configuration section of the software. At this point, the user may define the hardware and software emulation components that will constitute the elements of the network that is to be tested.

When the user enters the network configuration section, the TN defaults to a null network with no nodes. Through a series of questions and responses, the user can configure any type of network (Mil-Std-1553, MAP or Ethernet) that is desired. From all the NIC's available, The MCN configuration software will then select only those nodes that interface to the network type selected. All other NIC's are disabled.

If the user has one or more nodes that contain a network interface to external hardware, these nodes are specified next. When the user dedicates a node to a hardware interface node, an absolute NIC address must be provided for that node so that the MCN can match the down-loaded software to the hardware attached to that NIC. As nodes are added to the configuration, the network configuration data will be updated to indicate the presence of all such nodes connected to the TN.

After hardware interface nodes (if any) are selected, the software emulation nodes (if any) can then be specified. With software emulation modes, the user needs only to specify the type of node that is to be emulated, and the appropriate software for this node will be down-loaded to any available NIC.

After all operating network nodes have been specified, the monitor requirements for the test in question are specified. Monitor specification includes both passive (spy) nodes and active (probe) monitors. The amount and type of data that is to be collected from the network is indicated as the monitor nodes are specified.

Once the configuration of the TN has been established, the user can enter the test initiation section of the MNOS. At this point the user will be given options as to the start time of the test operation, the length of the test and other parameters. When all the parameters are set, the test can begin.

When the test operation begins, the user can view performance statistics on the MCN console. This will include graphical displays of network performance. The user will be given the option of selecting any of several different screen displays that present various network performance statistics. The user can also interact with the TN to dynamically alter the topology and configuration of the network while the test progresses. The user can also terminate the test or extend the test duration from the MCN.

When the test terminates, the user will have access to the data that was collected by the monitor nodes during the test operation. This data can be analyzed to determine system performance during the test. The data analysis phase will be conducted through the software development system facilities rather than through the MCN.

5.2 A Hypothetical Example

Perhaps the easiest way to get a general understanding of the operation of MAST is to consider a simple example that will reflect most of the options available for a MAST test. In this example, we assume that a new hardware device is to be tested for inclusion on an already existing Mil-Std-1553 network.

For our example, we assume that the existing network contains an engine health monitor (EHM), several transducers, several sensors and a flight controller. This network has been tested previously, and there are existing software emulators and hardware interface modules stored in the MAST module library. Network configuration information is also already stored in the library. The purpose of the test is to evaluate the effectiveness of an new improved EHM that is to replace the old EHM.

In most cases, it will be desirable to model the predicted behavior of the new EHM through a software emulation module before the actual hardware for the EHM is begun. In this situation, the initial step would be to develop a software module to model the expected behavior of the new EHM. This module could be fairly realistic with respect to the volume and type of traffic that would be generated by the new EHM.

In performing the emulation test, the test conductor would first load the old network configuration from the MAST library. This configuration would then be modified by deleting the old EHM emulator and replacing it with the new EHM. A complete test would be run using emulation modules for the entire network or by using a mixture of emulation modules and previously tested hardware interface controllers.

After the emulation module tests indicate that the new unit is operating satisfactorily, the experiment would proceed to test the proposed hardware itself. In some cases, the emulation test might be skipped and the test would begin directly with the hardware interface.

For a hardware interface test, the first step would be is to develop the software to interface the new EHM to the network. This software is designed to read the digital and/or analog inputs from the EHM hardware and format the input data into the appropriate protocol for the network being used. This data is placed onto the network in accordance with the contention algorithms used by the network in question.

When any component is replaced in a test configuration, it is also likely that the monitor parameters will need to be modified. In the example in question, all the system components except the new EHM have already been tested, and can be assumed to be operational. Collecting data on these units would probably be unnecessary. The performance characteristics of the new EHM are unknown, however, and the monitor might need to be instructed to observe and record all traffic to and from the new EHM node. At the same time, general network data such as number of messages carried, number of collisions, average message length, etc., might also be collected in order to assess the effects that the new EHM may have on the network as a whole.

After the network is configured, the user enters the test initiation phase. Here the start and stop time for the test is established. These time can be clock time, or network events. For example, if the new EHM was in the first stage of a flight vehicle, the test could begin at the initiation of the countdown sequence and end with separation of the first stage. Both these events could be identified from network traffic.

During the test, the user would monitor the operation of the new EHM by selecting to display on the console a summary of messages to and from the EHM. Alternatively other displays of network traffic could be viewed as desired.

After the test is completed, the user can leave the MCN and examine the test statistics from one of the terminals in the software development system. Modifications can be made to the emulation module and/or the hardware interface module if desired, and the test can be repeated with different parameters.

5.2.3 Network Test Summary

In performing network tests, the MAST system can perform two basic operations on any of three (or more with new hardware) networks. These experiments consist of testing a node emulation module or a node hardware interface. With any number of nodes possible, and with three different network interfaces permanently connected, there is an almost unlimited number of test configurations that can be established on the MAST system.

In addition, the network parameters, such as packet length, token hold time, backoff algorithms, etc., can also be tested. Hence, the MAST system provides the capability to perform almost any conceivable network test or experiment. Further, the experiments can be configured and executed with minimum set-up time between different experiments.

6 Development Plan

The development of the MAST system can be divided into roughly five phases. These phases are somewhat interdependent, and overlap to some extent. The major phases are:

- System operating specifications development
- System hardware specification and procurement
- System software specification and procurement
- System assembly and testing
- System operation and maintenance

The development of the MAST system is already in progress. The system operating specifications have been developed, and most of the hardware specifications have been completed. Much of the hardware for MAST has been procured and is already in place in EB33.

There are two major tasks remaining before the MAST system can enter the operational phase. These tasks are: software specification and procurement, and system assembly and testing.

6.1 System Operating Specifications Development and System Hardware Specification and Procurement

As noted above, the majority of the operating specifications for the MAST system have been completed. Minor revisions to accommodate omissions and corrections will still be necessary, but the major work is completed.

In addition, the system hardware specifications are complete. The MCN and NIC device specifications have been determined, and the equipment procured. The ProNET 80 hardware is also in house, as well as a number of interface components for the three initial TNs. Minor components remain to be fabricated or procured, but, in general, the hardware for MAST is in place.

6.2 System Software Specification and Procurement

By far the largest task remaining in the MAST program is the development of the specifications for and the procurement of all the software required for the testbed. This includes the development of the MNOS as well as the emulation and hardware interface modules, the monitor software and the statistical routines for the test result analysis.

6.2.1 Networking Capability

The networking capabilities required for the MAST system are extensive. Each NIC will have to interface to two different networks. A survey of current literature reveals little or no information on this type of dual ported network nodes. Of course, there are bridge and gateway systems that interface to two networks, but these are an order of magnitude less complex than the MAST NIC's will be. The development of software to control access to two heterogeneous networks will present an interesting challenge.

The software will have to interact with both networks simultaneously. During major portions of the test, the operation of the CN will be invisible to the NIC, but under special circumstances, the NIC will have to recognize and respond to commands from the CN while maintaining full operation on the TN. This implies the extensive use of interrupt facilities on the NIC hardware.

During test configuration there will have to be extensive down-load capability built into the CN. It will be essential for the MCN to be able to control the operation of each NIC. This includes up-loading and down-loading of software, initiation of test runs, network reconfigurration in real time and premature termination of tests.

The equipment on hand for MAST varies greatly in the type and amount of software provided by the vendor. MAP interface boards come with the full seven level OSI protocol, at least according to the supplier. The ProNET 80 boards, on the other hand, require software interfacing at the register and buffer manipulation level. Software for other boards lies somewhere between these levels. It will be a significant undertaking just to get all the interfaces to all these different networks operational at some common user- acceptable level.

6.2.2 Software Development System (SDS)

Most of the software for the SDS is already in place. This software was purchased as an option with the UNIX operating system for the MCN. The software contain a C compiler, an assemblers, linkers, debuggers, device drivers, etc. Unless a more user-friendly environment than UNIX is desired, this software should suffice for most software development operations.

Since it is desirable to have the data analysis software available from any SDS terminal rather than just from the MCN, some sort of interface to the test data storage will have to be made available from the SDS. This could be anything from a locally written system to a statistical package such as SPSS or a simple data base program. The requirements for this program are currently unspecified.

6.2.3 MAST Network Operating System (MNOS)

Developing the software for the MNOS offers an interesting challenge. The software required ranges all the way from a very high level operating system user interface (MNOS) itself all the way down to some rather primitive register manipulation procedures in the hardware interface modules. Not only does the development of the software present an interesting challenge, but the prospect of having up to 18 network nodes, two monitor nodes and the MCN operate in harmony over two different networks simultaneously adds to the complexity of the task.

MAST will require all the features and facilities normally associated with computer network operations, but will require everything four times over for the four different networks used on the system (the CN and three TNs). Even utilizing Internet Protocols (IP), the task will be formidable. At some point, NIC's will be used as bridges and gateways, and MAST will evaluate the effectiveness of internet communications operations. Monitoring these bridges and gateways will present another opportunity for exciting software development.

6.2.3.1 Test Configuration Section

The test configuration section is software that is unique to the MAST system, and will have to be written specifically for the MAST system. The driver portion of the software will be the user interface to the set-up of a network experiment. The program will be utilized by users that are generally unfamiliar with the overall MAST system, and will therefore have to be robust and easy to use.

The configuration software will have to be menu driven, and if possible should have graphics capabilities. The user should be able to select options from various menu lists in order to configure the network to be tested. The graphics capability would be beneficial in displaying the configured network for inspection and modification.

As a part of the configuration section, the various modules associated with the NIC nodes must be available. This includes the hardware interface modules and the software emulation modules. Development of the software emulation modules will represent a sizable software engineering task. There will have to be software modules to emulate the performance of any component that might be placed on the network. This includes engine health monitors, transducers, sensors, flight controllers, RF data links, etc. Such simulation modules will have to be developed for many different kinds of the same devices, since the performance will vary from one contractor to another.

Hardware interface modules will have to be developed for any hardware that is to be interfaced to a TN. A few of these may need to be developed initially as generic interfaces, but in general, these will be one of a kind interfaces that will have to be developed as they are needed. Since each interface is specific to the hardware involved, it is likely that the MAST user will develop these interfaces as they are needed for a specific test.

6.2.3.2 Test Parameter Initialization

The parameter initialization section of MNOS permits the user to specify the specifics of a single network experiment after the configuration of the network has been established. Software will have to be written that will interact with the user as the experiment parameters are entered. This software should also be developed as menu driven software.

There should be provisions for setting abnormal test conditions that will halt the test, and also for setting network "breakpoints" that will permit stopping the test at certain points for observation of the status of the network.

6.2.3.3 Test Operation and Monitoring

During the actual test, the MNOS software will be required to monitor the operation of the network and display summary data to the user. This software must also provide interactive capability to modify the test parameters while the test is in progress. The software will receive data from the monitor nodes and dispatch commands to all nodes via the CN.

6.2.3.4 Metric Collection and Statistical Analysis

While the test is in progress, the software in the monitor nodes will collect and store network performance data. As necessary this data will be transferred to the MCN over the CN for storage on disk. Parameters to control the data collection are specified as the test is initiated, but can be changed as necessary during the test operation. Data is stored for analysis later under facilities provided in the software development system.

6.3 System assembly and testing

The MAST system hardware is to be assembled and initial software development initiated in the very near future. Since the building that will eventually house the MAST system will not be completed until 1991, the assembly process will be in two or three phases. The initial phase will assemble the MCN and two or three NIC's in room B-240 of building 4487. The final phase will occur when the MAST system is moved to its permanent location in a new building proposed for that purpose.

Depending on the completion date of the new building and progress made in developing software during the first phase, a second phase may be necessary. In this case, a secondary location in building 4487 will be used to set up a more complex MAST development system consisting of about ten nodes plus the MCN. A location in A wing of 4487 has been tentatively identified for this purpose.

6.4 System operation and maintenance

Once the system is developed, operation and maintenance will begin. Operation of the testbed will normally be under the direction of the organization that wishes to test a network of a specific network node or interface. Development of specialized software for the interfacing of specific hardware devices will probably remain the responsibility of the organization proposing the hardware.

Software emulation modules will have to be developed for a wide variety of devices that will need to be emulated on the network. Development of these modules will probably remain the responsibility of the organization operating the MAST facility. It is reasonable to assume, however, that users of the MAST system who develop emulation modules in the normal process of testing their own designs will make such modules available to other MAST users through the NOS library.

6.5 Development Time Table

A very tentative time table for the development of the MAST system is included in Figure 8. Tentative completion time is December 1991. The software development and system testing will occupy the majority of the project time.

7 Conclusions and Recommendations

The preliminary specifications for the MAST system are essentially complete, and the majority of the hardware is in place. Some preliminary hardware tests have been conducted this summer. The hardware chosen appears to be appropriate for the task specified, and seems to operate at the specified levels.

Software development has not yet been initiated. Due to the complexity of the undertaking, and due to the fact that there is little current literature relating to a testbed as complex as MAST will be, there is a significant software development task ahead. System programming should not begin until such time as system software specifications have been completed.

The design of the software component of the MAST system needs to be addressed in the very near future in order to insure a timely development of the entire system.

REFERENCES

Frank Ingles, John Owens, Stephen Daniel, F. Ahmad, and W. Courrellion, " A Study of the Marshall Avionics System Testbed, Final Report, Phase 1", Electrical Engineering Department, Mississippi State University, Sept 30, 1989.

D. Jackobson, S. Gaitonde, J. Kim, J. Lee, D. Rover, M. Sarwar, M. Shafiq, "A Master/Slave Monitor Measurement Technique for an Operating Ethernet Network", IEEE Network, Vol. 1, No. 3, pp 40-48, July, 1987.

David C. Wolfe, Jr., "Baseband Emulators for the Protocol Testing of Multimedia Communications Networks", IEEE Network, pp 8-13, November, 1988.

Lillian N. Cassel and Paul D. Amer, "Management of Distributed Measurement Over Interconnected Networks", IEEE Network, Vol. 2, No. 2, pp 50-55, March, 1988.

Howard Salwen, "How to Get Ready for FDDI", Proc. 1987 IEEE 12th Conference on Local Computer Networks, pp 12-13, 1987.

Ilan Kolnik and Joseph Garodnick, "First FDDI Local Area Network", Proc. 1987 IEEE 12th Conference on Local Computer Networks, pp 7-11, 1987.

Larry Green, "Planning for FDDI", Proc. 1987 IEEE 12th Conference on Local Computer Networks, pp 4-6, 1987.

William M. Seifert, "Bridges or Routers", Proc. 1987 IEEE 12th Conference on Local Computer Networks, pp 118-129, 1987.

John Hart, "Extending the IEEE 802.1 MAC Bridge Standard to Remote Bridges", IEEE Network, Vol. 2, No. 1, pp 10-15, January, 1988.

Floyd Backes, "Transparent Bridges for Interconnection of IEEE 802 LANs", IEEE Network, Vol. 2, No. 1, pp 5-9, January, 1988.

Michael Soha, "A Distributed Approach to LAN monitoring using Intelligent High Performance Monitors", IEEE Network, Vol. 1, No. 3, pp 13-20, July, 1987.

Dick C. A. Bulterman and Eva Manolis, "Application Level Performance Prediction Tools for Network-Based Systems", IEEE Network, Vol. 1, No. 3, pp 6-12, July, 1987.

R. Aronoff, K. Mills, M. Wheatley, "Transport Layer Performance Tools and Measurement", IEEE Network, Vol. 1, No. 3, pp 21-31, July, 1987.

Donna Ritter and Marilyn Seale, "A Multi-Purpose, Distributed LAN Traffic Monitoring Tool", IEEE Network, Vol. 1, No. 3, pp 32-39, July, 1987.

B. Lewis Barnett, III, Kanishka Abeynayake, Miroslaw Malek, "LANCET: Local Area Network Comprehensive Evaluation Tool", Proc. 1988 Computer Networking Symposium, IEEE Press, pp 340-347, 1988.

B. Lewis Barnett, III and Michael K. Molloy, "The Local Area Network Testbed", Technical Report TR-85-25, Department of Computer Sciences, University of Texas at Austin, May 1985.

John McConnell, "Internetworking Computer Sustems", Prentice Hall, New Jersey, 1988

William Stallings, "Data and Computer Communications", second edition, Macmillan Publishing Company, New York, 1988.

Andrew S. Tannenbaum, "Computer Networks", second edition, Prentice Hall, New Jersey, 1988.

Excellan Inc., "EXOS 302 Intelligent Ethernet Controller for VMEBus Systems Reference Manual", Publication Number 4200090-00, Excellan, Inc. 1988.

Motorola, Inc. and VMEBus International Trade Association (VITA), "VMEBus Specification Manual, Revision C1", Motorola, Inc. and VMEBus International Association, Oct 1985.

Proteon, Inc., "ProNET Model p1580 VME Local Network System Installation and Programming Guide", Proteon, Inc., Westborough, MA, April 1986.

Proteon, Inc., "ProNET Model p3280 ProNET-80 Counter-Rotating Ring Fiberoptic Interface Installation and Operation Guide", Revision A, Proteon, Inc., Westborough, MA, June 1988.

Heurikon Corporation, "Heurikon HK68/V20 (Including HK68/V2F and HK68V2FA) User's Manual", Heurikon Corporation, Madison, WI, May, 1988.

Heurikon Corporation, "Heurikon Hbug Monitor User's Manual", Revision D, Heurikon Corporation, Madison, WI, Oct., 1987.

IEEE Standards Board, "An American National Standard IEEE Standards for Local Area Networks: Token Ring Access Method and Physical Layer Specifications", 1985.

IEEE Standards Board, "An American National Standard IEEE Standards for Local Area Networks: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications", 1985.

IEEE Standards Board, "An American National Standard IEEE Standards for Local Area Networks: Logical Link Control", 1984.

IEEE Standards Board, "An American National Standard IEEE Standards for Local Area Networks: Token Passing Bus Access Method and Physical Layer Specifications", 1984

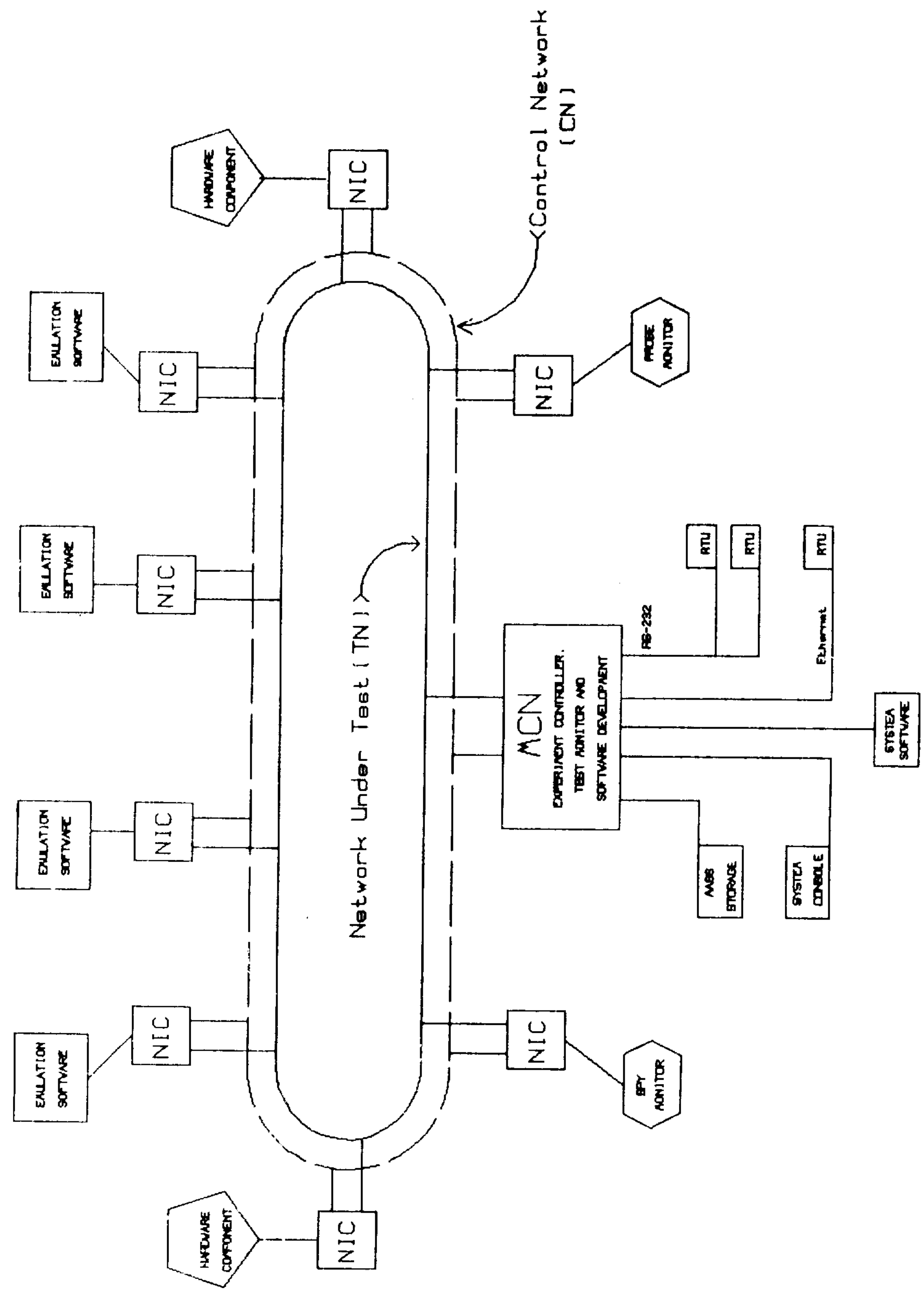


Figure 1: The Marshall Avionics System Testbed

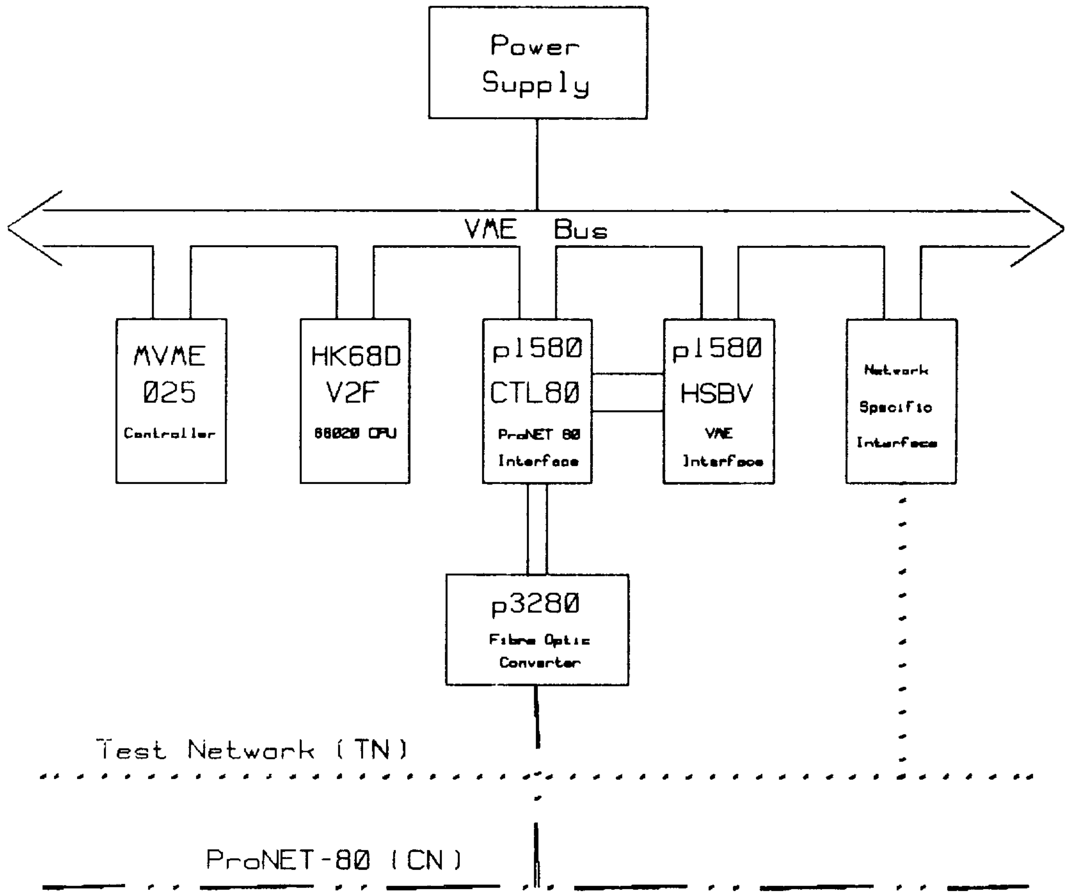


Figure 2: Node Interface Controller Hardware

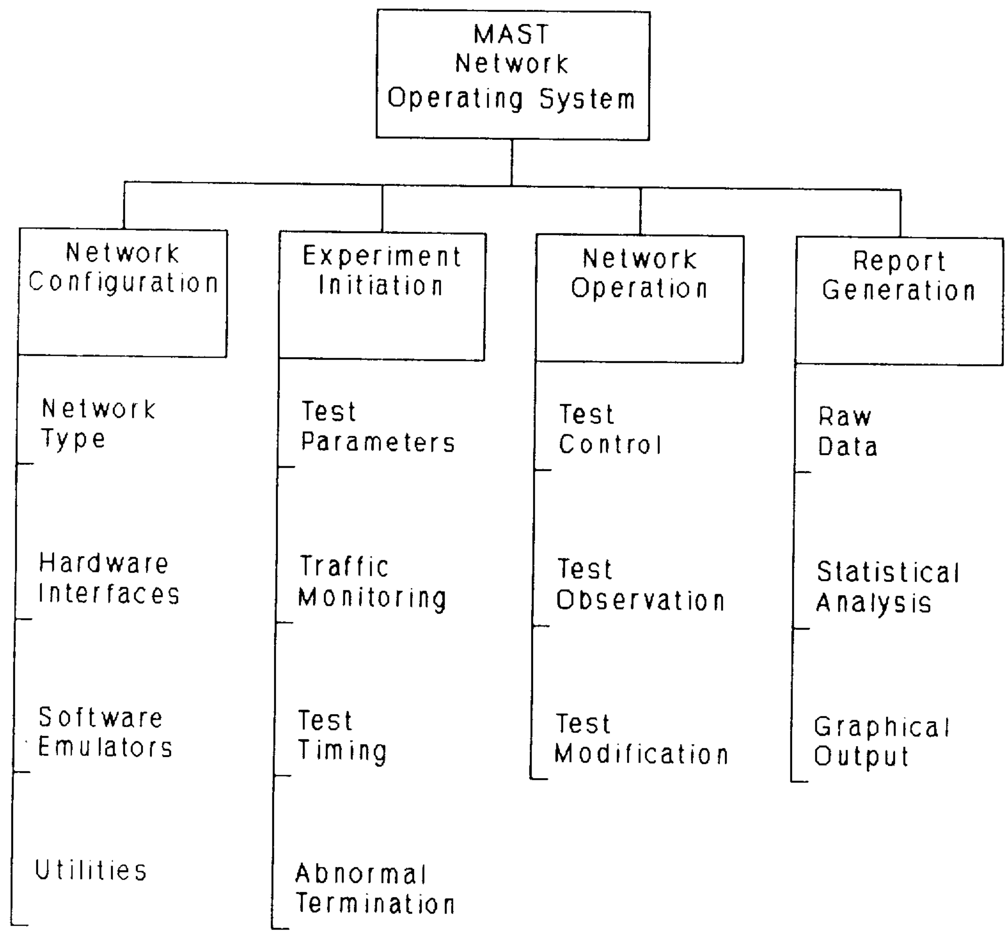


Figure 3: Mast Network Operating System Software Overview

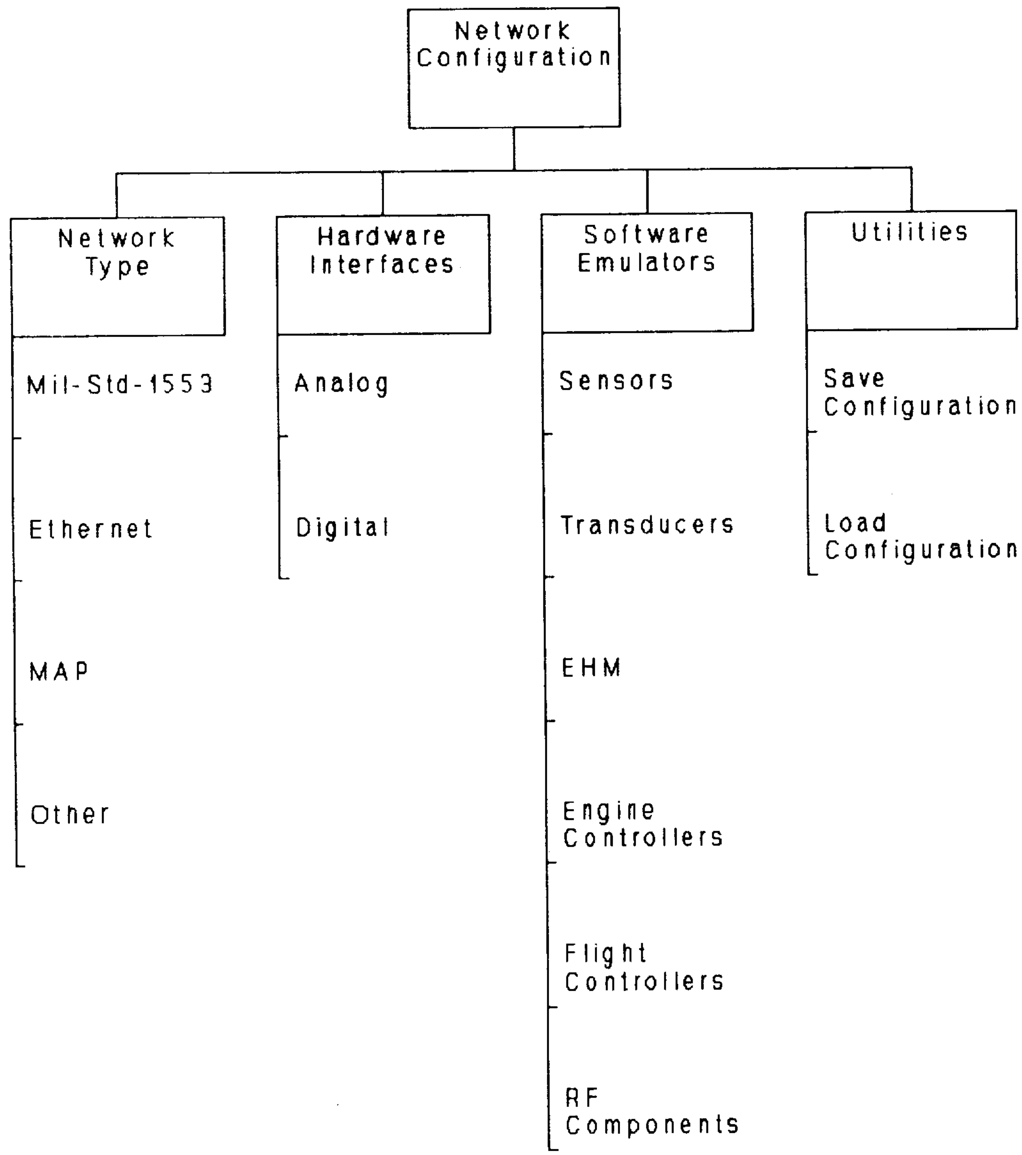


Figure 4: Test Configuration Software

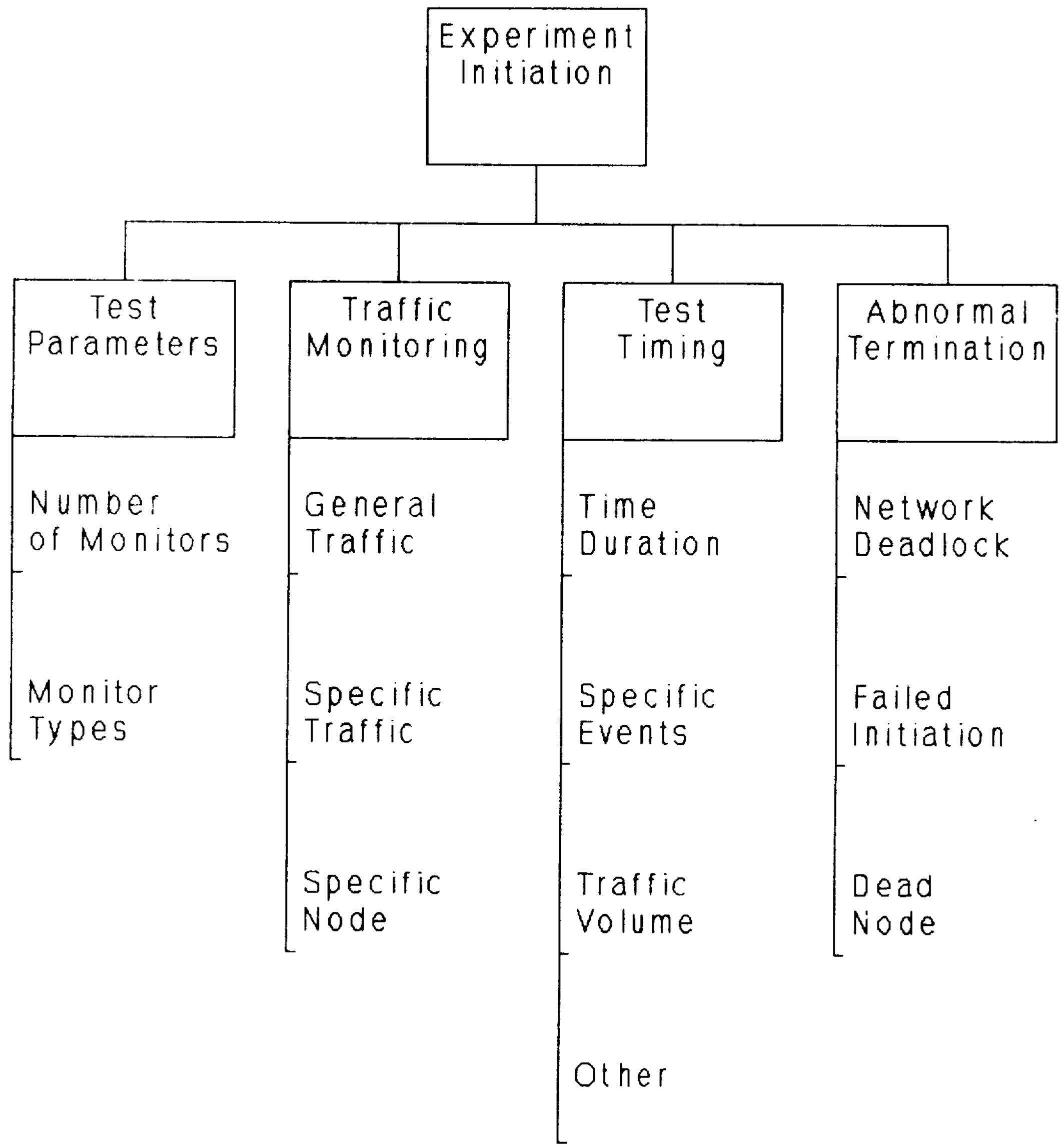


Figure 5: Test Initiation Software

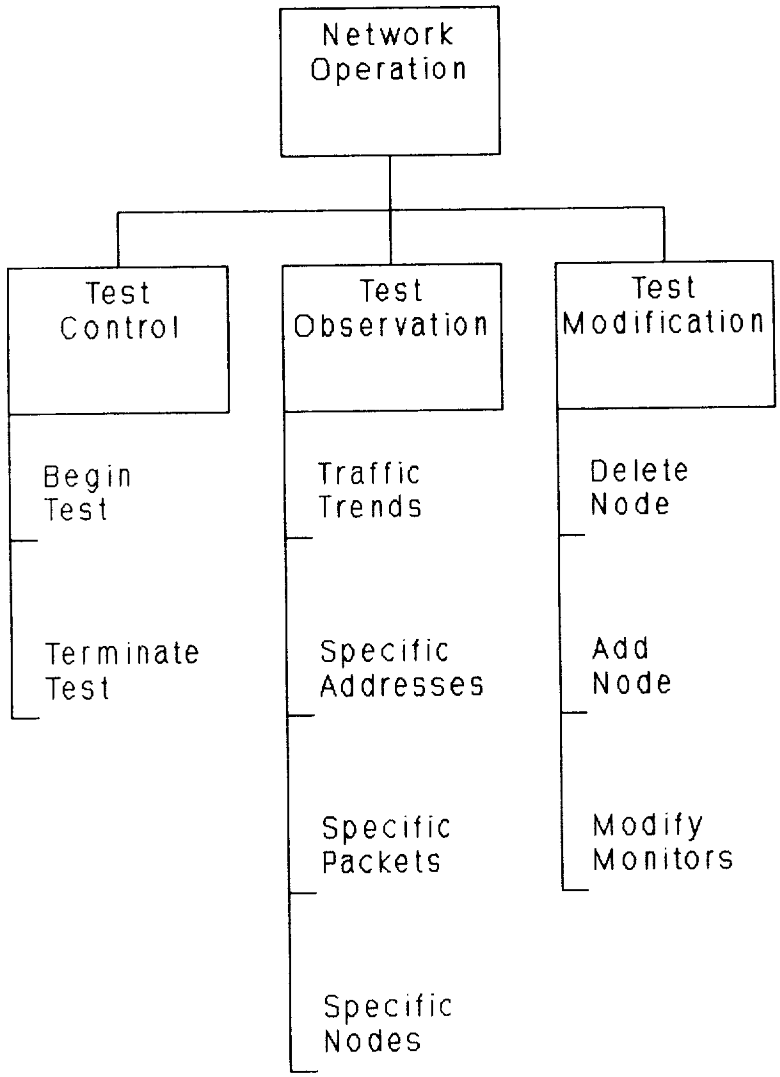


Figure 6: Network Operation Software

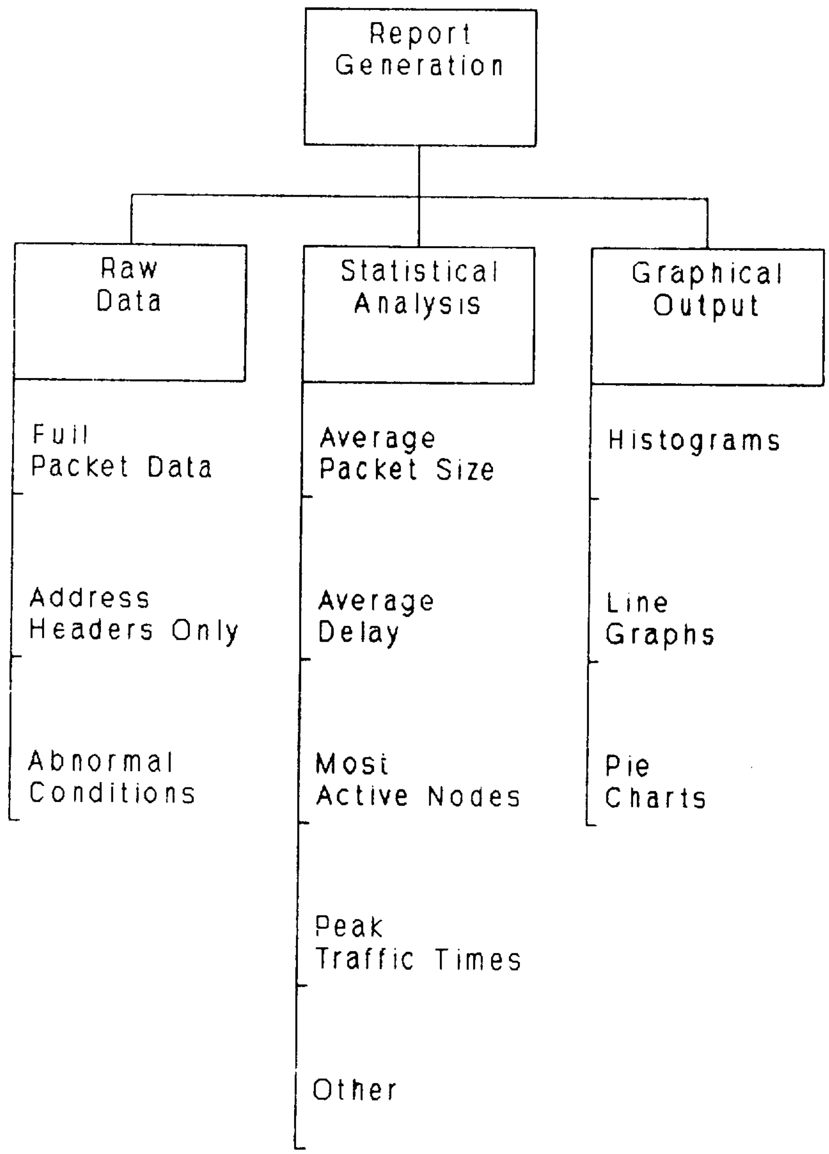
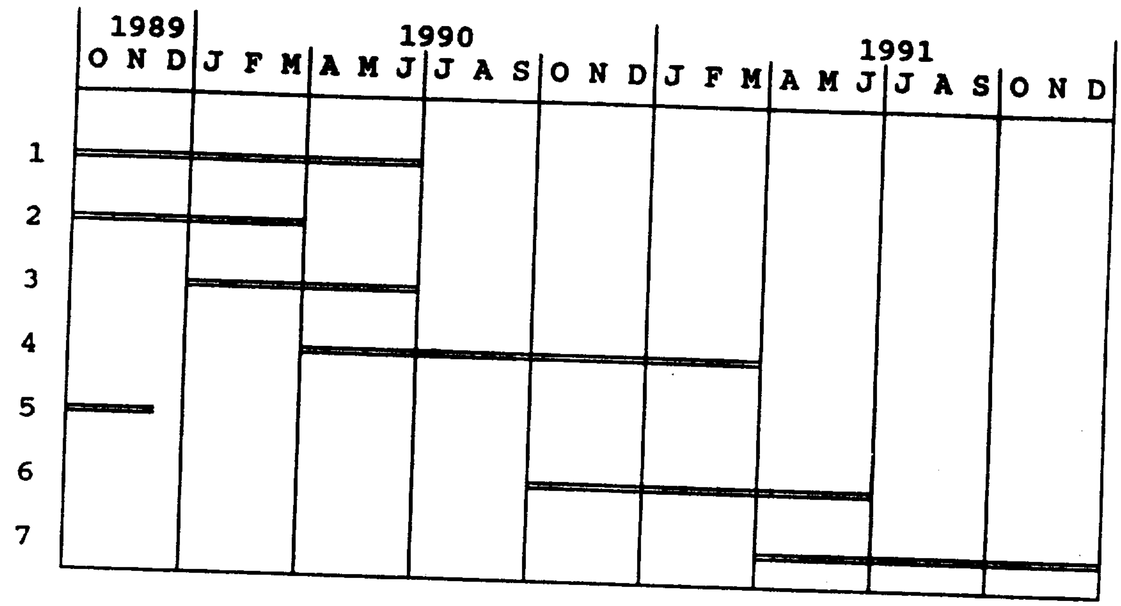


Figure 7: Network Test Report Generation



Task Definitions:

1. Write specifications for MNOS software
2. Procure[1], install and test ProNET software
3. Procure, install and test other network software
4. Procure, install and test MNOS software
5. Assemble three node system hardware
6. Assemble ten node system
7. Assemble full system in permanent location

Figure 8: Tentative MAST Development Timetable

---

[1]Commercial software may not be available that meets specifications. Software may need to be written in-house or obtained through contract.