NBS TECHNICAL NOTE 912

U.S. DEPARTMENT OF COMMERCE / National Bureau of Standards

The Network Measurement Machine—

A Data Collection Device for Measuring the Performance and Utilization of Computer Networks

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] was established by an act of Congress March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau consists of the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Institute for Computer Sciences and Technology, and the Office for Information Programs.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of the Office of Measurement Services, the Office of Radiation Measurement and the following Center and divisions:

Applied Mathematics — Electricity — Mechanics — Heat — Optical Physics — Center for Radiation Research: Nuclear Sciences; Applied Radiation — Laboratory Astrophysics[2] — Cryogenics[2] — Electromagnetics[2] — Time and Frequency[2].

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement, standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; and develops, produces, and distributes standard reference materials. The Institute consists of the Office of Standard Reference Materials, the Office of Air and Water Measurement, and the following divisions:

Analytical Chemistry — Polymers — Metallurgy — Inorganic Materials — Reactor Radiation — Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services to promote the use of available technology and to facilitate technological innovation in industry and Government; cooperates with public and private organizations leading to the development of technological standards (including mandatory safety standards), codes and methods of test; and provides technical advice and services to Government agencies upon request. The Institute consists of the following divisions and Centers:

Standards Application and Analysis — Electronic Technology — Center for Consumer Product Technology: Product Systems Analysis; Product Engineering — Center for Building Technology: Structures, Materials, and Life Safety; Building Environment; Technical Evaluation and Application — Center for Fire Research: Fire Science; Fire Safety Engineering.

THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in improving cost effectiveness in the conduct of their programs through the selection, acquisition, and effective utilization of automatic data processing equipment; and serves as the principal focus within the executive branch for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Institute consists of the following divisions:

Computer Services — Systems and Software — Computer Systems Engineering — Information Technology.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS and other agencies of the Federal Government; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System; provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:

Office of Standard Reference Data — Office of Information Activities — Office of Technical Publications — Library — Office of International Relations — Office of International Standards.

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.

[2] Located at Boulder, Colorado 80302.

The Network Measurement Machine— A Data Collection Device for Measuring the Performance and Utilization of Computer Networks

Robert Rosenthal

Don E. Rippy

Helen M. Wood

Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

U.S. DEPARTMENT OF COMMERCE, Elliot L. Richardson, *Secretary*

James A. Baker, III, *Under Secretary*

Dr. Betsy Ancker-Johnson, *Assistant Secretary for Science and Technology*

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Acting Director*

Issued April 1976

National Bureau of Standards Technical Note 912

Nat. Bur. Stand. (U.S.), Tech. Note 912, 91 pages (Apr. 1976)

CODEN: NBTNAE

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1976

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402
(Order by SD Catalog No. C13.46:912). Price $2.00. (Add 25 percent additional for other than U.S. mailing).

CONTENTS

FIGURES

TABLES

DISCLAIMER

Certain commercial products are identified in this technical note in order to adequately specify the Network Measurement Machine. In no case does such identification imply recommendation or endorsment by the National Bureau of Standards, nor does it imply that the products identified are the best available for the purposes described.

THE NETWORK MEASUREMENT MACHINE -- A DATA COLLECTION DEVICE FOR MEASURING THE PERFORMANCE AND UTILIZATION OF COMPUTER NETWORKS

Robert Rosenthal
Don E. Rippy
Helen M. Wood

ABSTRACT

The Network Measurement Machine (NMM) is a device used to acquire data for the performance measurement of computer network systems and services. By focusing on the service delivered to network customers at their terminals, measurements can be made that are directly relevant to user needs and to management concerns.

This technical report presents the details of a data acquisition device. The device is a minicomputer-based system that employs regular (off-the-shelf) and special purpose hardware under the control of a specially written software acquisition system. The technical aspects of inserting the NMM into the data communications portion of a computer network are discussed. The detailed nature of both the hardware and the software data acquisition system is presented with a discussion of the important design decision trade-offs.

Keywords: Computer networks; data acquisition; man-machine interaction; minicomputer; network measurment; performance measurement

INTRODUCTION

More and more people are making use of interactive computer systems through terminals that operate in a conversational mode. This type of interactive computer access is characteristic of most current computer networks.

As discussed here, a computer network consists of one or more computers, one or more human users employing an interactive terminal, and the communications facilities which interconnect them [BLA 74]. The user of a computer network is a customer of computer services. The communications portion of the network is only the delivery mechanism for the service. However, as a delivery mechanism, the communications facility may have a significant effect on the service received by the terminal user.

Recognizing the lack of a methodology for measuring the service delivered by an interactive computer network to the user at his terminal, the NBS Institute for Computer Sciences and Technology has developed a system for measuring the performance of computer networks and the services they render. The system consists of a data acquisition device called the Network Measurement Machine and a Data Analysis Package (DAP) for generating reports about the quality of network service delivered to interactive terminal users as well as a characterization of user demands and network communication facility utilization. This total system -- the acquisition system and the DAP -- is called the Network Measurement System and represents the implementation of a new approach to the performance measurement and evaluation of computer networks and the services they render. Abrams and Cotton introduced "The Service Concept Applied to Computer Networks" [ABR 75] which includes background information necessary for understanding the need for and use of the Network Measurement System.

The Network Measurement Machine (NMM) is the data acquisition device used to acquire data for the Network Measurement System. Implemented on a mini-computer, the NMM employs regular and special purpose hardware controlled by a specially written software acquisition system. The regular hardware includes the processor (a Digital Equipment Corporation PDP-11/20), an operator's console, disk and magnetic tape storage, two programmable clocks, and data communications interfaces. Special purpose hardware is employed to connect the NMM to the network that is to be measured. This hardware includes an automatic calling unit and line selector (ACU/LS) for computer-controlled origination and answering of data calls, and a specially designed communications line interconnection device called a "data probe".

The special software acquisition system is a real-time interrupt-driven scheduler incorporating various drivers and handlers for the standard and special purpose peripherals attached to the NMM. The software is written in assembly language and uses the manufacturer's regular software -- editor, assembler, and linking loader -- in its preparation.

Programming is modular, with extensive use of semaphores for task synchronization. A command interpreter serving the operator's console makes it possible to view the status of the NMM and to view the status of selected ongoing communication.

This technical note presents the details of the NMM -- the hardware and software components. The technical aspects of inserting the NMM into the data communications facility are discussed. The detailed nature of the data acquisition system is presented with a discussion of important design decision trade-offs.

Why Measure

An increasingly large number of computer terminal users are applications oriented. Accordingly, their only interest in the computer is in terms of the service it can provide them. It follows that when such customers solicit computer services, they are highly interested in the level of computer service delivered to the terminal. Thus one of the main purposes of measurement is to aid in the evaluation of service provided to the terminal user. For a more comprehensive treatment of the motivations for measurement see Abrams and Cotton [ABR 75].

What to Measure

In the data communications area, the basic events to be observed are the occurrences of bits on a link in some time sequence. With proper synchronization of the measurement instrument, appropriate clusters of bits may be recognized and recorded as characters. Along with each character can be recorded its time of occurrence (time-tag) and, if the link is bidirectional, the source of the character.

A record of each character with its source and time-tag is all that is necessary to reconstruct the physical events being observed, namely the conversation or "dialogue". Other less important communication events can be measured and time-tagged. These include the ring indicator and carrier detect control signals from the modem that indicate the start and finish of the interactive dialogue and these measured events may be used when reconstructing the physical events being observed.

Where to Measure

Although measurements may be taken anywhere in the computer network, certain points are favored because of their convenience or their logical relationship to the measurement objectives. When the measurement objective is to determine the quantity or quality of service rendered to the user at his terminal, the best point of measurement is at the terminal itself, or at least at the terminal's interface to the network. When the objective is to determine the demands placed on a network host computer and its responses to those demands, the best point of measurement is in the interface between the computer and the network. When the objective is to determine the effect of the network on demands and service, both of the preceding points of measurement may need to be employed, preferably together and simultaneously.

Unfortunately, practical limitations frequently interfere with the selection of a point of measurement. The (non) portability of the measurement device exerts a strong influence on the point of measurement, as does the ease of attachment to the system being measured. To compensate for the (non) portability of the measurement device, a specially designed remote data probe is employed. This remote data probe sends data back to the NMM from the site of the computer system being measured.

How to Measure

When the point of measurement is at the interface between a host computer and the data communications element of the computer network, measurement could be performed within the computer itself. Certainly all the information needed is available. However, to collect this information in the form desired would require additional software (and possibly hardware) in the host computer [BOI 74]. This approach especially is undesirable because it perturbs the system being measured.

When the point of measurement is elsewhere, a special measurement device must be provided. Once developed, such a device could also be used at the interface between the computer and the network. This approach has the appeal of generality as well as overcoming the previous objection to perturbing the system being measured.

System Overview

This minicomputer-based data acquisition system is suited for use at any of the external measurement points that have been discussed. The data are not analyzed as they are collected; rather, they are stored on magnetic tape and processed later on a large scale computer that executes the Data Analysis Programs (DAP).

These analysis programs accept as input the data recorded by the NMM and produced as output -- after appropriate structuring and analysis -- measures of the service delivered to the user. The design and use of these analysis programs is discussed by Watkins and Abrams [WAT 75]. In a companion paper, Abrams discusses the measurement of interactive service using the NMM during the procurement of network systems and services [ABR 74].

This report describes only the data acquisition portion of the Network Measurement System -- the Network Measurement Machine. When connected to a data communications facility, the NMM collects, characterizes, and records events on the facility. The physical circuits that make up the communications facility can be voice grade lines or wideband circuits owned or leased from common carriers, dial-up circuits, or hard wired private facilities. The NMM connection point to these circuits is always at an EIA RS-232 [EIA 69] standard interface.

Attaching the NMM to these circuits requires specially designed hardware including a data probe and an Automatic Calling Unit and Line Selector ACU/LS. To understand these two devices, and the role they play, it is necessary to present a basic description of the RS-232 interface as it relates to the interactive terminal environment. The section on DATA COLLECTION presents a comprehensive and detailed description of this communications environment.

In general, a user connects to the NMM through a data probe which then completes the connection to the network host computer. This procedure is refered to as the Initial Connection Procedure or ICP. The data probes facilitate this connection procedure by switching the user terminal data paths (under NMM program control) from the NMM to the service computer while still maintaining a receive data path from both. Conceptually, this forms a "T-connection" between a user terminal, a computer, and the NMM. Modifications to this procedure are possible and are discussed in the section on ICP.

The data probe physically connects to the mini-computer through communications interfaces. By design, the probe routes several of the RS232 control leads away from the NMM and into the network host computer. As a convenience feature, a special "mode" switch has been provided that logically removes these alternate routings. This feature is used when the minicomputer is supporting other tasks in the experimental computer facility.

DATA COLLECTION

The hardware system required for this specialized data acquisition process consists of a minicomputer (DEC PDP-11/20) employing both standard and specially designed communications peripherals and related equipment. This section presents a technical discussion of the design considerations, hardware realizations, and operating environment of the major hardware components of the system peripheral to the minicomputer. Also presented are examples of interconnection methods used in the data acquisition process.

Interactive Environment

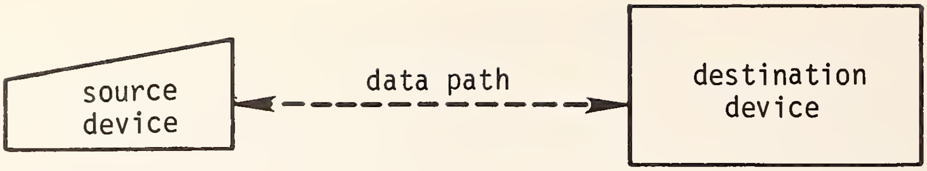
An interactive communications environment can be depicted as shown in Figure 1a. A source device (usually a terminal) is connected to a destination device (usually a computer) by an electrical data path which at some point conforms to the RS-232 interface specification (1). The total data path is usually complex, incorporating such communications facilities as local hardwired digital transmission lines, modems for digital to analog signal conversion of different types, and common carrier public or leased telephone lines of varying quality and speed. This data path supports binary, bit serial, and full or half duplex data transmission. The NMM currently measures only character asynchronous transmission. Measurement techniques for character synchronous and bit synchronous transmission disiplines are presently being investigated.

Communication Interconnection

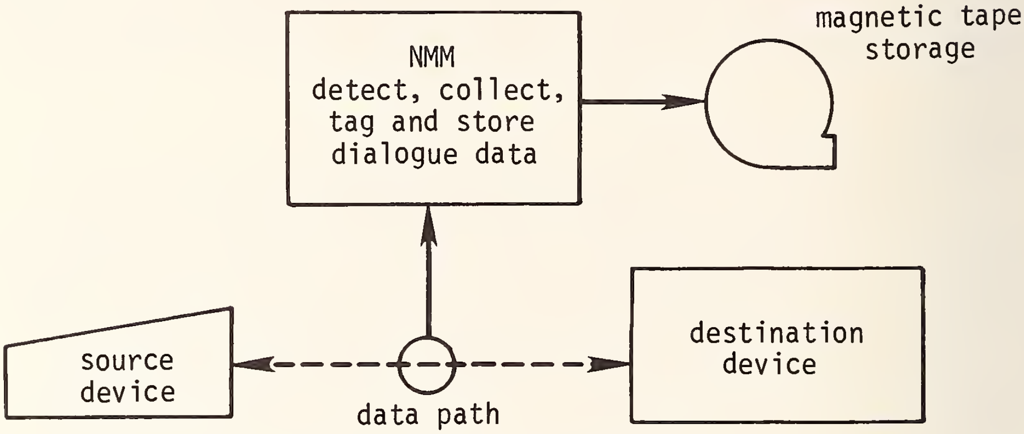
Since it is the transmission of data along a contiguous data path to be measured, it is sufficient to select one convenient point along that path for the purpose of inserting the NMM. Interconnections along the data path which conform to the EIA RS-232 interface standards for digital data communications provide relatively convenient and uniform sets of electrical data, control and status signals which are compatible with available terminals, modems, and computer interfaces. The RS-232 interface standard specifies the electrical, functional, and mechanical parameters of interconnection. Insertion of the NMM into a data path is designed around these specifications.

---

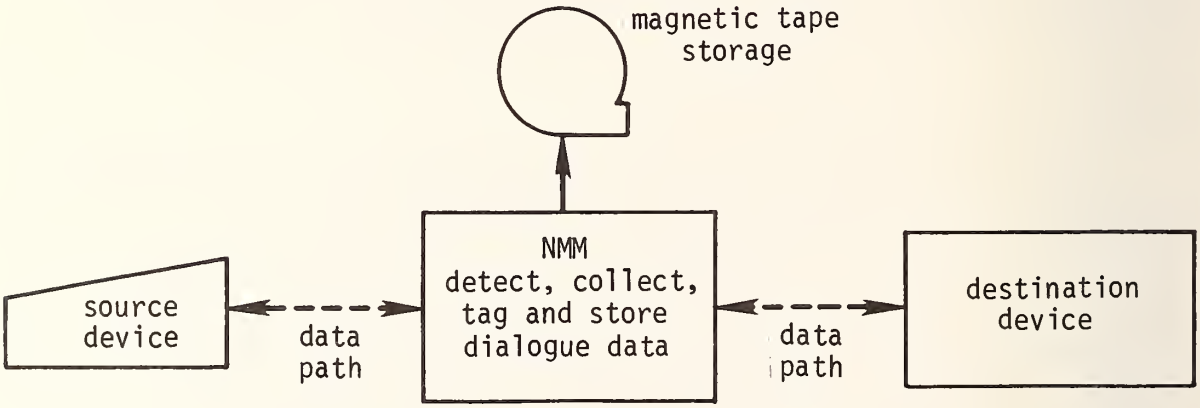
(1) For the purposes of this report, the data "source" is the originator of the data dialogue. The data "destination" is the intended responder to the data source.



a. The Interactive Environment



b. Transparent Data Collection



c. Store-and-Forward Data Collection

Figure 1

Generalized Concepts

Data interchange hardware is classified as Data Terminal Equipment (DTE) and Data Communications Equipment (DCE) as defined within the RS-232 specification. These may be loosely referred to as "terminals" and "modems," respectively. The term "modem" refers to that class of DCE used to interface terminal equipment to carrier (public or private) communications facilities. Table 1 lists a basic subset of the data exchange signals defined by the RS-232 standard. The vast majority of all asynchronous data communications devices utilize this group or a subset for their operation. By judiciously swapping signal lines (e.g., Transmit Data (XD) for Receive Data (RD)), terminal devices can be made to appear as modem devices and vice versa. This capability to exchange the function of a terminal or modem device to its opposite is fundamental to the design and operation requirements of the specialized NMM interconnection hardware described in the following sections.

PIN NO.	SIGNAL MNEMONIC	SIGNAL DEFINITION	SIGNAL SOURCE
2	XD	transmitted data	terminal
3	RD	received data	modem
4	RTS	request to send	terminal
5	CTS	clear to send	modem
6	DSR	data set ready	modem
7	SG	signal ground	-----
8	DCD	data carrier detect	modem
20	DTR	data terminal ready	terminal
22	RI	ring indicator	modem

TABLE 1. PRIMARY RS-232 INTERFACE CONNECTOR PIN ASSIGNMENTS

Computer Interface Operation

Data traffic at the RS-232 interface level can be full duplex, where data traffic occurs independently and simultaneously in both directions. It is, therefore, necessary to incorporate two appropriate computer interfaces operating in a receive-only mode to capture the full duplex data dialogue. Each pair of interfaces and the associated specialized interconnection hardware for inserting the NMM into a selected data path is collectively called a data probe -- symbolically represented in Figure 1b.

The DEC PDP-11/20 minicomputer used incorporates DC11 and DL11 asynchronous serial line interfaces [DEC 73] for monitoring control and status signals, receiving data, and controlling the data path. DC11 and DL11 interfaces have essentially identical operating characteristics with one significant difference: the DC11 incorporates program-selectable data rates, commonly used over voice grade telephone lines, of 10, 13.45, 15, and 30 characters per second (CPS) whereas the DL11 is a fixed speed interface which may be manually adjusted from 10 to 960 CPS. DC11 interfaces are used to probe data paths for which the data rate cannot be predetermined. Speed recognition software in the NMM sets the correct operating speed of DC11 interfaces during the initial connection procedure (ICP). DL11 interfaces are used to probe data paths not incorporating modems. Such data paths do not include the RS-232 Ring signal (Table 1) and, therefore, require a different ICP. Such data paths are also assumed to operate at a known, fixed speed.

Hardware Design Requirements

Three requirements are placed on the NMM data probe by the interactive environment. These include:

1) transparency

2) fail-safe operation, and

3) multi-purpose use of equipment.

This section addresses the interrelated hardware design requirements of the data probe and its implementation.

Transparency

There are two basic methods by which the required data acquisition can be accomplished -- "store-and-forward" and "transparent," (Figure 1). Implementation of a store-and-forward technique involves physically breaking the data path between source and destination devices and directly inserting the NMM, creating two separate data paths directly dependent on the NMM processor, as shown in Figure 1c. For example, source device data is received on an incoming line, stored in computer memory and then, at some future time, "forwarded" (transmitted) to the destination via the appropriate output line. Outgoing data, status, and control signals to both source and destination devices are directly under processor control via the involved communications interface and only indirectly reflect the condition of their counterpart incoming signals. To preserve the logical connection between source and destination device, NMM data communication interfaces must receive and retransmit data, status, and control signals. Software overhead, program and data buffer memory requirements, and processor I/O bandwidth must increase accordingly to accommodate retransmission.

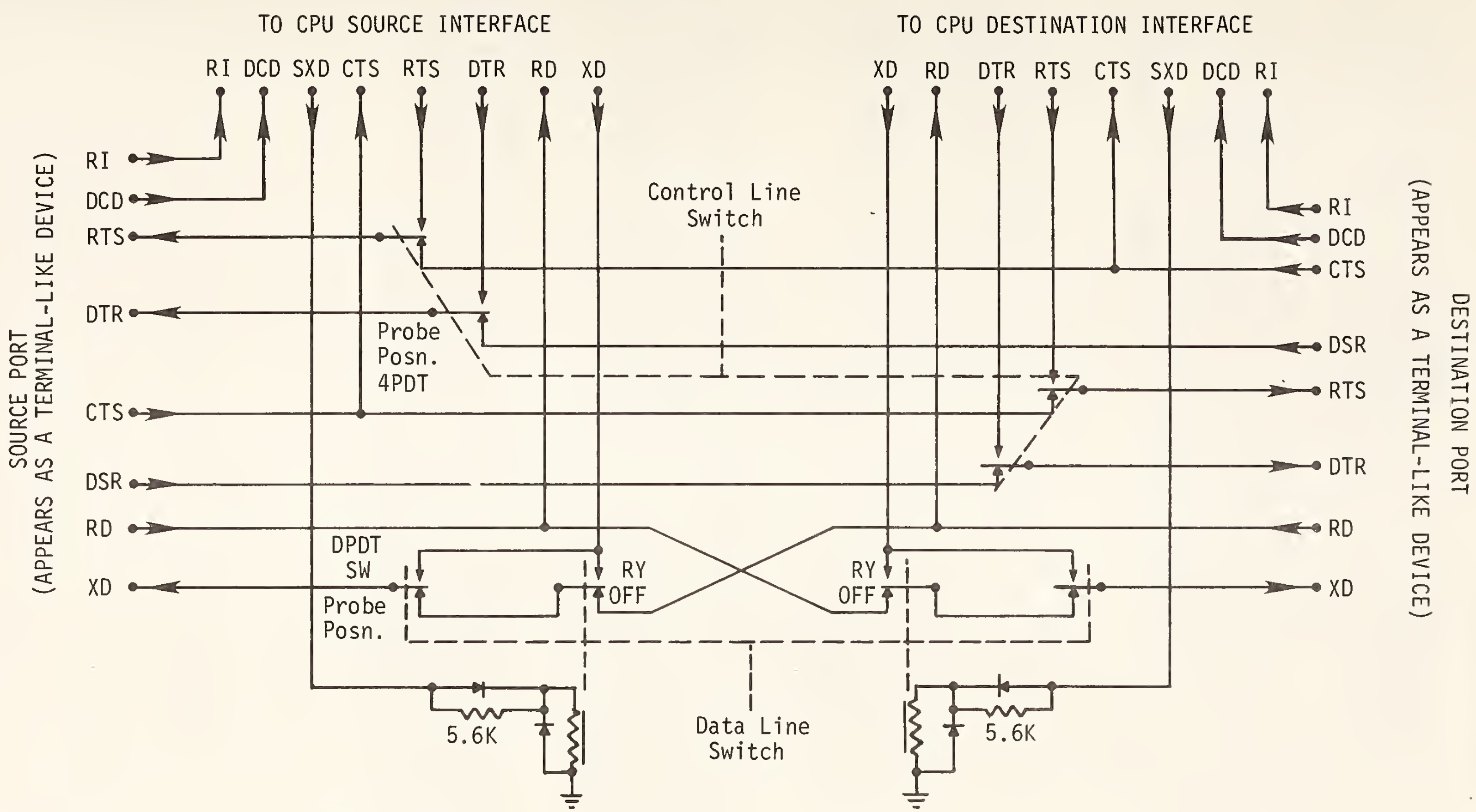
In transparent data acquisition the data probe is "transparent" to the ongoing dialogue with respect to the data, status, and control signals involved. The "hard-wire" connection at this point between source and destination devices is preserved (Figure 1b) and the NMM communication interfaces assume a basically passive, receive-only role relative to the data path. Outgoing interface status and control lines have no influence on the data path, since they are not connected to it. In contrast to the store-and-forward method, source and destination devices communicate directly through an intermediate device (the data probe).

Considering the above factors, significant advantages of the transparent method of data acquisition are apparent. Interference with the data path between source and destination is minimized. Program and buffer space requirements are minimized, reducing memory size and software overhead and I/O bandwidth requirements are reduced. Therefore, we chose to implement this transparent method of data collection.

Fail-Safe Operation

Communication interruption or errors introduced into the data dialogue by the data probe cannot be tolerated. It is therefore essential that the operational characteristics of this specialized interconnection hardware preserve the integrity of the communications path between source and destination devices within the limits of practicability. This will assist in protecting the communications path being probed from inadvertent power failure, computer and/or interface hardware failure, and software crashes. Such capabilities are made possible by the use of the transparent method of data collection and are largely dependent on the means by which the NMM is connected to the data path to be measured.

The specialized interconnection hardware requires no electrical power for its operation other than the RS-232 signals normally provided by the attached DC11 or DL11 interfaces. Data signal routing is handled by relays under direct NMM interface control. These data control relays, when de-energized, maintain the data signal connections between source and destination devices. A schematic of the data probe interconnection hardware showing all control, status, and data signals is presented in Figure 2. The functions of the switches and relays shown in Figure 2 are described in the following section.



Probe Mode:
SXD = 0: XD THRU MONITOR (RY OFF)
SXD = 1: XD FROM INTERF. (RY ENERGIZED)
SG NOT SHOWN

Figure 2

Data Probe Interconnection Hardware Circuit Diagram

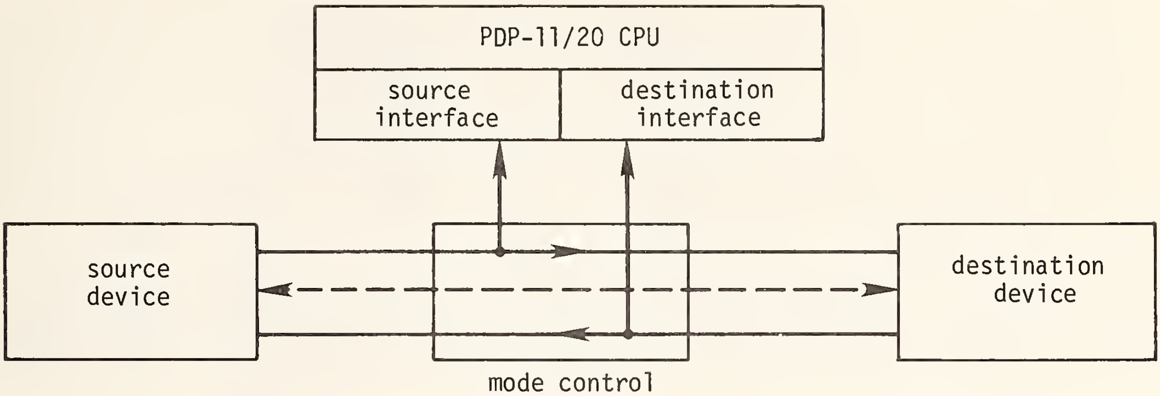
Multipurpose Use of Equipment

When not running as part of the NMM, the PDP11/20 is used for other applications where bit-serial, asynchronous interfaces are required. Therefore, the data probe hardware has been designed to permit multi-purpose use of the interfaces involved. Two manual switches, called control line switches, are used to effect mode control (Figure 2). With these switches, two basic modes of interconnection hardware operation can be configured as shown in Figure 3: data probe mode and interface mode. This permits the related interfaces to function either as a data probe (in pairs) or as separate interfaces for other general purpose I/O uses as required.

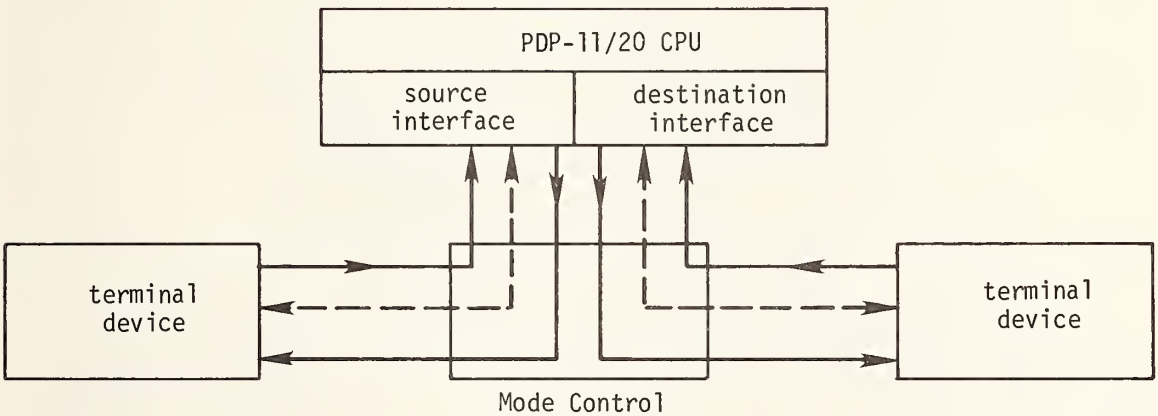
The data line switch selects the Transmit Data (XD) signal routing as shown in Figures 2 and 3. With the data line switch in the interface mode position, the associated data control relay is bypassed and the XD signal originates from the respective computer interface regardless of the state of the associated data control relay. This function is incorporated specifically to prevent the data lines from being affected by changes in the state of the data control relay during interface mode operation.

With the data line switch in the probe mode position, the XD signal source is governed by the data control relay. Secondary Transmit Data (SXD), an otherwise unused RS-232 interface data line, is used for direct computer control of the relay through its respective interface during probe mode operation. Figure 4 illustrates the combination of data paths used by the data probe.

The control line switch selects Request To Send (RTS) and Data Terminal Ready (DTR) routing as shown in Figure 5. Note that the Data Set Ready (DSR) signal line is not routed to the interfaces during interface mode operation because the DC11 and DL11 interfaces have no DSR inputs.



a. Configured as a Transparent Data Probe



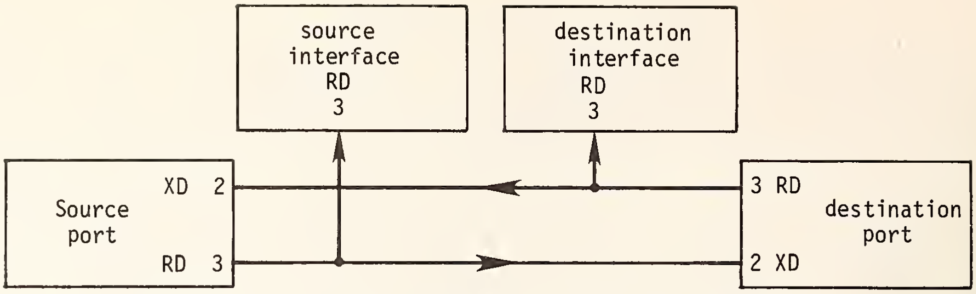
b. Configured for Normal Interface Operation

NOTE: ----: Control and status signal paths

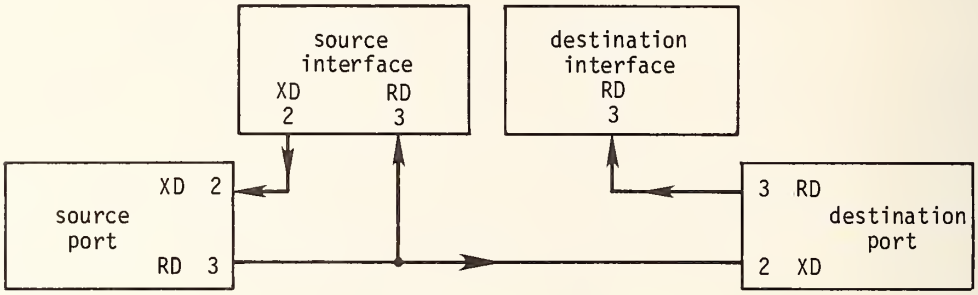
____: Data signal paths

Figure 3

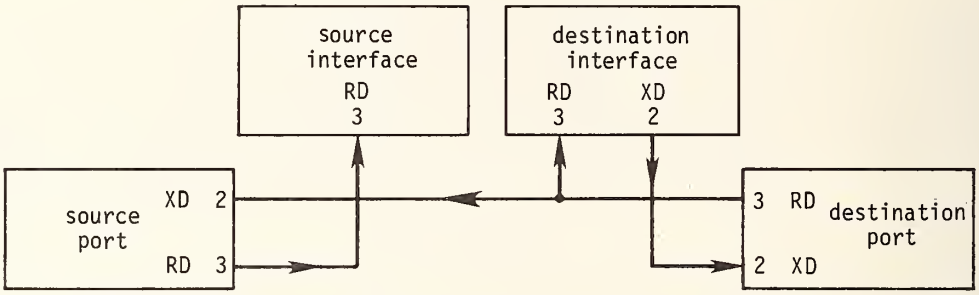
Data Probe Basic Configurations



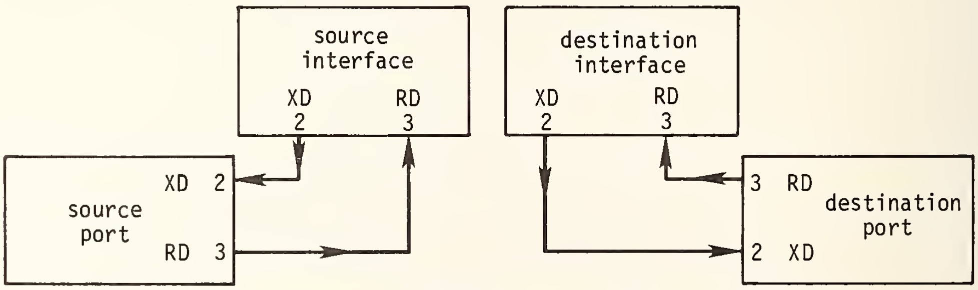
a. Both data control relays de-energized (normal probe configuration, both SXD = 0)



b. Source probe data control relay energized (source interface SXD = 1)



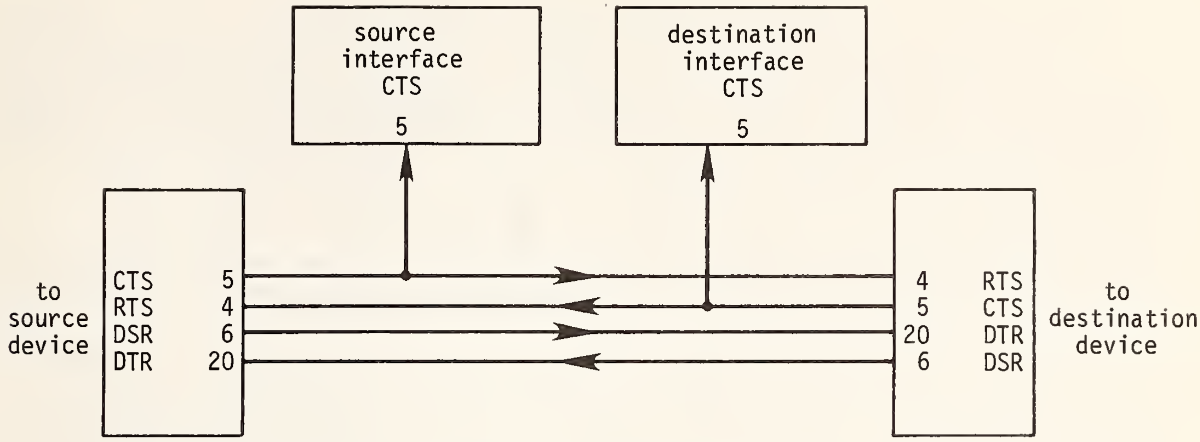
c. Destination probe data control relay energized (dest. interface SXD = 1)



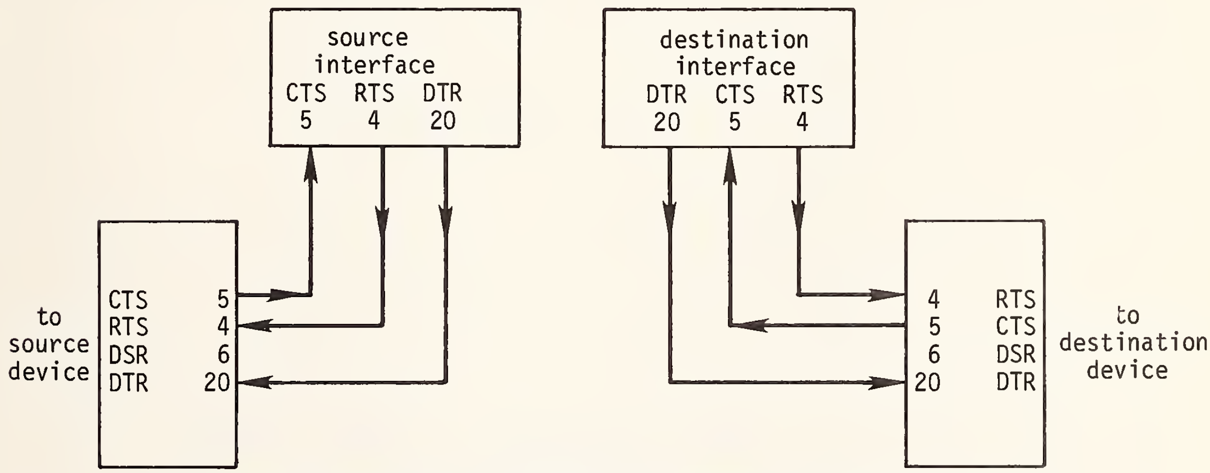
d. Both probe data control relays energized (both interface SXD = 1)

Figure 4

Program Control of Data Path (Probe Mode)



a. PROBE MODE (Data path is controlled by interface SXD)



b. INTERFACE MODE (Data path fixed -- interface SXD state ignored)

NOTE: 1. RI, DCD and data paths omitted for clarity.
2. Numbers refer to RS-232 connector pin designations.
3. DSR is not used by the DEC DC11 and DL11 interfaces.

Figure 5

Manual Mode Control of Data Probe

Hardware Operating Environment

As stated earlier, virtually all modem and terminal equipment designed for bit-serial asynchronous data communication in the non-military environment subscribes to the RS-232 interface specifications. Nevertheless, wide variation exists from device to device with respect to utilization and logical implementation of the individual signals defined by the RS-232 standard. The basic set of signals listed in Table 1 represents the common denominator.

The NMM specialized interconnection hardware has been designed around this basic signal set in response to the wide variety of interconnection configurations anticipated. The data probe concept is sufficiently flexible to permit virtually any type of interconnection between asynchronous terminal and modem devices. Additionally, a special patch cable, wired as shown in Figure 6, permits like devices such as two terminals to interconnect.

The variety of data probe interconnections fall into five categories;

1) Local

2) Dial-in

3) Dial-out

4) Dial-through

5) Remote

All except the local data probe category require the use of at least one modem and related common carrier communications service. Dial-in, dial-out, and dial-through are the variations of the data probe which must provide modem control. Remote data probes constitute a group of special cases which are described, along with the other four general purpose groups of data probes, in the following three sections.

Local Interconnections

Figure 7a outlines the simplest form of data probe operation -- interconnecting source and destination devices which are both in the immediate vicinity of the NMM and which do not require the use of modems. This is frequently the case for an in-house computer system with local terminal access. For example, in the environment where the NMM was developed, local interconnections involve communications terminals (source) located within the same building as the NMM and the ARPA Network TIP [ORN 72] (destination).

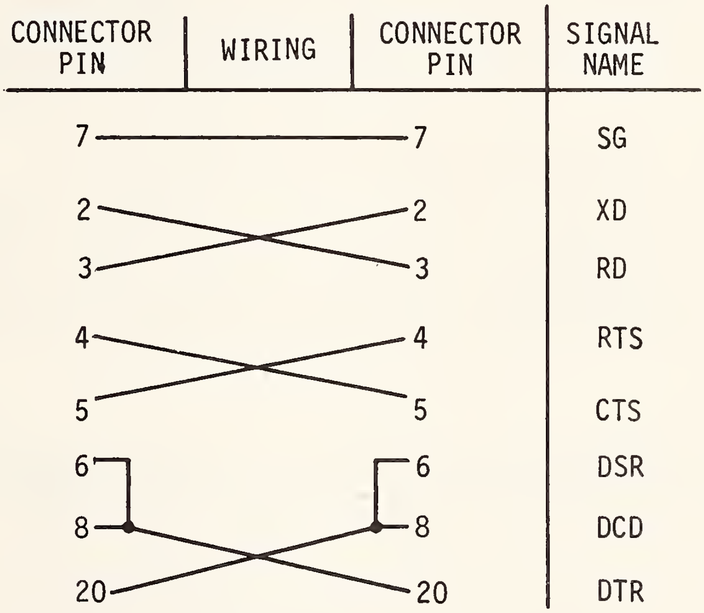


Figure 6

Special Patch Cable Wiring

Within the local working environment, where hard-wire interconnections are the rule, only four RS-232 signal lines are used between the source device and the data probe. These signals consist of Signal Ground (SG), Transmit Data (XD), Receive Data (RD), and Data Terminal Ready (DTR). The range of these local terminal connections typically extends to several hundred feet from the data probe over which CRT terminals normally communicate, at data rates up to 2400 bits per second. Remaining RS-232 control and status signal requirements are satisfied by looping DTR back to DSR, Clear To Send (CTS), and Data Carrier Detect (DCD) as required at the source device. The source device DTR signal then becomes the only control necessary between source, NMM, and destination devices. Data probe interconnections redistribute this signal as appropriate to the CTS, DSR, and DCD inputs of the associated NMM interface and destination device.

Modem Service

The majority of NMM use presently involves dial-in or dial-through data communications. In both cases the immediate hard-wire data probe connection is to one or two modems which are provided either by a common carrier (e.g., Bell 103, 113) or an independent modem manufacturer (e.g., VADIC VA305D).

Dial-in service involves only one modem (logically connected to the source device) and requires only that the modem be of the answer-only type (e.g., Bell 113B) or be in an auto-answer mode (e.g., Bell 103) and that the computer, through the data probe, provide the necessary ring signal detection and call answering procedures. Interface operation of the control and status signal lines is required as shown in Figure 5b. This requires the control line switches (Figure 2) to be in the interface mode position while the data line switch remains in the probe mode position. Required destination device control and status signals are similarly supplied by the data probe destination interface.

Dial-out capability is provided by a communications subsystem consisting of a computer controlled automatic calling unit and line selector (ACU/LS) and 15 addressable originate/answer modems. Contrary to common practice, only one ACU is required to service all 15 modems. The ability of the ACU/LS to uniquely address and control each modem for dialing purposes under program control provides this capability. Each modem can be used to answer an incoming call or, alternatively, to originate a call. The ACU/LS is program controlled by the NMM using a standard PDP-11 interface unit (DR11-A) modified to accommodate interface signals and operational characteristics unique to the ACU/LS.

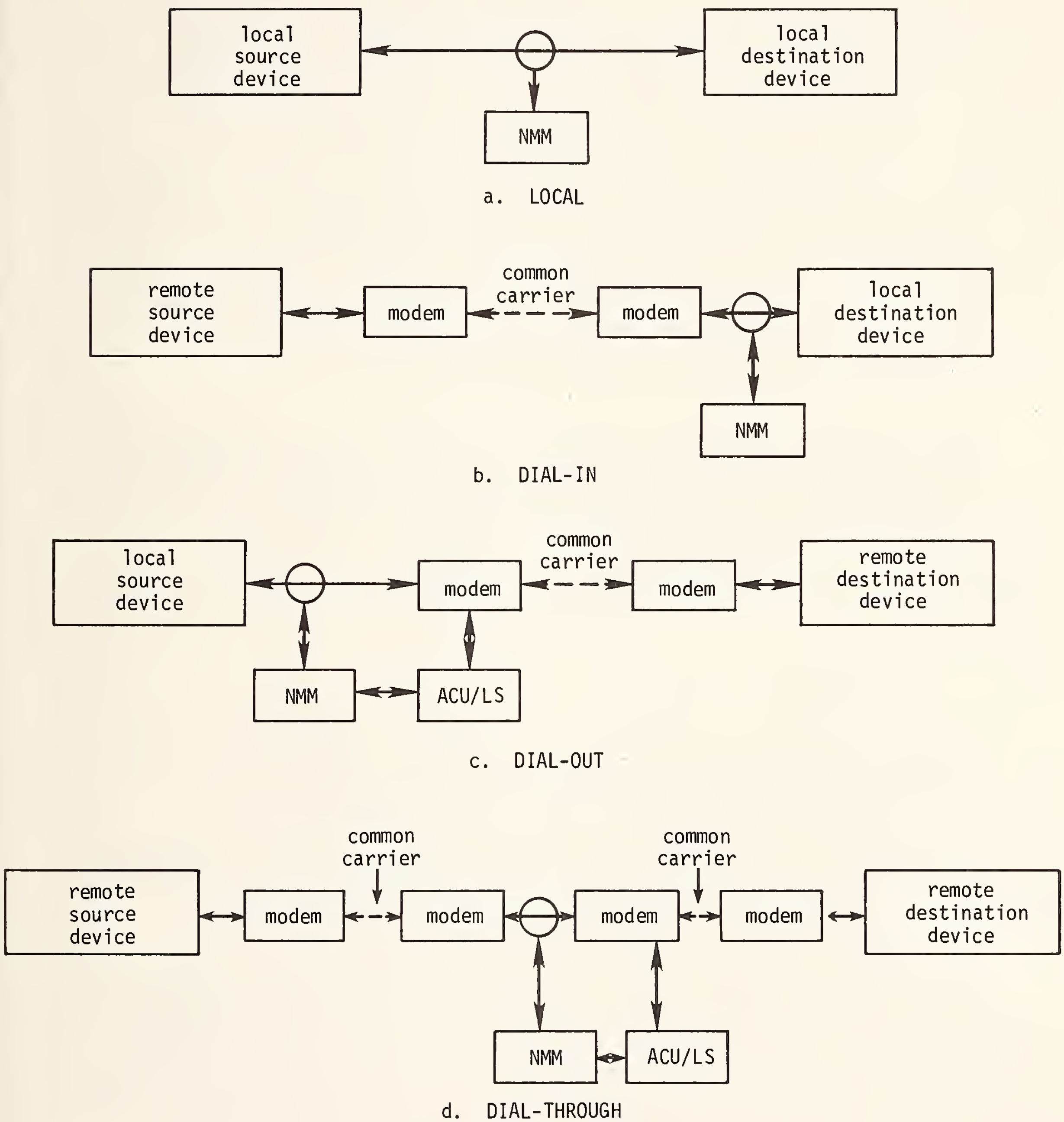


Figure 7

Basic Types of Interconnections

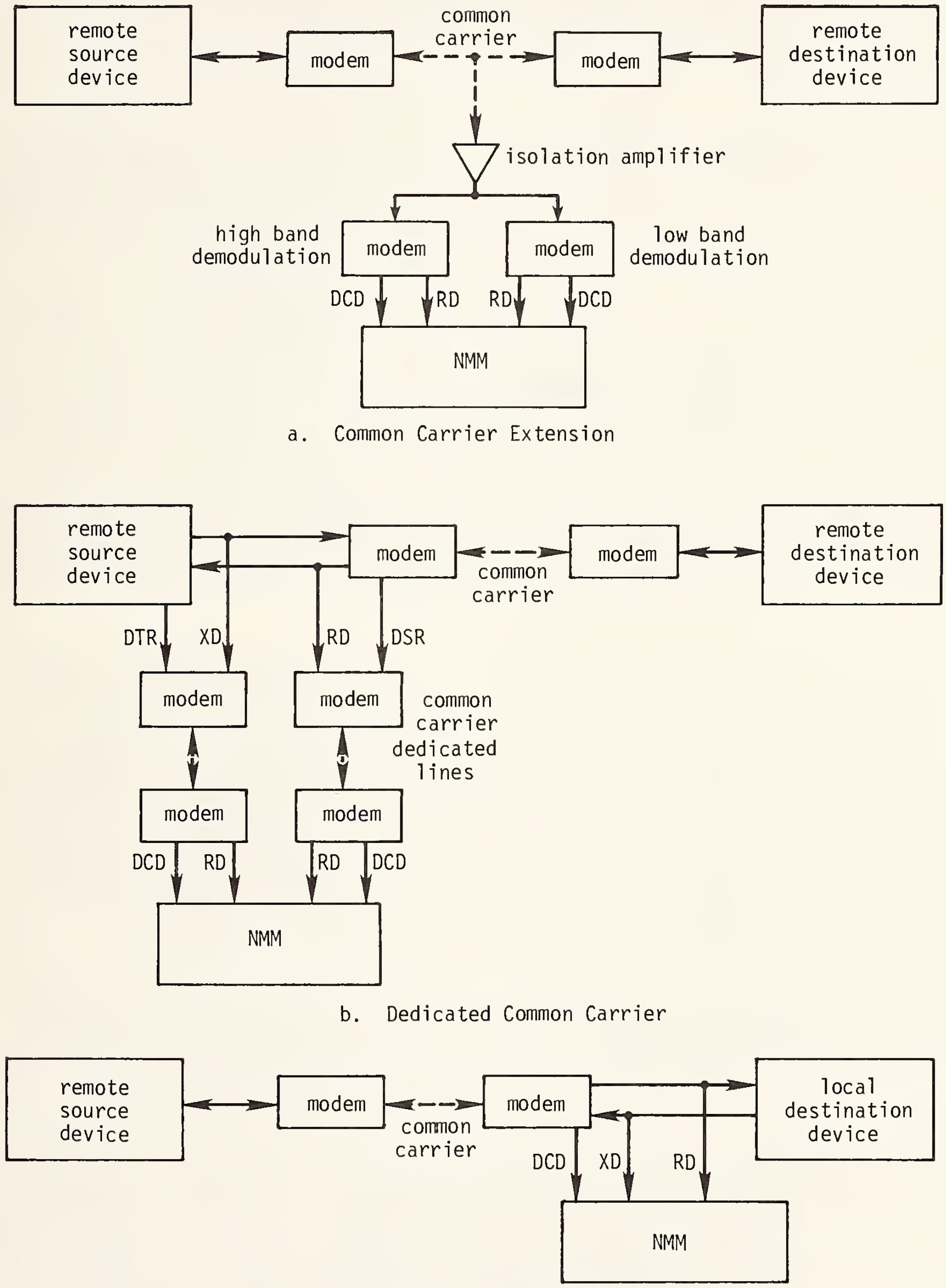
Dial-out is initiated by addressing a modem through the ACU/LS. If the modem is busy (on-line), the driving program is so notified and, if appropriate, another line (modem) is selected and the process is repeated. Otherwise, the ACU/LS gains control of the modem and places it on-line, analogous to picking up the hand set on an ordinary telephone prior to manual dialing. Once the on-line condition is established and communicated back to the driving program, dialing is begun under program control and continues until the ACU/LS receives a unique end-of-number code following the last digit of the telephone number being dialed. At this point dialing is completed and line control is automatically passed to the addressed modem by the ACU/LS. The ACU/LS is then free to address another modem and proceed with another dialing operation. Meanwhile, the driving program monitors the status of the calling modem to determine whether connection has been completed within a predetermined period of time (i.e., the called modem has answered properly and the proper carrier signals are received and are stable) and if not, "hangs up" (disconnect) the modem and, if appropriate, tries the call again.

Dial-through service, Figure 7d, involves two modems, one for answering the incoming call and one for dialing out to the destination device, effectively combining the dial-in and dial-out interconnections of Figure 7b and 7c, respectively. Affected status and control lines of both modems are controlled by the data probe interfaces and the out-going call is handled in the same manner as for dial-out operation.

Remote Data Probes

Several methods of remote connection ("T-connections") may be used for data collection where insertion of the data probe as described above is either impractical, inconvenient, or undesirable. Such T-connections are characterized by having no data, status or control signals passing through the specialized interconnection hardware as depicted in Figure 5a, only to it. Data probes thus arranged operate only in the interface mode.

Three basic types of T-connections currently in use are shown in Figure 8 -- common carrier extension, dedicated common carrier, and local T-connection. There are several variations of the methods shown. For example, the actual point of T-connection in Figure 8b may be between the remote destination device and its associated modem instead of between the remote source device and its modem as shown. Since additional hardware and resultant costs are involved (particularly in the cases of Figures 8a and b), connection is usually made at the most economic and/or convenient point.



a. Common Carrier Extension

b. Dedicated Common Carrier

c. Local T-connection

Figure 8

Remote Data Probe Configurations

Figure 8a represents an on-site telephone line extension connected to a dial-in port of the NBS Univac 1108 service computer located in another building. The extension incorporates an isolating amplifier with appropriately high impedence inputs to prevent any line control or interference by the NMM. This extension is terminated by two VADIC modems operating in receive-only mode, wired to never disconnect (hang-up). One modem is strapped to always demodulate the high band, the other to always demodulate the low band (2).

The received data (RD) and carrier (DCD) signals from each modem are communicated to the data probe in the normal manner with the data probe operating in the interface mode. Data is accepted by the NMM only when the carrier is present in both modems. Because computers generally permit different speed users to dial in on the same line, speed recognition software is used with this arrangement. The conversation is assumed to begin when carrier (DCD) is detected and to end when carrier is lost. Ring Indicator (RI) is ignored for the following reason: If both the 1108 modem and one or both of the associated data probe modems were to attempt to answer the incoming call (i.e., go "on-line" and transmit the high band carrier), the conflicts in connection procedures created at both the modem and controlling device levels would prevent proper establishment of the intended user/1108 connection.

Figure 8b depicts a remote T-connection wherein the communications path to be probed is geographically distant from the NBS facility. This configuration can be used to probe a communications path between the National Library of Medicine's (NLM) MEDLINE medical bibliographic information network and a port on the TYMNET computer network, both of which are in the same room at NLM, approximately 15 miles from the Network Measurement Machine. These network connections are described by McCarn in a short paper on MEDLINE communications [McC 74].

Readily available, off-the-shelf equipment is used, minimizing interconnection requirements. Further, the modems involved (Bell 108) and the two full duplex, dedicated lines used have an additional purpose. They

---

(2) As implemented in Bell 103 compatible modems, Frequency Shift Keying (FSK) employs two frequency bands used in low speed, full duplex data transmission. The low band frequency is modulated by the call originator (modem of the source device) and the high band is modulated by the call responder (modem of the destination device).

provide two independent full duplex connections to the NBS ARPA Network TIP for MEDLINE access by ARPA Network users when not being used in the T-connection configuration. When used in this manner, the ARPA/MEDLINE connections can be probed in the normal manner (Figure 7a).

As in the previous example, the carrier signals (DCD) and the associated received data signals (RD) are connected to the data probe interfaces in the normal manner via the special interconnection hardware. Since the Bell 108 modem cannot detect a ringing signal (and indeed, there is none -- these modems are always "on line"), carrier signals, rather than ring signals, are used to determine the beginning and the end of a data conversation.

The third example, Figure 8c, illustrates a hard-wire T-connection used where it is impractical or undesirable to bring the interconnection between modem and controlling device to the data probe for connection. Figure 8c illustrates a data probe connection to an ARPA Network TIP dial-in modem as one example where this method is used. Here, the ring signal is not used and the carrier signal is distributed to the DCD inputs of both source and destination interfaces of the data probe. Similar to case 8a, speed recognition is required since TIP dial-in ports are not normally constrained to a preselected data rate for dial-in users.

Portable T-connection

An auxiliary unit called a portable T-connection has recently been developed for use in situations where data probe connections as previously described are impractical, such as at a remote computer site incorporating numerous data communications links to be probed. It consists of a transmitting unit (figure 9) and an NMM based receiver (Figure 10) which function together as a remote T-connection as shown in Figure 11. The unit allows an operator at a remote location to select one of four possible communication links and transmit the selected link over common carrier (dial telephone) lines by use of an integral acoustic coupler (Figure 9) to the NMM.

The transmitting unit is housed in an ordinary briefcase to provide full portability. It is light-weight (approximately 8 lbs) and reasonably immune to physical damage from ordinary handling; no incompatibility is expected with present-day airline security policies. External requirements include A.C. power, a portable printer unit for data monitoring (another briefcase-sized device), and a nearby telephone for establishing a data connection to the receiving unit.

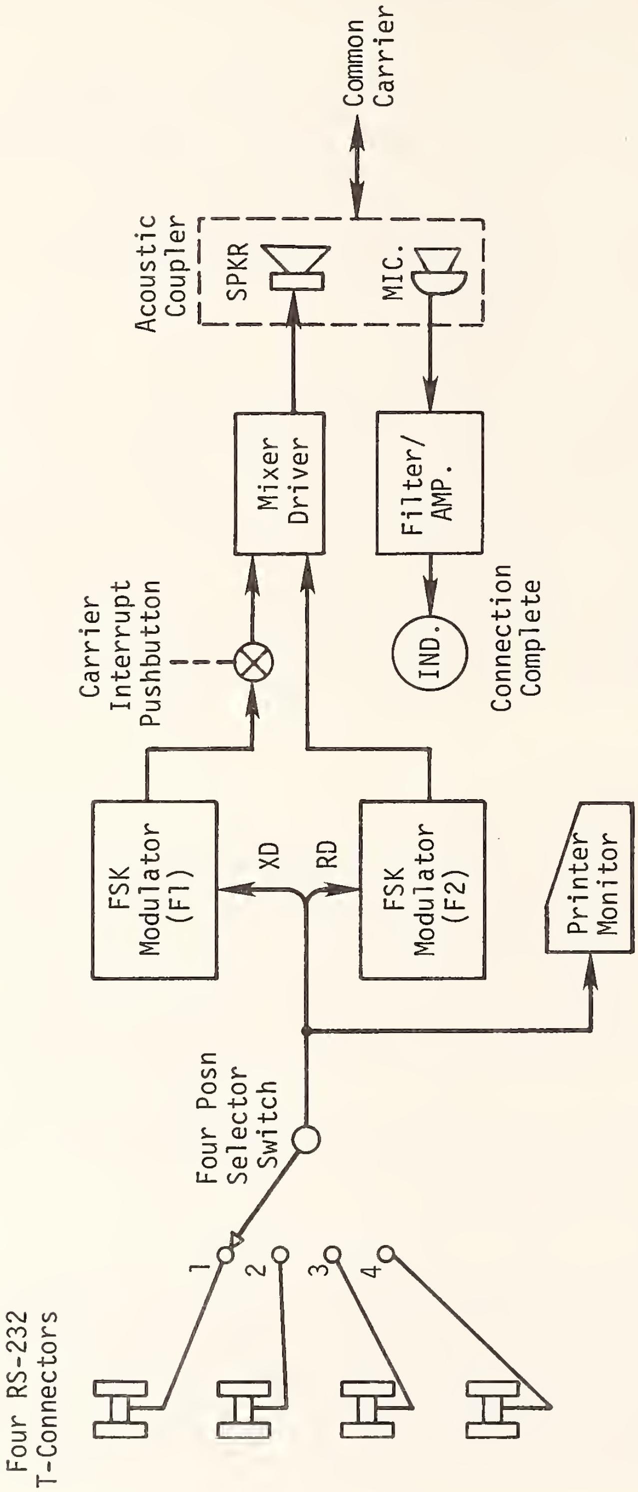


Figure 9.

Portable T-Connection Transmitter

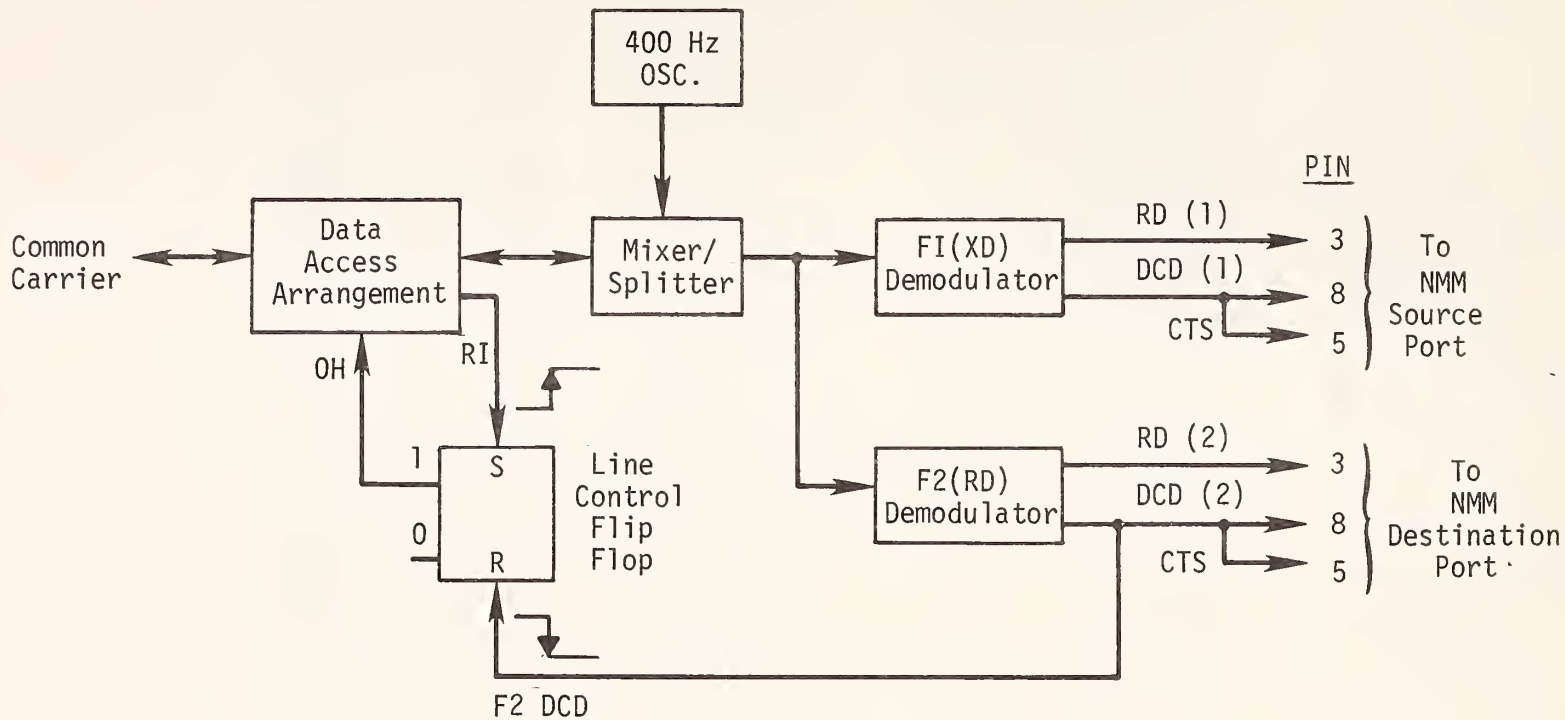


Figure 10.

NMM Receiving Unit For Portable Transmitter

FSK modulation is used to transmit activity on the XD and RD data lines as shown in Figure 9. XD modulates the low band (F1) and RD modulates the high band (F2). These signals are mixed and appropriately amplified to drive an acoustic coupler speaker for use with an ordinary telephone handset. The F1 carrier can be manually interrupted to signal the end of a dialogue. This capability is used when switching from one T-connector to another to inform the NMM that a different link is being measured or when a probed connection is terminated.

Although the data transmission mode used by this unit is dual-simplex, an oscillator within the NMM receiving unit is used to provide a low frequency (400 Hz.) third carrier which is detected by the remote transmitting unit. This in turn drives an indicator notifying the operator of the transmitting unit that the NMM receiver has answered the call and is on-line. The NMM receiving unit demodulators are well filtered and are therefore blind to this third carrier. Also, the transmitting unit indicator circuits are tuned to 400 Hz. and are similarly blind to the FSK signals to prevent data traffic on the line from activating the indicator.

A data access arrangement (DAA) provides line control, D.C. isolation and FSK coupling between the common carrier line and the NMM receiving unit. Modulated signals from the transmitter are demodulated and conveyed to an NMM data probe as shown in Figure 11. To avoid interaction between source and destination ports, the involved data probe is configured for interface mode operation as shown in Figure 3b.

The flip flop shown in Figure 10 provides line control through the DAA as follows: an incomming ring signal is detected by the DAA and used to set the line control flip flop. This, in turn, causes the receiving unit to go on-line through the DAA analogous to picking up the handset of an ordinary telephone. When the transmitting unit operator hangs up the associated telephone, the resultant loss of F2 carrier at the receiving unit resets the line control flip flop. This, in turn, causes the DAA to "hang up" and await another incoming call.

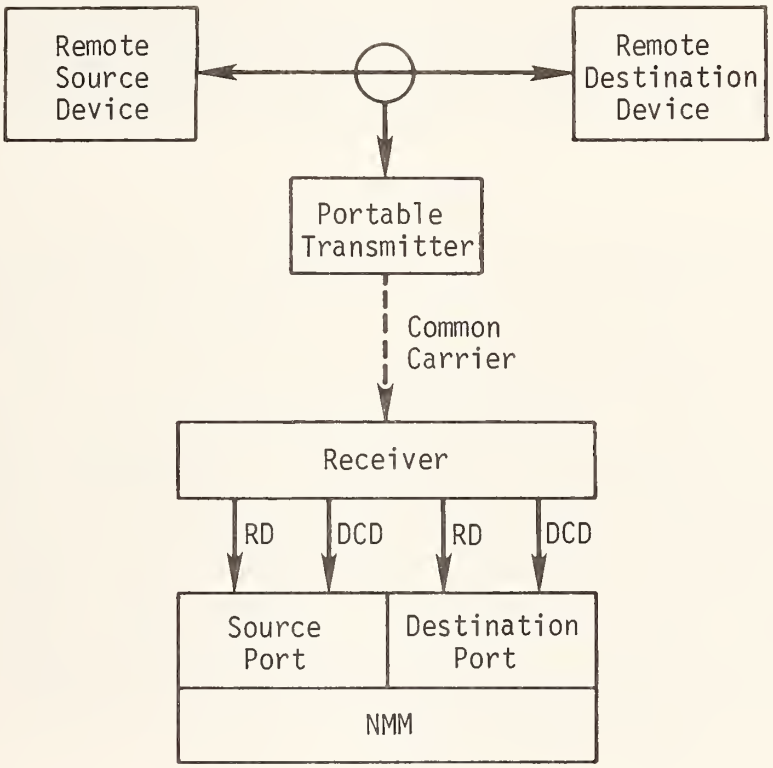


Figure 11.

Protable T-Connection Configuration

NMM SOFTWARE STRUCTURE

The acquisition system requires software to control and manage the resources of the NMM -- the data probes, the data recording device (magnetic tape), and the operator's console. This software is a comprehensive set of program modules that together form a software operating system.

The system is written in the PDP 11 assembly language (MACRO-11) and is supported and maintained on the manufacturer's disc-based operating system (DOS). The DOS text editor, macro assembler and linking loader are used to prepare new versions of the NMM for execution. Once loaded, the acquisition system takes over the machine and overwrites a portion of DOS. This is required to handle real-time interrupts from the data probes and the Automatic Calling Unit and Line Selector (ACU/LS).

Configuration

The minimum hardware configuration needed to support the NMM includes the PDP-11 processor, 8192 words of main memory, two real-time clocks, a secondary storage medium (magnetic tape), an operator's console, and at least two asynchronous interfaces connected to a data probe. The operator's console dynamically presents and logs the data probe interface connection status. As an input device to the NMM, this console is used by the operator to initialize, modify, and terminate the NMM operating system. The current system employs an ACU/LS for the automatic dialing of data calls and the current system utilizes a line printer to monitor any one of several on-going data dialogues. This configuration can be expanded to include many interfaces. The current system utilizes 14 interfaces forming 7 data probes; as the majority of the operating system is re-entrant, only a small amount of core is utilized for tables when adding additional interfaces.

System Requirements

The data acquisition software is a real-time operating system designed to fulfill two requirements:

1) to provide an accurate record of the data traffic and control signals with time tags for later analysis and

2) to provide an efficient method for inserting the data probes into the data communications path.

These two requirements form the basis of the design specification for the Network Measurement Machine.

Software Overview

The NMM software is an interrupt driven operating system incorporating various device drivers and interrupt service routines for the standard and special purpose peripherals attached to the system. Semaphores (described in the section on Data and Control Structures) are used extensively throughout the system for task synchronization and for controlled access to queued data.

To meet the requirements outlined above, the NMM software is organized into four separable sections that are assembled and linked together into an executable operating system. These four sections are:

-- interrupt service routines,

-- scheduled procedures,

-- utility routines

-- global information; messages and buffers.

This operating system will be described by tracing the data flow from the interrupt service routines through the scheduled procedures to the logging tape. Time tagging, an integral part of the data probe interrupt service routine, is described later. This routine satisfies the first requirement -- to accurately time-tag and record the data traffic and control signals for later analysis. The second design requirement (to provide an efficient method for inserting the data probe) is addressed in the section on the ICP.

Data Flow

Data flows through the operating system from three interrupt service routines to the logging device -- an industry-compatible 7-track magnetic tape. When the interrupt service routines place enough data into a buffer, the scheduler starts a procedure responsible for writing the contents of the buffer onto the logging device. The three interrupt service routines that drive the system are:

1) the Data Probe Routine,

2) the Time Tag Clock Routine, and

3) the Time of Day (TOD) Clock Routine.

The Data Probe Interrupt Service Routine (Figure 12a) identifies, time tags, and buffers communication interrupts including received data, carrier transition, ring, and overrun interrupts. Since it is re-entrant, all of the data probes can be serviced by this routine.

The Time Tag Clock Routine (Figure 12b) services the crystal controlled clock used to provide 24 bits of timing data. The clock hardware counts the low order 16 bits in a synchronous counting register that causes an interrupt on overflow. The service routine software increments an 8 bit byte in core to produce the 24 bits. Further, when this byte overflows, an entry is placed in the buffer destined for the logging tape. The clock counts at a 10kHz. rate, providing an interval timer with 100 microsecond resolution.

A second clock, the Time of Day (TOD) Clock, maintains timing functions for both the ACU/LS and the time of day (TOD) software clock which is updated every second (Figure 12c). Whenever a modem that is connected to a data probe senses a transition in the carrier detect signal or ring indicator signal, a timing function is required to control the NMM activity on that line. For instance, whenever the ring indicator makes an off-to-on transition, the NMM will set DTR, causing the modem to answer the phone. A 10 second timer is started which, if allowed to time-out, would reset all of the status on that data probe line. A carrier transition from off-to-on resets this timer prohibiting a status reset. Also, after sensing a carrier transition, a 200 millisecond timer is started and allowed to time-out. While this timer is active, the control and data leads from the modem are in transition and have not yet reached steady-state. Consequently, any data received in this time period is ignored and considered "noise". On placing calls, the ACU/LS is allowed to ring the intended receiver for 25 seconds before abandoning the call. In order to cater to this wide range of times (200 milliseconds to 25 seconds), a 100 millisecond programmable clock drives the Time of Day Clock Routine software. Counters and flags within this routine are used to control all of the timing functions.

1. Disable the interrupts and time tag the event
   a) Data character
   b) Ring indicator
   c) Carrier transition
   d) Receive overrun
2. Enable the clock interrupts
3. Buffer the data
   a) Character
   b) Time tag
   c) Interface status
4. "V" MAKE 6
   a) Enable character interrupts
   b) Reformat and buffer for magnetic tape output (possibly switch buffers and schedule TAPE)
   c) Schedule RCVME
   d) Disable character interrupts
   e) "P" MAKE 6
5. Return the processor to the state before the interrupt occurred

Figure 12a. Data Probe Interrupt Service Routine

1. Disable the interrupts
2. Increment the high order byte
3. Schedule TD on the high order byte overflow
4. Return the processor to the state before the interrupt occurred

Figure 12b. Time Tag Clock Interrupt Service Routine

1. Disable the interrupts
2. ACU/LS service
   a) Carrier transition service
   b) Ring indicator service
3. Time-of-day service
   a) Update the TOD counters
   b) Schedule ASTIM
4. Return the processor to the state before the interrupt occurred

Figure 12c. Utility Clock Interrupt

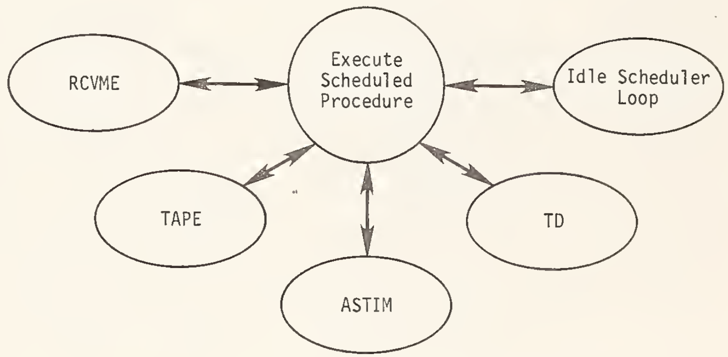


Figure 12d. Scheduler Functions

Figure 12. Interrupt Service Routines and Schedule Functions

These three interrupt service routines schedule processes that the NMM Scheduler runs. Figure 12d illustrates the scheduler and its relationship to these processes. Whenever the NMM is not servicing interrupts and is not running a process, the IDLE Scheduler loop is executing tests on process "wake-up-flags". Each process owns a flag which is set by an interrupt service routine and which is reset by the process completion software: A process starts to run after the Scheduler acknowledges the "wake-up-flag"; the process runs to completion; executes the process completion software, and resets the "wake-up-flag" before returning control to the IDLE Scheduler loop.

Initialization and Termination

An initialization routine prepares the logging tape for a recording session by positioning to the appropriate file mark. The operator is required to provide the date and time of the recording session and to insure that all of the data probe connections are properly set up. At the end of the recording session, the operator requests that the NMM write an end-of-file mark; rewind, and unload the logging tape. A simple command language (Appendix i) enables the operator to initialize and terminate the session.

Data and Control Structures

Semaphore

A semaphore is a special-purpose integer variable used to control the synchronization of parallel processes. All semaphores in the NMM are initialized to minus 1 when the NMM is loaded into core and prepared for execution. Only two operations are performed on a semaphore -- the "P-operation" and the "V-operation" [DIJ 68]. A Process "Q" that performs the operation P, decreases the value of the semaphore by 1. If the resulting value of the semaphore is greater than or equal to zero, the process "Q" can continue. If the value of the semaphore is less than zero the process "Q" is blocked until further notice -- until a "V-operation" is performed on this same semaphore.

A process "R" that performs the "V-operation" on a semaphore, increases the value of the semaphore by 1. If the value of the semaphore is greater than zero, the "V-operation" is complete. On the other hand, if the value of the semaphore is equal to zero, any process waiting execution on this semaphore is allowed to proceed.

Semaphores are used extensively in the NMM to control synchronization of the NMM processes that reformat data acquired by the interrupt service routines. This acquired data is stored in a circular buffer.

Circular Buffer

The Network Measurement Machine uses a circular buffering scheme to handle input interrupts from data probes. A buffer header containing the buffer start address, the buffer stop address, a "next in" pointer, and a "next out" pointer is used to control access to the data in the buffer. Entries are made in the buffer by placing the value of the data item indirectly through the "next in" pointer. Then, the size of the data item is added to the pointer. A check is made to insure that the pointer is within the bounds of the buffer. If the pointer is outside the bounds of the buffer, it is set to point at the first element. Entries are removed from the buffer by obtaining a data item from an indirect access to the "out pointer". An identical strategy is used to update the "out-pointer" as was used for the "in-pointer". This circular buffer provides a fast, efficient method for storing the data accumulated by the data probe interrupt routines.

A slight variation of this scheme is used to queue buffers for output to the logging device. Each time a buffer is filled (indicated by having the "in pointer" address an item outside the current buffer), a new buffer is selected. The filled buffer is queued to be written on the magnetic tape. The new buffer address is obtained from a list of free buffers. This list of free buffers is maintained in a pre-assigned circular buffer.

Data Probe Interrupt Service Routine

The objective of the first software design requirement (to accurately capture and time tag all events at the data probe) is accomplished in the data probe interrupt service routine. This routine uses the circular buffer described above to make a copy of the received character along with a time-tag and other interface status bits. This routine then V's a semaphore controlling a process which reformats the information for eventual entry onto the logging tape. The rest of this section describes how these interrupts are handled by the NMM -- the responsibility that the software has to the hardware and the responsibility that the NMM has to the data.

While received data character interrupts occur most often, other interrupts must be serviced by the Data Probe Routine. The DC11 asynchronous interface generates three other interrupts besides the received character interrupt: ring indicator interrupt, carrier transition interrupt, and receive overrun interrupt. The DL11 interface is similar to the DC11 interface and it generates the same interrupts, but it has two others -- the clear-to-send interrupt and the framming error interrupt. To satisfy the hardware, the software must be responsive to these interrupts by clearing the data from the interface -- this is the responsibility of the interrupt software.

Responsibility to the Hardware

The data probe interrupt software removes assembled characters from the hardware interface, time tags them, and records other relevant status information for later analysis. Each received character interrupt sets a "status" bit indicating the presence of an assembled character. This bit is cleared by moving the character from the interface to the circular buffer. Ring interrupts are generated whenever the interface is connected to a modem and the modem ring indicator line becomes true. The software responds to this condition by setting Data Terminal Ready (DTR). This signal is passed through the interface to the modem which then goes on-line -- analogous to manually answering a ringing telephone. Some terminals or remote systems are hardwired to the interface, and no ring signal is available. When such a connection is used, a carrier transition interrupt is generated when the equipment comes "on-line". The interrupt software also responds to this condition by setting DTR. Receive Overrun and Framing Error interrupts are ignored by the NMM, but, the occurrence of these errors are recorded for latter analysis.

Time-Tagging

Whenever an interrupt from a device occurs, a record of information is generated in the circular buffer entry. This record contains a 24-bit interval clock value, the interrupting unit number (each interface is assigned a unit number), the interface status and control word contents, and the contents of the character buffer register. This buffer entry is later reformatted for output to the logging device.

The time-tag clock value is the output of the 24-bit interval counter. The low order 16 bits are supplied in the hardware clock count register and the high order 8 bits (maintained in core) represent the overflow. The counting rate is governed by a 10KHz, crystal-controlled, programmable real-time clock (DEC KW11P) -- each clock tick represents .0001 seconds. To avoid taxing the system this 16 bit clock is only allowed to interrupt the processor when it overflows. At 10KHz this occurs once every 6.5536 seconds. At each interrupt the high order byte contents are incremented.

During the data probe interrupt sequence, the 16 bit low order clock register is sampled and saved along with the high order 8 bits. This is accomplished while the processor is locked out from servicing other interrupts. This avoids processing a clock overflow interrupt while attempting to time tag a data probe interrupt. While a complete description of timing considerations is presented later, it is worth noting here that the time it takes to service a data probe interrupt and time tag the event while the processor is locked out is 76 microseconds.

Flow Control

Program flow through the data probe interrupt service routine is shown in Figure 12a. Once entered, the service routine assembles the time tag, the unit number, the control and status bits, and the character into a circular buffer entry. Then, the processor is made interruptable so other character interrupts may occur. Next a process called MAKE6 is "V′ed". The V′ed process reformats the circular buffer entry by creating a 6-byte (48 bit) record (Appendix ii) destined for the logging tape.

Having created a 6-byte record destined for the logging device (and possibly writing a full buffer to the logging tape), a routine called RCVME, which only operates on the character and interface status, is entered. This routine is responsible for performing the initial connection procedure (ICP). The ICP gives users access to the ACU/LS, recognizes the user′s terminal speed, and sets or resets Secondary Transmit Data (SXD), the associated data line relay control.

MAKE6, V′ed by the interrupt routine, comes to completion by P′ing itself. In this way, the total interrupt process is self-regulating. That is, the more interrupts there are to process, the faster the interrupt routine can return. This happens because the initial overhead of saving the machine state during the interrupt need only be done once -- for the first interrupt. The price paid for this convenience is the additional core buffer space required to store the data from intermediate interrupts. The advantage of such a scheme is evident in the time tagging process, where the sooner the time tag can be associated with the interrupt, the more accurate it will be. The CPU is locked out from other data probe interrupts only during the period of time in which the clock is copied. The interrupt routine can then continue even though future interrupts might occur. Time tag accuracy is thus maximized.

Initial Connection Procedure

The logical interconnection of a data probe to a data path involving modems and common carrier facilities (Figure 7) is accomplished during the Initial Connection Procedure (ICP). The ICP consists of answering the in-coming call for dial-in users, recognizing the terminal speed when appropriate, and connecting to the remote system for the user by either dialing out with the ACU/LS or connecting through a hard wire connection to a local computer network. To handle the wide variety of terminal types, terminal connections, and remote system connections that are possible, a table of ICP states is defined which identifies a particular ICP for a given data probe. Thus, the method used to insert the data probe is predefined.

ICP Table

The table used to describe the ICP for all interfaces connected to a data probe is defined in a macro used to help generate the system. The ICP entries in this table control access to portions of the ICP software. For example, if speed recognition is desired, the speed recognition software subroutine is used. If speed recognition is not required, then this subroutine is bypassed. The decision is made (for speed recognition and other ICP functions) by examining the entries in the ICP table. The ICP table entries are changed by the software to reflect the state of a connection as it progresses through the ICP. ICP table entries are first used by the initializing program to load initial values into the control and status registers of each device. This also includes initial states of the data probe line control relays. Table entries exist for the initial values of the control and status registers of both the transmitter and the receiver for every data probe interface.

Another table entry called the "use" flag, indicates what subroutines should be accessed during the ICP. Bits within the use flag direct the NMM to observe incoming data characters during the ICP, to observe carrier transitions during the ICP and to observe ring indicator interrupts during the ICP. A special entry indicates whether or not a herald message is required after the data rate has been established.

Overall control of the ICP is represented by a system recognition indicator. This indicator reflects two important Boolean ICP states: (1) whether or not the speed of this user is recognized, and (2) whether or not the remote system connection is made. Each of the appropriate flags or variables is examined for every event such as a character interrupt, ring indicator interrupt or carrier transition interrupt (Table 2).

ENTRY	DESCRIPTION
IV	Interrupt vector location for this interface
CSR	Control and status register address for this interface
TYPE	Designator for interface DC11 = 1, DL11 = 2
UNIT	Unit number of this interface
T-CONNECTION	Letter designator for pairs of interfaces defining a data probe
SPEED DESCRIPTOR	One word descriptor where each bit represents a possible interface speed
USE FLAG	Indicates which ICP routines are applicable for a given interface
INITIAL CSR ENTRIES	Initial transmit and receive control and status register values
SYSTEM RECOG-NITION INDICATOR	Two Boolean variables: 1) speed recognized, and 2) remote system connection established

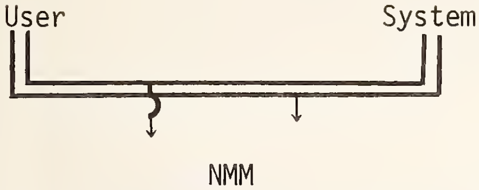
TABLE 2. THE INITIAL CONNECTION PROCEDURE TABLE

The progression of the ICP through its various states is dependent upon which subroutines are required. The simplest ICP (Figure 13a) would have the Boolean variables set to indicate that the remote system connection is established and that the speed is recognized. This combination is used when the data probe is already inserted into the data path and the user interface and system interface are already established. A typical example of this ICP is the use of a remote T-connection as in Figure 8.

The next most complicated ICP involves the connection of a hard-wired terminal to a hard-wired system. Here, carrier transition at the user terminal is used to cause the NMM to output a herald message. The associated data line relay is then reset completing the connection to the remote system (Figure 13b). Again, if the DC11 interface is used, speed recognition could take place before the herald message is output to the user.

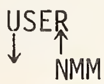
The most complicated ICP (Figure 13c) occurs when a dial-in activity requires speed recognition and dial-out service through the automatic calling unit to a remote system. First, the user calls the NMM. The resultant ring interrupt causes the proper subroutines to place the answering modem on-line. A ten second timer is then started which, if allowed to time-out, would disconnect the modem (hang-up). A carrier transition to the true state, indicating that the modem-to-modem common carrier communication link is properly established, aborts this timer. Since data carrier is not established during a voice call, this timer prevents non-data calls from tying up the data communication link. Once the carrier is established, a 200 millisecond timer is started. All characters received during this 200 millisecond window are considered noise and are ignored by the ICP routines. However, interrupts are recorded, reformatted and written to the logging tape even though they occured in this noise-window. The first character received after the 200 millisecond timeout is treated as a speed recognition character. The proper data rate is established in the DC11 interface after this character is interpreted. A herald message soliciting the telephone number of the required remote access system is then sent to the calling device. The user then enters the telephone number and the automatic calling unit software is employed to cause the ACU/LS to dial the requested number. When this connection is established, an "on-line" message is sent to the caller, and the data line control relay is reset, completing the ICP.

These initial connection procedures satisfy the second design requirement -- to provide an efficient method for inserting the data probes into a data communications path. Though complicated by the large variety of different connection classes, the ICP table facilitates the organization and catagorization of these classes, making them more manageable and convenient to use.



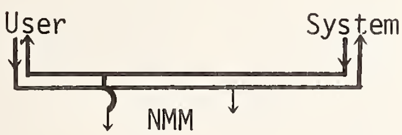
1. Local hardwire connection with speed preset and system configuration established.

Figure 13a. Simple ICP



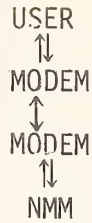
1. User switches hardwired terminal online.

2. Carrier transition informs the NMM to begin ICP.



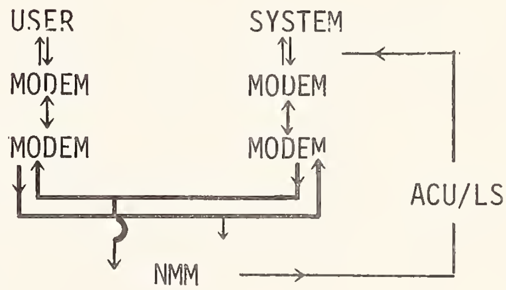
3. NMM completes connection to remote system and resets STD forming data probe "T-connection".

Figure 13b. More Complex ICP



1. A user dials the NMM

   a) Telephone rings; NMM answers.
   b) Carrier detected and stable.
   c) First character after 200 ms. delay used for speed recognition.
   d) Herald message to user solicits telephone number of remote system.



2. NMM dials requested number with ACU/LS.

   a) Remote carrier is detected and stable.
   b) NMM resets STD forming data probe "T-connection".

Figure 13c. Complex ICP

Figure 13. Sample Data Probe/ICP Configurations

Timing Considerations

Once a connection is established, the primary responsibility of the NMM is to accurately time-tag the events on the communications line.

Time-Tag Accuracy

To accurately time tag the events occuring at the data probe, a special re-entrant interrupt service routine makes use of the priority interrupt structure of the minicomputer architecture. As mentioned earlier, the critical section of interrupt code executes in 76 microseconds. In this time period, a circular buffer entry is made in which the character, the time tag, the unit number, and interface status are stored. The interrupt service routine then allows other character interrupts to occur while the previous interrupt finishes execution. For the next 500 microseconds of the interrupt period, data, stored in the circular buffer mentioned above, is reformatted and prepared for eventual output to the logging device. Finally, an 18 microsecond period of critical uninterruptable code is executed while the interrupt service routine updates pointers in the buffer. Figure 14a shows the interrupt status of the processor while servicing a single interrupt.

The interrupt structure of the PDP 11/20 allows for eight levels of program priority. The lowest priority software runs at level zero and any peripheral device is allowed to interrupt the CPU. The highest priority, level 7, is given to software which must execute without interruption from other peripheral service routines. Intermediate levels of priority are assigned to software so other peripheral devices may interrupt as necessary. Any peripheral assigned a priority level equal to or lower than the current software priority must wait for interrupt service; as soon as the software lowers the running priority, pending interrupts are acknowledged.

During the first 28 microseconds of the character interrupt service, the processor priority is raised to its highest level (7) while the clock value is copied. Once copied, the processor priority is then lowered to 5. This prevents other data probe events from interrupting but allows clock interrupts (priority level 6) to occur.

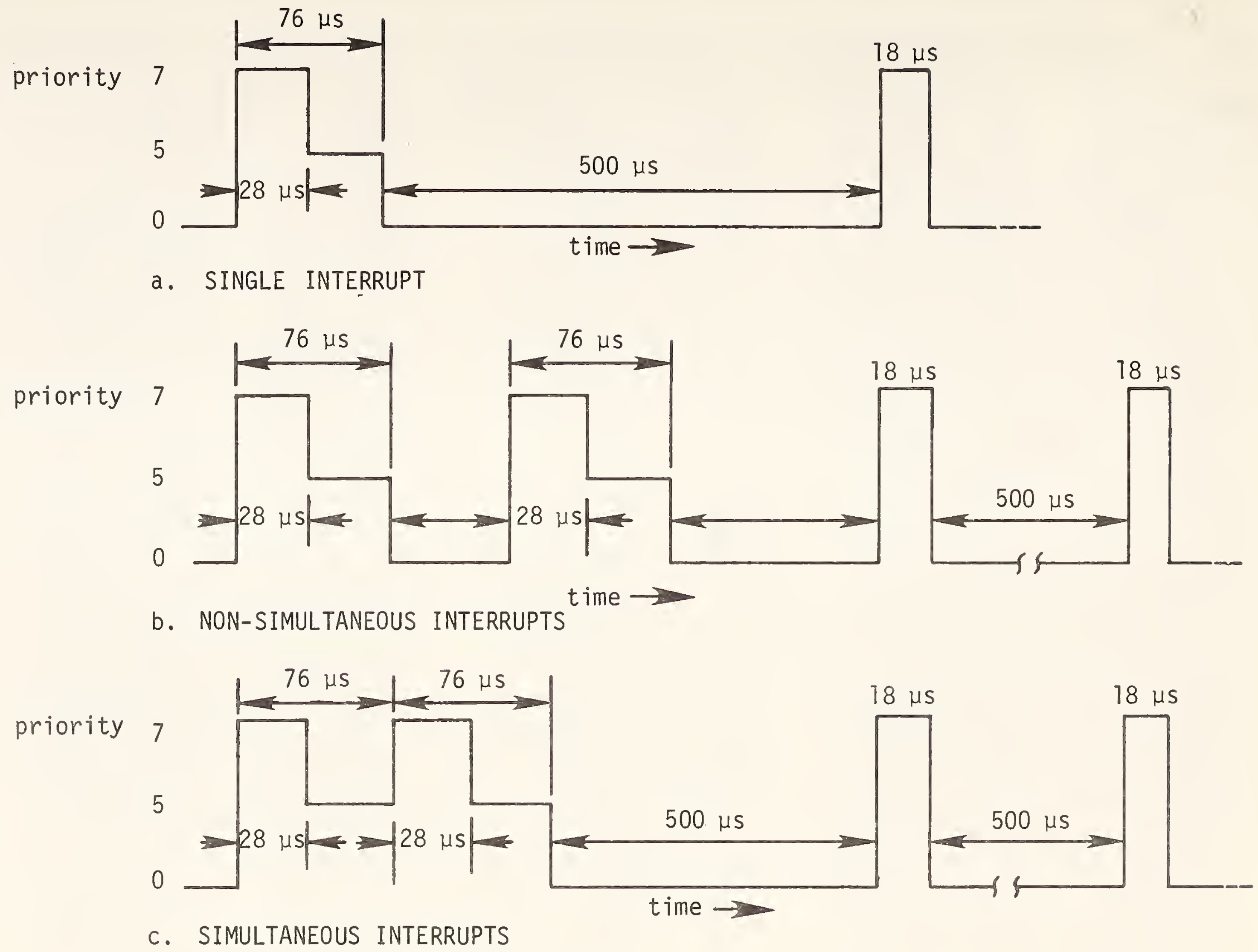


Figure 14

Interrupt Timing Sequences

If another interrupt occurs during the 500 microsecond interruptable section of code, no delay is experienced in obtaining the next time tag as illustrated in Figure 14b. If, however, another interrupt occurs simultaneously, one interrupt is postponed by 76 microseconds, as illustrated in Figure 14c. So, the maximum (or worst case) error in a time tag is a function of the number of devices allowed to interrupt simultaneously. In fact, if N devices can interrupt simultaneously, then, the worst case timing error is 76(N-1) + 100 microseconds. The 100 microseconds represent the resolution of the clock. For eight full duplex terminal connections -- the maximum number currently supported by the NMM -- sixteen interrupts can occure simultaneously (n=16). The maximum worst case delay is:

$$76\ (\ 16-1\ ) + 100 = 1240 \text{ microseconds}$$

The probability that this delay will occur -- that all terminals will interrupt at the same time -- is very small.

Data Transfer Rates

Data transferred into the NMM must not exceed the bandwidth of the interrupt service routine or "receive over-run" interrupts indicating loss of data will result. Each input interrupt is processed in 76 + 500 + 18 = 594 microseconds as shown in Figure 14. The maximum input rate at saturation would be: 1/.000594 = 1684 interrupts/second (characters/second).

The transfer of data out of the NMM onto the logging device is also of concern. The magnetic tape drives used by the NMM (7-track/556 bpi) are only capable of transferring 48,000 bits/second (the equivalent of 6000 characters/second). Each input interrupt character generates 48 bits (the equivalent of 6 characters) as previously described. Therefore, a maximum continuous input character rate of 1000 characters per second, causing 6000 characters/second to be transferred to the logging tape, would saturate the system. Conservatively, then, a total continuous input bandwidth of 960 CPS can be accommodated. This bandwidth can be distributed across all of the interfaces connected to data probes. A typical NMM configuration might support the following terminals:

	6	30CPS terminals	(12 interfaces)	360
	1	240CPS CRT	(2 interfaces)	480
	1	60CPS terminal	(2 interfaces)	120
total	8			960

In the current system, the logging device limits the total throughput to about 960 character interrupts per second. By replacing the current tape subsystem with a faster device, the interrupt saturation bandwidth of 1684 CPS could be realized. This would require a logging device capable of outputing data at 1684 * 6 = 10104 CPS. This bandwidth is not unreasonable for high density magnetic tape or disc cartridge systems.

System Validation

To become a viable tool, the NMM must consistently perform its intended function -- to identify, to characterize and to record selected activity on a data communications link. There is an obvious need to insure that the NMM is both accurately timing the activities on the link and that the machine is functioning properly to identify and record the selected communication line activities.

Criteria

The character time tagging functions of the NMM are verified by inputting a known, controlled sequence of data and examining the recorded results. A favorable comparison is in itself sufficient to insure that the timing mechanism in the NMM is working properly, and that other functions such as recording the data on magnetic tape are being performed.

This comparison is achieved with the NMM installed in its normal configuration. A character generator connected to the NMM transmits characters at a known bit rate and at a known interval. The NMM, if functioning properly, will record these characters. A comparison between the known characters, their rates, and intervals and their values as recorded on magnetic tape reveals the operability and accuracy of the measurement machine.

Equipment

An ASCII character generator modified to allow the operator to select inter-character intervals of .1, 1., and 10. seconds is used in the test procedure. The generator is capable of transmitting characters at 10 CPS, 30 CPS, and 120 CPS through an RS-232 interface into any of the data probes.

Figure 15 shows this generator with all of its switches and buttons. When the switch near the outboard timer wire is set to EXT (external clock), the precision timer triggers the generator. The INCR (increment) position of the INCR - NORM switch is chosen to allow the generator to increment through the entire ASCII character set. The NORM (normal) position transmits the character placed in the data switchs on the upper left panel. The CONT(continuous) - Single Step toggle switch should be set to Single Step. The CONT position will cause the generator to continuously output characters defeating the purpose of the outboard precision timer. The data rate switch is set to the rate expected by the NMM ICP software.

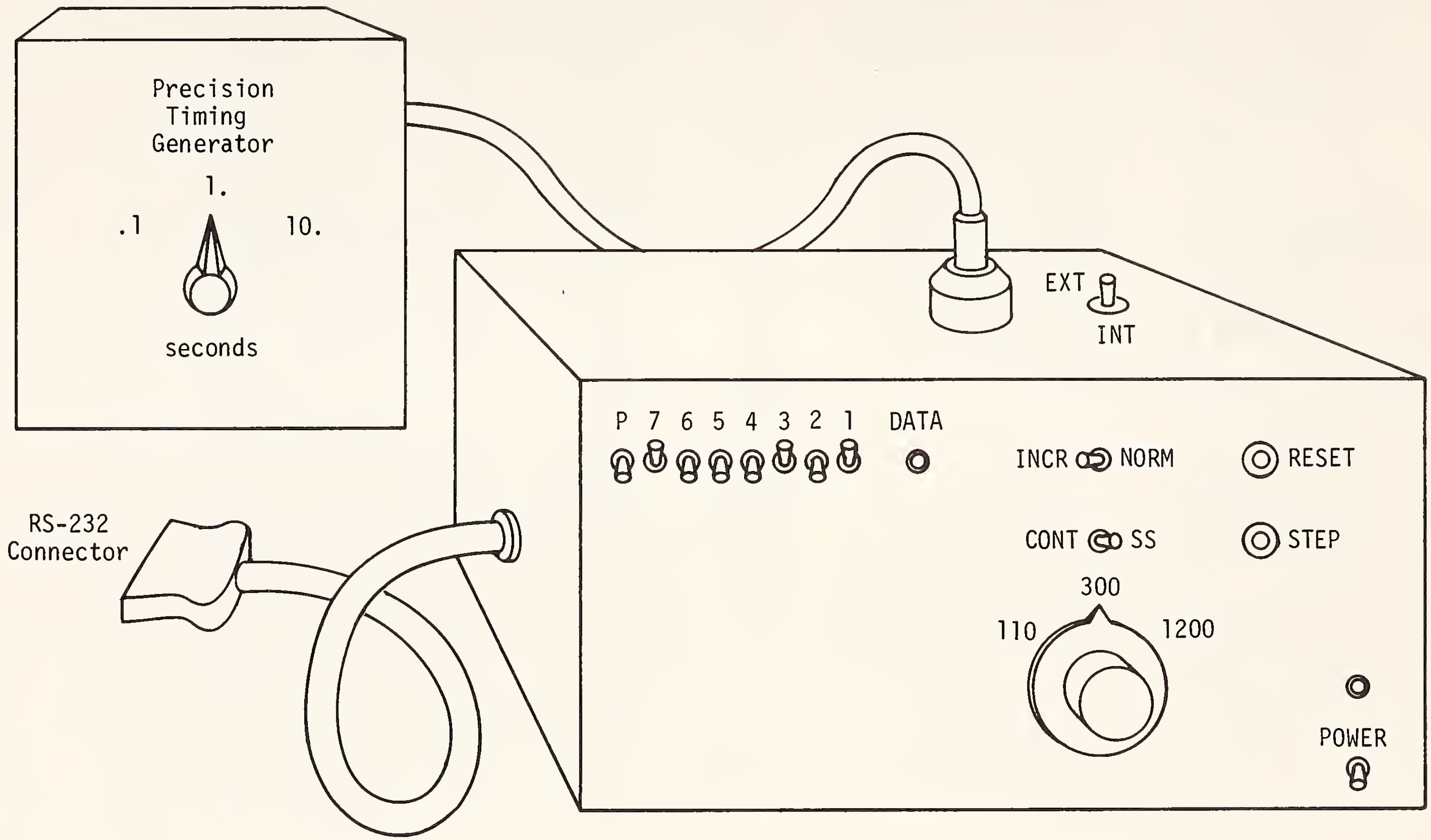


Figure 15

ASCII Code Generator with Outboard Inter-character Rate Selector

Several NMM interfaces will dynamically recognize the user (generator) character transmission speed. This does require the generator to send the ASCII "E" as its first character. To send an "E", bits 1, 3, and 6 are toggled into the DATA switches and the other switches are set to the following positions: INT, NORM, SINGLE STEP. Then, the Single Step button is pushed once. After the speed recognition character is sent, the normal switch positions are selected. Table 3 summarizes these two uses of the generator.

Normal Use	Speed Recognition
EXT	INT
INCR	NORM
SINGLE STEP	SINGLE STEP
Data Rate Select	Data Rate Select
	DATA Switches 1, 3, and 6
	Single Step (once)

TABLE 3. SWITCH POSITION SUMMARY FOR ASCII CHARACTER GENERATOR

Observations

This testing system is designed to verify that the machine is operating correctly before an experiment is actually run. Other uses are made of the timing device while the NMM is running. The operator can monitor the port as the generator sends data by using the operator command "PK" (Appendix i). If the monitored port is misbehaving, the operator can assume that the NMM is malfunctioning. The analysis routines can also make use of the data generated by this device. They can use the data from the port to verify that the NMM behaved properly for the duration of the experiment by comparing the inter-character differences. If the analysis routines detect any discrepancies, the entire data collection experiment should be suspect. In general, the NMM will seldom fail partially. Usually, any error is catastrophic. Thus, if the timing port satisfies the analysis routines, other ports are likely to contain good data.

CONCLUSION

As the use of interactive terminals increases, the need to measure and compare the services rendered to the users of these services will become increasingly important. Tools which measure the performance and utilization of network services are beneficial for the study of, the selection of, and the procurement of service providers. The Network Measurement System -- of which the NMM is a part -- is one such tool which has been successfully tested and used.

The tool was constructed using a DEC PDP-11 mini-computer with standard character asynchronous interfaces. A special hardware data probe was designed and built to insert the NMM into a communications link between a user terminal and the service computer. A special automatic calling unit and line selector assists in placing data calls when modems are utilized. These hardware peripherals are controlled by a special minicomputer operating system which records selected activity on the data communication link to be measured.

Verifying the data recorded by the NMM from the data communications link has been given special attention. Use of a specially modified crystal controlled character generator as a test driver for the NMM assures the operability of the data probes and the accuracy of the time-tagging software. As part of the larger Network Measurement System that requires the use of modeling and analysis routines, the NMM is responsible for providing these assurances to the data collected.

Other efforts taken in the design and implementation of the NMM help to maintain and to assure data validity. The use of a dedicated minicomputer with relatively simple operating system software rather than a time shared or multi-programmed system with large and complex operating system software eliminates the sometimes unpredictable behavior of the later. In the dedicated environment of the mini-computer, careful attention can easily be payed to the data probe interrupt level software where the particulars of both the interrupt timing and the interrupt strategy employed are easily studied. A combination of the reliable data probe hardware and the dedicated minicomputer make the Network Measurement Machine a viable tool for acquiring data on the performance of network services and service providers.

REFERENCES

AB75 Abrams, M.D., and Cotton, I.W., The Service Concept Applied to Computer Networks, NBS Technical Note 880, August 1975.

AB74 Abrams, M.D., "Measuring Service Delivered in Interactive Computing," Proc. of the 2nd Jerusalem Conference On Information Technology, Pp. 747-754, August 1974.

BLA74 Blanc, R.P., Review of Computer Networking Technology, NBS Technical Note 804, January 1974.

BOI74 Boies, S.J., "User Behavior on an Interactive Computer System," IBM System Journal, No. 1, 1974, Pp. 2-18.

DEC73 PDP-11 Peripherals Handbook, Digital Equipment Corporation, Maynard, Mass., 1973.

DIJ68 Dijkstra, E.W., "The Structure of "THE"-Multiprogramming System," Comm. ACM Vol 11, No. 5 May 1968, Pp. 341-346.

EIA69 "Interface Between Data Terminal Equipment and Data Communications Equipment Employing Serial Binary Data Interchange," Electronic Industries Association, Engineering Department, Washington, D. C., August 1969, EIA RS-232-C, 29p.

McC74 McCarn, D.B, "The Communications Jungle as Seen by the User," Computer Networks: Trends and Applications, Institute of Electrical and Electronic Engineers, Inc., New York, 1974, Pp. 7-8. (Proceedings of the 1974 Symposium sponsored by the National Bureau of Standards, Gaithersburg, Maryland, and the IEEE Computer Society)

NE74 Neumann, A.J., et.al., A Technical Guide to Computer-Communications Interface Standards, NBS Technical Note 843, August 1974.

ORN72 Ornstein, S. M., et.al., "The Terminal IMP for the ARPA Computer Network," Proc. SJCC 1972, Pp. 243-254.

WA75 Watkins, S.W., and Abrams, M.D., Interpretation of Data in the Network Measurement System, NBS Technical Note, in press.

APPENDIX i

OPERATOR'S VIEW

The operator is responsible for starting the Network Measurement Machine, mounting tapes, and other machine-room procedures. He interfaces with the data acquisition system through the operator's console and communicates with the system using the NMM command language.

The command language used by the operator is defined with the following syntax:

<message> is any character string

<unit> is any preassigned unit number

All operator commands end with carriage return.

The operator commands used most often are:

a. END

This command empties the contents of any partially full buffers onto the magnetic tape. Then an "end-of-tape" information record is written followed by three end-of-file marks. Finally the tape is rewound and the tape drive placed off-line. A new copy of DOS is loaded into core by simulating the action of the bootstrap loader.

b. PK <unit>

This command is used by the operator to observe the input character stream from a particular unit. Use of this command again on the same unit will stop the output. Output is directed to the line printer.

c. TO <unit> <message>

This command allows the operator to send a message to the unit specified only during the ICP.

Several commands allow the operator to change the ICP table entries while the acquisition system is operating:

d. NLM <unit>

e. TIP <unit>

f. DIAL <unit>

g. 1108 <unit>

These commands specify which type of connection each data probe will be involved in. These result in appropriate entries being made on the logging tape as required by the analysis routines. Normally, each data probe has a default value in the ICP table.

Several commands are available as diagnostic aids to the operator:

h. MD <unit>

This command will cause the message "DIALING" to be sent to the unit specified. Also, the modem attached to the destination port of the data probe is prepared for ACU/LS dialing activities.

i. SET <unit>

This command causes the message "ON-LINE" to be sent to the unit specified. Also, the data probe data line control relay is de-energized (Secondary Transmit Data (SXD) = 0).

j. ACU <unit> <modem number> [<telephone number>]

This command assigns a unit to a modem for the automatic calling unit and overrides the default assignment. If the optional telephone number is specified, it is automatically dialed by the ACU/LS.

APPENDIX ii

OUTPUT DATA FORMATS FOR LOGGING TAPE

The interface between the analysis routines and the data acquisition system is the logging tape. This tape contains entries defining the acquisition system hardware configuration, software version number, time and date of the data collected, the characters collected with their time tags, the modem status indicators with time tags, and an indication of the end of the tape.

The logging tape is organized into record of information. Each record is 6 bytes (8 bits/byte) long. The first byte of each record is used to identify the record type. Table 1 shows the record types used along with the identifying byte.

Reference	Description	Code (Byte 1)
1.	Character information record	any positive byte
2.	Time and Date record	-1
3.	Configuration record	-2
4.	Destination record	-3
5.	Version/Title record	-4
6.	End of Logical File	-0
		2's complement byte arithmetic is used. -0 refers to the sign bit of the byte being set.

TABLE 1. LOGGING TAPE RECORD TYPES

A description of each record follows.

1. Character Information Record

The Character Information Record contains all of the information available from the hardware asynchronous interface at the time an interrupt is acknowledged. Interrupts are caused by one of the following four conditions at a data probe interface:

1) Receive Data Interrupt

2) Carrier Transition Interrupt

3) Ring Interrupt, and

4) Receive Overrun or Framing Error.

This record is organized in bytes and packed in 16 bit words in the following manner:

Character Information Record

INFO	CHARACTER
UNIT	HIGH CLOCK
MIDDLE CLOCK	LOW CLOCK

The Character byte identifies this record because the high order (sign) bit of the byte is forced to 0 (a positive byte), therefore only seven bits are relevant. The INFO field contains bits which reflect the status of the data probe interface at the time of the interrupt. These bits are defined below (bits are numbered from the right starting at 0). The Unit number specifies which data probe interface caused the interrupt. The High, Middle and Low Clock entries together form the 24 bit interval time-tag.

INFO for DL11 Interfaces

```
         |-------------------------|
bit      | 7  6  5  4  3  2  1  0 |
         |-------------------------|
           |  |  |  |  |  |  |  |
           |  |  |  |  |  |  |  ---Carrier Detect
           |  |  |  |  |  |  ------(unused)
           |  |  |  |  |  ---------(unused)
           |  |  |  |  ------------Parity
           |  |  |  ---------------Over Run Indicator
           |  |  ------------------Framing Error
           |  ---------------------(unused)
           ------------------------Secondary Transmit Data
```

INFO for DC11 Interfaces

```
         |-------------------------|
bit      | 7  6  5  4  3  2  1  0  |
         |-------------------------|
           |  |  |  |  |  |  |  |
           |  |  |  |  |  |  |  ---Carrier Detect
           |  |  |  |  |  |  ------Low order Speed Bit
           |  |  |  |  |  ---------High order Speed Bit
           |  |  |  |  ------------Parity
           |  |  |  ---------------Over Run Indicator
           |  |  ------------------Ring Indicator
           |  ---------------------Carrier Transition
           ------------------------Secondary Transmit Data
```

While the DL11 and DC11 interfaces are different, similar signals are available in the interface for recording by the NMM. Notable exceptions are the speed bits used in the DC11. Two bits allow four different speeds to be defined. The Configuration Information record defines the actual speeds available for this interface. The parity indicator in the DL11 is used to represent any discrepancy between the expected parity (strap option in the DL11) and the received character parity. In the DC11 the parity indicator is a parity check defined in the following manner: if the bit is set, odd parity checks or even parity faults, and if the bit is reset, odd parity faults or even parity checks. Note that the DC11 interface does not have framing error detection.

2. Time and Date Information Record

A Time and Date Information Record is written on the logging tape whenever the 24 bit interval clock overflows. This record is identified by a -1 (octal 377) in the 1st byte of the record.

Time Date Information Record

```
|----------------------------|
|   SPARE     |  OCTAL 377   |
|----------------------------|
|        TIME OF  DAY        |
|----------------------------|
|   JULIAN     DATE          |
|----------------------------|
```

The Time of Day is represented using 4 hexadecimal digits (16 bits) indicating the hour of the day and the minute of the hour. A 24 hour clock is used. The Julian Date is an integer (5 digits) with the least significant 3 digits representing the day of the year and the 4th and 5th digits specify the year modulo 70 (1970). The Julian Date 6002 corresponds to January 2, 1976.

3. Configuration Record

The Configuration Record describes to the analysis routines the configuration of the NMM. A record is written on the logging tape for each interface connected to a data probe, and for the operators console.

Configuration Record

CODE	OCTAL 376
DESIGNATOR	UNIT
SPEED DESCRIPTOR	

Octal 376 (-2) identifies this record as a Configuration record. CODE is an integer which identifies the interface type. A DC11 interface is assigned the CODE 1, a DL11 interface is assigned the CODE 2. The UNIT identifies the device and is the same unit used in the Character Information Record. A DESIGNATOR is used to assign two interfaces to a single data probe. Designators are ASCII characters from the set A through Z. The special character 0 (zero) is used to identify the operator's console. The SPEED DESCRIPTOR is a word used to indicate the baud rate of the interface. The following table is used to decode these bits:

Bit Assignments to Interface Baud Rates

BIT	Baud
0	75
1	110
2	134.5
3	150
4	300
5	600
6	1200
7	1800
8	2400
9	4800
10	7200
11	9600

These bits represent all the speeds a NMM interface is capable of running at, so, for a DC11, 4 bits from this set should be set.

4. Destination Information Record

Destination Information Records are used only by the analysis routines. Their function is to indicate the beginning and ending of conversations. A conversation is defined to be all of the data recorded between occurences of this record. This record is written on the logging tape whenever a carrier transition occurs on a dial out data probe, or whenever the operator specifically requests a Destination Information Record be written.

Destination Information Record

SPARE	OCTAL 375
DESTINATION	UNIT
STATUS	

UNIT is the unit number of the source interface of the data probe. The destination is coded in the following manner:

Destination Codes

Code	USE
0	Null destination; probe inactive
1	Dial-out data probe was dialled
2	Data Probe used for TIP connection
3	Data Probe used for NLM connection
4	Data Probe used for 1108 connection
5	Data Probe used for remote T-connection

The STATUS word contains the low order 8 bits of the receiver status register.

5. Version/Title

In order to inform the analysis routines of changes in the NMM software, a record identifying the version number and title is written as the first block on the logging tape. This record has the following format:

Version/Title Record

```
|-----------------------------|
|   VERSION   |   OCTAL 374   |
|-----------------------------|
|            TITLE            |
|-----------------------------|
|            TITLE            |
|-----------------------------|
```

The version number is an integer used to identify minor changes in the NMM software. The analysis routines check this number to ensure compatibility between the acquired data and the data reduction routines. A more coarse indicator of the NMM software is the TITLE which is a 32 bit field into which six ASCII characters are packed in a form known as RAD50.

The RAD50 format for packing characters is defined as follows:

Character	RAD50 Equivalent (octal)
(space)	0
A-Z	1-32
$	33
.	34
0-9	36-47

The RAD50 equivalents for characters C1, C2, and C3 are combined as follows to produce a 16 bit packed word:

RAD50 value = ((C1*50)+C2)*50+C3

For example the RAD50 value of ABC is:

((1*50)+2)*50+3 or 3223 (base 8)

6. End of Logical File

The logical end of a NMM run is defined by the octal number 200 in the first byte of the record identifier position. When the operator enters the END command, the NMM fills the remainder of the current output buffer with this value before writing it to the logging tape.

APPENDIX iii

PROGRAM DOCUMENTATION

Contents

APPENDIX iii

PROGRAM DOCUMENTATION

I. Introduction

The software documentation contained herein is intended to provide systems programmers, familiar with the Network Measurement Machine (NMM), with the necessary tools for modifying and expanding the NMM code. Included are descriptions of the program routines and macros. The routines are presented in the order that they appear in the program listings, with an alphabetical index.

II. Documentation of Macros

A. Detailed Descriptions

In order to facilitate the programming of the Network Measurement Machine, a set of user defined, global macros was developed. This macro library includes code which specifies a rigid calling sequence for subroutines as well as a set of primitive functions such as the push and pop stack operations. A thorough description of the PDP-11 macro assembler used may be found in the DEC <u>Macro-11 Assembler Programmer's Manual</u>. Each macro operation in the library is described below.

The macro MYMAC identifies the elements of the user macro library. Also at assembly time it prints the assembly version number with the interface configuration descriptions.

Since the PDP-11 is a stack-oriented machine, the macros PUSH and POP were written to facilitate adding and removing arguments from the processor stack SP. Up to five arguments can be handled by each of these macros.

The macro definition of PUSH is as follows:

```
1.  .MACRO       PUSH        A0$,A1$,A2$,A3$,A4$
2.               .IRP        A$,<A0$,A1$,A2$,A3$,A4$>
3.               .NCHR .     A$.<A$>
4.               .IF         NE,A$.
5.               MOV         A$,-(SP)
6.               .ENDC
7.               .ENDM
8.  .ENDM        PUSH
```

Line 1 defines the macro PUSH.

Lines 2 through 7 push the value of the arguments onto the processor stack.

Line 8 ends the macro definition.

The macro definition of POP is:

```
1.  .MACRO       POP         A0$,A1$,A2$,A3$,A4$
2.               .IRP        A$,<A0$,A1$,A2$,A3$,A4$>
3.               .NCHR       A$.,<A$>
4.               .IF         NE,A$.
5.               MOV         (SP)+,A$
6.               .ENDC
7.               .ENDM
8.  .END         POP
```

Line 1 defines the macro POP.

Lines 2 through 7 pop the values of the arguments off of the processor stack.

Line 8 ends the macro definition.

The stack is used in the NMM to provide subroutine linkage that allows:

(1) formal parameter passing by value
(2) re-entrancy and recursion
(3) single value parameter returns, if required, or
(4) no parameter return to calling routine, if desired

Four macros are used to provide this linkage: PROC, RETURN, CALL and TCALL.

PROC and RETURN are used to declare a portion of code to be a procedure. They create a program environment in which formal parameters are passed to the subroutine via the stack. Here register 5 (R5) is used as a parameter linkage register and the program counter (PC) is used as the subroutine linkage register. When arguments are passed on the stack, they are referenced with R5.

The macro definition of PROC is:

```
 1.  .MACRO      PROC        NAME  ARG1,ARG2,ARG3,ARG4,ARG5
 2.              THISPROC=4
 3.  NAME:       PUSH        R5
 4.              MOV         SP,R5
 5.              .IRP        ARGLST,<ARG1,ARG2,ARG3,ARG4,ARG5>
 6.                          .NCHR     NCHR,    <ARGLST>
 7.                           .IIF     EQ,NCHR,     .MEXIT
 8.              ARGLST=THISPROC
 9.              THISPROC=THISPROC+2
10.              .ENDM       ;ENDS   THE   IRP
11.  .ENDM       PROC
```

Line 1 declares the macro and its argument list. NAME is the label to be used as the entry point.

Line 2 initializes a variable to point to the first stack relative address, i.e., the contents of register 5 plus the number 4 will point to the address (stack) whose contents are ARG1.

Lines 3 and 4 register 5 to the stack pointer. This allows recursion and re-entrancy.

Lines 5 through 10 equate the actual names of the variables used in the argument list to stack relative addresses. The terminating condition is the absence of an argument - line 7.

Line 11 ends the macro definition.

The macro RETURN simply delinks register 5 from the stack and uses the normal return from subroutine instruction with linkage register PC - program counter.

```
1.        .MACRO        RETURN
2.                      MOV         R5,SP
3.                      POP         R5
4.                      RTS         PC
5.        .ENDM         RETURN
```

Line 1 defines the macro RETURN.

Lines 2 and 3 delink register 5, restoring the SP to its previous value.

Line 4 is the return from subroutine.

Line 5 ends the macro definition.

The macro CALL is used to setup the stack with the values of the formal parameters. Since all formal parameters are passed by value, the parameter list is removed from the stack when the called PROC returns.

```
 1.  .MACRO      CALL NAME ARG1,ARG2,ARG3,ARG4,ARG5
 2.              ARGCNT=0
 3.              .IRP     ARGLST,<ARG5,ARG4,ARG3,ARG2,ARG1>
 4.                       .NCHR     NCHR,<ARGLST>
 5.                       .IF       NE,NCHR
 6.  PUSH        ARGLST
 7.              ARGCNT=ARGCNT+2
 8.                       .ENDC
 9.              .ENDM
10.              JSR      PC,NAME
11.              ADD      #ARGCNT,SP
12.  .ENDM       CALL
```

Line 1 defines the macro CALL. The transfer address NAME must represent the PROC entry point.

Line 2 initializes the argument count to zero. ARGCNT will be used to remove any formal parameters from the stack.

Lines 3 through 9 push the values of the arguments on the stack in reverse order.

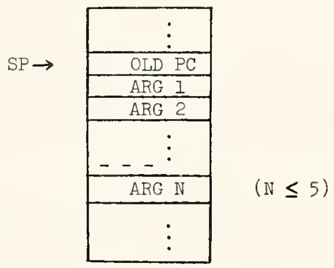
Line 10 causes a jump to the subroutine.

Line 11 is the return point from the subroutine. It clears the stack of the parameters passed.

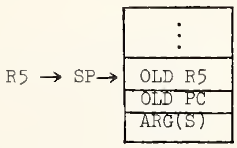
Line 12 ends the macro definition.

Thus when line 10 is executed, the processor stack looks like this.

Processor Stack



The location of the label NAME is the entry procedure as specified for that routine by PROC. Thus after the PROC code has been executed, the stack would appear as follows:



The only difference between the macro CALL and the macro TCALL is the reservation of one stack word for a return parameter.

```
1.  .MACRO  TCALL  NAME  ARG1, ARG2, ARG3, ARG4, ARG5
2.          CLR    -(SP)
3.          CALL   NAME  <ARG1> , <ARG2> , <ARG3> ,<ARG4> , <ARG5>
4.  .ENDM   TCALL
```

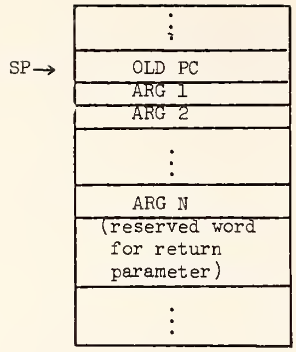
Line 1 defines the macro TCALL.

Line 2 reserves one stack word into which the PROC may place a return value. It becomes the responsibility of the calling program to remove the returned value from the stack.

Line 3 invokes the macro CALL described above.

Line 4 ends the macro TCALL.

Thus the resulting stack contents, after the jump is made to the subroutine (in the macro CALL) will be:



Certain conventions must be followed in the use of these four macros.

(1) The argument list is expandable; however, the number of arguments is restricted to the number that will fit on one line of code. Up to five arguments are allowed now.

(2) The variables THISPROC and ARGCNT are reserved.

(3) No parameter count checks are made.

(4) General register use within the body of a procedure is limited to registers 0 through 4, and the saving and restoring of these registers is the responsibility of the procedure. R5 is the stack relative address pointer for formal arguments.

(5) A procedure which returns a parameter expects the calling program to POP the parameter off of the stack. Therefore, upon return the SP is left pointing to that parameter's location on the stack.

An example of the above defined methods for subroutine linkage is a procedure that is called to add two arguments and return their sum.

```
PROC    SUM     X, Y
        PUSH    RO                      ;Save Register RO
        MOV     X(R5), RO               ;Copy X into RO
        ADD     Y(R5), RO               ;ADD Y
        MOV     RO,THISPROC(R5)         ;Put the result on stack
        POP     RO                      ;Return old RO
        RETURN                          ;Return to Caller
```

The PROC "SUM" would be called as follows:

```
TCALL   SUM #2, #3
POP     ANSWER
```

The macro LOCKOUT disables all interrupts except NPR requests (see PDP-11 PROCESSOR HANDBOOK, page 16 for description of NPR). UNLOCK restores the PDP-11 to its previous processor status.

```
.MACRO LOCKOUT
PUSH   @ #PSW
MOV    #340, @ #PSW
.ENDM  LOCKOUT

.MACRO UNLOCK
POP    @ #PSW
.ENDM  UNLOCK
```

Macro definitions of table entries are utilized in the NMM code. This structure provides an easy method for creating tables and subtables consisting of items from the main table. Also, should there exist a need to change a value contained in many subtables, only the main table value as specified by the macro need be changed. Then when the system is reassembled the new value is properly placed in all tables.

Consider the need to define a large number of interactive terminals. The defining parameters for these terminals can be organized in one macro from which other macros can build tables at assembly time. For instance, assume we have 3 terminals. The macro TRMTBL might be defined to identify these terminals, i.e.,

```
.MACRO TRMTBL NAME

NAME    300    174000    DC11
NAME    360    175610    DL11
NAME    150    176500    KL11
NAME    340    174770

.ENDM TRMTBL
```

The macro parameter NAME is used by other macros to access the table entries defined in TRMTBL. Here the two numeric fields used in the example define an interrupt vector and CSR for the specified terminal of type DC11, DL11 or KL11.

To demonstrate the usefulness of such a macro consider the following macro and its invocation.

```
1.  .PRINT ; THE FOLLOWING TERMINALS ARE USED
2.  .MACRO PRNTME IV, CSR, TYPE
3.  .PRINT ; TERMINAL TYPE IS AT IV, CSR
4.  .ENDM PRNTME
5.  TRMTBL PRNTME
```

Here the assembler expands the macro call TRMTBL (in line 5) with the macro parameter PRNTME associated with the argument NAME. The macro PRNTME is then invoked once for each entry in the list in the body of the macro TRMTBL and the macro PRNTME is expanded causing the assembler to produce a list of all the terminals.

Macros similar in structure to PRNTME can be used to declare storage for the terminals and generate code for the terminals. Most important, however, is that all terminal definitions are contained in one table. Therefore, it is a simple matter to change the NMM configuration parameters simply by changing one table, TRMTBL.

The macro BITS equates commonly used names to their respective numeric value. For example, the magnetic tape control commands are encoded so that the rewind command RWND becomes the number 3 at assembly time. Thus an item in the macro bits might read:

```
.MACRO BITS
   .
   .
   .
RWND = 3
   .
   .
   .
.ENDM BITS
```

The macro SCHEDULE clears the entry in the NMM scheduler's table for the scheduled procedure NAME. This will result in the procedure being executed when its turn comes, i.e., in the round robin schedule. Thus to schedule a routine to direct the ACU to dial a telephone number the statement SCHEDULE DIAL$ would be used.

The macro MAGTAPE was written to allow the systems programmer to give commands to the magnetic tape device in an English-like syntax. For example the command to rewind tape unit one could be expressed:

```
MAGTAPE ONE REWIND
```

The variables ONE and REWIND must have been assigned values in the macro BITS.

The macros MTCALL and TPARG are used within MAGTAPE to interpret the argument list for the MAGTAPE macro call and execute the code necessary to carry out the command.

B. Index to Macro Library

III. DOCUMENTATION OF ROUTINES

A. Detailed Descriptions

T-Connection Interrupt Control Software

Upon receipt of an interrupt from a data probe interface, the following interrupt sequence takes place:

(1) the current program counter (PC) and processor status (PS) are pushed onto the processor stack;

(2) the new PC and PS are loaded from a pair of locations (the interrupt vector) in addressed memory, unique to the interrupting device.

(Note: Consult the DEC PDP-11 Peripherals Handbook for more details).

In the NMM, the input interrupt software causes a jump to a section of non-reentrant code labeled RCVxx (where xx is the unit number for this interrupting device). Here the circular buffer entry is built, consisting of the item of data received (DBR contents) the Control and Status Register (CSR) contents, and the clock value.

Then a jump is made to the reentrant portion of the data probe Input Interrupt Control Software, BLDSOM.

A similar set of actions results when an output interrupt is received; however, the new PC contents causes a jump to a section of non-reentrant code labeled XMTxx, which after some intermediate instructions causes a jump to OUTINT, the reentrant output interrupt processor.

Input Interrupt Service Routines

Entry Point:
RCVxx (where xx is the unit number of the interrupting device as specified in TRMTBL)

Entered from:
hardware interrupt service

Functions:
1. Saves register RØ
2. Stores clock value, DBR contents, unit number, and CSR contents in CIRBUF
3. Exits to reentrant input interrupt service routine

Exits to:
BLDSOM

Contents of Registers on Exit:
RØ - Input Pointer after entries made in CIRBUF

Note: The "RCV" sections of code are the non-reentrant portion of the input interrupt service routine

* * *

Entry Point:
BLDSOM

Entered from:
RCVxx (where xx is the unit number of the interrupting device)

Contents of Registers on Entry:
RØ - New CIRBUF Input Pointer

Routines Called:
MAKE6
IRCVD

Functions:
This is the first section of the common interrupt code for the data probe interrupt processor. Each time that an interrupt occurs on a data probe interface an entry is made in the circular buffer CIRBUF by RCVxx. BLDSOM is responsible for controlling access to the CIRBUF entries. The functions of BLDSOM are:
(1) Updates the input pointer INP by testing for end-around conditions.
(2) V's the semaphore that controls access to the data. If the semaphore indicates that NXTCHR, the access routine, is busy, then RØ is restored and BLDSOM is exited thru RTI. Otherwise a jump is made to NXTCHR. At NXTCHR, the following steps occur:

1. Lowers processor priority to 0
2. Calls MAKE6 and IRCVD
3. Updates the CIRBUF pointers
4. Locks out character interrupts
5. P's the semaphore to see if there are any more data waiting to be processed. If so, go to NXTCHR. Else exit to BLDONE in BLDSOM routine.

Exits to:
RTI

Contents of Register to Exit:
RØ is restored to pre-RCV state

Output Interrupt Service Routines

Entry Point:
XMTxx (where xx is the unit number of the interrupting device as specified in TRMTBL)

Entered from:
hardware interrupt service

Functions:
1. Saves register RØ
2. Loads RØ with a unit index
3. Exits to reentrant output interrupt service routines

Exits to:
OUTINT

Contents of Registers on Exit:
RØ - Units index, created by doubling the unit number of the interrupting device. Previous value of RØ saved on stack.

Note: The "XMT" sections form the non-reentrant portion of the output interrupt service routine.

* * *

Entry Point:
OUTINT

Entered from:
XMTxx (where xx is the unit number of the interrupting device as specified in TRMTBL)

Contents of Registers on Entry:
RØ - unit index

Routines Called:
TRO
INBUFINITIALIZE

Functions:
OUTINT is the common output interrupt routine which is entered after either an external device interrupt or a pseudo interrupt from the terminal output (TRO) procedures. OUTINT outputs characters until the output buffer contents are exhausted. After each output interrupt is serviced the controlling semaphore is P'd to check for any other buffers that need service.

Exits to:
Interrupt routine via RTI

Contents of Registers on Exit:
RØ - restored from the stack to original contents before the interrupt

General Utility Routines

Entry Point:
PKBOO

Entered from:
CALL PKBOO <ACHARACTER>

Contents of Registers on Entry:
R5 - Linking register

Functions:
This routine outputs <ACHARACTER> to the data buffer register PKDBR for the assigned "Peekaboo" device.

Exits to:
RETURN to caller.

Contents of Registers on Exit:
All registers used are restored

* * *

Entry Point:
OCTOUT

Entered from:
CALL OCTOUT <ANUMBER> <ABUFFER><ACOUNT>

Contents of Registers on Entry:
R5 - Linking register

Functions:
OCTOUT converts <ANUMBER> , an octal integer, to ASCII and stores the results in <ABUFFER> , right justified, with the number of digits in < ACOUNT > .

Exits to:
RETURN to caller.

Contents of Registers on Exit:
All registers used are restored

* * *

Entry Point:
UNIBUS

Entered from:
TCALL UNIBUS <ADDRESS>

Contents of Registers on Entry:
R5 - Linking register

Functions:
If the designated address exists, this routine returns true (-1), else a false (0) is returned. An erroneous address is detected by a CPU trap resulting from trying to reference the given address.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

* * *

Entry Point:
SPEEDSET

Entered from:
TCALL SPEEDSET <ACSR> <ACHARACTER>

Contents of Registers on Entry:
R5 - Linking register

Functions:
This routine is used to determine the speed of the interrupting interface. This routine applies only to DC-11 interfaces. If the speed is found the procedure returns a true condition (-1) via the stack. Otherwise a false (0) is returned. It is assumed that the XMITTER and RCVR speeds are the same. This routine changes the contents of the CSR if necessary to modify the hardware speed indicator bits.

Exits to:
RETURN to caller

Contents of Registers on Exit
All registers used are restored

<u>Procedure MAKE6</u>

Entry Point:
MAKE6

Entered from:
CALL MAKE6

Routines Called:
TYPE
PUTINBUF

Functions:

First MAKE6 takes data out of the location in CIRBUF indicated by OUTP and reformats and packs this data into a character information packet in a temporary buffer called CHRSKL. Then PUTINBUF is called to enter this character packet into the logging tape's buffer.

Exits to:

RETURN to caller

Contents of Registers on Exit:

All registers used are restored

Procedure PUTINBUF

Entry Point:

PUTINBUF

Entered from:

CALL PUTINBUF <PTR> <COUNT>

Contents of Registers on Entry:

R5 - Linking register

Routines Scheduled:

TAPE$(TAPEOUT)

Functions:

This procedure transfers indicated data to the logging device's output buffer area, checking for buffer overflow. Pointers to the ping-pong table buffers are updated. Should overflow occur, the tape output routine TAPE$(TAPEOUT) is scheduled and incoming data is routed to another buffer area. Interrupts are locked out while the controlling semaphore is V'd.

Exits to:

RETURN to caller

Contents of Registers on Exit:

All registers used are restored

Procedure IRCVD

Entry Point:

IRCVD

Entered from:

CALL IRCVD

Routines Called or Scheduled:

RCVME$

Functions:

IRCVD is responsible for placing the unit number, CSR contents, and DBR contents onto the circular queue RCVBEG.

This routine is activated each time that a character interrupt occurs.

IRCVD also utilizes the semaphore RCVCNT to control the process RCVME$ which is scheduled if necessary.

Exits to:

RETURN to caller

Contents of Registers on Exit:

All registers used are restored

Terminal Output Procedures

Entry Point:

TRO

Entered from:

CALL TRO <MSGADR> <OUNIT>

Contents of Registers on Entry:
R5 - Linking register

Routines called:
TROH

Functions:
TRO calls TROH and also keeps track of the maximum number of messages in any one output queue for debugging purposes only.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

Terminal Output Routines

Entry Point:
TROH

Entered from:
TCALL TROH <ADDRESS> <INDEX>

Contents of Register on Entry:
R5 - Linking register

Routines Called:
PUTINQ
NEXTFROMQ

Functions:
TROH is responsible for controlling the output of terminal messages. A semaphore controlling access to a circular queue is utilized to handle multiple output message requests.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

I/O Utility Routines

This group of routines and procedures deals with the manipulation of semaphores and circular buffers used by output interrupt processors and buffer handlers.

Entry Point:
PUTINQ

Entered from:
TCALL PUTINQ <ITEM> <QNAME>

Contents of Registers on Entry:
R5 - Linking register

Functions:
This routine enters a designated item into the queue identified by <QNAME> and adjusts the number of items counter. Queue overflow (end-around) is tested for.

This routine is useful for controlling access to procedures that run off of a queue, such as TRO.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

Clock Utility Routines

Entry Point:
ATGTHR

Entered from:
CALL ATGTHR

Functions:
Packs the transformed contents of the T-ARRAY into the time/date block ACLOCK.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

* * *

Entry Point:
FIXCLK

Entered from:
CALL FIXCLK

Functions:
Using the macro FIX, FIXCLK is called to transform the contents of the T-ARRAY into a meaningful 24-hour a day clock

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are saved and restored

* * *

Entry Point:
APART

Entered from:
CALL APART

Functions:
Breaks apart the contents of the time date block and stores them in the 4 byte array named "TO," also called the T-ARRAY

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are saved and restored

Utility Procedures for Extraction of Data from Configuration Blocks

Entry Point:
CONDEX

Entered from:
TCALL CONDEX <UNITNUM>

Contents of Registers on Entry:
R5 - Linking register

Functions:
Passes the address of the configuration block for the unit specified by <unit number> to the calling routine through the stack

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

* * *

Entry Point:
TYPE

Entered from:
TCALL TYPE <UNITNUM>

Contents of Registers on Entry:
R5 - Linking register

Routines called:
CONDEX

Functions:
Passes a byte code for the type of interface of the indicated unit number to the calling routine through the stack. The codes are as follows:

Interface	Code
DC11	1
DL11	2

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

I/O Utility Routines

Entry Point:
NEXTFROMQ

Entered from:
TCALL NEXTFROMQ <NAME>

Contents of Registers on Entry:
R5 - Linking register

Functions:
This routine returns an item from the designated queues and passes this item to the calling routine via the stack

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

General Utility Routines

Entry Point:
EXOR

Entered from:
TCALL EXOR <ARG1> <ARG2>

Contents of Registers on Entry:
R5 - Linking register

Functions:
EXOR is an exclusive OR function which accepts two arguments, calculates the result of the exclusive OR operation on these arguments and returns this value to the caller via the stack.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

*　*　*

Entry Point:
SETIMER

Entered from:
CALL SETIMER <ARG><AUNIT>

Contents of Registers on Entry:
R5 - Linking register

Functions:
This procedure is used to set the carrier or ring timer control word, TIMERS. Since the cell CLKISRUNNING must be incremented whenever a TIMERS entry exists, a test is made to see if an entry already exists before incrementing CLKISRUNNING.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

*　*　*

Entry Point:
TSTCARRIER

Entered from:
TCALL TSTCARRIER <UNIT>

Contents of Registers on Entry:
R5 - Linking register

Routine Called:
TYPE

Functions:
The carrier state for a DC11 or DL11 interface unit is tested. If carrier is off, False (0) is returned. If carrier is on, True (-1) is returned. Also, if the interface under test is a DL11, the result of the carrier state is stored.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

The Scheduled Procedure TAPE$

Entry Point:
TAPEOUT

Entered from:
SCHEDULED TAPE$

Routines Called or Scheduled:
TAPE$
TRO

Functions:
This procedure is schedule for execution whenever the switch section of PUTINBUF fills one of its buffers. TAPE$ writes the buffer out onto tape.

Immediately upon entry to TAPE$ the routine removes itself from the Scheduler's list of waiting routines. Then the tape unit is tested to see if the last operation is busy. If it is, then TAPE$ schedule itself again and exits thru RTS PC. Otherwise, I/O error conditions are tested for and if all is in order the buffer is output.

This routine is controlled by the semaphore TAPCNT.

Exits to:
RTS PC - Return to Scheduler

Command Interpreter Utility Routines

Entry Point:
SEEIT

Entered from:
CALL SEEIT <ACOUNT> <APOINTER> <AUNIT>

Contents of Registers on Entry:
R5 - Linking register

Routines Called:
NUMBER
DIAL
TRO

Functions:
SEEIT is called for each command input to NMM while the user is in ICP (initial connection procedures). It is used to see if 7 or more numbers are in the user input string. If this is the case, then the number is assumed to be a telephone number, the number is dialed, and the user is informed of this action.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

* * *

Entry Point:
PNTNUM

Entered from:
TCALL PNTNUM <BASE> <CONTROL>

Contents of Registers on Entry:
R-5 Linking register

Routines Called:
NUMBER

Functions:
PNTNUM (point to number) is used by ACUCALL to point to the next numeric contained in the ACU command string.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

* * *

Entry Point:
ACUCALL

Entered from:
TCALL ACUCALL <STRING><ACOUNT>

Routines Called:
GETU
PNTNUM
NUMBER
DIAL
DTRRESET

Functions:
ACUCALL parses the parameters associated with the ACU command and carries out the appropriate actions. These parameters include <UNIT> , <VADIC modem address> , and <telephone number> . If a phone number is given, that number is dialed (via the call to DIAL). If not, the modem unit number is assigned to the specified unit.

Exits to:
RETURN to caller

* * *

Entry Point:
TPCLOSE

Entered from:
CALL TPCLOSE

Functions:
TPCLOSE writes 3 end-of-file marks, closes the magtape file, and takes it off-line.

Exits to:
RETURN to caller

* * *

Entry Point:
GETU

Entered from:
TCALL GETU <TEXT> <COUNT>

Contents of Registers on Entry:
R5 - Linking register

Routines called:
NUMBER

Functions:
GETU is called by USER$ routines and is used to obtain the unit number from the operator input command arguments.

If there exists a unit number in the command string, then this number is returned via the stack, else a false (0) condition is returned.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

* * *

Entry Point:
NUMBER

Entered from:
TCALL NUMBER <AT>

Contents of Registers on Entry:
R5 - Linking register

Functions:
NUMBER is called by USER$ routines to determine if a command parameter is numeric. If it is numeric, true (-1) is returned to the calling routine via the stack.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

The Scheduled Procedure RCVME$

Entry Point:
RCVME

Entered from:
SCHEDULE RCVME$

Routines Called or Scheduled:
USER$
SPEEDSET
TRO
TYPE
SETIMER
EXOR
PKBOO

Functions:
RCVME$ is scheduled after a 6-byte packet destined for the logging device is created. RCVME$ manipulates the control function of the DC-11 and the DL-11 interfaces used in the NMM system.

It is responsible for:

(1) performing the initial connection procedure (ICP);
(2) recognizing user's terminal speed;
(3) setting or resetting the T-connection switch.

RCVME$ is controlled by a semaphore. Before returning to the scheduler, it checks the semaphore for any remaining work. In this way it is given the highest priority of the schedulable processes. RCVME$ gets is parameters from a circular buffer built by the routine IRCVD. The contents of RCVOUT point to the next parameter packet in the buffer. The parameter data is organized in the queue into three word packets:

(1) unit number,
(2) control register contents,
(3) data buffer register contents.

Exits to:
RTS PC - return to Scheduler

The Scheduled Procedure USER$

Entry Point:
USERPARSER

Entered from:
SCHEDULE USER$

Routines Called or Scheduled:
ACUCALL
GETU
FIXCLK
ATGTHR
PUTINBUF
TAPEOUT (immediate entry forced)
TPCLOSE
TRO
EXOR
DESTPUT
SEEIT
OCTOUT

Functions:
USER$ is the command string interpreter for the NMM. It parses user input strings and takes the appropriate action.

Exits to:
RTS PC - return to Scheduler

The Scheduled Procedure TD$

Entry Point:
TIMDAT

Entered from:
SCHEDULE TD$

Routines Called:
PUTINBUF

Functions:
This routine places the time/date information block onto the logging tape. Interrupts are locked out before PUTINBUF is called and enabled after return is made.

A count TDCNT is maintained to indicate the number of times a time/date block was written.

Exits to:
RTS PC - return to Scheduler

Destination Block Utility Routine

Entry Point:
DESTPUT

Entered from:
CALL DESTPUT <AUNIT> <ACODE>

Contents of Registers on Entry:
R5 - Linking register

Routines Called or Scheduled:
DEST$

Functions:
This routine is responsible for placing the unit number <AUNIT> and destination code <ACODE> into the circular buffer DESTBEG. DESTPUT then adjusts the buffer pointer and schedules DEST$.

This routine is controlled by the semaphore DESTCNT.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

The Scheduled Procedure ASTIM$

Entry Point:
ASTIM

Entered from:
SCHEDULE ASTIM$

Routines Called:
APART
FIXCLK
ATGTHR

Functions:
This routine updates the ASCII time/date information block.

Exits to:
RTS PC - returns to Scheduler

The Scheduled Procedure DEST$

Entry Point:
DSTINATION

Entered from:
SCHEDULE DEST$

Routines Called:
PUTINBUF

Functions:
This routine is scheduled by DESTPUT and is responsible for putting the destination block on the logging tape.

Interrupts are locked out before the call to PUTINBUF and enabled after the return.

Exits to:
RTS PC - return to Scheduler

The Scheduled Procedure DIAL$

Entry Point:
DIALIT

Entered from:
SCHEDULE DIAL

Routines Scheduled:
DIAL$

Functions:
DIAL$ controls access to the ACU handler which dials the indicated telephone number.

DIAL$ also includes code at ACUHLD which checks the timer for data terminal ready. This timer must have been "on" for at least 6 seconds before redialing is permitted. When the 6 seconds are up the modem address, CRQ, and wake up indicator DSSTIMER (a software counter) are set up and then DSSTIMER causes the ACU to be activated after a 100 ms. delay. The automatic calling unit interrupts are then handled at INTPND and INTDSS.

This routine is controlled by the semaphore DILCNT.

Exits to:
RTS PC - return to Scheduler

Dial Routines

Entry Point:
DTRRESET

Entered from:
CALL DERRESET <U>

Contents of Registers on Entry:
R5 - Linking register

Functions:
This routine resets the 6 second timer for data terminal ready at DLCLK indexed by the pointer U for the device. The counter DLCLOCKCNT is then incremented to tell the clock that DTR bits are pending a reset for 6 seconds. Interrupts are locked out when the counter is incremented.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

Entry Point:
DIAL

Entered from:
CALL DIAL <AUNIT> <ACRCNT>

Contents of Registers on Entry:
R5 - Linking register

Routines Called or Scheduled:
DIAL$

Functions:
The unit <AUNIT> and number of times to abandon call and retry <ACRCNT> are passed to this routine. It then packs these two parameters into an entry in the circular buffer at DILBEG, locks out interrupts, increments the semaphore DILCNT, schedules DIAL$, unlocks interrupts, and exits.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

Common Block Area

An area in NMM is set aside for buffers, semaphores, and the like. These particular areas and a description of their contents are as follows:

Data Area	Contents
Common Block for T-Connections	Circular buffers, semaphores controlling access to procedures, destination information blocks, telephone number tables, clock interrupt variables.
Common Block for Clocks	Common clock control words, T-ARRAY, time/date skeleton information block, clock interrupt service, ACU timing functions.
Common Block for Logging Device	Ping-pong buffers for PUTINBUF, data set data buffer.
Common Block for Idle	Common file for all of the schedule and idle tables.
Common Block for Idle Statistics	Scheduler statistics (number of times procedures were scheduled).
State Variables and Constants	All state variables used, including tables for T-connection, input interrupts and output pointers and controls for message output during ICP, controlling semaphores for each unit.

The interrupt routines included in the Common Block area are outlines in more detail below.

Clock Interrupt Routines

Entry Point:
DTCLKENB

Entered from:
JSR PC, DTCLKENB

Functions:
This section enables the data clock and loads the interrupt vector with the address DTCLKINTERRUPT.

Exits to:
RTS PC return to Scheduler.

Entry Point:
DTCLKINTERRUPT

Entered from:
Interrupt vector

Routines Called or Scheduled:
TD$

Functions:
Increments low order bits of data clock and schedules TD$ if specified number of clock overflows have occurred.

Exits to:
RTI - Return the interrupt

* * *

Entry Point:
ASCLKENB

Entered from:
JSR PC, ASCLKENB

Functions:
This section enables the ASCII clock, sets it for 100 ms interrupts, and sets the interrupt vector to ASCLKINTERRUPT.

Exits to:
RTS PC

* * *

Entry Point:
ASCLKINTERRUPT

Entered from:
Clock interrupt

Routines Called or Scheduled:
DTRRESET
TRO
TSTCARRIER
INTERFACEINITIALIZE
DESTPUT
OCTOUT
ASTIM$

Functions:
This is the interrupt service routine for the 100 ms ASCII clock interrupt. It includes code which checks to see if any 6 second DTR timers need decrementing, or if it is time to activate (wake up) the ACU. Also the carrier timers and ring timers are maintained.

Exits to:
RTI

Contents of Registers on Exit:
All registers used are restored

The Initializer

Entry Point:
INTERFACEINITIALIZE

Entered from:
CALL INTERFACEINITIALIZE <U>

Contents of Registers on Entry:
R5 - Linking register

Routines Called:
UNIBUS
INBUFINITIALIZE
DESTPUT

Functions:
This procedure initializes the indicated interface <U>. The CSR's are initialized with the appropriate preset values, speed/system recognition indicator is set, and all relevant state variables are cleared.

Interrupts are locked out while CLKISRUNNING is decremented.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

* * *

Entry Point:
INBUFINITIALIZE

Entered from:
CALL INBUF INITIALIZE <U>

Contents of Registers on Entry:
R5 - Linking register

Functions:
This procedure is used to initialize a receiving buffer for device <U> after each input line and at system start up.

Exits to:
RETURN to caller

Contents of Registers on Exit:
All registers used are restored

The Start

Entry Point:
START

Entered from:
JMP START (first executable instruction in NMM)

Routines Called:
PUTINBUF
SETIV

Functions:
The Start module determines the status of the magnetic tape device. If there are no error conditions then the NMM configuration blocks are written out to tape. Exit is made to SETIV to force the system to initialize itself.

Exits to:
JMP SETIV

* * *

Entry Point:
INTVECINITIALIZE

Entered from:
CALL INTVECINITIALIZE <VECTOR> <RCVADDRESS> <XMITADDRESS> <U>

Contents of Registers on Entry:
R5 - Linking register

Routines Called:
INTERFACEINITIALIZE

Functions:
This procedure sets up the interrupt vector for the receiver and transmitter portions of the T-connection specified by <U>.

Exits to:
RETURN to caller

The Initializer

Entry Point:
SETIV

Entered from:
JMP SETIV

Routines Called:
TRO

Functions:
This routine initializes and starts the data clock. It then jumps to IDLE, which is simply an infinite loop which retains control until an interrupt occurs.

Exits to:
JMP IDLE

B. Index to Routines

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBS TN-912	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE The Network Measurement Machine--A Data Collection Device for Measuring the Performance and Utilization of Computer Networks	5. Publication Date April 1976 6. Performing Organization Code
7. AUTHOR(S) Robert Rosenthal, Don E. Rippy, and Helen M. Wood	8. Performing Organ. Report No.
9. PERFORMING ORGANIZATION NAME AND ADDRESS NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234	10. Project/Task/Work Unit No. 6502130 11. Contract/Grant No.
12. Sponsoring Organization Name and Complete Address *(Street, City, State, ZIP)* Same as 9.	13. Type of Report & Period Covered Interim FY '75 14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

The Network Measurement Machine (NMM) is a device used to acquire data for the performance measurement of computer network systems and services. By focusing on the service delivered to network customers at their terminals, measurements can be made that are directly relevant to user needs and to management concerns.

This technical report presents the details of a data acquisition device. The device is a minicomputer-based system that employs regular (off-the-shelf) and special purpose hardware under the control of a specially written software acquisition system. The technical aspects of inserting the NMM into the data communications portion of a computer network are discussed. The detailed nature of both the hardware and the software data acquisition system is presented with a discussion of the important design decision trade-offs.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Computer networks; data acquisition; man-machine interaction; minicomputer; network measurement; performance measurement.

18. AVAILABILITY	19. SECURITY CLASS (THIS REPORT)	21. NO. OF PAGES
[X] Unlimited [ ] For Official Distribution. Do Not Release to NTIS [X] Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13. 46:912 [ ] Order From National Technical Information Service (NTIS) Springfield, Virginia 22151	UNCLASSIFIED 20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED	91 22. Price $2.00

USCOMM-DC 29042-P74

☆U.S. GOVERNMENT PRINTING OFFICE: 1976 210-801/230 1-3

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH reports National Bureau of Standards research and development in physics, mathematics, and chemistry. It is published in two sections, available separately:

- **Physics and Chemistry (Section A)**

Papers of interest primarily to scientists working in these fields. This section covers a broad range of physical and chemical research, with major emphasis on standards of physical measurement, fundamental constants, and properties of matter. Issued six times a year. Annual subscription: Domestic, $17.00; Foreign, $21.25.

- **Mathematical Sciences (Section B)**

Studies and compilations designed mainly for the mathematician and theoretical physicist. Topics in mathematical statistics, theory of experiment design, numerical analysis, theoretical physics and chemistry, logical design and programming of computers and computer systems. Short numerical tables. Issued quarterly. Annual subscription: Domestic, $9.00; Foreign, $11.25.

DIMENSIONS/NBS (formerly Technical News Bulletin)—This monthly magazine is published to inform scientists, engineers, businessmen, industry, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on the work at NBS. The magazine highlights and reviews such issues as energy research, fire protection, building technology, metric conversion, pollution abatement, health and safety, and consumer product performance. In addition, it reports the results of Bureau programs in measurement standards and techniques, properties of matter and materials, engineering standards and services, instrumentation, and automatic data processing.

Annual subscription: Domestic, $9.45; Foreign, $11.85.

NONPERIODICALS

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a world-wide program coordinated by NBS. Program under authority of National Standard Data Act (Public Law 90-396).

NOTE: At present the principal publication outlet for these data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St. N. W., Wash. D. C. 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The purpose of the standards is to establish nationally recognized requirements for products, and to provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Federal Information Processing Standards Publications (FIPS PUBS)—Publications in this series collectively constitute the Federal Information Processing Standards Register. Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service (Springfield, Va. 22161) in paper copy or microfiche form.

Order NBS publications (except NBSIR's and Bibliographic Subscription Services) from: Superintendent of Documents, Government Printing Office, Washington, D.C. 20402.

BIBLIOGRAPHIC SUBSCRIPTION SERVICES

The following current-awareness and literature-survey bibliographies are issued periodically by the Bureau:

Cryogenic Data Center Current Awareness Service

A literature survey issued biweekly. Annual subscription: Domestic, $20.00; foreign, $25.00.

Liquefied Natural Gas. A literature survey issued quarterly. Annual subscription: $20.00.

Superconducting Devices and Materials. A literature survey issued quarterly. Annual subscription: $20.00. Send subscription orders and remittances for the preceding bibliographic services to National Bureau of Standards, Cryogenic Data Center (275.02) Boulder, Colorado 80302.

Electromagnetic Metrology Current Awareness Service

Issued monthly. Annual subscription: $24.00. Send subscription order and remittance to Electromagnetics Division, National Bureau of Standards, Boulder, Colo. 80302.