AD733049

ARPA ORDER NO.: 189-1

R-664-ARPA
September 1971

ARPA Network Series: I. Introduction to the ARPA Network at Rand and to the Rand Video Graphics System

T. O. Ellis, E. F. Harslem, J. F. Heafner and K. U. Uncapher

A Report prepared for

ADVANCED RESEARCH PROJECTS AGENCY

Rand
SANTA MONICA, CA. 90406

46

MISSING PAGE NUMBERS ARE BLANK AND WERE NOT FILMED

This research is supported by the Advanced Research Projects Agency under Contract No. DAHC15 67 C 0141. Views or conclusions contained in this study should not be interpreted as representing the official opinion or policy of Rand or of ARPA.

DOCUMENT CONTROL DATA

1. ORIGINATING ACTIVITY

The Rand Corporation

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

3. REPORT TITLE

ARPA NETWORK SERIES: I. INTRODUCTION TO THE ARPA NETWORK AT RAND AND TO THE RAND VIDEO GRAPHICS SYSTEM

4. AUTHOR(S) (Last name, first name, initial)

Ellis, T. O., E. F. Harslem, J. F. Heafner, K. U. Uncapher

5. REPORT DATE	6a. TOTAL NO. OF PAGES	6b. NO. OF REFS.
September 197	48	26

7. CONTRACT OR GRANT NO.	8. ORIGINATOR'S REPORT NO.
DAHC15 67 C 0141	R-664-ARPA

9a. AVAILABILITY/LIMITATION NOTICES	9b. SPONSORING AGENCY
DDC-A	Advanced Research Projects Agency

10. ABSTRACT

An overview of the Advanced Research Projects Agency's experimental computer network, and a technical description of the Rand Video Graphic System that links Rand computing resources into the network. Based on principles of distributed communications without a central control point, set forth in a 1964 Rand publication series, the ARPA network has 18 nodes, located at 10 university sites, 4 research institutes (Rand, SDC, SRI, Mitre), 2 manufacturers (BB&N, Burroughs), and Rome Air Development Center. Computers of different make, model, size, speed, hardware, and software are interconnected by small special Interface Message Processor computers at each site. SRI handles all network documentation. UCLA analyzes performance statistics. BB&N coordinates maintenance and testing. Rand will experiment with information processing techniques. Network control programs are being written at each site. The Video Graphic System and its relation to the network are described and illustrated.

11. KEY WORDS

Advanced Research Projects Agency
Information Systems
Networks
Video Graphic System
Computer Graphics

ARPA ORDER NO.: 189-1

R-664-ARPA
September 1971

ARPA Network Series: I. Introduction to the ARPA Network at Rand and to the Rand Video Graphics System

T. O. Ellis, E. F. Harslem, J. F. Heafner and K. U. Uncapher

A Report prepared for

ADVANCED RESEARCH PROJECTS AGENCY

Rand
SANTA MONICA, CA. 90406

PREFACE

This report is the first of a series that will describe experiments conducted on the ARPA Experimental Network by the ARPA Information Processing Techniques (IPT) Project at Rand. Sponsored by ARPA, the project is an integral part of both the client's and Rand's overall program to explore the utilization of computer resources applicable to military environments.

The problem addressed by the ARPA Network, and by Rand as a participating node, is how to economically share heterogeneous computer resources that are separated geographically. Network development is divided into two phases: (1) research and experimentation, and (2) provision of services to an extended ARPA research community. We are approaching the second phase, that is, some services are being offered on a limited basis, but experimentation and development will continue.

As introductory material, this report gives an overview of the initial work completed at Rand. Its readers include Rand researchers who will use the Network to access remote resources, and users at remote sites who will use Rand-provided resources. Background and overview of the Network are given for potential users who are not familiar with its operation and scope. A description of the Rand Video Graphics System is included for those unfamiliar with our configuration.

Because no services are yet offered by Rand, this report is neither a tutorial nor a user's guide to services. Rather, the functional organization and operation of the Network is explained and reference is made to the *kinds* of uses (services) to which Rand experiments will be directed. The motivation and results of future experiments will be contained in subsequent reports of this series.

SUMMARY

This report is an introduction to the ARPA Network at Rand. Overviews of the Network and of the Rand Video Graphics System are presented.

The ARPA Network is a distributed digital network with nodes located on the premises of various ARPA contractors. It can be viewed as a research project to examine ways of sharing computer resources among different kinds of computers and operating systems. It can also be viewed as a facility that provides services to a large group of users.

The results of previous Rand studies on distributed digital communication systems were useful as a foundation for the design of the ARPA Network. The ARPA Network was designed by an Interface Message Processor Group consisting of representatives from many of the ARPA contractors along with ARPA itself, and was implemented by Bolt, Beranek and Newman, Inc.[†]

The structure and operation of the Network are described. A subnet is formed by interconnecting small computers called Interface Message Processors (IMPs) through wideband communication lines. The various host-computers of the ARPA contractors are then connected to the IMPs to form the Network. Hosts converse via the IMPs. The IMPs switch standard-size message packets by a store-and-forward technique. There are provisions for multiplexing messages, acknowledging receipt of messages, and finding the beginning and end of a message.

Protocol between the host-computers is managed by Network Control Programs (NCPs) that maintain logical message paths and control the rate of information flow between processors. The NCP operations were specified by representatives of the participating sites. NCPs are now being implemented at each site.

Hardware comprising the Rand Video Graphics System and its relation to the Network is described. Implementation strategy of the NCP within the framework of our current system is stated. The NCP behaves as a mini-system controlled by a task supervisor and several types of asynchronous (software) interrupts.

[†]Located at 50 Moulton Street, Cambridge, Massachusetts, 02138.

The short-range goals for user programs are identified. A Network Service Program, currently being implemented at Rand, is offered as an approach to combining experimentation and services to users into a common program.

ACKNOWLEDGMENTS

The authors would like to thank the following persons for their careful review and helpful comments on this report: Dr. Robert Kahn of Bolt, Beranek and Newman, Inc.; Dr. Gabriel F. Groner and Mr. Michael Warshaw of The Rand Corporation.

CONTENTS

FIGURES

I. INTRODUCTION AND OVERVIEW

THE NETWORK CONCEPT, PURPOSE, AND GOALS

The nationwide ARPA Network [1-5] is composed of ARPA contractors at geographically separated sites. Different computers at these sites are interconnected by small, standardized computers and 50-Kbit communication lines. The computers vary in make, model, size, speed, and other hardware and software features, as shown in Table 1. The Network is a distributed network (as opposed to a centralized network with a central control node) and traffic routing is governed adaptively by the standardized computers over redundant network paths. Via this Network, each participant can reliably access the various remote resources, such as programs, data, and unique hardware facilities. Individual programs at the sites control information flow.

Of primary concern are the fundamental intercommunication problems inherent in the marriage of autonomous hardware and software. No attempt has been made to provide compatible equipment in order to, for example, transfer large programs as a means of resource sharing. On the contrary, the necessity for large program transferability can often be eliminated by transmitting small messages in a predetermined format over standard interfaces, to be operated on by programs written to run in the remote environment. This approach greatly minimizes reprogramming efforts and reduces concern for developing standard languages and for improving hardware compatibility. The Network will make technological and human resources more readily available at less expense. An obvious outcome of the Network environment is increased cross fertilization of research products.

There are several other motivations for this research. The total number of computers at multiple computer installations may be reduced. Such tangible Network facilities as mass storage may be remotely accessed and more efficiently utilized. And the area of interprocess communication can be more thoroughly explored.

A project goal, then, is to discover and validate techniques and mechanisms attendant to uniform and easy access to all available resources, independent of hardware and software dissimilarities. The

Table 1

NETWORK NODES
(October 1971)

CONTRACTOR AND LOCATION	COMPUTER(S)
Bolt, Beranek and Newman, Inc. Cambridge, Massachusetts	PDP-10 PDP-10 DDP-516 Terminal IMP
Burroughs Corporation Paoli, Pennsylvania	B6500
Carnegie-Mellon University Pittsburgh, Pennsylvania	PDP-10
Case Western Reserve University Cleveland, Ohio	PDP-10
Harvard University Cambridge, Massachusetts	PDP-11 PDP-10 PDP-1
Massachusetts Institute of Technology Lincoln Laboratory Lexington, Massachusetts	TX2 IBM 360/67
Massachusetts Institute of Technology Project MAC Cambridge, Massachusetts	GE 645 PDP-10
Mitre Corporation McLean, Virginia	Terminal IMP
NASA Ames Research Center Moffett Field, California	IBM 360/67 Terminal IMP
Rome Air Development Center (ISIM) Rome, New York	GE 635/645 Terminal IMP
Stanford Research Institute Artificial Intelligence Group Menlo Park, California	PDP-10
Stanford Research Institute Augmentation Research Center Menlo Park, California	PDP-10
Stanford University Computation Center Stanford, California	PDP-10
System Development Corporation Santa Monica, California	DDP-516 (IBM 360/67)
The Rand Corporation Santa Monica, California	IBM 1800 (IBM 360/65) (PDP-10)
University of California at Los Angeles Computer Science Department Los Angeles, California	Sigma 7 IBM 360/91
University of California at Santa Barbara Santa Barbara, California	IBM 360/75
University of Illinois Center for Advanced Computation Urbana, Illinois	PDP-11
University of Utah Computer Science/IRL Salt Lake City, Utah	PDP-10

inventory in Table 1 shows the variety of computer types in the present Network configuration.

More specifically, we aim to make remote services as easily accessible as local ones, without noticeably degrading overall performance. Although services could be shared as in a time-sharing system, the distance between user and service could be vastly extended. Likewise, the number and size of services would not be constrained, as in a typical time-sharing system. Another goal is to allow more flexibility in the use of programming languages. Because services would be offered remotely, compatible languages allowing program transferability would not be required.

DISTRIBUTED COMMUNICATIONS

In the early 1960s, Rand began a study [6-17] of distributed digital communications that included system design, communications routing, and performance measurement aspects of distributed digital communications networks. Such a network was designed as the ARPA Network [18-19] by many of the contractors listed in Table 1, along with ARPA and others. American Telephone and Telegraph designed and implemented the circuits, ARPA provided the "plan," Network Analysis Corporation selected the topology, Honeywell fabricated the subnet equipment, and Bolt, Beranek and Newman, Inc. (BB&N) was responsible for most of the subnet design, installation, and checkout.

PLANNED USES OF THE NETWORK

Such a network has many uses. Of greatest interest, however, are those that readily allow exploration of communication methods among different operating systems. One such generic use is *program sharing*, in which data are transmitted to a remote program and results are returned. Another is *data sharing*, in which small programs or algorithms are transmitted to operate on a large, remotely located data base. Also of general interest are *remote services*, in which a remotely located data base and program are queried. Other possible candidates are *load sharing* and *message services*, neither of which appears to be as interesting at present.

Several examples clarify prospective benefits. Program sharing will occur when the ILLIAC-IV (to be included in the Network) acts as a remote facility for simulation and modeling functions. This machine performs such functions in a fraction of the time required by conventional hardware. Parameter data will be transmitted from remote sites as input to large simulation programs that run on the ILLIAC-IV. Results of the simulations will be returned to the remote sites for analysis. A similar form of program sharing is planned by Rand: researchers will run simulation programs (too large for Rand's machines) on the Remote Job Service (RJS) at the University of California at Los Angeles (UCLA). The data-storing and post-processing of the simulation results can be performed via Remote Job Entry (RJE) at the University of California at Santa Barbara (UCSB). The data base produced by the RJS can be remotely stored and--in an example of data sharing--accessed via RJE. In this instance, small post-processing programs are transmitted to UCSB to operate on the simulation results. The graphical output of the post-processing programs are returned to Rand for local display.

Several types of remote service are already in use, but like RJS and RJE, these services will be more available and less costly when incorporated into the Network. For example, UCSB's On-Line System provides a remote service by permitting human interaction in mathematical analysis; Stanford Research Institute's (SRI) TODAS allows remote construction and modification of arbitrary text files.

USE OF THE NETWORK BY BROADER DISCIPLINES

The current user population consists of about 2000 persons, whose interests are primarily in computer sciences technology. Some ARPA-sponsored projects in this area that will make use of the Network are artificial intelligence and computer system architecture. However, ARPA-sponsored research that will benefit by Network use is variegated, such as seismology, climate dynamics, and behavioral science.

For example, the *modus operandi* of an ongoing climate dynamics project at Rand involves cross-town courier service for computer runs on a machine just adequate for the problem. The effects of this operation are long turnaround time and less than optimal data output, because of machine size limitations. The Network can eliminate or reduce both problems. It will be possible to submit computer jobs and obtain results from the researcher's office, thus eliminating courier service. Sufficient machine power will be available via the Network to produce the desired results in less time.

A TRANSFER OF NETWORK TECHNIQUES

Finally, large corporations would, hypothetically, employ Network principles to save the expense of many terminals and communication facilities to access various services. These principles would allow a single terminal to access those services. Services that are now prohibitively expensive for small companies could be made economical. Likewise, the central processor requirements of these companies could be reduced.

PARTICIPANTS WITH SPECIAL NETWORK-ORIENTED FUNCTIONS

Most of the current and impending Network sites shown in Table 1 are expected to contribute a unique resource to the Network community. In particular, the operations of three current members will fulfill the following special functions.

SRI is responsible for a Network Information Center (NIC), which will provide access to documentation of Network experiments as well as information preliminary to the use of remote resources. For example, a researcher might use this service, via the Network, to discover what programs are available and how to use them. SRI provides a partial library in hardcopy to each site.

Another special operation is the Network Measurement Center (NMC) at UCLA. Because the Network is experimental, it was programmed to sample its own performance and send these statistics to UCLA for analysis. The two classes of measurements are message tracing through nodes and measuring the activity of a single node. From these data, for example, one can observe the performance of the Network message routing algorithm under varying Network loadings.

The third special function is the Network Control Center (NCC) maintained by BB&N. The NCC coordinates Network maintenance and testing.

The Network is programmed to send status information on itself and on the communication lines to BB&N, which monitors Network performance and operations.

ACCESSING THE NETWORK FROM RAND

Rand will provide access to the Network from the Rand Video Graphics System (VGS) consoles [20]. The VGS consists of such shared electronic components as scan converters. Costs are amortized over many concurrent but different applications. Twenty-six low-cost graphics consoles are available throughout Rand.

Figure 1 illustrates the Rand configuration. Users of VGS consoles communicate with the Network by programs operating in the IBM 360 service machine. The communication paths pass through an 1800, which is used as a message switch.

Perhaps the best way to describe how these consoles will be used as a Network access is to describe some current in-house uses of VGS. Thus far, the VGS has been used mainly to support research in man-machine communications. Thus, the system's ability to provide real-time response of machine interpretation of human gestures (via the Rand Tablet-stylus input device) is vital. This capability is used for such functions as real-time automatic handprinted character recognition and supplying highly responsive full-graphic editing. However, the computer researchers and the programming staff of the Rand Computation Center also need remote job entry, program debugging, and program editing features. These are currently supplied via IBM's Conversational Programming System (CPS II) [21], retrofitted with a graphic interface for VGS (keyboard and Tablet input, and graphic display output), and other support software programs.

In addition, the VGS is used extensively where problem-solving is enhanced by superimposed computer-generated pictures and noncoded information from a TV camera.

Figures 2-5 represent the various VGS uses. Figure 1 shows a portion of a displayed "page" of programming text used with graphic CPS II. Figure 3 shows graphs of simulation results using a biomedical modeling

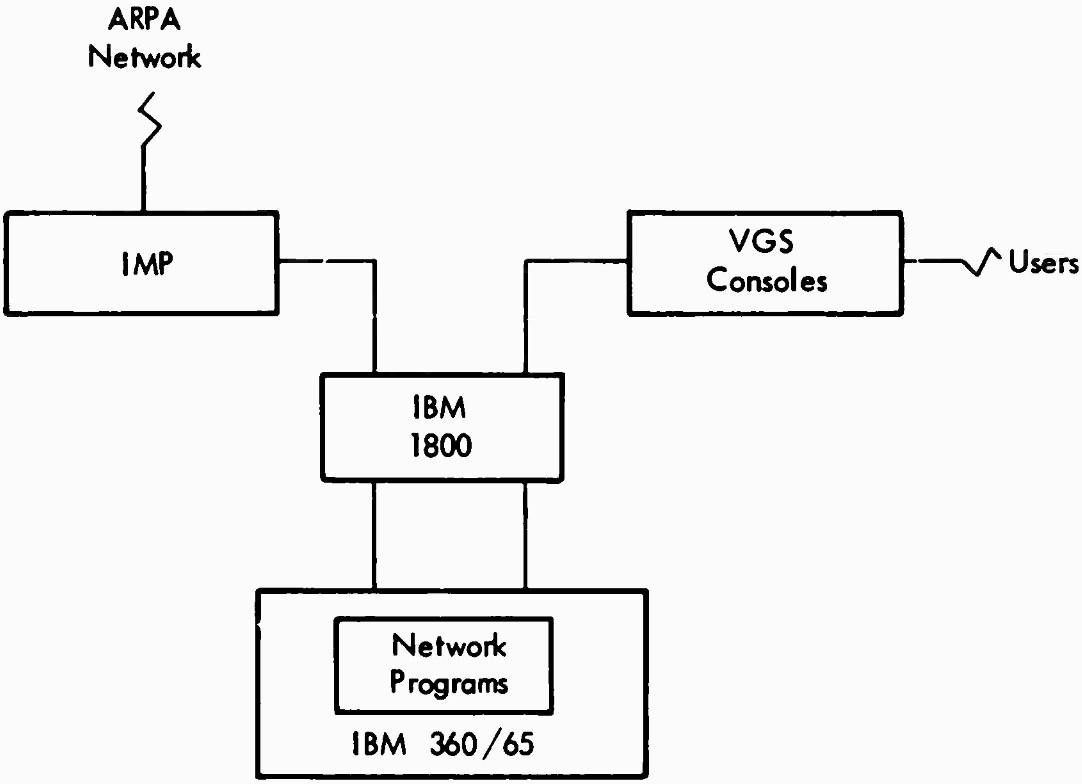

Fig. 1—VGS console access to ARPA network

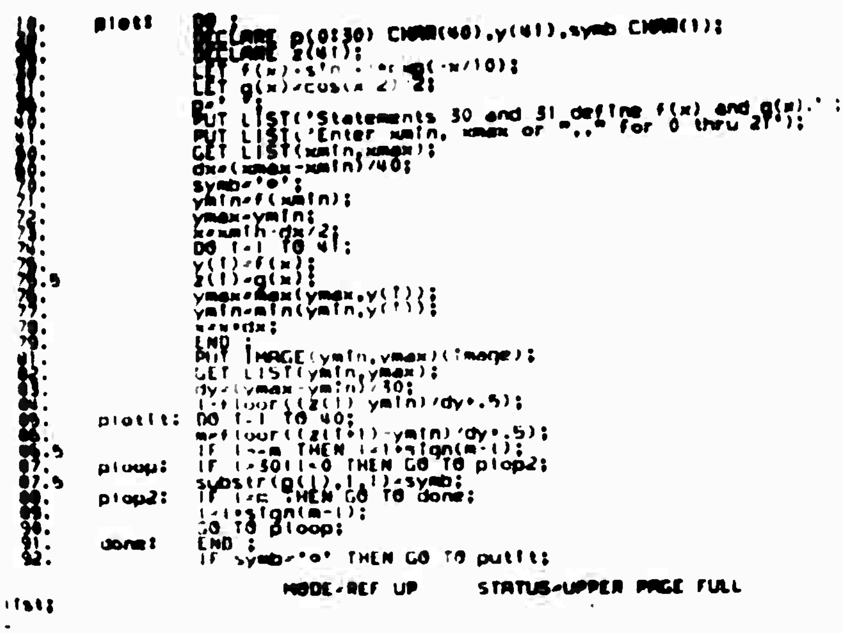

Fig. 2—Sample of CPS program text printed on screen at 74 characters per line by 52 lines per screen height

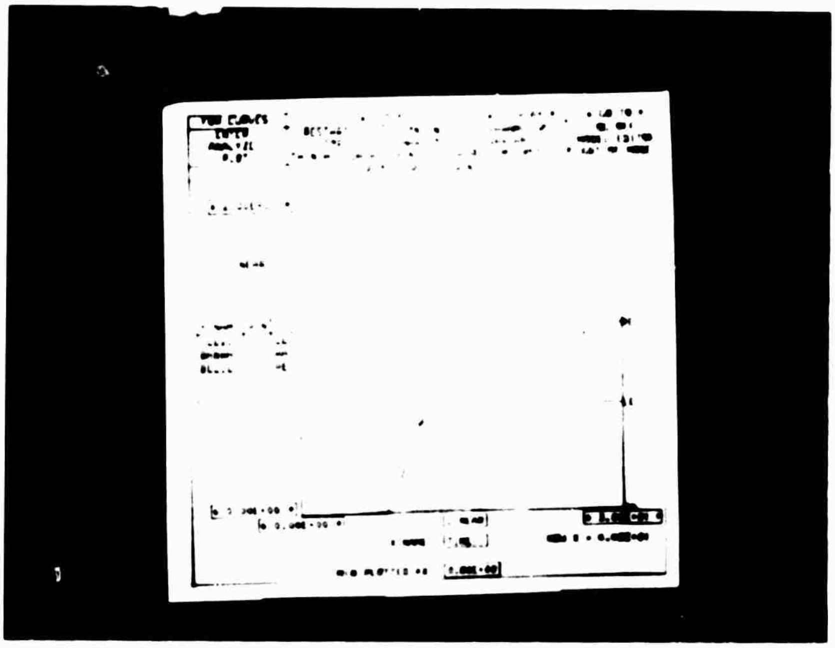

Fig. 3 — Graphs of simulated results of biomedical modeling system

NOT REPRODUCIBLE

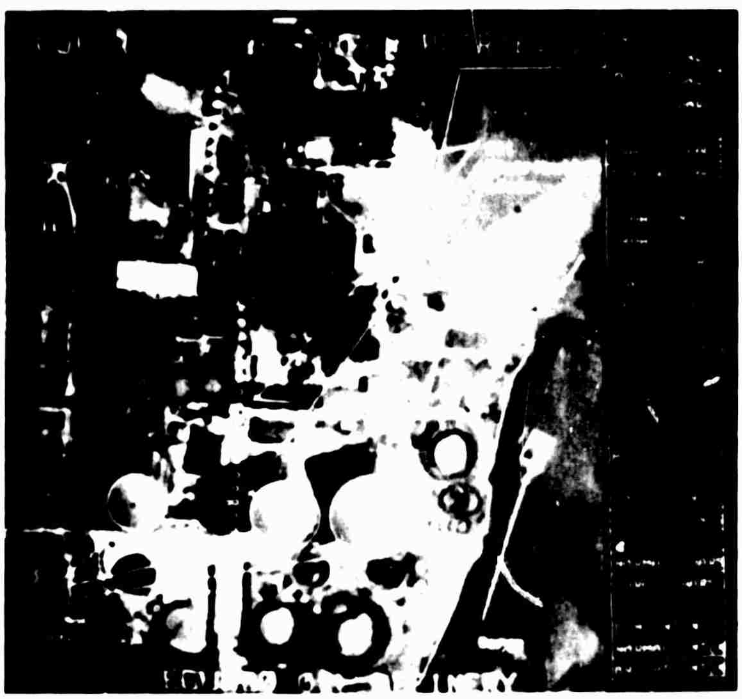

Fig. 4 — VGS application of annotation of photograph

system [22]. Figure 4 shows the application of VGS to the photo-interpretation process used in an intelligence application. The picture is a composite of an image from a TV camera looking at a photograph and user-described annotation that is digitally generated as a result of tablet-stylus action. The right side of the picture shows a column of computer-generated "display buttons" that allow choice of line brightness, line width, and solid or dashed lines, and special graphic structures to permit on-line annotation of the picture via the Rand Tablet stylus used as the input device. For this application, there is generally no need to digitally store the complex grey scale picture. Annotation is user-generated, based on features important to the interpreter. The annotation is either generated and stored on-line and available for later review, or a transparent hardcopy overlay can be generated by a film recorded for use with the original photograph.

Figure 5 shows a partially constructed flowchart description of a biological model. The flowchart elements are drawn using a Tablet stylus.

The design and development of VGS was supported by ARPA. The image display portion of the system was developed under a subcontract with IBM, Advanced Systems Development Division, Los Gatos.

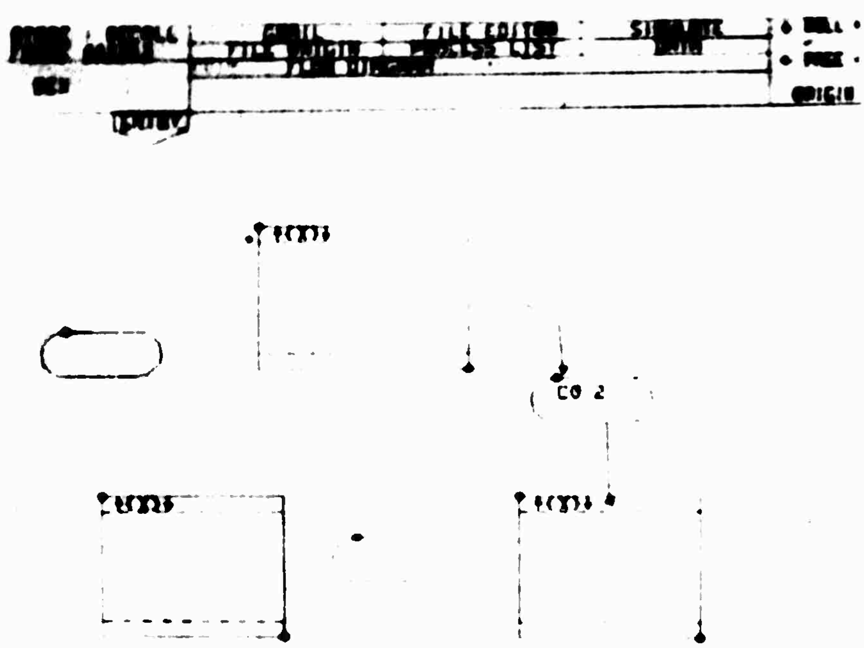

Fig. 5—Close-up view of picture display from a block diagram program

II. HOST-TO-IMP AND IMP-TO-HOST COMMUNICATIONS

NETWORK CONFIGURATION

The different computers at each site, called hosts, are each connected to a small computer, called an Interface Message Processor (IMP), on the host's premises. The Network is completed by interconnecting IMPs through 50-Kbit full duplex communications lines leased from the common carriers (Fig. 6). The IMPs employ a heuristic routing and store-and-forward switching algorithm to pass messages. A message is sent from a host to its IMP, passed from IMP to IMP, ultimately goes to the destination IMP, and then to the destination host.

The IMP is a modified Honeywell DDP-516; the standard interface is built by BB&N (Fig. 7). Other IMP equipment includes a teletype, paper tape reader, and one to four modems[†] for connection to the communication lines.

MESSAGES

A host communicates directly with the IMP on its premises by sending and receiving small, fixed-length control messages. Hosts communicate with each other through the IMPs by transmitting regular messages of variable length. Messages consist of a control leader and text. Reference 19 gives a complete description of leader formats and their meanings.

LINKS

A link is a logical simplex connection[‡] between two programs in remote hosts. The logical links allow multiplexing outgoing messages and distributing incoming messages within the hosts. Programs may have multiple concurrent links.

[†] A modem is a modulator/demodulator device that converts data between a data processing form and a transmission facilities form.

[‡] A simplex connection permits transmission in one direction only, as opposed to a half-duplex connection, which permits transmission in both directions but not simultaneously.

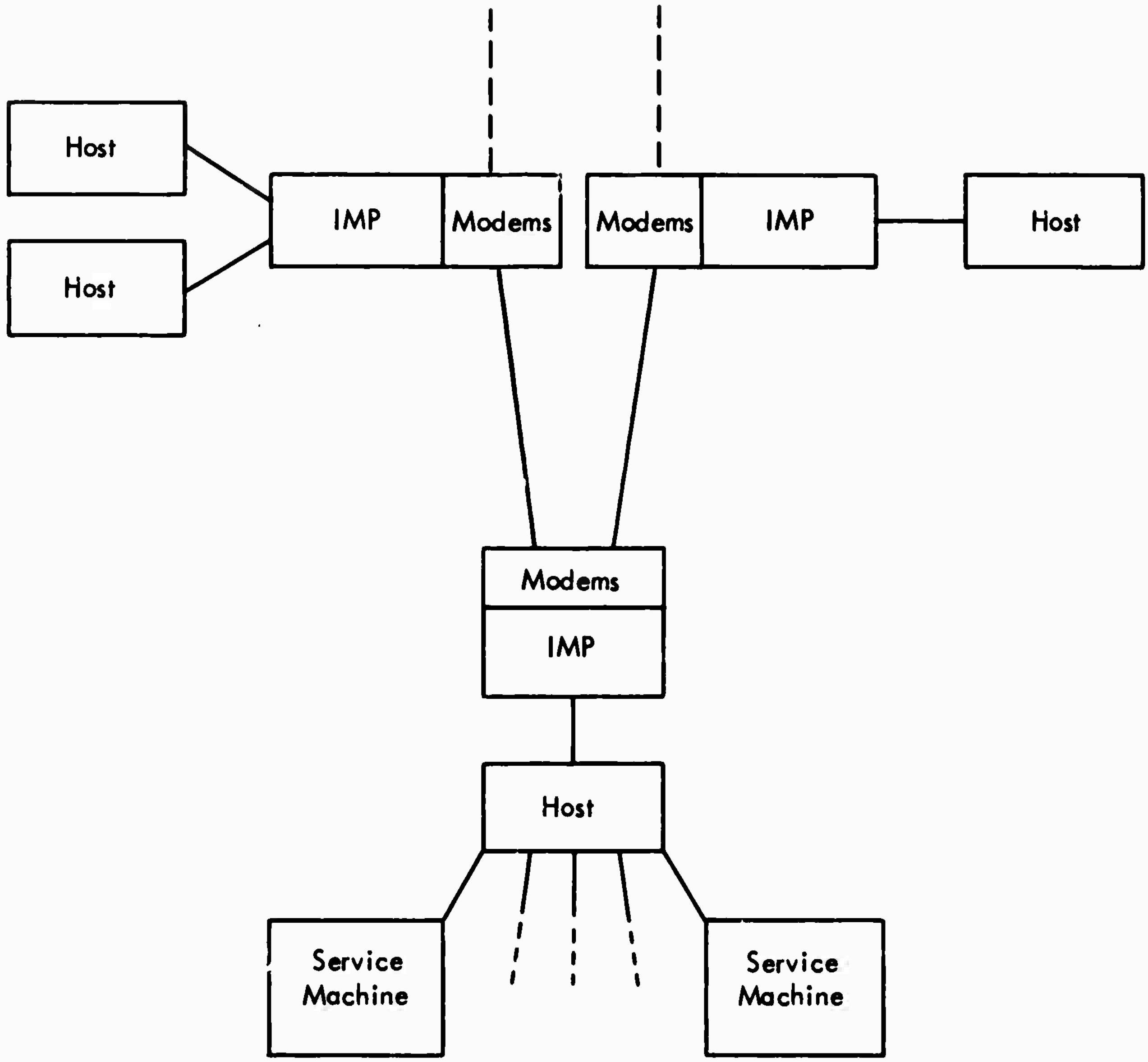

Fig. 6--Network Configuration

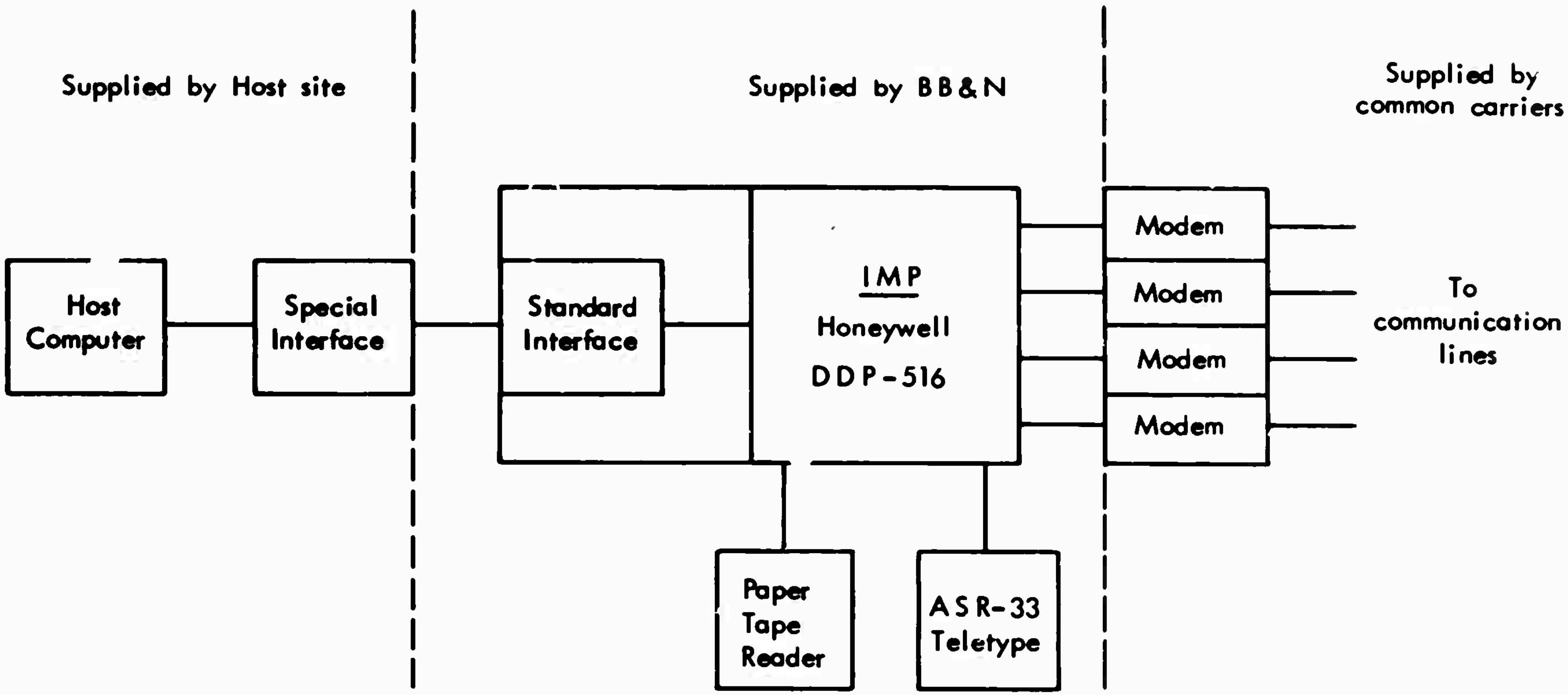

Fig. 7--Typical Equipment at Each Node

A link is uniquely identified by a source host number (predefined for each host and known to the IMPs), a destination host number, and an identification (multiplexing) number.

ACKNOWLEDGMFNTS

During transmission of a message over a link, the link is unavailable for other use. After the message has been delivered by the destination IMP to its host, a ready-for-next-message acknowledgment is returned to the source host by the destination IMP and the link is made available for further use.

END-OF-MESSAGE DETECTION

The end of a message is "padded" by the interface and IMP hardware to permit a receiving host to locate the lest meaningful bit of the message. The padding consists of an interpretable bit pattern that terminates on the receiving host's word boundary.

HARDWARE ANOMALIES HANDLED BY THE HOSTS

The hosts, rather than the IMPs, are responsible for resolving two hardware difference problems. First, where processor word lengths differ, reformatting is done by the receiving host. Second, "marking" the beginning of a message (again due to mismatched word lengths) is done by the sending host. Similar to padding the message end, the sending host marks the start of the message beyond the leader by an interpretable bit pattern that terminates on the sending host's word boundary.

III. HOST-TO-HOST COMMUNICATIONS

THE NETWORK WORKING GROUP (NWG)

An NWG, composed of representatives from ARPA and from each Network site, coordinates experimentation and development of the Network and provides a convenient forum for discussing and formulating Network protocol. The initial task of the NWG has been to functionally specify the Network Control Program (NCP)--the standard protocol manager between hosts (discussed below). This specification has been completed and is being implemented by each site. Current endeavors of the NWG include studying the requirements for problem program communications.

NCP DEFINITIONS AND FUNCTIONS

A standard host-to-host protocol makes the study of program-to-program communications more convenient. It interfaces the program/Network at the host level rather than at the IMP level and also allows a larger set of identifiers for message paths (than that provided by the IMPs) to be defined. (If programs are to operate independently, they must have their own unique set of identifiers for Network logical message paths.)

This standard protocol manager is the NCP[†] provided by each site, usually, but not in all cases, within its system software. The NCP can accept and generate Network messages in a standard format. The messages include the link information described in Sec. II. For Network transmission, the NCPs multiplex outgoing messages over the links and distribute incoming messages to programs.

The NCP functions, implemented independently in each host computer, insure a common dialogue at the host-to-host level throughout the Network. The NCPs support establishing or terminating connections, message-flow control, interrupts, echoes, and notification of detected errors. Typically, NCPs are embedded in the host's system software and their functions invoked through system calls.

[†] Reference 5 describes an earlier version of this protocol.

One NCP-supported function, the connection, is a simplex message path between two remote programs. It is similar to the link of the IMP-to-host protocol, but provides a larger range of identifiers. The respective programs identify a connection by means of a pair of identifiers, called sockets (one at each end). Because a connection is a simplex, one socket is designated a receive socket and the other a send socket.

Another NCP-supported function, the message-flow control technique that exists between user programs, differs from that employed at the IMP-to-IMP, IMP-to-host level, where message rate is controlled by acknowledgments. The same message-transmission acknowledgment is present at the host-to-host (program-to-program) level, but flow control over a connection must be further regulated by the programs themselves. The receiving program supplies the sending program with a buffer-allocation count for messages to be received. When the allocation is exhausted, the sending program's NCP must force temporary suspension of message flow until the receiver replenishes the allocation.

Because implementation of NCPs and higher-level software is installation dependent, the next section limits the discussion to the Rand setting.

IV. THE RAND VGS

OVERVIEW

Rand's computer hardware related to this report consists of four major subsystems: (1) the IBM 360/65 service machine, (2) the VGS, which includes an IBM 1800 with a picture generator and a distribution system, (3) the consoles, and (4) the IMP.

Note in Fig. 8 that Rand's Network host-computer is an IBM 1800. Typically used for process control, it serves here as a communications control interface to other service machines (currently one IBM 360/65) in support of the VGS. The IBM 1800 is responsible for (1) buffering and routing information from consoles and IMP to the service machine, (2) buffering and routing information from the service to the picture generator and the IMP, and (3) providing appropriate high-rate feedback to such input devices as line-at-a-time buffering, editing for keyboards, and Rand Tablets. By connecting the IMP to the host IBM 1800, existing message queueing and routing services were used to provide Network access as well.

The VGS is designed to serve many highly interactive users and to load share across a spectrum of applications in a time-shared, multi-user, problem-solving environment.

DESCRIPTION

Figure 8 indicates the gross aspects of the system now operating at Rand. The basic system concept is characterized by a central set of shared hardware currently connected to the 360/65 and 26 consoles [23-25].

Perhaps the most technically interesting portion of the system is the Image Distribution System (IDS) that lies between the 1800 and the consoles (Fig. 9). The IDS consists of a microprogrammed graphic display control, a picture generator, 3 scan converters, a 32-channel central buffer store, a distribution panel, a graphic compare, and a raster compare. The more important components are described below.

The picture generator is a high-quality line and character generator. It converts the graphic order codes and data to x, y, and I

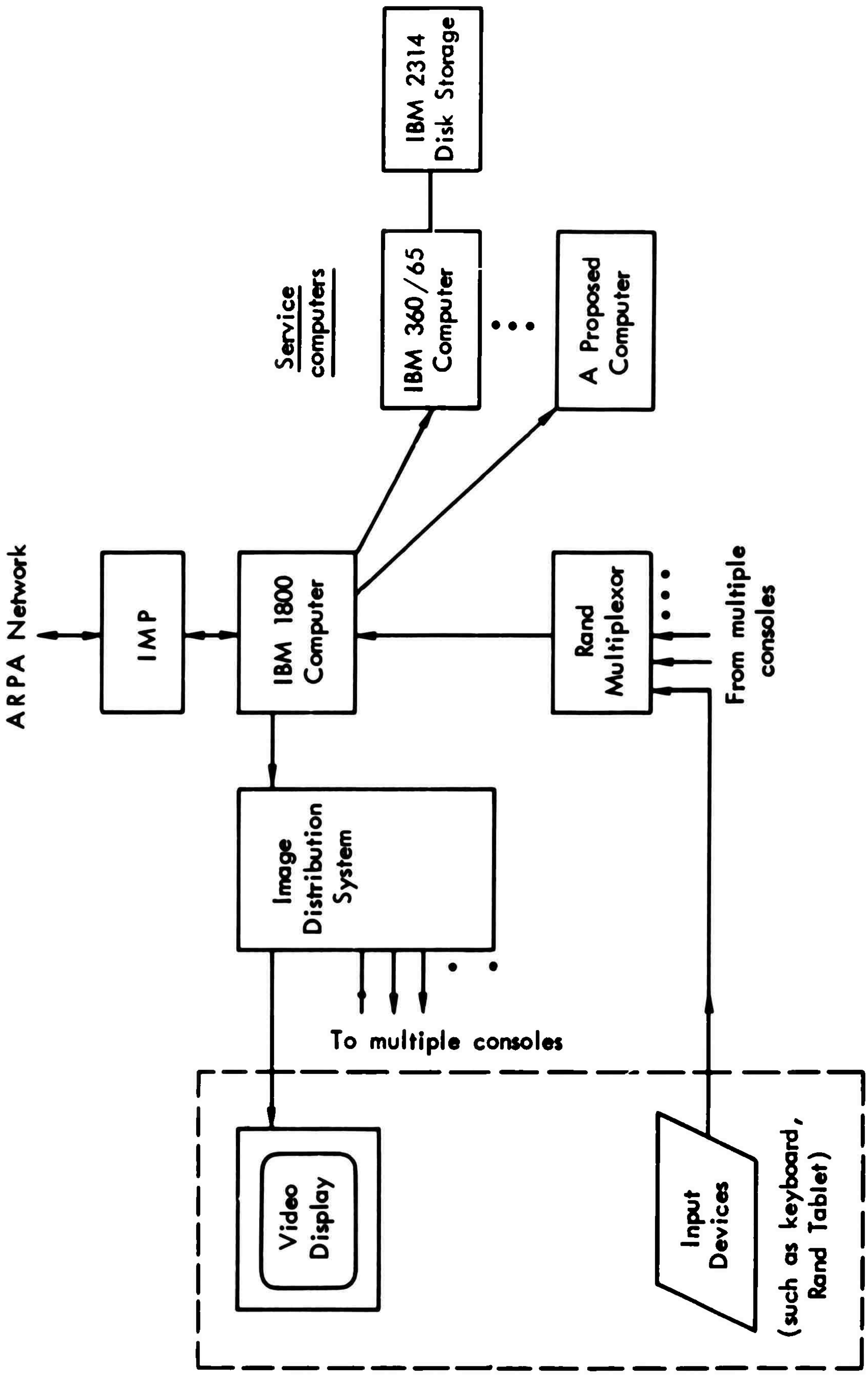

Fig. 8--Rand Hardware Configuration

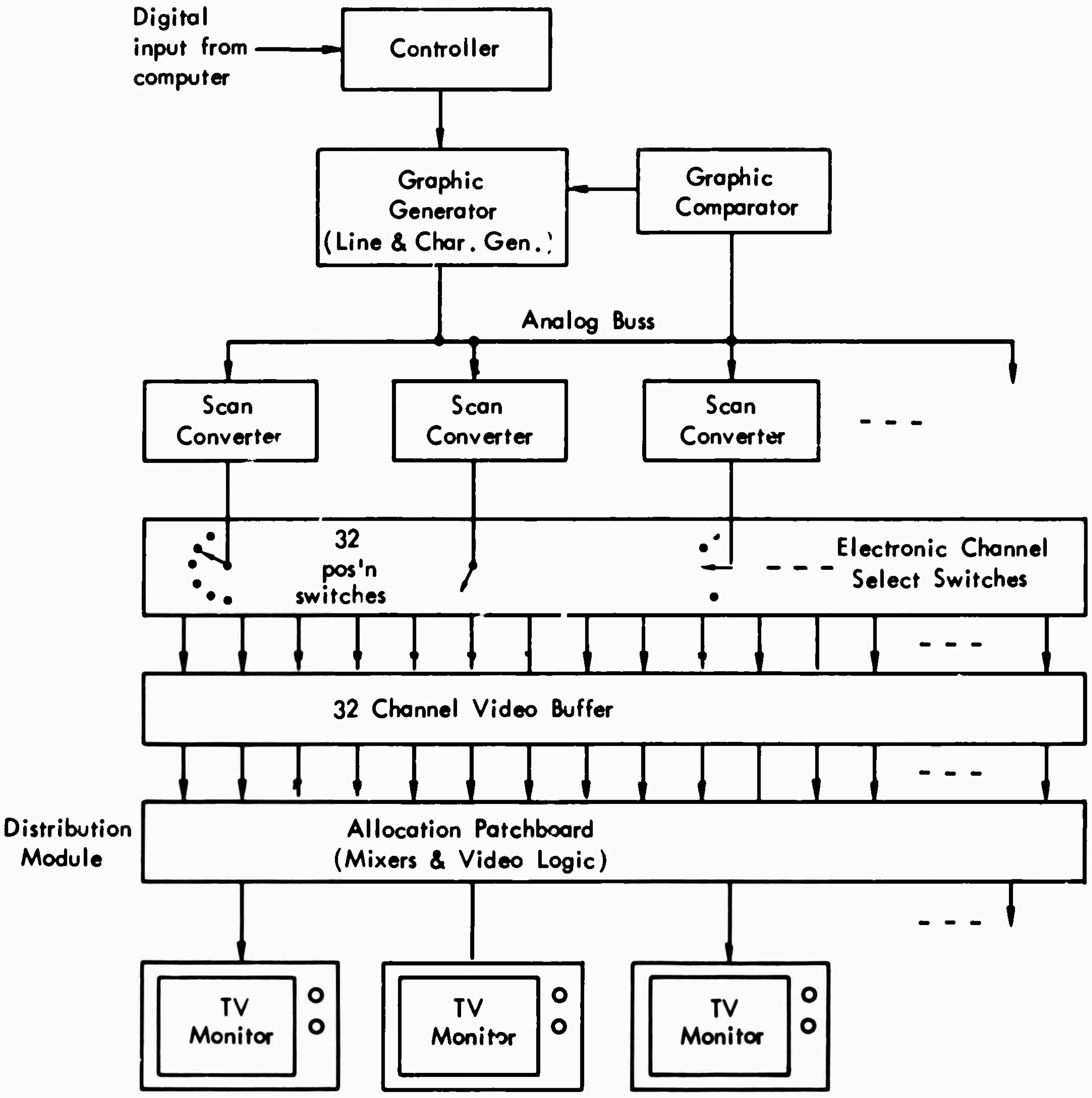

Fig. 9--Image Distribution System

(intensity control) analog vectors to be written ("painted") on a selected scan converter.

The scan converter translates the x, y, and I signals to an electronic charge pattern on the target of a recording vidicon scan-conversion tube. A picture may be painted in any sequence desired. Once the charge pattern is painted on the scan converter target, the picture generator is available for generating another user's picture on another scan converter.

Before the paint cycle, any previous image on the target scan converter is erased, using a high-intensity strobe light to discharge the photoconductor target. Once a picture is painted, the scan converter is switched to the raster scan mode and the stored image is scanned off (read) in an 873-line raster format. The scanned signal is stored on the selected video buffer channel (used for centralized refresh of the consoles) and displayed on the selected console. The scan converter is then ready for another paint cycle.

The common video buffer is a 32-track, single read-write-head-per-track disk. Each channel stores and refreshes one TV picture.

Dynamic portions of a picture are updated by generating a change in a service computer (the 360/65). For some applications, only the changed information is routed via the 1800 to update a track. Therefore, for text use only, one buffer channel per console is sufficient, because a partial rewrite feature may be used to replace a single line of text. The partial rewrite feature allows rewriting any size horizontal band (from one line to full screen) of the picture scan. This band may be positioned at any vertical position desired. For dynamic, full graphic pictures, two channels per console may be used, one for static portions of the picture and one for dynamic portions. This eliminates rewriting the static part each time the dynamic part is changed.

Signals from several buffer tracks can be summed linearly for transmission to the appropriate console. Noncoded information, such as natural pictures from a television camera, can be summed with coded pictures.

The distribution module is the output element of the IDS. Its functions are

1. To connect consoles to the appropriate IDS channel;
2. To linearly mix, as needed, signals from the buffer channels as well as externally generated TV signals.

In certain types of graphic applications, it is desirable for programs to have the system hardware examine a picture to determine if any visible elements occur in a certain area of the display. This function is called graphic compare; it allows a program to detect the presence and identity of an image element within a variable program-defined rectangular "window" within the display area. The system can perform this test with or without concurrent track rewrite.

CONSOLES

The consoles are designed to act as the user's personal property--that is, they are inexpensive enough to be dedicated to a person with moderate to heavy use rates. Consoles are designed to operate in the user's most productive problem-solving environment, usually his office. Two coax cables connect each console to the distribution module.

Figure 10 diagrams the console. The console modem allows the user a choice of up to eight simultaneous input devices. Thus, the user has input flexibility to match the output flexibility of graphically displayed information. Current input devices available are a selectric keyboard (standard on all consoles), a Rand Tablet, a Graf Pen tablet, a light pen, and a fan-out device that will accommodate up to 32 data-entry keyboards per station.

The keyboard has upper- and lower-case symbols, a third case for special symbols, and a fourth case for user-described function keys. The console has a small set of control buttons and lights to constantly inform the user of system status.

Figure 11 pictures the 17-in. CRT version of the console. This console can display 52 lines of characters in a 74-character line, flicker-free format. Each station can handle full graphics and continuous gray scale. The TV monitors are provided with a polarity

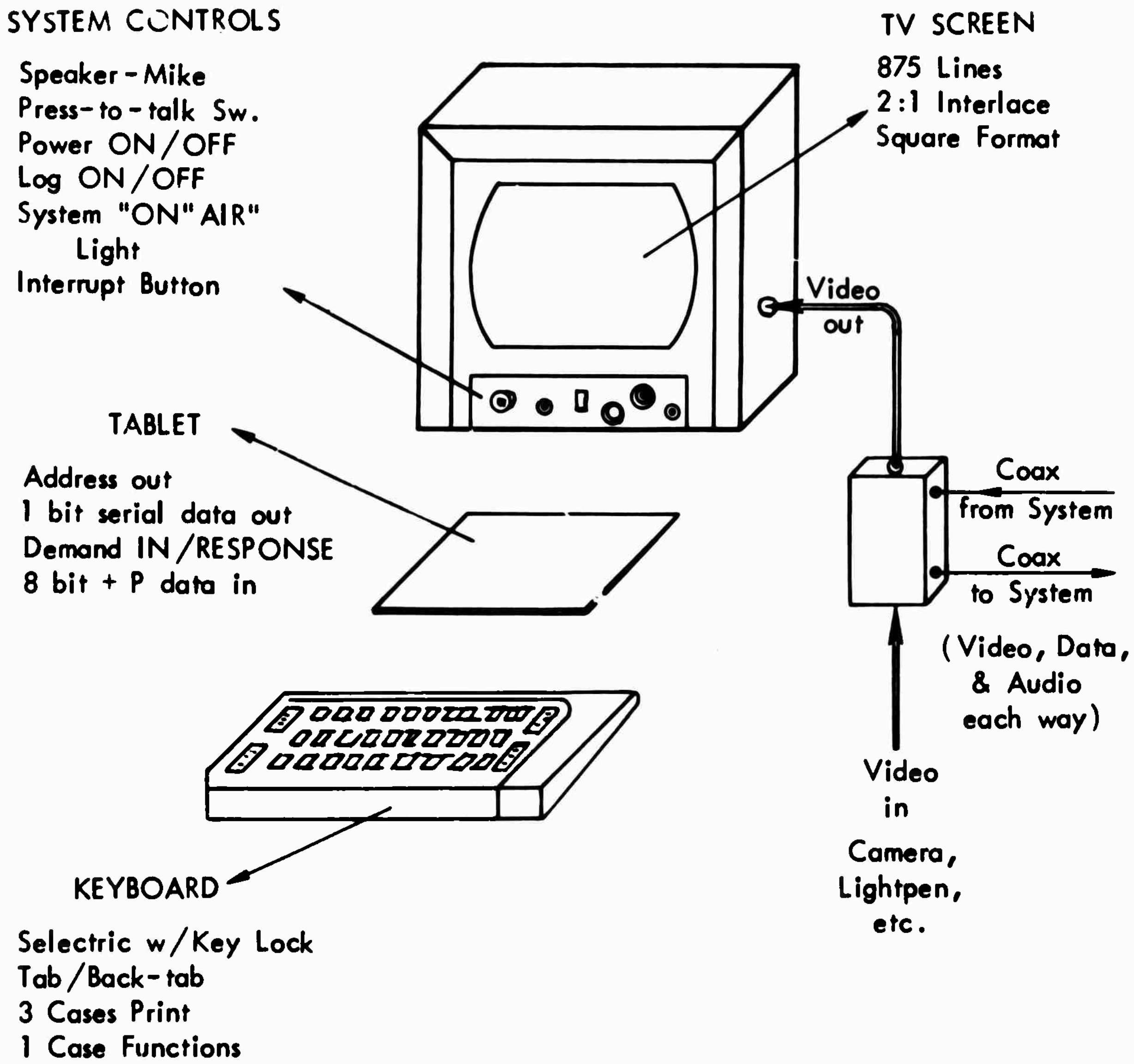

Fig. 10--Block Diagram of a Console

Fig. 11--Console with Rand Tablet and Keyboard

switch, which allows the user a choice of black on white or white on black formats.

SYSTEM PERFORMANCE

The major design criterion is a sufficiently responsive and sharable system that allows any of several concurrent users to utilize an interaction capacity appropriate to his problem. The averaging power of many concurrent users allows appropriate economic sharing of the hardware. Uniform line-widths, uniform brightness, and line closure are important, and are state-of-the-art.

The VGS was installed at Rand in October 1968. It has been operational virtually 24 hr a day, 7 days a week since installation. Overall reliability has been remarkably good. Downtime has been trivial and maintenance requirements minimum.

V. THE RAND SYSTEM SOFTWARE

Rand's system software packages consist of the IBM 1800 System Software, the IBM 360/65 MVT Operating System (OS), the Video Operating System (VOS), the Integrated Graphic System (IGS), the NCP, page and graphic interfaces to IBM's CPS, and a set of FORTRAN-callable routines, called VPACK, for interactive and graphic needs (see Fig. 12).

IBM 1800 SOFTWARE

Users treat the 1800 software as an extension of the hardware. The 1800's primary functions are to set up, protect, and manage traffic through the logical information flow paths of the console-service machine. Other functions relate to error analysis and recovery, rote task and high interrupt-rate processing, and buffering to relieve service-machine loads.

An example of the latter is the single line-text-editing function. One way for a service-machine program to unlock a console keyboard (allowing character input) is by a command to the 1800 with initial contents of a text line that is to be displayed. The 1800 updates the given text line as modified by individual key strokes at the keyboard. When a terminal key is stuck, the keyboard is physically locked and the 1800 sends the resultant text line back to the service machine. Thus, the service machine sees only one interrupt per line rather than one per key stroke. The 1800 software system design is optimized for this type of job, so it can handle many such tasks with very quick response, allowing a multitude of simultaneous interactive displays.

1800 SOFTWARE STATUS

The 1800 software has been running since 1968, supporting all the devices described, as well as regular maintenance functions. Revisions have recently been made to support a connection to the ARPA Network and some experimental terminal devices.

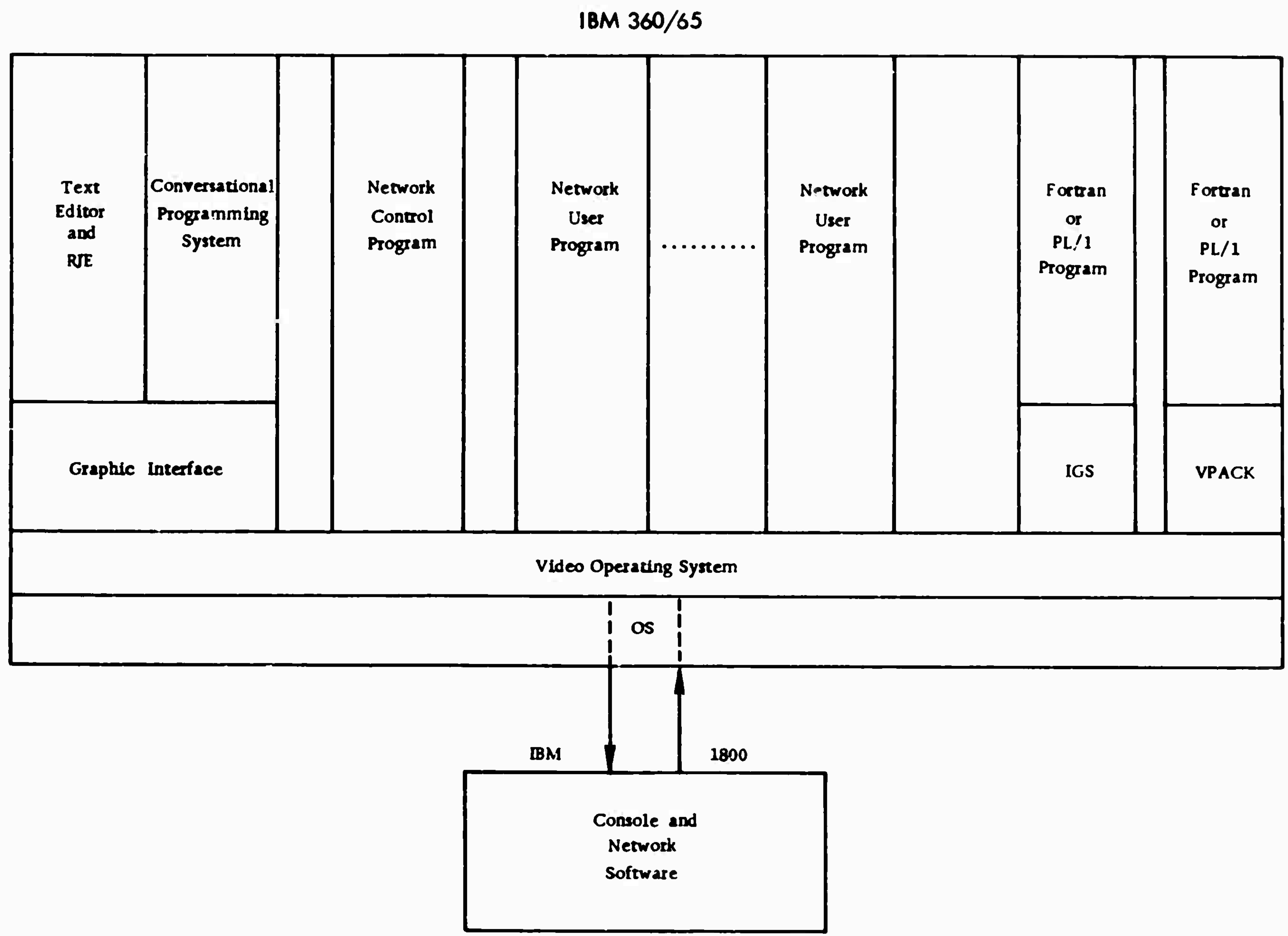

Fig. 12--The Rand System Software

THE VOS

This software resides in a general OS multiprogramming environment operating under IBM OS MVT as a high-priority task.

Its major functions are

- o To act as the general communication agent for each program, in a multiprogramming environment;
- o To create, manage, and polic· ›nsole-to-program and program-to-console connections according to requests on the part of consoles and programs (the IMP is treated as a console);
- o To receive, queue, route, and concentrate messages to and from the 1800;
- o To manage and distribute control messages.

VOS acts as a communications controller within a service machine to pass messages between programs in the service machine, VGS consoles, and the Network. That is, program-to-program, program-to-console, console-to-program, console-to-console, Network-to-program (NCP), and program-to-Network (NCP) communications are possible. The programs may be multi-access subsystems, which communicate concurrently with multiple consoles or with the Network.

Figure 13 shows the component routines of VOS. Their functions are to receive, route, enqueue, and deliver messages. The routines operate independently; some are executed as subtasks whereas others exist as shared subroutines callable by either system or user programs. Each subtask is enabled by and responds to one or more of the following event types:

- o Acceptance of an 1800 attention.
- o Completion of a read operation by OS.
- o Successful completion of the input of a message from the 1800.
- o Acceptance of a message from a program.
- o Appearance of a message on the queue of messages to be transmitted to the 1800.
- o Completion of a write operation by OS.

In Fig. 13, the receiver reads messages transmitted to the service machine from the 1800. The distributor is the central component of VOS.

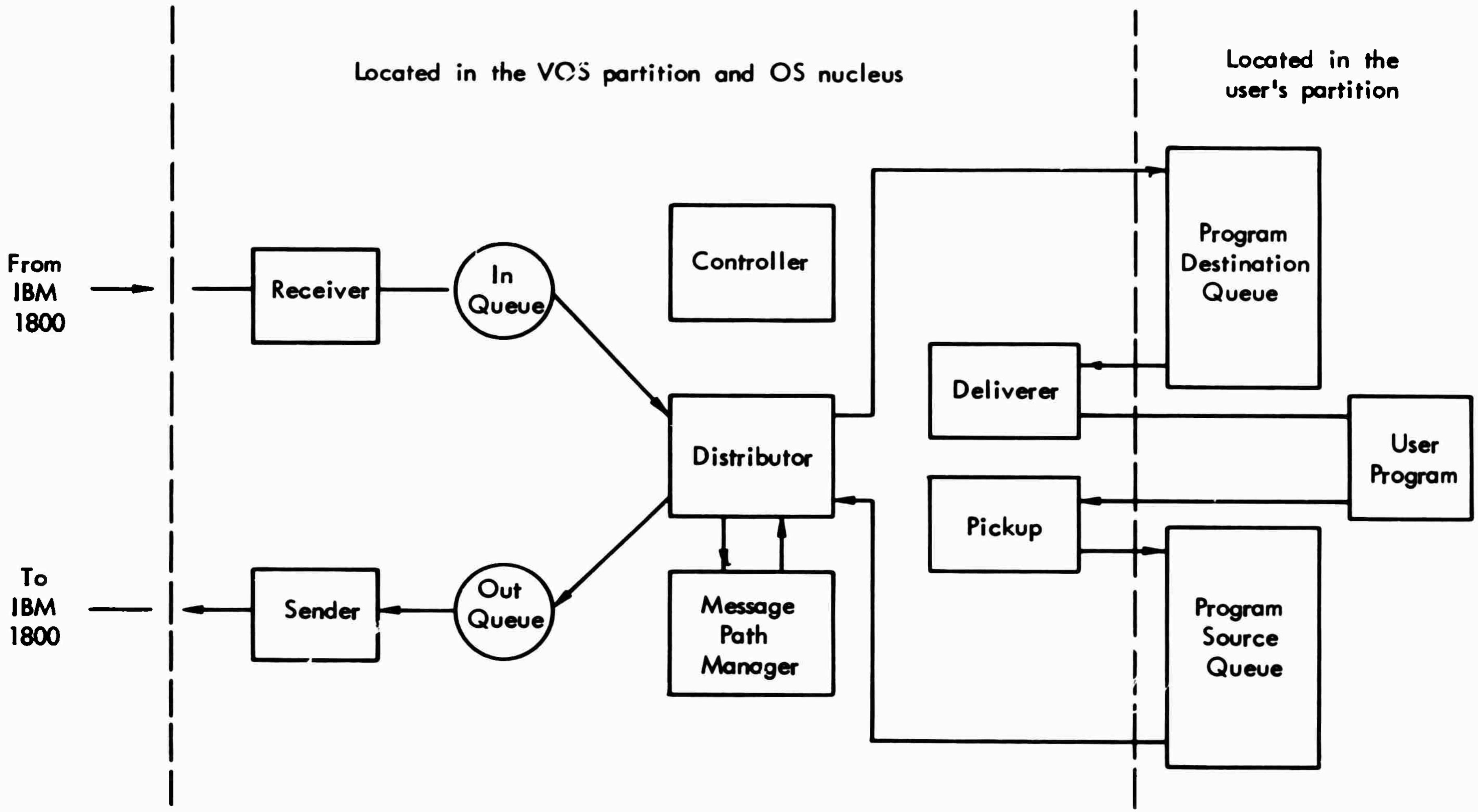

Fig. 13--Components of the Video Operating System

It responds to the occurrence of a "message-to-go" event from either the receiver or the pickup. The deliverer, shared by all destination programs, manages the access to messages, which are either waiting in its program destination queue or expected to arrive there. The pickup, shared by all programs acting as message sources, notifies the distributor of the existence of a message (originating from a proper source) to be distributed. The sender transmits messages to the 1800. The message-path manager, shared by all programs, services requests for initiation, completion, blocking, or deletion of message paths.

IGS

IGS is a graphic subroutine package callable from PL/1 or FORTRAN [26]. It is OS compatible and display-device independent (i.e., it can support an IBM 2250, VGS consoles with several input devices, and the Stromberg Carlson 4060 hardcopy recorder). IGS allows existing programs to be transferred unchanged from the IBM 2250 to the VGS. Presumably, the reverse is also true; however, this has not been tested.

The IGS package currently supports all former 2250 display calls and input-device management and messages, including the character-recognition package associated with a Tablet. A large portion of IGS is currently being revised to be re-entrant for more efficient multi-user use.

NCP

We stated that the NCP is normally a part of the basic system software. In the interim, Rand has treated the NCP as a standard OS user program rather than including it in the host IBM 1800 operating system. As a user program, the NCP may be debugged and manipulated without the inconveniences associated with verifying system code in an operational environment.†

†Treating the NCP as a user program was a good choice. Implementation time has been short, compared to other sites where the NCP was written as part of the basic system software. The short implementation time can be attributed to (1) treating the NCP as a user program, (2) code checking with the use of an on-line debugger, and (3) the existing facilities provided by VOS that allowed access to the IMP.

The purpose of the NCP is to (1) extend the VOS connection facility to initiate and switch logical connections (message paths) between local programs and remote Network programs, using the Network dialogue, and (2) switch messages between service-machine programs and remote Network programs in the Network dialogue.

The NCP consists of a task dispatcher, three asynchronous routines, tasks to interpret and generate Network protocol, and miscellaneous subroutines and data tables (Fig. 14).

The NCP's interfaces are the asynchronous routines, namely, the VOS message-path transition routine, the VOS message-arrival routine, and the interval-timer routine. By placing messages in the circular queues shown in Fig. 14 and posting an asynchronous event, these routines affect the scheduling of instances of the Network tasks, via the task dispatcher.

The task dispatcher is the "control program" of the NCP. It is initiated when an event is posted by an asynchronous routine and, in turn, initiates a task based upon messages placed in the circular queues.

A description of the NCP and its use will be contained in forthcoming reports.

SERVICE-MACHINE PROGRAMS

Service-machine programs may communicate with local consoles, other service machine programs, and remote Network programs by establishing the appropriate VOS and NCP message paths. Figure 15 shows message paths for typical programs. Many paths between programs and consoles may be maintained concurrently.

Section VI describes our current work in the area of program and console communications.

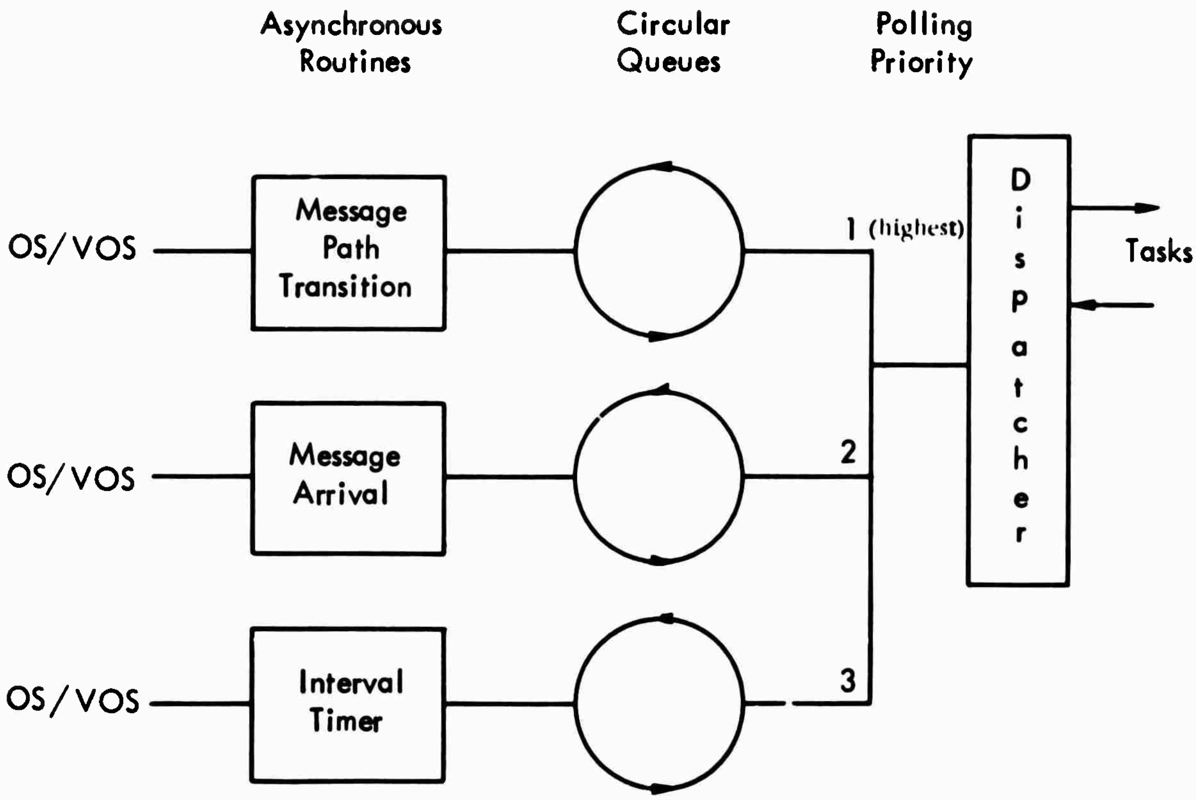

Fig. 14--NCP: Interface between the Dispatcher and the Asynchronous Routines

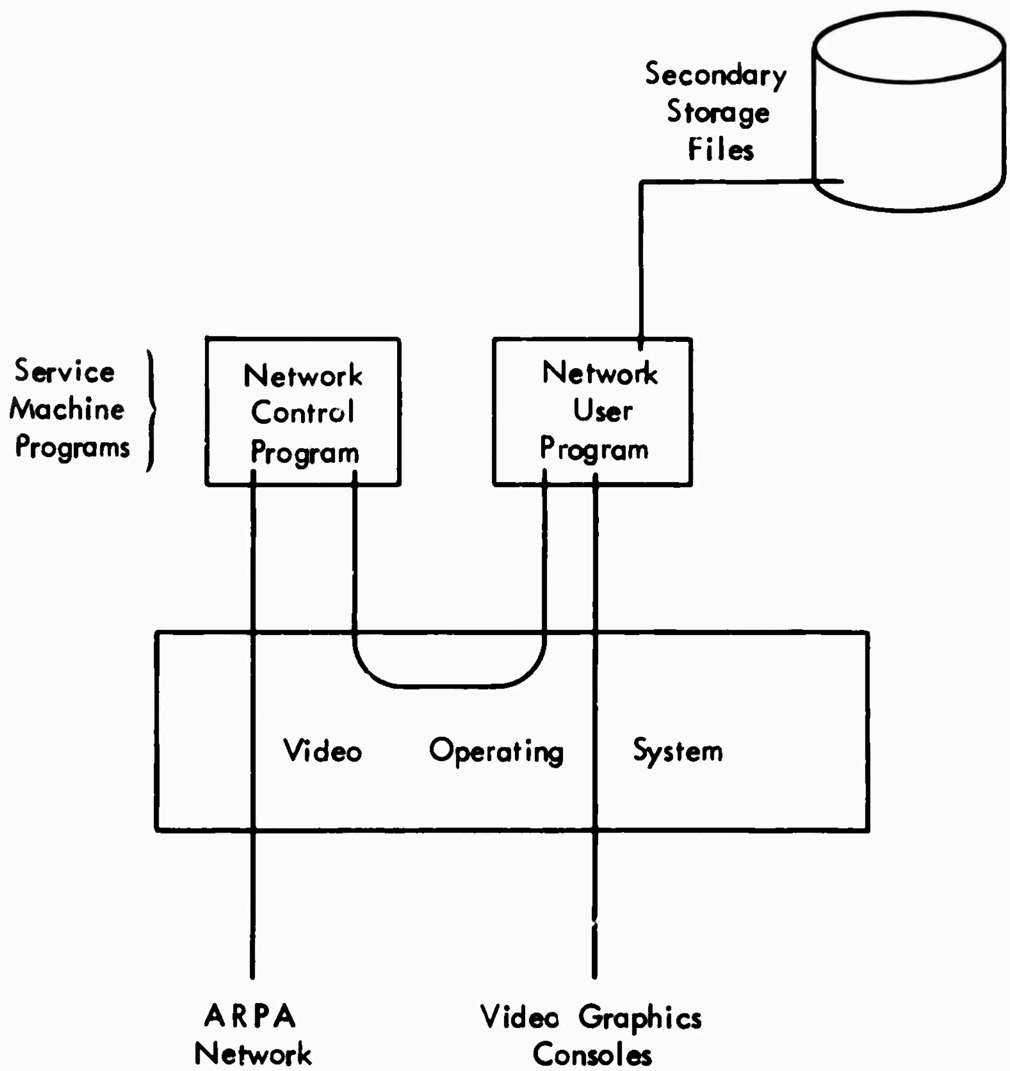

Fig. 15--Typical Message Paths Through VOS

VI. RAND PROGRAM AND TERMINAL COMMUNICATION

GOALS OF USER PROGRAMS

Many Network sites will soon undertake meaningful experiments and provide useful services. Rand is attempting to combine much of these two objectives into a common user program.

The typical Rand researcher is unfamiliar with the details of the VGS or the Network and usually has little time to learn them. Thus, there is an increasing tendency to consult a programmer, who serves as a buffer between the VGS/Network and the researcher. Our objective is to eliminate the need for this buffer--to integrate the researcher smoothly into an on-line, remote environment.

A second site objective is to provide an environment that allows an investigator to address the problems of disparate software intercommunications. It is desirable to adapt to Network use problem programs that were not planned with the Network in mind and that will not readily yield to Network standards existing at the time of their inclusion. We do not wish to add an interface to each such program so that it meets Network standards. A similar problem confronts those sites with a minimal host computer that wish to offer a variety of dissimilar Network services to their terminal users.

There are two problems--that of protocols above the NCP level for parties or processes to contact one another, and that of data formats amenable to the parties at both ends of an established connection. One approach to the protocol and data-format problems is to provide an adaptable mechanism (at strategically located hosts throughout the Network) that can generate, on demand, custom-protocol sequences for message-path management and that can reconfigure data passing between connected parties.

THE NETWORK SERVICES PROGRAM (NSP)

Rand has combined remote-service support and experimentation into a common program, the NSP, which allows Rand researchers to access remote resources from VGS. In particular, provisions are included to satisfy the previously described needs of the Rand researchers. NSP also includes test modules by which we will investigate general protocol and data-format handling.

The NSP is a multi-access program, driven either from console or programs, that operates as an OS user program (Fig. 15). The NSP nucleus is identical to the one described for NCP (Fig. 14), with two additional asynchronous routines and queues for internally generated messages among NSP modules and for interrupts resulting from secondary storage transmissions.

Figure 16 shows the tasks (or modules) currently being implemented for NSP. Their functions are described below:

- The Console/Program Manager interfaces NSP to the VGS consoles and other service-machine user programs, via VOS; it handles the requisite console dialogue.
- The NCP Manager handles Network communication protocol through the NCP to establish and maintain connections to remote programs.
- The File Manager provides the NSP access to disk data sets. It can dynamically open, close, read, write, and create data sets as directed by NSP console users or Network programs.
- The UCSB Manager permits on-line use of RJE/RJS at UCSB.
- The RJS Manager will satisfy a particular research group's need by conducting a specially needed protocol dialogue.

A comprehensive description of NSP and its use will be contained in forthcoming reports.

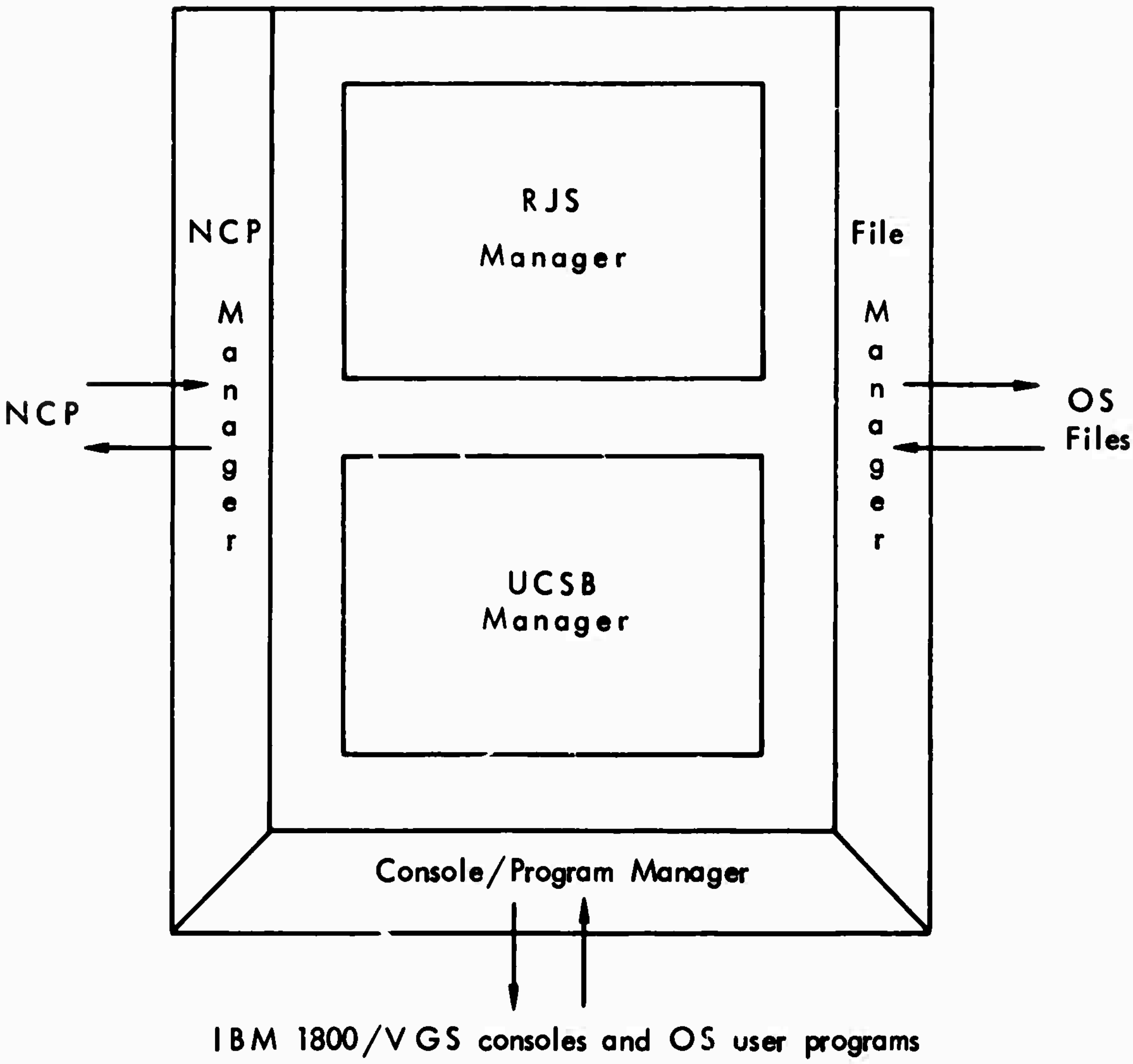

Fig. 16--Network Services Program

REFERENCES

1. Roberts, L. G., and B. D. Wessler, "Computer Network Development to Achieve Resource Sharing," *AFIPS Conference Proceedings*, Vol. 36, 1970, pp. 543-549.
2. Heart, F. E., R. E. Kahn, S. M. Ornstein, W. R. Crowther, and D. C. Walden, "The Interface Message Processor for the ARPA Computer Network," *AFIPS Conference Proceedings*, Vol. 36, 1970, pp. 551-567.
3. Kleinrock, Leonard, "Analysis and Simulation Methods in Computer Network Design," *AFIPS Conference Proceedings*, Vol. 36, 1970, pp. 569-579.
4. Frank, H., I. T. Frisch, and W. Chou, "Topological Considerations in the Design of the ARPA Computer Network," *AFIPS Conference Proceedings*, Vol. 36, 1970, pp. 581-587.
5. Carr, C. S., S. D. Crocker, and V. G. Cerf, "HOST-HOST Communication Protocol in the ARPA Network," *AFIPS Conference Proceedings*, Vol. 36, 1970, pp. 589-597.
6. Baran, Paul, "On Distributed Communication Networks," *IEEE Transactions on Communication Systems*, Vol. CS-12, March 1964.
7. -----, *On Distributed Communications: I. Introduction to Distributed Communications Networks*, The Rand Corporation, RM-3420-PR, August 1964.
8. Boehm, S. P., and Paul Baran, *On Distributed Communications: II. Digital Simulation of Hot-Potato Routing in a Broadband Distributed Communications Network*, The Rand Corporation, RM-3103-PR, August 1964.
9. Smith, J. W., *On Distributed Communications: III. Determination of Path-Lengths in a Distributed Network*, The Rand Corporation, RM-3578-PR, August 1964.
10. Baran, Paul, *On Distributed Communications: IV. Priority, Precedence, and Overload*, The Rand Corporation, RM-3638-PR, August 1964.
11. -----, *On Distributed Communications: V. History, Alternative Approaches, and Comparisons*, The Rand Corporation, RM-3097-PR, August 1964.
12. -----, *On Distributed Communications. VI. Mini-Cost Microwave*, The Rand Corporation, RM-3762-PR, August 1964.
13. -----, *On Distributed Communications: VII. Tentative Engineering Specifications and Preliminary Design for High-Data-Rate Distributed Network Switching Node*, The Rand Corporation, RM-3763-PR, August 1964.
14. -----, *On Distributed Communications: VIII. The Multiplexing Station*, The Rand Corporation, RM-3764-PR, August 1964.

15. Baran, Paul, *On Distributed Communications: IX. Security, Secrecy, and Tamper-Free Considerations*, The Rand Corporation, RM-3765-PR, August 1964.

16. -----, *On Distributed Communications: X. Cost Estimate*, The Rand Corporation, RM-3766-PR, August 1964.

17. -----, *On Distributed Communications: XI. Summary Overview*, The Rand Corporation, RM-3767-PR, August 1964.

18. *Interface Message Processor: Operating Manual*, Bolt, Beranek and Newman, Inc., Report No. 1877, February 1970.

19. *Interface Message Processor: Specifications for the Interconnection of a HOST and an IMP*, Bolt, Beranek and Newman, Inc., Report No. 1822, October 1970.

20. Uncapher, K. W., *The Rand Video Graphic System--An Approach to a General User--Computer Graphic Communication System*, The Rand Corporation, R-753-ARPA, April 1971.

21. *Conversational Programming System (CPS) Terminal User's Manual (360D-03.4-016)*, International Business Machines Corporation, Form No. GH200758-0, January 1970.

22. Groner, G. F., R. A. Berman, R. L. Clark, and E. C. DeLand, *BIOMOD: A User's View of an Interactive Computer System for Biological Modeling (A Preliminary Report)*, The Rand Corporation, RM-6327-NIH, August 1970.

23. Ball, N. A., H. Q. Foster, W. H. Long, I. E. Sutherland, and R. L. Wigington, "A Shared Memory Computer Display System," *IEEE Trans. Electronic Computers*, Vol. EC-15, October 1966, pp. 750-755.

24. Macaulay, M., "Low Cost Terminals Using Television Techniques," *National Radio and Electronics Engineering Convention Abstracts*, IREE (Australia), May 1967, p. 222.

25. Kennedy, J. R., "A System for Time-Sharing Graphic Consoles," *1966 Fall Joint Computer Conference, AFIPS Conference Proceedings*, Vol. 29, Washington, D.C; Spartan, 1966, pp. 211-222.

26. Brown, G. C., and C. H. Bush, *The Integrated Graphics for the IBM 2250*, The Rand Corporation, RM-5531-ARPA, October 1968.

BIBLIOGRAPHY

Engelbart, D. C., and W. K. English, "A Research Center for Augmenting Human Intellect," *1968 AFIPS FJCC Proceedings*, Vol. 33, Part 1, pp. 395-410.

Gay, T. R., Jr., "A Queueing Model for Scan Conversion," *1969 AFIPS FJCC Proceedings*, Vol. 35, pp. 553-560.

[illegible]l, Thomas, and L. G. Roberts, "Toward a Cooperative Network of [illegible]-Shared Computers," *AFIPS Conference Proceedings*, Vol. 29, 1966, pp. [illegible]25-431.

Metzger, R. A., "Computer Generated Graphic Segments in a Raster Display," *1969 AFIPS SJCC Proceedings*, Vol. 34, pp. 161-172.

Rose, G. A., "Integraphic, a Microprogrammed Graphic Interface Computer," *IEEE Transactions on Electronic Computers*, Vol. EC-16, No. 6, December 1967.

Sutherland, W. R., J. W. Forgie, M. V. Morillir,,"Graphics in a Time-Sharing; A Summary of TX-2 Experience," *1969 AFIPS SJCC Proceedings*, Vol. 34, pp. 629-636.