5-00 [illegible]

SYSTEMS COMMUNICATIONS MODULE

```
¤¤¤¤¤  ¤   ¤  ¤¤¤¤¤  ¤¤¤¤¤  ¤¤¤ ¤¤¤
¤      ¤   ¤  ¤      ¤   ¤  ¤ ¤ ¤ ¤
¤¤¤¤¤  ¤¤¤¤¤  ¤      ¤   ¤  ¤ ¤¤¤ ¤
    ¤    ¤    ¤      ¤   ¤  ¤     ¤
¤¤¤¤¤    ¤    ¤¤¤¤¤  ¤¤¤¤¤  ¤     ¤
```

Software Operational Guide

Preliminary

13 September 1978

BURROUGHS CORPORATION

COMPUTER SYSTEMS GROUP

SANTA BARBARA PLANT

B1800/B1700 Series
Systems

Company Confidential

Title: SYSTEMS COMMUNICATION MODULE
Identity: Software Operational Guide

Burroughs Corporation believes the information described in this document to be accurate and reliable, and much care has been taken in its preparation. However, the Corporation cannot accept any responsibility, financial or otherwise, for any consequences arising out of the use of this material. The information contained herein is subject to change. Revisions may be issued from time to time to advise of such changes and/or additions.

Release:	Level Number	- 7.0
	Revision Level	- Original
	Date of Release	- 13 Sep 1978
	Date of Printing	- 12 Sep 1978
	I/D Number	- Preliminary

PREFACE

SECTION I

INTRODUCTION TO SYCOM

SECTION II

CONCEPTS AND FACILITIES

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

SECTION III

COMMANDS

TABLE OF CONTENTS (continued)

System SPO/Program Commands (continued)

SECTION IV

OPERATING PROCEDURES

SECTION V

GENERATING THE NETWORK CONTROLLERS

SECTION VI

APPENDICES

SECTION VII

GLOSSARY

LIST OF ILLUSTRATIONS

SECTION II

CONCEPTS AND FACILITIES

LIST OF TABLES

SECTION III

COMMANDS

REVISION HISTORY

Original Release - September 1978

DOCUMENTATION NOTES

Do to the information available when this document was written, descriptions of functions and facilities for a remote host system assumes a B1800/B1700 series system. When information becomes available for implementations of the Systems Communications Module (SYCOM) on "other" Burroughs systems, this Software Operational Guide will be updated to reflect those capability differences that are provided by B4800/B4700 and B6800/B6700 systems.

Although this document describes the "Program-to-SYCOM" and "Program-to-Program" features, these capabilities are not yet implemented in this system release level.

PREFACE

MANUAL OBJECTIVES AND READER ASSUMPTIONS

The intent of this software operational guide is to enable its user to successfully operate the Systems Communications Module (SYCOM) under control of the B1800/B1700 Master Control Program (MCP). As a means to this objective, this document defines scope and objectives, describes inherent and command driven functions, and presents basic implementation characteristics. Additionally, provides operating procedures at both system and terminal levels. (Terminal operating procedures are limited to those capabilities within the scope of SYCOM.)

The Systems Communications Module can be used for a multitude of applications and by users with varying degrees of proficiency in computer operations. Although this guide has a tutorial slant, it does not claim to train operators. The reader is assumed to have a basic knowledge of the B1800/B1700 system and of the Master Control Program (MCP). The reader is also assumed to be familiar with B1800/B1700 operating procedures, to have had experience operating the system Supervisory Console (SPO) and peripheral devices, and to have a basic vocabulary of computer related terms. The user need not however, need any knowledge of programming or of the Network Definition Language (NDL).

This guide is designed to be self-contained. However, if the user wishes to have a deeper insight into the system's construction or precise details as to the various internal system functions, a reading of the "Network Definition Language Reference Manual" and the "System Software Operational Guide" is strongly recommended.

STRUCTURE OF THE DOCUMENT

Section I presents an overview of the Systems Communications Module along with the functions it performs. This section also identifies any limitations which might be applicable to the usefulness of the system and lists associated documentation.

Section II describes concepts and implementation characteristics employed by the Systems Communications Module. This includes system configurations, functional specifications, relationships in inter-system communications, and intra-system relationships with other system software elements. Pictorial overviews are presented where applicable to elaborate points of interest. Presentation of the Functional Description progresses from basic introductions, to what is involved, thru detailed "how it works" explanations.

Section III describes directive commands available to the user from either a Work Station (terminal) or the User SPO. With these commands, the user directs SYCOM to perform specific functions to satisfy a user's processing requirements. Examples are offered for each command which includes both commanc input and reply messages. The commands are presented in alphabetical order for ease of reference. The command language corstruction is presented in detail; including keyword/parameter semantics.

Section IV contains operating procedures for a SYCOM system (operating independently or in conjunction with the Supervisory Message Control System (SMCS)) and for local system terminals. Under the heading "System Operating Procedures", useful SYCOM operating procedures, commands, and operational hints/pointers are presented. Under the heading "Terminal Operating Procedures", terminal operational commands and functions are covered which assume either a TD820 or TD830 Video Display Terminal (VDT). However, the system utilizes any Burroughs supported VDT, (see "Limitations").

Section V presents instructions for the preparation of a Network Controller program required by the SYCOM system. It is basically comprised of a sample NDL coded input file for the Controller's generative process. Various comments are included throughout the sample listing to explain variable parameter data which could be different for each installation. The procedure used to invoke the generative process for an operational Network Controller program tailored to the needs of individual systems is included.

Section VI contains the appendices to this cocument, each of which is presented below:

Appendix A provides lists of error messages and their meanings. These lists are broken down into two major categories; those error messages which may be displayed in response to commands and those error messages which are displayed on the User SPO which identify a processing exception condition within SYCOM.

Appendix B contains the "character set" defined for terminal input/output and data transmissions.

Appendix C lists the "reserved words" recognized by SYCOM either as a command or command option keyword.

Appendix D contains a "command summary" which lists each command and briefly describes its purpose.

Appendix E provides a list of all control messages used by the communications protocol and includes their associated format attributes.

Appendix F presents a list of restricted MCP commands which are permitted from a terminal or SPO on the User System.

Appendix G describes the PARAMETERS file attributes and details the parameters record format.

Appendix H provides a summary of system files; including names and purposes.

Appendix I contains sample application programs which show the mechanics of interfacing with SYCOM and the system's terminals for program-to-program and terminal-to-program communications. These sample programs can also be utilized in the system test procedures once the appropriate Network Controllers are created and the systems installed.

Section VII is a glossary of terms and acronyms used throughout this document.

SECTION I

INTRODUCTION TO SYCOM

SECTION I

INTRODUCTION TO SYCOM

GENERAL DESCRIPTION

The Systems Communications Module (SYCOM) program provides the B1800/B1700 user with 1) the capability to submit jobs to run on another system, 2) to transfer files between two systems, and 3) in general, to interactively obtain services and information from another system with the ability to communicate with interactive programs in a remote system. Job run output can be returned to the submitting system (upon request) or directed to a peripheral device on the processing system itself. Interactive output data is returned to the User System directly, via Network Controllers.

SYCOM along with its associated Network Controller is responsible for all data communications with other systems. This includes performing the necessary functions to establish line connections, receive/transmit data, and execute the correct line termination procedures.

The Systems Communications Module maintains a secondary level of control (MCP has primary control) over local peripheral devices such as the card reader, card punch, line printer, and magnetic tapes. It reads and transfers card, tape, and disk files, accepts input from and displays messages to the Supervisory Console (SPO) and receives data files/messages for printing, punching, display, or writing to either tape or disk. It provides the mechanism by which a terminal or application program in one system is able to communicate with specific application programs in another system. Program selection is based on the servicing system which dictates the processing requirements (programs) optioned to run in that system.

SCOPE AND OBJECTIVES

The purpose of the Systems Communications Module is to increase the use of data processing facilities (systems) either local or remote to the target system. The scope of SYCOM is to provide all of those features previously afforded by the Remote Job Entry (RJE/TERMINAL) system and additionally provide enhancements with the following expansive communications features: file transfer, terminal-to-program, program-to-program, and program-to-SYCOM. The first two functions assume inter-system communications with the program-to-SYCOM function assuming intra-system communication which is provided by the MCP. The Systems Communications Module with these encompassing capabilities, considerably expands the flexibility of a RJE Station; henceforth referred to as a "User System".

In addition to the above design objectives, several mandatory objectives have been specified. They are: 1) Retain compatibility with the standard Burroughs data communications line discipline and message format conventions and 2) maintain standard Remote Job Entry (RJE) command compatibility with existing RJE program products for Burroughs small, medium, and large systems.

With this software system, the objectives are also to provide the user with the most benefits in terms of flexibility and ease of use. Flexibility in that the user is free to design and install system configurations suited to the requirements of the company for distributed processing. For example, a user can install a B1800/B1700 in the same installation to increase/share resources with an existing system; in a remote installation to place some resources in a remote facility; or in a network of systems to provide distributed processing services. Ease of use in that all capabilities of the system can be selectively utilized by any user thru either command initiated or inherent system functions.

FUNCTIONS PERFORMED

Since SYCOM applications imply transmission of large data files between two or more systems, line throughput maximization, error detection, and error recovery are standard features. Following is a comprehensive list of the functions performed; including those that are under user control. They are presented in three groups, Basic, Line Throughput, and Error Detection and Recovery.

BASIC FEATURES

For SYCOM to accomplish the functions generally described for the system, the following basic features are provided; full details are presented in Section II, "Concepts and Facilities".

A. File control: SYCOM verifies all file transfers between two systems.

B. Input: Introduction of programs from an input device on the local system for compilation and/or execution on a remote system. Introduction of data from an input device on the local system for processing on the remote system.

C. Command Input: SYCOM commands as well as SMCS/MCP type commands are permitted from the system SPO, terminals, and specifically tailored application programs running in that system.

D. Output: Dispersion of data and/or programs produced by the remote system to the local systems peripheral output devices, SPO, terminals, and application programs.

E. Monitoring: Monitoring and controlling programs on the remote system via a local system's Remote Supervisory Console (RSC); terminal, SPO, or application program (requires SMCS on remote system).

F. File Transfers: The transfer of certain types of files between a local system and a remote system; in both directions.

G. Program-to-Program: Support of communication between a program running in a local system and a program running in a remote system. Also, support a local application program communications link to SYCOM itself.

H. Virtual Terminals: Provide a means by which a terminal, SPO, or application program in a local system can become a virtual terminal entity to a remote system. By design, virtual terminals are automatically assigned the status of a Remote SPO when connected to a SMCS in the remote system.

I. Communications: Supports a full line of communications; asynchronous, synchronous, and binary-synchronous data transmissions (including broadband up to 50,000 bps).

LINE THROUGHPUT

Maximization of line throughput is assured by a point-to-point message transfer procedure which:

A. Specifically acknowledges receipt of a message with the transmission of either another message block or a "numbered acknowledgement" (#ACK) message. This reduces both line time and turnaround time.

B. Blocks records for transmission, (i.e., sends multiple data records per message block.

C. Compress and expand blanks in data records. It is common for card input and printer output to have a large number of imbedded blanks. Eliminating transmission of these blanks significantly improves line throughput.

ERROR DETECTION AND RECOVERY

The following features are provided in error detection/recovery areas to assure reliable service to the system user:

A. A specific numbered acknowledgement (either in the next message or in a #ACK) guards against the possibility of a lost message by providing an unambiguous acknowledgement for the transmitted message block.

B. The use of modulo 2 by the transmission numbering facility for the detection of duplicate message blocks.

C. Dynamic reduction (by command) of the transmission message block size provides the potential for reducing error rates and retransmissions.

D. The convention of different timeout values for local and remote systems alleviate line collision problems.

E. The message block transfer procedure eliminates the ambiguous use of certain ASCII control characters (for example, "ENQ" is used only during "establishment"; and no where else).

F. Data transmission error detection is provided thru the use of vertical and horizontal parity in the systems hardware and software.

LIMITATIONS

The Systems Communications Module (SYCOM) is intended to provide a combined system of RJE, file transfer, program-to-SYCOM message transfer and terminal-to-program/program-to-program communication capabilities. To provide these services anc still maintain a certain degree of compatibility with existing Remote Job Entry (RJE) systems, certain limitations become inherent to the system. Therefore, applicable limitations are defined as follows:

A. SYCOM does not, and will not, support Cassette tapes as an optional peripheral device.

B. In reference to "file transfers", the implementation currently in effect does not save, or forward, symbolic file headers. Therefore, certain file characteristics would be lost on file transfers between unlike systems.

C. Program-to-Program limitations involve an application program being assigned a "virtual terminal" position which reduces the number of positions available for terminal attachments.

D. Interactive application programs in a remote system serving virtual terminals on a local system cannot make use of "tallies" or "toggles". This prevents CANDE from from being a viable application program in this system.

E. The implemented version of SYCOM which this document descrites, does not have the "file transfer recovery" capability. Therefore, when the cccasion arises when an inter-system connection is lost, any in-process file transfer operations are aborted anc must be restarted from the beginning.

F. The most noticeable limitation is the fact that SYCOM only supports a single line connection for RJE and File Transfer data communications; at any one time.

NOTICE

The specifications of the software system described in this document are subject to change without notice. The availability or performance of some features may depend on a specific configuration of equipment such as larger memory or communication line speed. Customers should consult their Burroughs Sales Representative for details.

ASSOCIATED DOCUMENTATION

Some of the details of the Systems Communications Module system and its composite software elements are contained in other system manuals. Following is a list of associated documents referenced in this guide. Readers and users of this document may refer to any or all of these listed documents for additional information.

Company Confidential

Document Title	Identification Number
B1800/B1700 SUPERVISORY MESSAGE CONTROL SYSTEM	P.S. 2219 0482
MESSAGE CONTROL SYSTEM INTERFACE	P.S. 2212 5447
B1700 NETWORK DEFINITION LANGUAGE	P.S. 2212 5223
B1800/B1700 NDL LIBRARY	P.S. 2212 5215
B1700 DATA COMM AUDIT	P.S. 2212 5421

Corporate Released

Document Title	Form Number
B1800/B1700 SYSTEMS SYSTEM SOFTWARE OPERATIONAL GUIDE	1068731
B1700 SYSTEM NETWORK DEFINITION LANGUAGE (NDL) REFERENCE MANUAL	1073715
B1700 SYSTEMS DATA COMMUNICATIONS INFORMATION MANUAL	1089992
BURROUGHS CORPORATE DATA COMMUNICATIONS STANDARDS	1284 9006
B1800/B1700 SYSTEMS REMOTE JOB ENTRY (RJE) REFERENCE MANUAL	1090602
B1800/B1700 SUPERVISORY MESSAGE CONTROL SYSTEM SOFTWARE OPERATIONAL GUIDE	Preliminary

SECTION II

==

CCNCEPTS AND FACILITIES

SECTION II

CONCEPTS AND FACILITIES

GENERAL DESCRIPTION

To elaborate upon the design concepts applied to SYCOM, the full range of system facilities are presented in logical progression. First to be covered are actual system hardware configurations. This provides the base for further concept elaborations. Next, descriptions of the data communication networks involved and the relationship of a User System to a Server System are identified. Following this, implementation concepts and functional aspects of the system are described.

THE SYSTEM CONFIGURATION

A User System oriented to SYCOM utilizes a hardware configuration similar to the one shown in Figure II-1 (next page). A system is usually comprised of the following hardware elements: processor, memory, SPO, disk, plus standard communication controls/adapters. Optionally it can be equipped with an assortment of peripherals as per a user's needs (e.g., card reader, card punch, disk packs, line printer, magnetic tapes, etc. and terminals).

The SYCOM program is responsible for managing availability and use of peripheral devices which are used for SYCOM functions. It also controls all input/output operations for those peripheral devices as well as transmitting and receiving data files. (Data file is a general term used here to encompass all file types acceptable to SYCOM such as data, program source/object, printer backup, etc..)

With this hardware configuration, "all" SYCOM capabilities for a User System are initiated thru the system SPO. An "extended" User System (configuration with local terminals) adds a new dimension to the SYCOM system in that it's capabilities are no longer confined to initiation from the SPO but are now also available to local terminals as well. Commands entered from a system SPO can initiate any SYCOM function; including establish the connection for virtual terminal communications. Terminals are, however, restricted to an explicit command repertoire. (Details of this concept are presented later in this section.)

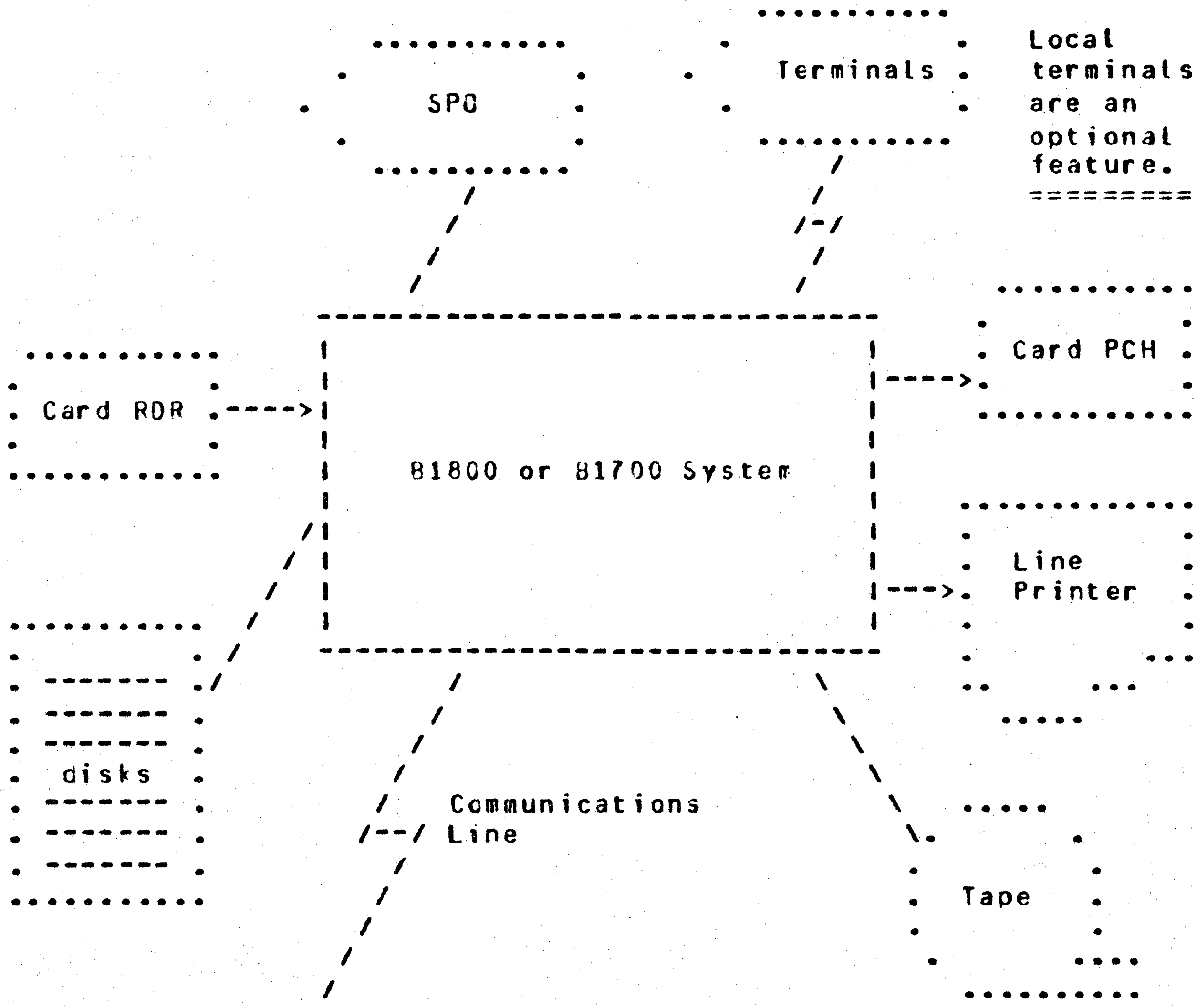


Figure II-1: User System Configuration

SYSTEM COMMUNICATIONS

Described here are the communications network aspects of a User System and its relationship to a Server System. SYCOM and its associated Network Controller supports User Systems using either a point-to-point non-switched or point-to-point switched network to a Server System.

The data communications procedure utilized is a subset of the Burroughs conversational point-to-point line procedure. It is defined in a document entitled "Burroughs Corporate Data Communications Standards"; refer to "associated documentation". That document fully describes the characteristics and functional attributes for a non-transparent half-duplex mode of data communication to which this system subscribes.

Point-to-Point Switched Network

In a switched network environment, a User System communicates with a Server System without requiring dedicated communication lines; see Figure II-2 (on the following page). The computers are connected by telephone lines to the common carrier exchange serving their respective locations. A temporary connection (data path) is established between the computers only for the period of time in which data transmission is actually required. The connection is established by dialing the telephone number of the Server System. Usually the dialing procedure is manual, however, SYCOM can cause this function to be done automatically upon command directive.

In switched line networks, a line refers to a discrete data path between a Server System and the User System (including terminals) thru the common carrier exchange. The services provided by this common carrier are usually on a time-used basis.

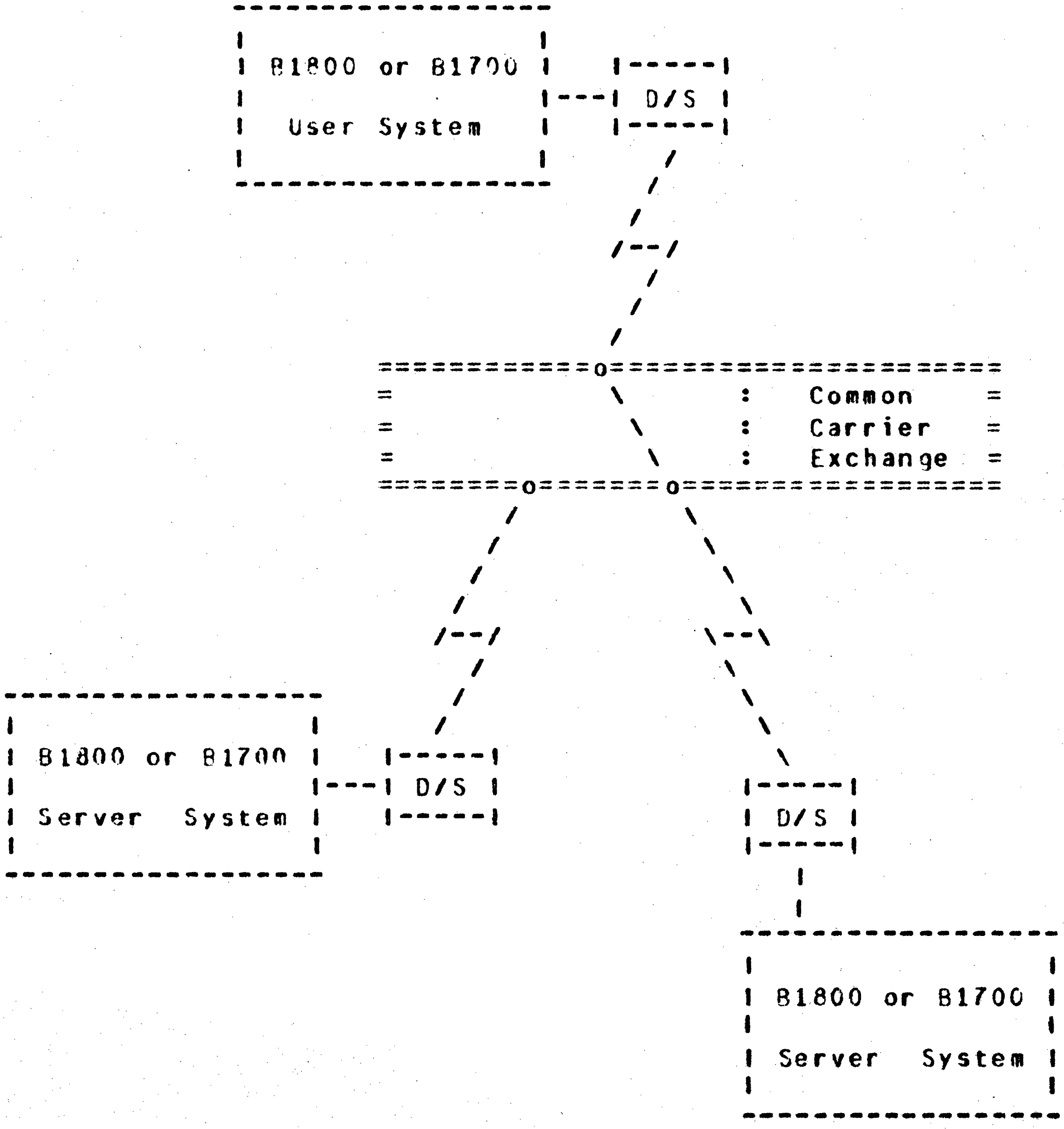


Figure II-2: Point-to-Point Switched Network

Point-to-Point Non-switched Network

A point-to-point non-switched network, as shown in Figure II-3 (below), consists of a private or leased communications line that connects the User System with a Server System. In this network, the computers are physically connected, that is, the circuits making up the communications line are permanently established, and data may be transmitted over the line at any time. No dialing is required for this type of network.

Communication lines in a non-switched network are alternately known as private, leased, or dedicated lines. In this document they are referred to as "dedicated lines". These lines may be privately owned (e.g., internal direct connection in the same physical location), or provided by a common carrier on a contract basis. This dedicated connection between two specific locations would provide a line for exclusive system use.

```
------------------   |-----|                   ------------------
|                |---| D/S |------             |                |
| B1800 or B1700 |   |-----|    /    |-----|   | B1800 or B1700 |
|                |             ------| D S |---|                |
|  User  System  |                   |-----|   | Server  System |
|                |   |-----|                   |                |
|                |---| D/S |                   ------------------
------------------   |-----|
                         \
                          \
                           \
                            \
                          \--\
Note: Direct               \
      connections           \
      do not                 \        ------------------
      require                 \       |                |
      Data Sets.            |-----|   | B1800 or B1700 |
                            | D/S |---|                |
                            |-----|   | Server  System |
                                      |                |
                                      ------------------
```

Figure II-3: Point-to-Point Non-switched Network

Communications Procedure

With the conversational point-to-point communications procedure assumed by SYCOM, inherent provisions incluoe the capability to "reverse" message block transmissions without having to invoke separate establishment and termination procedures. The essential characteristic of this capability is its ability to specifically "acknowledge" receipt of the previous message block. This is accomplished by including a special acknowledgement character in the header of the "next" message block transmitted. This special character is the Acknowledge Received Message (ARM) character.

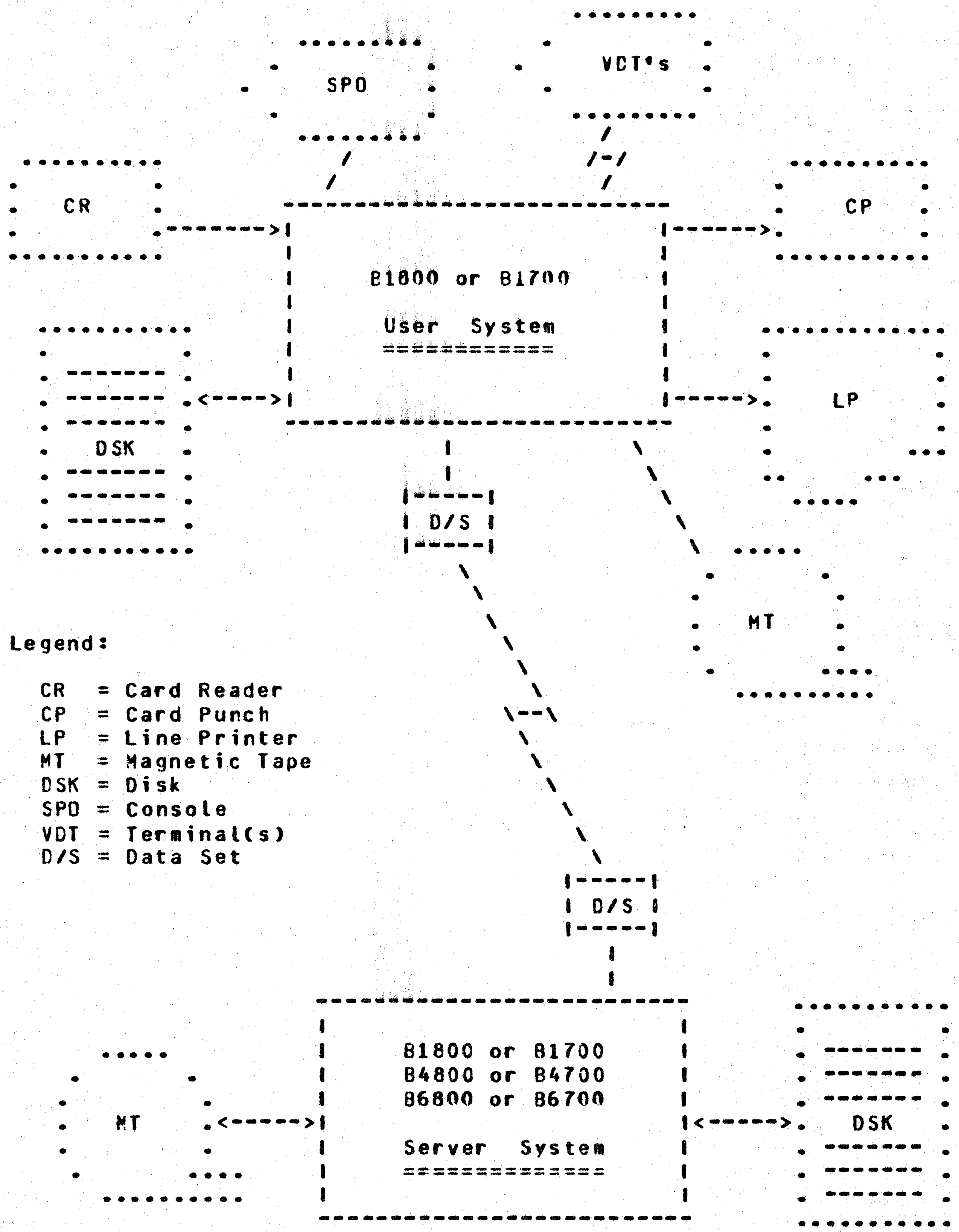
The usefulness of the ARM character is twofold: First, it provides a specific acknowledgement to the receipt of a previous message block by utilizing the "numbered ACK" facility. Secondly, when the receiving system receives a message block that it cannot accept (no buffer space or a specific peripheral is not-ready), the ARM character permits an implicit NAK of that message block. This is achieved by transmitting the NAK in the next outgoing message block.

User System - Server System Relationship

The concept of a User System - Server System relationship is quite fleeting, especially when both systems are in the same physical location. The important point to remember, however, is the basic principles of this relationship (described below) and how it applies to this system.

Referring to Figure II-4 (following page), the User System is viewed as the system in the network which serves as a point of data collection and data distribution. For example, RJE type card input, card output, line printer output, and/or terminal input/output messages.

A Server System on the other hand is generally considered to be the data processing system; often referred to as a Host System or Central System. It normally receives large volumes of data from a User System to be processed or program source files to be compiled. In the case of program compilations, the printout and/or program object code files can be transmitted back to the User System (upon request) for printing or possible program execution. Interactive output is returned directly to the initiating program or terminal via the Network Controllers.



Legend:

CR = Card Reader
CP = Card Punch
LP = Line Printer
MT = Magnetic Tape
DSK = Disk
SPO = Console
VDT = Terminal(s)
D/S = Data Set

Figure II-4: User System - Server System Schematic

The actual physical size of a computer however, has relatively little importance when considering a system as a Server System. The B1800/B1700 system can be either a User or Server System; depending on the angle of view and the services required at the time. It can also be, under certain circumstances, both a User System and a Server System.

SYCOM AND ITS ENVIRONMENT

HARDWARE REQUIREMENTS

The following list of equipment must be present for SYCOM system operations. However, the listed equipment is not dedicated to SYCOM (or supporting system software) and is available for use by other programs.

Hardware Type	Usage
B1800/B1700 Central Processor	System Control
98 KB Main Memory (minimum)	System Software
Secondary Memory (disk storage) (head-per-track, pack)	Program/Data Files
120 or 132 Position Printer	Line printer output
80 or 96 Column Card Reader	Control/RJE Type Input
80 or 96 Column Card Punch	Control/RJE Type Output
System SPO	MCP/SYCOM/SMCS Operations
Single or Multi Line Control	System/Terminal Communications
Line Adapters as Required (leased, switched, direct)	System/Terminal Communications
Data Sets and Cables (as required)	System/Terminal Communications
TD820/TD830 Terminals	As Required

MEMORY REQUIREMENTS

Provided here is a list of System Software elements and their associated memory requirements. They can be used to approximate memory requirements for an interactive B1800/B1700 SYCOM system; operating with or without a Supervisory Message Control System (SMCS):

- 64.0 KB Master Control Program (MCP).
- 15.0 KB Network Controller (basic).
- 3.9 KB Each additional line in the Controller.
- 16.0 KB Supervisory Message Control System (SMCS).
- 18.0 KB Systems Communications Module (SYCOM).
- 22.0 KB Command AND Edit system (CANDE).
- 2.0 KB Each additional terminal logged-on CANDE.

After determining the basic memory requirements for the system, additional memory must be included for any other programs that are to be executed in conjunction with SYCOM. (See appropriate reference manuals for those memory requirement estimates.)

SOFTWARE REQUIREMENTS

In order for SYCOM to become operational, certain system software elements are required. Although these elements support the operation of SYCOM, they are general purpose and in fact support the entire system. Each software element identified here is individually documented for its capabilities and requirements. Following is the list of those pertinent system software elements required to support SYCOM operations.

- Master Control Program (MCP)
- SYSTEM/MAKEUSER program
- DMPALL program
- Network Controller program

Master Control Program (MCP)

The MCP is an operating system which manages the demands and resources of B1800/B1700 systems. It reduces programming efforts by providing a family of commonly needed functions and services. The MCP requires strict user adherence to program structures and procedures in order that there may be communication between the different programs, interpreters, and the MCP itself.

Its primary function is to optimize the productivity of a system. External intervention is held to a minimum, and maximum throughput is achieved by incorporating into the MCP the primary tasks of Input/Output (I/O) control, file handling, multiprocessing, interrupts, memory allocation, application programs, and operator interfaces.

SYSTEM/MAKEUSER Program

A utility program that is used to create, modify, and access the system's "usercode" file. This file contains all valid system usercode/password combinations. These usercodes-passwords are maintained in a file named "(SYSTEM)/USERCODE". With this file, the MCP can support, and enforce, a file security mechanism for disk files and programs/users that access them.

DMPALL Program

The DMPALL program is a general media conversion facility that allows reproduction of files from one medium to another. It also can list files from any medium with byte, digit, or combined format. Along with these abilities it can start or stop its reading of these files at a specified record. Specifications for these functions may be passed to DMPALL from a SPO, Card Reader, or local system terminal.

Network Controller

The Network Controller is a Network Definition Language (NDL) type program whcse prime function is to process and supervise the flow of messages between systems, terminals, and programs. It performs all data communication line discipline functions which are necessary for proper line/terminal control. The Network Controller handles the initiation and completion of communication I/O's and services exception conditions. In order to optimally provide these functions, the Network Controller is configured, by NDL generation, to the specific needs of the system on which it is to be run, (i.e., the number and type lines, number and type terminals, etc.). A product of this generative process is the Network Information File (NIF). This file contains all terminal and communication line attributes as well as logical aspects of the communicaticns network.

Basically, this Controller enables messages to pass between it and SYCOM as well as other terminal oriented programs executing in the system. This is accomplished via MCP supported Remote Files and Queue's along with the Controller's message switching capability. Each program that is to be part of the message flow must contain a "remote file" for input/output message traffic and it must be known to the Network Controller; this includes SYCOM.

SOFTWARE OPTIONS

The following software elements are not required but may be used in conjunction with SYCOM to service a particular user's needs.

- Supervisory Message Control System (SMCS)
- Application Programs
- CANDE Program (multiple-user text editing)

Although the Supervisory Message Control System is not a required system software element for the operation of a SYCOM system, it is definitely suggested for those systems which have terminals. A Terminal in a SYCOM system environment is permitted to initiate "file transfers" or become a "virtual terminal" entity on a Server System. Additionally, these terminals may be utilized for other applications such as data entry and program development. To maintain overall control of the terminals in these varied system environments, the SMCS becomes the controlling element.

Supervisory Message Control System (SMCS)

The Supervisory Message Control System (SMCS) is intended to be the "supervisor" for a data communications software system which includes such Burroughs software elements as "SYCOM", CANDE, and other online programs of either Burroughs or customer origin.

The SMCS provides to an interactive system, many functions and special features. To fully explore its capabilities, please consult the "SMCS Software Operational Guide". However, the highlights of some of the more pertinent functions are presented here. Relevant to the SYCOM system, SMCS provides some basic message switching (terminal to terminal, terminal to SPO, and SPO to terminal), attachment of terminals to application programs via a remote file, and information about the system's network.

Additionally, it controls terminal log-on security (checking of user security codes and granting access to user files) and approves (or denies) all remote file opens to terminals under its control. Moreover, SMCS provides a certain degree of resource allocation. By this is meant, thru security, job control, remote file control, and error handling, the SMCS can allocate some of the systems resources. For example, for every terminal controlled by the SMCS, it has final control over that device as a resource. It can therefore, temporarily "attach" terminals to SYCOM by means of the SMCS "SIGN ON" facility.

Application Programs

Effectively there are three types of application programs useful to the SYCOM system user, they are: 1) Interactive data entry, 2) system-to-system, and 3) command input to SYCOM. Each program type is briefly described here with details of how they actually work presented in the following "Functional Description".

A user can elect to develop an "interactive" application program tailored to specific needs of an installaticn. These specific needs could include such applicational functions as data entry, information retrieval, or any application desired.

If there is a requirement for an application program in a User System to communicate directly with another application program in a Server System, the capability is there. This communication link is directly thru the Network Controllers and SYCOM has no role in this message traffic. SYCOM does however, initially approve a "virtual terminal" lsn for identification and control purposes. This is a residual capability based on requirements for Network Controllers by SYCOM.

The third type of application program provides the source for a string of SYCOM commands to be performed on an automatic basis. Normally this would be for off-hours operation but could also be utilized for common everyday occurrences of a work flow pattern.

CANDE Program

The B1800/B1700 CANDE program provides generalized file updating and preparation capabilities in an interactive, terminal oriented environment. CANDE performs these file updating and text editing functions in program development and other such environments. For this system however, CANDE can only be utilized when SYCOM is not involved (i.e., for the creation and maintenance of "NDL source" files).

The CANDE system has been designed to run in a self-sustaining operation or in conjunction with the SMCS. (In either case, it requires a Network Controller for data communication functions.) It operates with a basic usercode-password type of security that is available to CANDE, and SMCS, from the MCP.

FUNCTIONAL DESCRIPTION

The operation of SYCOM is dependent upon support from a Network Controller program. This Network Controller provides interfacing links between SYCOM and the communications hardware (terminals and other systems). The controller furnishes those services and capabilities which are outlined in general with the preceding subject, "Software Requirements". Throughout this description, Network Controller functions are elaborated upon where it is necessary to maintain continuity of subject matter. Concluding this section, the functions of a Network Controller is outlined under the heading "Network Controller".

The following features of the system are functionally described in the order in which they are identified below: Standard RJE, File transfers, Virtual terminals, Program-to-program, and Network control. The virtual terminal capability contributes terminal-to-program, program-to-SYCOM, and Remote Supervisory Console (RSC) communications.

STANDARD RJE FUNCTIONS

The SYCOM program controls, at a secondary level, all typical RJE peripheral devices of a User System. It reads and punches cards, prints output, accepts SPO input, displays SPO output messages, and transfers file/control messages between the User System and a Server System.

Input can be directly from a card reader, from tape or disk thru file equates, or from the SPO. Card reader input is generally associated with RJF batch data transmissions; whereas magnetic tape and disk input more directly relates to file transfers. However, these operations in SYCOM are considered synonymous.

Commands can be alternately input from a specifically designed application program thru the "Program-to-SYCOM" facility. With this mechanism, the RJE input as well as associated output could be automatically controlled on a daily basis without requiring the system operator to enter a long series of commands to SYCOM.

Output can be made directly to a card punch, line printer, disk, magnetic tape, or SPO. However, in the case of a printer or card punch, the data received can be in the form of a "backup file" which is first written to disk. Subsequent MCP commands are entered to cause printing or punching as appropriate.

Furthermore, SYCOM gives certain directives to the Network Controller such as requesting it to establish or re-establish a connection to a Server System. SYCOM also causes the controller to enter an "idle" state while waiting for an incoming connection to be established and to terminate a connection when appropriate.

The transmission of RJE data files to/from a Server System occurs as follows: A user of the User System prepares a card deck for transmission to the Server System. The cards are read, compressed and blocked by SYCOM (upon command) and written into the Network Controller's remote file. (See writeup on "Queue Depth Control".) The controller begins immediately to transmit these card images (message blocks) to the Server System. This process continues until all cards have been read and transmitted. The card file, known as a "job stream", is then under control of the Server System's MCP which processes that job stream as per directives included in that file (compile and/or execute).

All SPO type messages which pertain to that job (BOJ, EOJ, and SPO messages) are transmitted back to the User System for display on the User SPO. Messages can be entered via the SPO to control jobs executing on the Server System or manipulate data files as required; providing a MCS is running in that system. SYCOM accepts these messages and writes them to the Network Controller for transmission to the Server System and subsequent processing.

When the job has finished, any associated line printer and card punch output created at the Server System is transmitted back to the User System. (This is dependent upon the software system in the Server System; HOST/RJE does this automatically where SYCOM does not.) The Network Controller receives these incoming message blocks and forwards them to SYCOM via SYCOM's remote file (input queue). SYCOM then appropriately prints and/or punches this data on a User System peripheral device.

The afore described process describes the flow of data between the User System and a Server System; often referred to as a "job stream". Any number of job streams may be initiated at the same time (multiple Server System MCP directives contained in one job file) from a User System. The number of jobs which can or will run on a Server System is limited only to its capabilities to process them.

FILE TRANSFERS

File transfers are supported for both disk and magnetic tape devices. Files may be transferred between any combination of these devices. Magnetic tape transfers are restricted however, in that only one file per tape may be transferred. Library tapes created by the "LOAD.DUMP" function are not supported by SYCOM. (These tape files are first loaded to disk and then transferred.) Between "unlike" systems, only program code files and 8-bit data files (includes text files) are permitted. On the other hand, "like" systems can transfer other files such as printer backup, punch backup, interpreter, intrinsic, and dump files. The data communications protocol allows transfer of only one file at a time; in each direction.

The user initiates a file transfer by entering a "COPY" command to SYCOM (PUT and FETCH are acceptable alternates). The command syntax allows source and cestination systems to be specified (PUT and FETCH excluded) for the transfer via attributes. These source and destination system identifiers must be the same as those specified in the "system-id" control message at connection initialization time.

If the specified source system is the User System, the transfer is a "PUT"; which is a transfer of a file "to" another system and is based on SYCOM's "SYSTEMID" parameter. When the source system is a Server System, then the transfer is a "FETCH"; which is the transfer of a file "from" another system, also based on SYCOM's "SYSTEMID" parameter.

The actual transfer of a file is initiated by exchange of control information between systems; this process is often referred to as "handshaking". All file transfers occur as "Put's". That is, if one system wishes to do a "Fetch", it sends a "Fetch message" to the other system. This causes the other system to initiate a "Put" sequence by sending a "Put Message" to the initiating system. Put and Fetch messages completely specify both source and destination file attributes. The Put message elicits a "Put Reply Message", signifying that data transfer may begin. The sender system then sends one or more data message blocks which are eventually followed by an end-of-file message.

Upon completion of the file transfer, the system that received the file informs the originating system of the completion. This notification is made with a "user-destined control message".

If either system wishes to abort an in-process file transfer (for any reason), that system sends either an "Abort Put" or "Abort Fetch" system control message to the other system. An abort is initiated by the appropriate command entered to SYCOM. Alternate "device not ready; suspend" and "device ready; resume" messages are used for flow control at any time for temporary stoppages.

A file transfer request that cannot be processed when entered, because another file transfer is in progress, is queued to disk and processed when next possible. Although there is a maximum depth to this disk queue (see "Restrictions"), it is sufficient to handle all but the most extreme cases. This feature prevents request rejections due to the fact that only one file transfer (in each direction) can be in-process at any given time. Requests are queued on a first come, first serve, basis. This mechanism assures an orderly processing sequence.

An important point to note, all file transfer requests are queued as "Puts". This is, if a user requests a data file "Fetch", that request is sent to the other system's SYCOM where it is queued as a "Put" on that system.

Maintaining Integrity of File Characteristics

A goal of the file transfer mechanism is to maintain as many of a file's characteristics as possible throughout the transfer. For example, if a COBOL source program is transferred, it should be capable of being compiled, without further modification, on the receiving system. Likewise, if a program code file is transferred to a "like system", it should be executable without modification.

Specifically, if a file is transferred between two like systems, even if via one or more unlike systems, all file characteristics must be preserved. To accomplish this, a symbolic version of the file's header (understandable to all systems) is transferred with the file. This symbolic file header is the "PUT-OR-FETCH-RECORD" as defined for SYCOM file transfers. It is prefixed in the data stream by a beginning-of-header control message.

In the receiving system, when a symbolic file header is received, as many as possible of the file characteristics are incorporated into the file being created/received. If the file is transferred to an "unlike" system, the symbolic file header and the original system type are saved with the file, to be forwarded with the file whenever it is again transferred. If the file is transferred to its original system type, the entire symbolic file header is incorporated to create a file identical in characteristics to the original file, and the symbolic header is not saved.

When a system sends one of its own files, it must include in the symbolic file header, enough information for it to recreate that file if returned. If this file is returned, it must be made to look identical to the original file, with the exception of "file name" (this is conveyed in the symbolic file header).

An unlike system created file may be changed or replaced locally, thus invalidating the saved symbolic file header. In this case, the symbolic file header describes the file according to its new creation system type.

NOTE OF IMPORTANCE: The "symbolic file header" discussed above is not implemented in this version of the system software. It will however, be available in a later release of the system.

VIRTUAL TERMINALS

To the SYCOM system, a "virtual terminal" can be either a local terminal, the system SPO, or an application program. Therefore, virtual terminals is a logical capability whereby a message path can be established between a terminal, SPO, or program on the User System and an appropriate interactive application program in a Server System. Additionally, utilize the inherent capability of controlling jobs in a Server System with a special MCP command repertoire. Basically, on the Server System end, this becomes a reality thru the capabilities of the SMCS. However, the main inter-system connections are established by SYCOM which makes it possible. Refer to Figure II-5 for a pictorial overview of a User System with virtual terminals. (Interactive application programs can be any program using a "remote file" but not using "tallies" or "toggles".)

The terminals are virtual in that all User System facilities as well as all Server System facilities are available to the system user. User System commands are local and are therefore, processed within the User System by either SYCOM, SMCS, MCP, or application program. Server System input messages are accepted by the User System and transmitted to the Server System (via the Network Controllers) for processing and subsequent reply messages.

In view of a "virtual terminal" in an interactive environment, Server System commands must be forwarded as soon as possible. To effect this requirement, SYCOM has the capability to "interleave" a "user destined control message" into the data transmission stream immediately if a file transfer is in progress. If SYCOM is not presently transmitting a file, the control message which contains the virtual terminal message is immediately transmitted on an individual basis to the Server System.

Terminal-to-Program Communications

When SYCOM is utilized in a larger system development or data entry environment (with terminals), then the User System depicted in Figure II-11 is applicable. In that hardware/software configuration, the complete spectrum of SYCOM capabilities are available to the system user. To be more specific, in addition to the capabilities previously identified, a terminal-to-program capability is available to the terminal user.

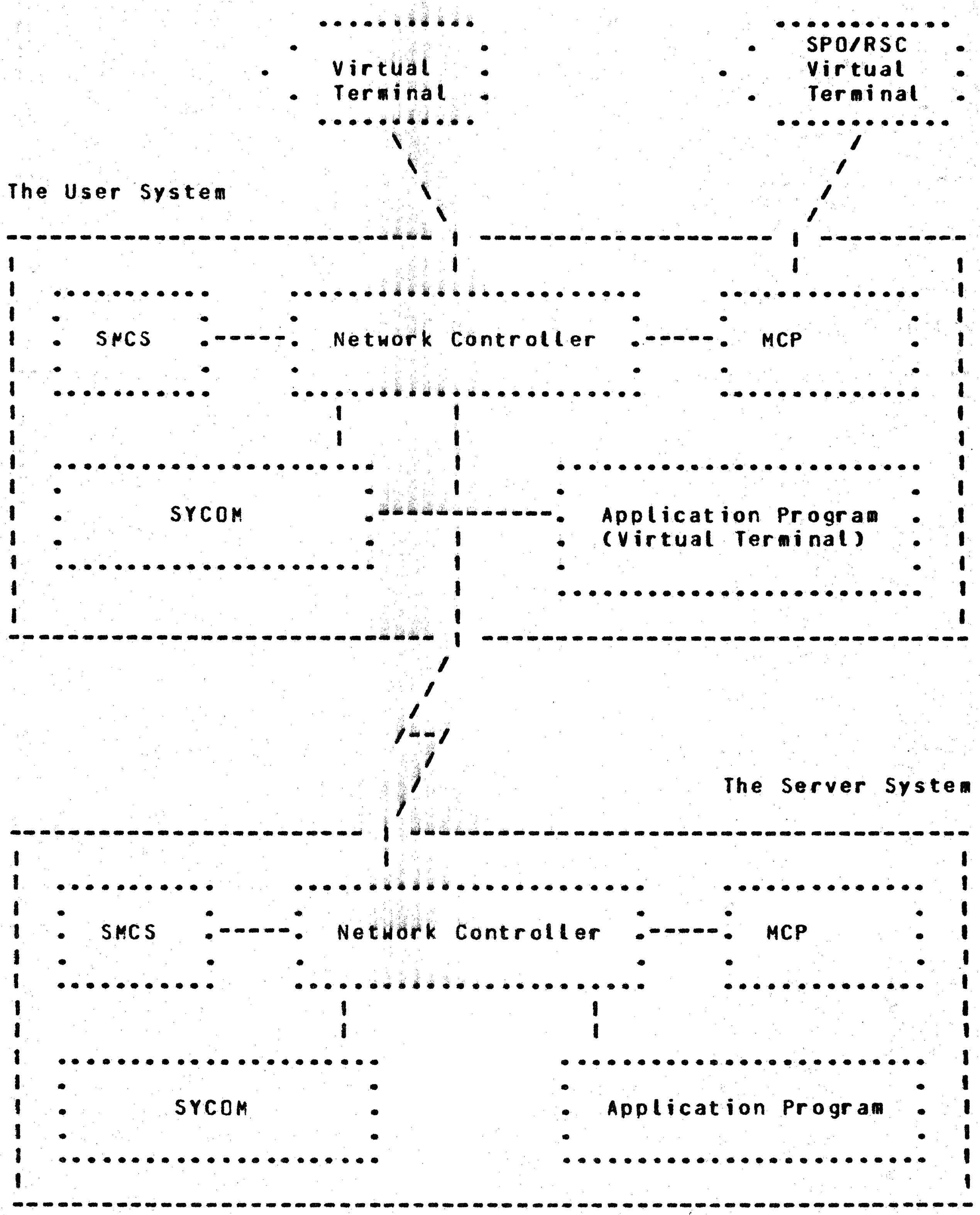


Figure II-5: Virtual Terminal Schematic

In this environment, SYCOM still communicates directly with the Network Controller for normal communications activity. However, the SMCS maintains elementary control over SYCOM and control over a system's terminals. To clarify this, the SMCS directly causes the execution of SYCOM when a terminal (or RSC) "signs-on" to SYCOM thru the SMCS. A terminal user can then direct commands to SYCOM in the User System or the SMCS in either the User or Server System. Exactly which commands and how the directives are given is covered in Section IV, "Terminal Operations". An important note; these capabilities are also available to the user who optionally elects not to utilize a SMCS in the system. (Refer to Figures II-9 and II-10 for configuration overviews.)

With the introduction of the SMCS software element into a system, all of its commands, functions, and capabilities become available to the terminal user. (Refer to the "SMCS Software Operational Guide" for details and capabilities.)

Before proceeding, it might be helpful to this discussion for the terms User System and Server System to again be clarified. A User System is the computer on which there are terminals/users wishing to engage in interactive computing sessions. The Server System is the computer that supplies those interactive services on a demand basis. (Refer to Figure II-4; back a bit.)

The communications protocol assumed by SYCOM supports User System terminals connected (logically) to an interactive application program on a Server System. Using this protocol minimizes the visibility of interfacing software elements so that to a user, it appears as though the terminal is connected directly to the application program. The terminal is then used as if it was locally attached to that program (see "Message Path Assignments" further on in this section).

Presently available interactive application programs may be used or the user can optionally elect to develop a special application program for terminal and/or other program interaction.

The method for conveying terminal data between two systems makes use of the "device address" field in the control message. This message specifies the terminal with which the message text is associated. The device address in the context is termed "virtual address", and is the name used by both User and Server Systems to reference a particular terminal. This addressing scheme follows:

A. The device address is comprised of two characters; one upper case and one lower case alphabetic letter.

B. The User System "transmits" a virtual terminal address as lower case/upper case letters. When it "receives" a response message from the Server System, the terminal address letters are reversed; upper case/lower case.

C. The Server System "receives" a virtual terminal message with a device address of lower/upper case letters (as transmitted by the User System). The virtual terminal addressing scheme then requires the Server System to "reverse" these device address characters prior to its transmission back to the User System. (These transmit/ receive device addresses are specified in the Network Controller's NDL generation process; "both" systems.

D. Assume that a User System supports a connection for five terminals to a Server System. The User System refers to these terminals by virtual terminal (transmit) address:

aA, bA, cA, dA, and eA

The Server System also refers to these same terminals by virtual terminal (transmit) address; but, as the Server Systems role is reversed, its transmit address is also reversed. For example:

Aa, Ab, Ac, Ad, and Ae

The following events represent the terminal-to-program procedure implemented by SYCOM. These events portray an interactive session between a virtual terminal on a User System and an application program on a Server System.

the user system

At initialization time, SYCOM issues a "remote file information" request to obtain, from the Network Controller, the Logical Station Numbers (lsn's) for the virtual terminals assigned to the system. These numbers are placed into a table which maps the actual "lsn" of each virtual terminal into a virtual "lsn" with which SYCOM can reference the corresponding virtual terminal.

When a user wishes a connection to a Server System, a ".CONNECT" command is entered to the User System's SYCOM. SYCOM finds the "next" available entry in its "virtual lsn table" and transmits a "connect message" to the specified Server System. This control message contains a virtual terminal address. Upon receipt of the "connect reply message", SYCOM replies with the results.

During the life of a connection, all data messages entered from a virtual terminal are sent to the Server System for processing. (Excluded are any local SMCS commands which are prefixed with the assigned signal character.) This is accomplished by doing a remote file write to the Controller using the terminal's virtual terminal "lsn". In processing this message, the Controller maps the virtual terminal lsn into the assigned virtual terminal address (two characters). The message is then transmitted to the Server System, according to SYCOM protocol, with the virtual terminal address. The terminal user can then "sign on" to any application program that is supported in the Server System; providing the capability is inherent in the form of a MCS.

Any incoming message to the Network Controller from a SYCOM line which contains a virtual terminal address (e.g., "Aa", "Ab", etc) is sent to SYCOM. These messages are written into SYCOM's remote file (LINK) by the Controller with a device address appropriate for the terminal. SYCOM upon reading the message, searches the virtual terminal address table for the actual system "lsn" for that terminal. SYCOM then puts the actual "lsn" into the reply message and writes it into the Network Controller's remote file (station queue) for distribution to the originating terminal.

In the case where the virtual terminal is actually an application program, the reply message is sent to that program via the queue mechanism for "program-to-SYCOM" communications.

When the user wishes to disconnect from the Server System, a ".DISCONNECT" command is entered to local SYCOM which transmits a "disconnect message" to the Server System, specifying the virtual terminal address to be disconnected. The terminal is notified of this action and is then free to establish another connection or work pattern as desired. (Virtual terminals are automatically disconnected when the system operator enters the ".EOJ" command.)

the server system

The Server System's Network Controller intercepts "connect" and "disconnect" messages (unlike all other control message which are routed to SYCOM). On receipt of a "connect message", the Network Controller verifies that the addressed virtual terminal exists, checks to see it it can support another connection, and that some interactive application program is ready to accept input from it. If all is okay, the Network Controller returns a "connect reply message" indicating successful completion; else the Controller sends back an error response in a "connect reply message".

If the connection is successful, the Server System makes this virtual terminal appear as a terminal local to that system (but not yet "signed-on").

During the life of the connection, any input data received by the Server System's Network Controller for an enabled virtual terminal is conveyed to the waiting application program thru its remote file (could be the SMCS). This message transfer transpires in exactly the same manner as for local terminals. Any output data generated by the Network Controller is transmitted to the User System, with the virtual terminal address specified.

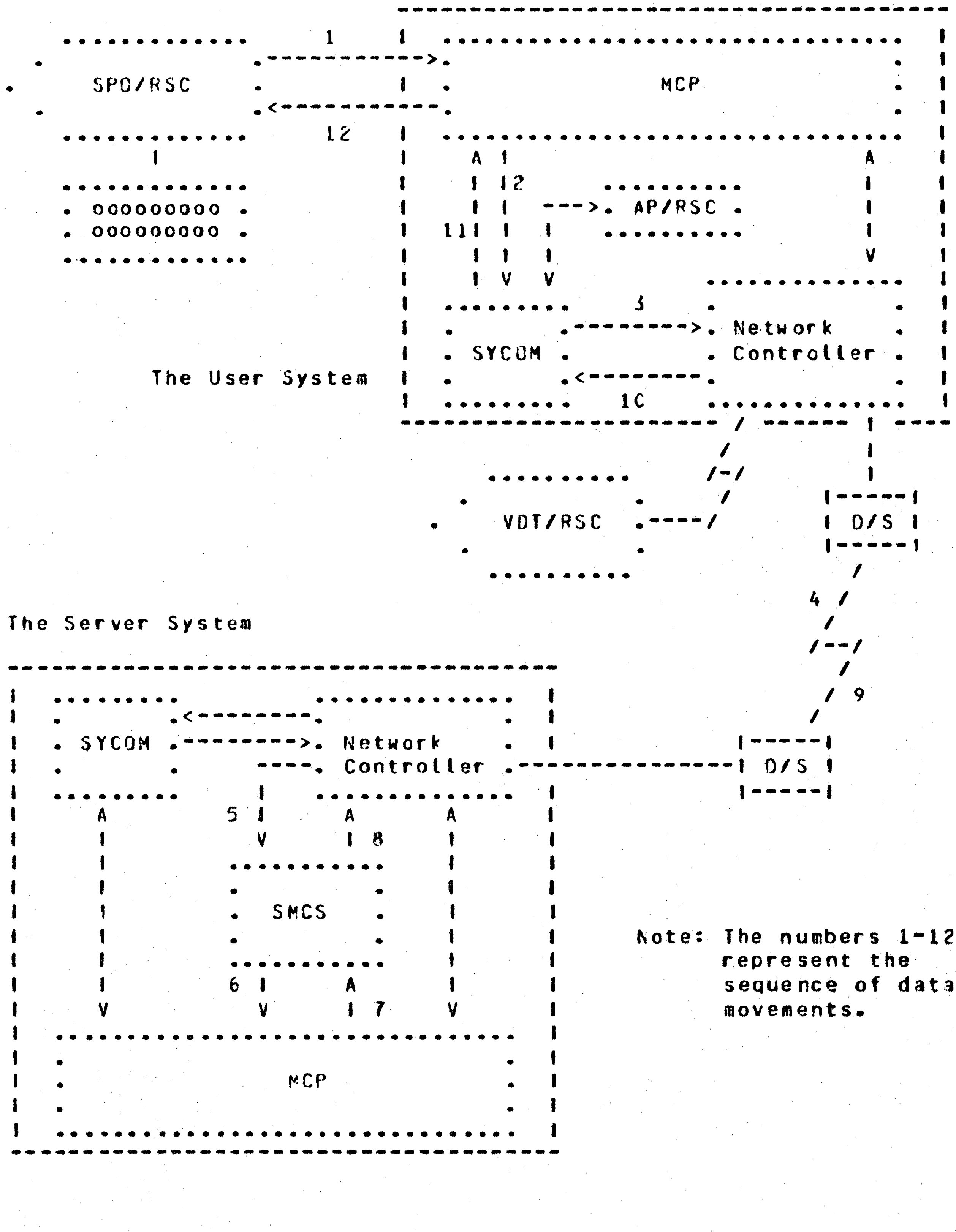
The Server System upon receipt of a "disconnect message", takes necessary action to detach that virtual terminal and respond to the User System with an appropriate "disconnect reply message".

Remote Supervisory Console (RSC)

It is an inherent capability of the system to utilize the SPO, an application program, or a terminal in a User System as a "Remote SPO" to a Server System. This only occurs when, as Figure II-6 shows, a copy of the SMCS is running in the Server System. For this system, the SPO is also referred to as a "Remote Supervisory Console", or simply the RSC. This term was selected to imply capabilities which transcend normal capabilities of a SPO. This interface infers communication (via the SMCS) with a Server System's MCP, hence the term, Remote Supervisory Console.

Messages input from the RSC to SYCOM are grouped into two basic types; Local SYCOM and Remote MCP. Local SYCOM input includes commands to initiate file transfers, request SYCOM status data, and other such functions. Additionally, respond to conditions within SYCOM's domain that cannot be preprogrammed; such as card reader not ready, clear peripherals, etc..

Remote MCP commands permitted from a RSC are basically directives used for control over the execution of jobs in the Server System; all via the Server System's SMCS. Only certain MCP commands are permitted from a RSC (reference Appendix F, "Restricted MCP Command Summary") for processing on a B1800/B1700 Server System. Should the Server System be other than a B1800/B1700 series computer, refer to the appropriate operating system reference manual for a list of MCP commands permitted from a Remote SPO.



Note: The numbers 1-12 represent the sequence of data movements.

Figure II-6: RSC Input/Output Message Flow

Output messages for RSC display can be grouped into four types; they are: Local SYCOM, Local MCP, Remote SMCS, and Remote MCP. Local SYCOM output messages are normally responses to commands but can also be error type messages reporting SYCOM problems. Local MCP output would include such messages as "NO PRINTER AVAILABLE" and other such information messages which pertain to operations of a SYCOM system. Remote SMCS messages would be replies to SMCS commands entered for processing by the Server System's SMCS. Remote MCP output messages include SPO type output resulting from jobs run on a Server System. These messages are routed back to the User System's SYCOM program for display.

In the case where the RSC is actually an application program, the response message is directed to that program via SYCOM's "program to SYCOM" queue mechanism.

Referring again to Figure II-6, assume for example, that a user wanted to execute a job on the Server System. Assume also that the subject program and all data requirements reside in the Server System. The appropriate command string is entered to the User System's SYCOM, via the SPO/RSC. SYCOM accepts this message from the MCP, formats and transmits it to the Server System; via the Network Controller. (The command string could also be entered from either a local terminal or application program. Although these message paths are slightly different, the end results are the same.)

In the Server System, the SMCS receives the MCP command message (from the Network Controller) and "zips" it along to the MCP. The Server System's MCP processes the request as asked. All output messages pertaining to that job are given to the SMCS which in turn transmits them to SYCOM in the User System. User System SYCOM then causes these messages to be displayed its SPO/RSC.

From this we can see SYCOM's involvement and the processing flow involved in this type of operation. All preceding functions described for the RSC are over and above the standard User SPO input/output messages used for normal User System operations.

Since SYCOM is symmetrical, both systems can perform an identical set of functions; providing of course, both systems are of the same series. Actual input and output messages are shown and exemplified in Section IV, "System Operating Procedures".

Program-to-SYCOM Communication

The same range of functions are possible for the program-to-SYCOM facility as for terminal-to-program communications. This feature is made possible by an adaptation of the terminal-to-program scheme. A mechanism (of implementation dependent nature) is provided which allows an application program in the User System to masquerade as a "terminal". This program formats messages to appear as if they were terminal keyboard input. These messages are then written into SYCOM's "INQ" queue for processing.

From this point on, at least until a reply message is processed, processing is the same as terminal-to-program commands. When SYCOM receives a reply message, it is written to its "OUTQ" queue where the application program in turn requests a "read" on this queue to obtain the message. These reply messages must also be formatted exactly as those created for terminal display; because they could be destined for terminal output. The application program can then either print, store, ignore, parse, or display select message text as appropriate for the program.

The schematic on the following page (Figure II-7) provides an overview of the mechanics for "Program-to-SYCOM communications": Communication between the application program and SYCOM is thru a pair of simple "queue files". The program gives a command to SYCOM by "writing" into SYCOM's "OUTQ" queue file. It obtains a response message by "reading" from SYCOM's "INQ" queue file.

To convey certain basic information with these input/output messages, a "header" is required which must contain specific information. This header is defined in Section VI; "Appendix E, Message Formats". There are no command restrictions enforced for this method of SYCOM/system control. However, this feature is controlled by the SYCOM parameter "PROGRAM". For this feature to be available, that parameter "must be" specified when the SYCOM parameters file is created.

```
                    The User System

                    -----------------------------------
                    |  .............................  |
                    |  .            MCP            .  |
                    |  .............................  |
                    |    |                       |    |
                    |    |        .........      |    |
                    |    |   ---->.  AP   .      |    |
                    |    |   |    .........      |    |
                    |    |   |                   |    |
                    |    |   |                   |    |
                    |    |   V        ..............  |
                    |  .........      . Network    .  |   |-----|
                    |  . SYCOM .<---->. Controller .------| D/S |
                    |  .........      ..............  |   |-----|
                    -----------------------------------     /
                                                           /
                                                          /
                                                         /
                                                        /
                                                       /
The Server System                                     /
                                                     /--/
-----------------------------------------               /
| ..................................... |              /
| .                 MCP               . |             /
| ..................................... |            /
|   |    |    |                     |   |          /
|   |    |    |                     |   |         /
|   |    |    |                     |   |        /
|   |    |  .........    .............. |       /
|   |    |  . SYCOM .<-->. Network    . |   |-----|
|   |    |  .........    . Controller .-----| D/S |
|   |    |               .............. |   |-----|
|   |    |                 A   A        |
|   |  ..........          |   |        |
|   |  .  SMCS   .<---------|   |        |
|   |  ..........              |        |
|   |                          |        |
|   |                          |        |
| ..........                    |        |
| .   AP   .<------------------|        |
| ..........                            |
-----------------------------------------
```

Figure II-7: Terminal-to-SYCOM schematic

PROGRAM-TO-PROGRAM COMMUNICATIONS

The program-to-program capability employs an entirely different concept for inter-system communication. As previously discussed, this feature is an inherent by-product of the system. SYCOM itself is in no way involved with the program-to-orogram message traffic and therefore, does not provide any services to either program. The only exception to this is that SYCOM does "aoprove" a virtual terminal position for the program communications path.

As shown in Figure II-8 (below), communications are directly thru the Network Controllers via a special "remote file" declared in each Controller; explicitly for that purpose. Both application programs must have mirror images of these remote files (one each) declared as well. This remote file is shown in both Controller examples (Section V) as well as both sample application programs (Appendix I).

Figure II-8 can be used to visualize how an application program in one system can communicate with a program in another system. The input/output path is for the program to accept input messages from the Network Controller and return outout reply messages for transmission to the other system/program. It would behove the developer of these application programs to design a specific set of messages which exclude any terminal connotations. This could greatly simplify development efforts extended for these programs.

```
The User System
------------------------------
! ......................... !
! .          MCP          . !        The Server System
! ......................... !        -----------------------------
!                       !   !        ! ........................ !
! ........  .........   !   !        ! .         MCP          . !
! .  AP  .  . SYCOM .   !   !        ! ........................ !
! ........  .........   !   !        !   !                      !
!    A          !       !   !        !   !            ........  !
!    !          !       !   !        !   !            .  AP  .  !
!    !          !       !   !        !   !            ........  !
!    !          !       !   !        ! ..............     A     !
!    !      ............... !   ---------. Network    .   !     !
!    !      . Network     . !   \    ! . Controller .<----!     !
!    ------->. Controller .--------  ! ..............           !
!           ............... !        -----------------------------
------------------------------
```

Figure II-8: Program-to-Program Schematic

To convey certain required basic information with these input/ output messages, a "header" is required and must contain specific information. This header is defined in Section VI, "Appendix E, Message Formats". There are no restrictions as to the text in these messages but the length is controlled to a degree be the "BUFFERSIZE" parameter. This parameter is in SYCOM's "parameters file" and can be specified at initialization time.

The usefulness of this feature is to be evaluated and decided upon by the system user. The capability is there to be utilized for whatever purpose.

NETWORK CONTROL

The SYCOM User System is normally considered to be the remote RJE Station and must initiate the transmission of data to the Server System. In a point-to-point non-switched network, this is accomplished automatically by enabling the line adapter and then exchanging handshaking control messages with the Server System.

In a point-to-point switched network, the User System must call the Server System. This is accomplished by either dial-out (automatic dialing by the computer) or by manual dial-up (dialing performed by the system operator). The dialing is done "after" the adapter is enabled. When the line connection is established, the exchanging of handshaking control messages with the Server System is carried out. In a point-to-point non-switched network utilizing a direct connection, no dialing is done; or possible.

Once this communication link has been established with a Server System, line communication is identical for the switched and non-switched networks. (The User System and the Server System must be compatible as to transmission rate and other such optional hardware capabilities.)

Conversational Communication Procedure

The following notes refer to the basic conversational line discipline employed by SYCOM. They apply to a point-to-point circuit, either dedicated or switched:

Message Transfer: All message blocks are delineated by special characters. Messages begin with a "SOH" character and end with a "BCC" character. The message control format is as follows: SOH, ARM, IM#, STX, DA1, DA2, message text (blocked records), ETX, and finally the BCC.

Retransmission of Messages: If the transmitting system does not receive a reply to its last transmission within a given period of time, the unacknowledged message block is re-transmitted.

Parity or Block-Check Error Handling: If the receiving system receives a recognizable message, but it contains either a parity or block-check error, it replies with a "NAK". The transmitting system should retransmit the message. This is repeated "n" number of times ("n" being user specified), and if the error persists, either system can initiate error recovery.

Message Idle State: If the receiving system receives a "good" message, but has no message traffic to send, it responds with an "ACK". If the sending system has more message traffic, it sends another message block. If the sending system has no more message traffic, it proceeds to the "message idle state".

Normal Termination: The normal termination of a conversation occurs when one system sends an "ACK" (indicating that it has no more message traffic), and the system receiving the "ACK" also has no more message traffic. The system receiving the "ACK" proceeds to the "message idle state".

Point-to-Point Contention Procedure

The essential characteristic of this procedure is that after the end-of-message transfer, as indicated by the transmission of the "DLE-EOT" sequence, both systems are in an idle state. Either system may then bid for master status by sending the "DLE-ENQ" sequence. This is normally initiated by the entry of a command to SYCOM.

Master Status: If either system is in the idle state and wishes to begin a message transfer, it may bid for master status by sending the "DLE-ENQ" sequence. If the same system had just terminated a message transfer by sending a "DLE-EOT" sequence, then there is a delay created before this system again sends the "DLE-ENQ" sequence.

Begin Message Transfer State: If the system bidding for master status receives an "ACK" (indicating ready to receive), it may commence message transfer immediately. If the response is "NAK" (not ready to receive), the "DLE-ENQ" sequence may be repeated up to a retry limit of "n" ("n" being user specified), and if no valid response is received, commence error recovery. If the system bidding for master status receives the "DLE-ENQ" sequence (indicating that the other system is bidding at the same time), it abandons its own bid and replies in accordance with its readiness to receive. In order to avoid any line collisions when both systems are bidding at the same time, the no-response time-out values must be set to different values. This is true whether not the systems are "like" or "unlike".

Termination: The "DLE-EOT" sequence is transmitted by the system directed to terminate to indicate the end of its session, and if using a switched line, to disconnect the modem.

Transmission of Packed Data

As in previous versions of this protocol, data records within a message block are separated by "RS" characters. Repeated occurrences of blank characters are compressed out using the "ESC-integer" mechanism. The method of representing non-printable characters has changed with SYCOM's implementation. Printable characters are those listed in "Appendix B, Character Set", with the exception of the "Line Feed" and "BELL" characters.

If the data records contain only printable characters, then the first character of the record "is not" a "DLE" character. If the data record contains any character not in the "character set", then the first character of the record must be a "DLE" character, and the remainder of the record contains translated data. The original data is translated such that every 4-bits is mapped into an 8-bit printable character. This mapping is accomplished by internal translations so that each 4-bits of data is prefixed with a 4-bit hexadecimal value "4". (Actual data transmissions occur in ASCII character mode.)

NETWORK CONTROLLER

The Network Contrcller provides interfaces with the User System's terminals and Server Systems at the hardware level. It is responsible all for direct communication with those terminals and Server Systems. It performs the actual data transmission (both transmitting and receiving) functions between two systems. The Network Controller accepts data message blocks and system control messages from SYCOM and certain application programs (program-to-program) and transmits them to the Server System. In reverse, it receives all messages from a Server System ard forwards them as required for handling control and distribution. The SYCOM program relies upon the MCP for the basic function of "message queuing" between it and the Network Controller and certain application programs. This facility provides the message paths between SYCOM, the SMCS, application programs, and the Network Controller.

Following is a normal sequence of events which transpires between SYCOM, the Network Controller, and the system's terminals:

In response to a "poll" by the Network Controller, a terminal transmits a message back to the Network Controller. The Network Controller accepts the message and appends a header, supplying pertinent information about the message, (e.g., terminal type, lsn, etc.). The Network Controller then determines which program to send this message to and writes it into the appropriate program's remote file (input queue). (Assume in this case it is SYCOM.) SYCOM reads its remote file (input queue) and obtains an input message. Should the message be a SYCOM command, it is processed and a reply message created. The reply message is then started down the line by SYCOM when it writes the output message into its remote file to the Network Controller. The Controller in turn reads the reply message and transmits it (devoid of header of course) to the originating terminal.

The description on the relationships of the SMCS and the Network Controller with SYCOM, which is also considered an application program, apply to any application program. Thus, any application program can enjoy the benefits of a sophisticated terminal network with message/terminal/program management thru the SMCS and Network Contrcller. In summation, the SMCS is the message controller for the system. It also provides a myriad of functions which direct Network Controller functions as well as special purpose MCP functions.

CONFIGURATIONS AND SOFTWARE ASSOCIATIONS

The Systems Communications Module has been designed to operate in either of three modes; 1) as an individual entity "without" local terminals, 2) as an individual entity, "with" local terminals, and 3) in conjunction with the Supervisory Message Control System (SMCS); also with terminals. Referring to Figures II-9, II-10, and II-11 (which follow), system configurations with supporting software elements are shown; and describec.

Entry Level System

This system configuration provides all of the SYCOM capabilities previously defined which includes the SPO as a virtual terminal. This is accomplished with the smallest memory size commitment. Although the command repertoire is generally confined to entry from the system SPO, special application programs running in the system can alternately inject commands to SYCOM for processing.

In the User System schematic which follows (Figure II-9), the communication links between system software elements are shown. Each of these links, "queues" and "remote files", have specific purposes which are briefly identified as follows:

A. "INQ" is a queue file used by SYCOM as output to an application program utilizing the "program-to-SYCOM" facilities.

B. "OUTQ" is also a queue file used by SYCOM as input from an application program utilizing the "program-to-SYCOM" facilities.

C. "LINK" is a remote file which is used in communications with a remote Server "host" System. All RJE and virtual terminal input/output message traffic moves thru this file. This remote file is directly related to the SYCOM communications line to a remote host.

D. "RFUSER" is a user defined name for a remote file used for program-to-program I/O message traffic, primarily directly thru the Network Controllers. All messages for this remote file move thru the same communications line as those identified for the "LINK" file.

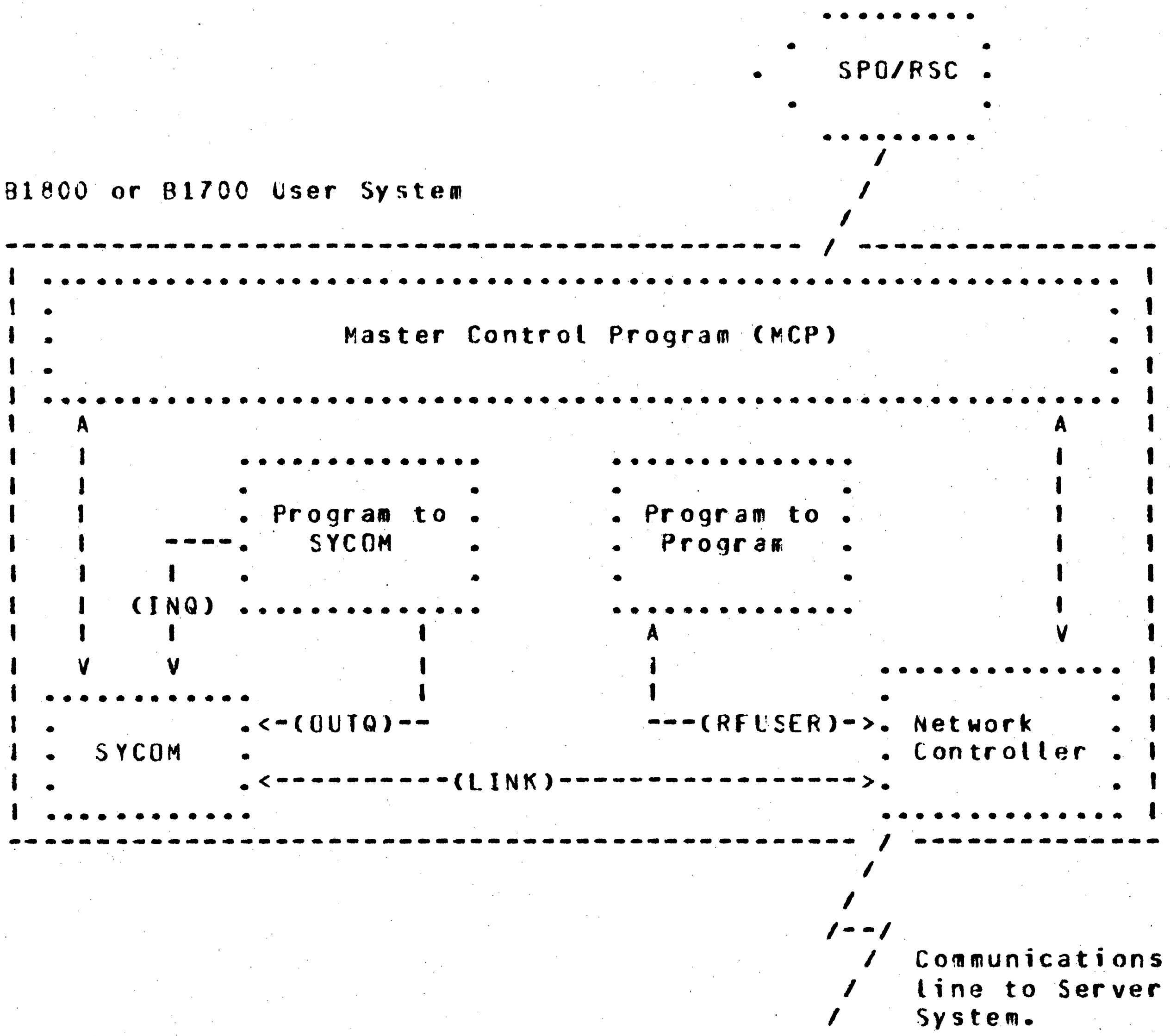

SPO/RSC
B1800 or B1700 User System
Master Control Program (MCP)
Program to SYCOM
Program to Program
(INQ)
SYCOM
<-(OUTQ)--
---(RFUSER)->
Network Controller
<----------(LINK)----------------->
Communications line to Server System.

Figure II-9: User System Schematic

Extended Level System

This system configuration also provides all SYCOM services as outlined for the Entry Level System and extends this to include local terminals. With terminals, SYCOM functions can be invoked from outside the computer room and away from the system SPO. Terminals on the system increase memory usage in proportion to the number of terminals attached to the system and the line or lines for terminals. The memory increase however, will be in the Network Controller program and not SYCOM itself.

Local terminals are exclusively owned and controlled by SYCOM. Although there is no operational SMCS required in a User System, it is "required" in a B1800/B1700 Server System when Remote SPO type activities are needed. The introduction of terminals in this system requires another remote file declared in both SYCOM and the local Network Controller:

A. "TERMINALS" is the remote file used by SYCOM for local terminal input/output message traffic between it and the Network Controller. As the contrcller handles all necessary terminal interfacing, SYCOM only reads input from and writes output to this remcte file.

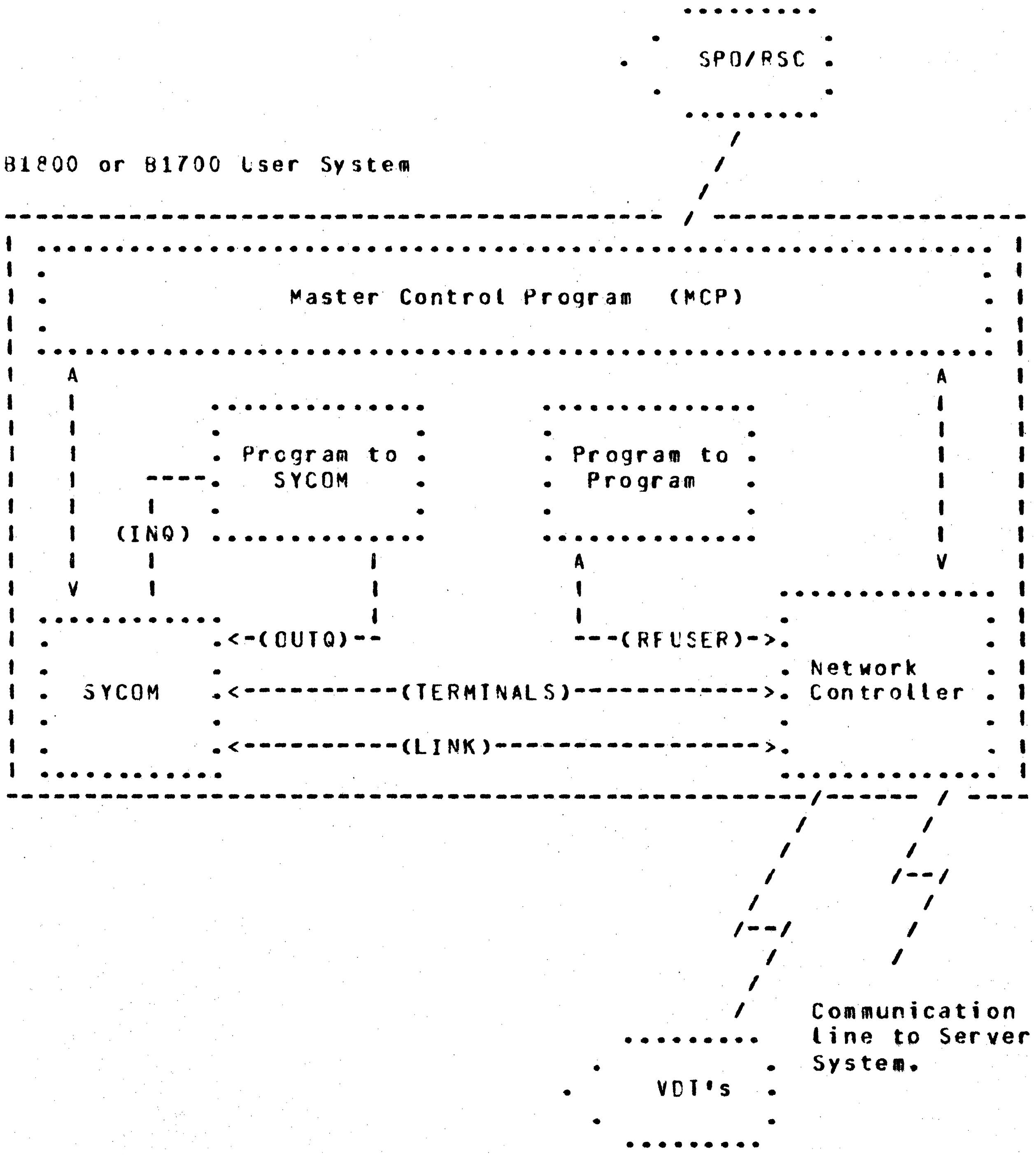


Figure II-10: User System Schematic (terminals)

Ultimate Level System

Using this configuration, not only are all of the capabilities provided as outlined for both Entry Level and Extended Level systems, it now provides the SYCOM terminal user with an access to all of the capabilities afforded by the Supervisory Message Control System (SMCS). Memory increases in this configuration are directly related to the SMCS and the Network Controller.

Local terminals in this system configuration are under direct control of the SMCS. Upon command (SMCS "sign on"), a terminal is temporarily "attached" to SYCOM for functional directives. When SYCOM activities are complete, the terminal is "detached" from SYCOM with the ".BYE" command and is again under the control of the SMCS. As such, an additional terminal I/O communications link is needed. Also, in addition to this link, the remote file "TERMINALS" is effected somewhat. Both are described as follows:

A. "MCSREMOTE" is a remote file used by the SMCS as a link to the system's terminals. All terminal I/O message traffic between SMCS and the Network Controller passes thru this file. During the time a terminal is "signed-on" to SYCOM, certain messages are "switched" to the SMCS for processing (those messages prefixed with a SMCS "signal character").

B. "TERMINALS" is SYCOM's remote file for terminal input/output message traffic between it and the Controller. The slight difference being that terminal input/output only passes thru this file for those terminals that are "signed-on" to SYCOM.

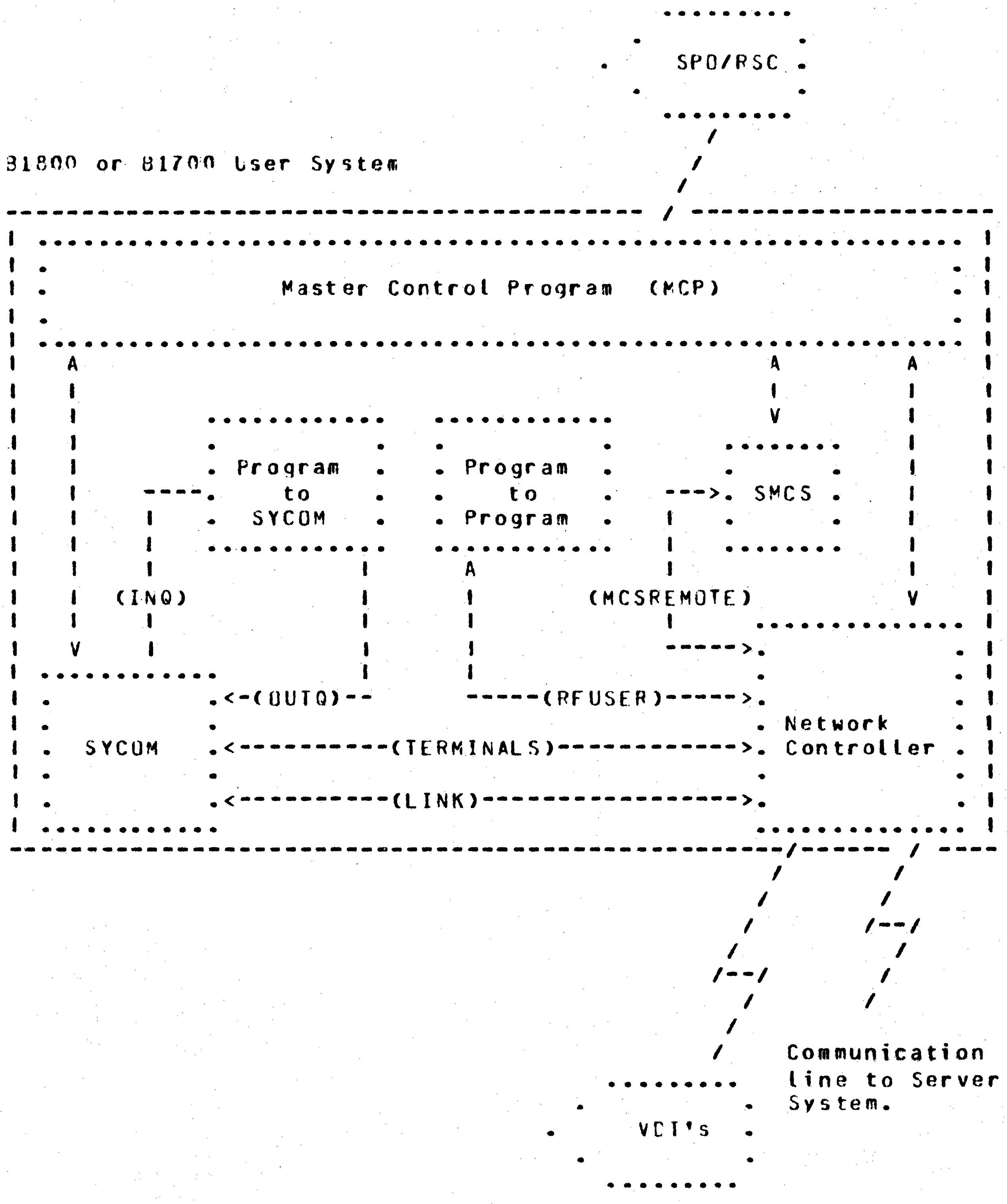

SPO/RSC
B1800 or B1700 User System
Master Control Program (MCP)
Program to SYCOM
Program to Program
SMCS
(INQ)
(MCSREMOTE)
SYCOM
<-(OUTQ)--
-----(RFUSER)----->
<-----------(TERMINALS)------------->
<-----------(LINK)----------------->
Network Controller
VCT's
Communication line to Server System.

Figure II-11: User System Schematic (SMCS; terminals)

MESSAGE PATH ASSIGNMENTS

In order to describe the potential message communication paths opened up by the virtual terminal capability, a special message path syntax was developed to pictorially present this myriad of possibilities. The message path syntax is easy to understand as one can visualize these logical paths thru the User and Server Systems. System software elements are easily recognized by the widely used acronyms (all defined in the "Glossary"). Although the directional flow arrows "-->" indicate a single direction, communications are two-way. The boxed in syntactical elements to the right indicate an alternate message path; also available thru the "virtual terminal" capability.

For the Entry Level System

SPO input:

For an "Entry Level System" (Figure II-9) there are three message paths for SPO input. The following examples explore each of these possibilities:

```
                            User System  |  Server System
-----------------------------------------+-----------------------
                                         |
                            .....................................
                            :                                   :
SPO  -->  MCP  -->  SYCOM  ---->  NC  -----                     :
                            :            /                      :
                            :           -----  NC  ----> SYCOM  :
                            .....................................
                                         |
     .<XXX>  ---->  SYCOM                |
      <XXX>  -------------------------------> SYCOM ---> SPO
                                         |
```

As shown in this diagram, all input command strings are given to User SYCOM by the MCP. Messages prefixed with a period "." are processed by User SYCOM. All other input messages are sent to Server System's SYCOM which displays them on that SPO. When the "virtual terminal" capability is invoked (by using the ".CONNECT" command), then the following message paths are established:

```
                            User System  I  Server System
-----------------------------------------+-----------------------
                                         I
     .CONNECT  -->  SYCOM                I
                                         I
                            .....................................
                            :                                    :
SPO  -->  MCP  -->  SYCOM  ---->  NC  -----                      :
                            :            /                       :
                            :           -----  NC  ---->  SMCS   :
                            .....................................
                                         I
     .<XXX>  ---->  SYCOM                I
     <XXX>   ------------------------------------------> SMCS
     ?<XX>   ------------------------------>  SMCS  ---> MCP
     ?<nnn>  ------------------------------>  SMCS  ---> MCP
                                         I
```

If there is no SMCS currently running in the Server System, then the message string is intercepted by the Network Controller which causes an error message to be displayed on that system's SPO.

When a SMCS is running in the Server System, the SPO operator can either "sign-on" to a program in that system or initiate RSC type commands; thru the SMCS. When the virtual terminal is signed-on to the specified application program, it is "attached" to that program. All further non-prefixed input is sent to that program for processing. (In the example below, the "$" character is assumed to be the Server System's SMCS "signal character"; it could be another character.)

```
                            User System  I  Server System
-----------------------------------------+-----------------------
                                         I
     SIGN ON AP  ---->  SYCOM            I
                                         I
                            .....................................
                            :                                    :
SPO  -->  MCP  -->  SYCOM  ---->  NC  -----                      :
                            :            /                       :
                            :           -----  NC  ---->  AP     :
                            :                         \-> SMCS   :
                            .....................................
                                         I
     .<XXX>  ---->  SYCOM                I
     <XXX>   ------------------------------------------> AP
     $<XXX>  ------------------------------------------> SMCS
     $?<XX>  ------------------------------>  SMCS  ---> MCP
     $?<nnn> ------------------------------>  SMCS  ---> MCP
                                         I
```

Program-to-SYCOM input:

For "Program-to-SYCOM" communications, the following message path assignments are available:

```
                            User System  |  Server System
-----------------------------------------+-----------------------
                                         |
                            .....................................
                            :                                   :
AP   ------------>  SYCOM  ---->  NC  -----                     :
                            :           /                       :
                            :           -----  NC  ----> SYCOM  :
                            .....................................
                                         |
     .<XXX>  ---->  SYCOM                |
     <XXX>   -------------------------------> SYCOM ---> SPO
                                         |
```

All input command strings are given to User SYCOM by the MCP via the "OUTQ" queue. Messages that are prefixed with a period "." are processed by User SYCOM. All other input messages are sent to Server System's SYCOM which displays them or that SPO.

When the "virtual terminal" capability is invoked (be entry of a "CONNECT" command), the following message paths are established:

```
                            User System  |  Server System
-----------------------------------------+-----------------------
                                         |
     .CONNECT  -->  SYCOM                |
                                         |
                            .....................................
                            :                                   :
AP   ------------>  SYCOM  ---->  NC  -----                     :
                            :           /                       :
                            :           -----  NC  ----> SMCS   :
                            .....................................
                                         |
     .<XXX>  ---->  SYCOM                |
     <XXX>   ------------------------------------------> SMCS
     ?<XX>   -------------------------------> SMCS  ---> MCP
     ?<nnn>  -------------------------------> SMCS  ---> MCP
                                         |
```

If there is no SMCS currently running in the Server System, then the message string is intercepted by the Network Controller which causes an error message to be displayed on that system's SPO. When a SMCS is running in the Server System, this program can "sign-on" to a program in that system or initiate PSC commands; thru the SMCS. When this virtual terminal (program) signs-on to a specific application program, it becomes "attached". All non-prefixed input is sent to that Server program for processing. (In the example below, the "$" is assumed as the Server System's SMCS "signal character"; it could be something else.)

```
                            User System  |  Server System
-----------------------------------------+-----------------------
                                         |
     SIGN ON AP  ---->  SYCOM            |
                                         |
                            .....................................
                            :                                   :
AP   ------------>  SYCOM  ---->  NC  -----                     :
                            :            /                      :
                            :           -----  NC  ----> AP     :
                            :                        \-> SMCS   :
                            .....................................
                                         |
     .<XXX>  ---->  SYCOM                |
     <XXX>   ------------------------------------------> AP
     $<XXX>  ------------------------------------------> SMCS
     $?<XX>  ------------------------------->  SMCS  ---> MCP
     $?<nnn> ------------------------------->  SMCS  ---> MCP
```

Program-to-program input:

The message path for program-to-program communications is limited to the following predetermined path. Messages can be any format and contain any text agreed upon by both application programs. The only prevailing restrictions are maximum "buffersize" and all messages must be prefixed with a "header" ("Appendix E").

```
                            User System  |  Server System
-----------------------------------------+-----------------------
                                         |
AP   -------------------------->  NC  -----
                                         /
                                        -----  NC  ----> AP
                                         |
     <XXX>   ------------------------------------------> AP
                                         |
```

For the Extended Level System

In this configuration (Figure II-10), the SPO and all terminals attached to, and owned by, SYCOM normally communicate with local SYCOM. When the virtual terminal capability is utilized, these devices are permitted to communicate with either the system software programs or appropriate application programs in a Server System.

The SPO, program-to-SYCOM, and program-to-program input message paths for this software configuration are the same as those defined for the "Entry Level System". The below message path assignments are for local terminals attachec to this system:

```
                                    User System  |  Server System
-------------------------------------------------+---------------------
                                                 |
VDT  -->  NC    -->  SYCOM                       |
                                                 |
                                                 |
     .<XXX>  ---->  SYCOM                        |
                                                 |
```

By entry of the ".CONNECT" command to SYCOM the virtual terminal capability is invcked. The following message paths are then established for the terminal:

```
                                                 |
     .CONNECT  ---->  SYCOM                      |
                                                 |
                                      ....................................
                                      :                                  :
VDT  -->  NC    -->  SYCOM  ---->  NC  -----                             :
                                      :          /                       :
                                      :     -----  NC  ---->  SMCS       :
                                      ....................................
                                                 |
     .<XXX>   ---->  SYCOM                       |
     <XXX>    ------------------------------------------->  SMCS
     ?<XX>    ------------------------------>  SMCS  --->  MCP
     ?<nnn>   ------------------------------>  SMCS  --->  MCP
                                                 |
```

If there is no SMCS currently running in the Server System, then the message string is intercepted by the Network Controller which causes an error message to be displayed on that system's SPO.

When a SMCS is running in the Server System, the terminal user can either "sign-on" to a program in that system or initiate RSC type commands; thru the SMCS. When a virtual terminal signs-on to a specified application program, it is "attached" to that program. All further non-prefixed input is sent to that program for processing. (In the example below, the "$" character is assumed to be the Server System's SMCS "signal character"; it could be another character.)

```
                             User System  |  Server System
------------------------------------------+-----------------------
                                          |
     SIGN ON AP   ---------------------------------------> SMCS
                                          |
                             ......................................
                             :                                    :
VDT  -->  NC   -->  SYCOM   ---->  NC  -----                      :
                             :            /                       :
                             :           -----  NC  ----> AP      :
                             :                        \-> SMCS    :
                             ......................................
                                          |
     .<XXX>   ---->  SYCOM                |
     <XXX>    -------------------------------------------> AP
     $<XXX>   -------------------------------------------> SMCS
     $?<XX>  --------------------------------> SMCS  ---> MCP
     $?<nnn> --------------------------------> SMCS  ---> MCP
                                          |
```

For the Ultimate Level System

In this configuration (Figure II-11), all local terminals are attached to, and owned by the SMCS. Therefore, the normal message communication path is to the User System's SMCS. When the virtual terminal capability is utilized, these terminals are permitted to communicate with either system software programs or appropriate application programs in the Server System. Additionally, these same terminals are permitted to communicate with any appropriate application program in the User System.

The SPO, program-to-SYCOM, and program-to-program input message paths for this software configuration are the same as those defined for the "Entry Level System". The following message path assignments are for all local terminals attached to the User System's SMCS. First the normal message paths for terminal input within the User System are presented:

```
                           User System  |  Server System
----------------------------------------+----------------------
                                        |
VDT  -->  NC   --> SMCS                 |
                                        |
                                        |
     <XXX>   ----> SMCS                 |
     ?<XX>   ----> SMCS  -->  MCP       |
     ?<nnn>  ----> SMCS  -->  MCP       |
                                        |
```

In order for the terminal user to gain access to the system's SYCOM, the terminal user must enter the SMCS "SIGN ON" command as shown below. (Although this example is for a sign-on to SYCOM, it could be any appropriate application program in the User System.)

```
                                        |
     SIGN ON SYCOM  --------->  SMCS    |
                                        |
```

The following message path assignments are then in effect for all terminal input.

```
                                        |
                            ....................................
                            :                                  :
VDT  -->  NC   ---->  SYCOM ---->  NC  -----                   :
                \->   SMCS  :           /                      :
                            :         -----  NC  ---->  SYCOM  :
                            ....................................
                                        |
     .<XXX>  ----->   SYCOM             |
     <XXX>   ------------------------------>  SYCOM  --->  SPO
     *<XXX>  ----->   SMCS              |
     *?<XX>  ----->   SMCS   -->  MCP   |
     *?<nnn> ----->   SMCS   -->  MCP   |
                                        |
```

The "virtual terminal" capability is invoked when the terminal user enters the ".CONNECT" command to SYCOM. Following are the message paths now established for the terminal; if a SMCS is not currently running in the Server System:

```
                                   User System  |  Server System
-----------------------------------------------+---------------------------
                                               |
     .CONNECT   ---->  SYCOM                   |
                                               |
                                   ......................................
                                   :                                    :
VDT  -->  NC   ---->  SYCOM  ---->  NC   -----                          :
               \->  SMCS   :                /                           :
                           :              -----  NC  ---->  SMCS       :
                                   ......................................
                                               |
     .<XXX>   ---->  SYCOM                     |
     <XXX>    ---------------------------------------------->  SMCS
     *<XXX>   ---->  SMCS                      |
     *?<XX>   ---->  SMCS   -->  MCP           |
     *?<nnn>  ---->  SMCS   -->  MCP           |
                                               |
```

If there is no SMCS currently running in the Server System, then the message string is intercepted by the Network Controller which causes an error message to be displayed on that system's SPO.

When the "virtual terminal" capability is invoked and there is a SMCS currently running in the Server System, then the following message paths are established for the terminal:

```
                                               |
                                   ......................................
                                   :                                    :
VDT  -->  NC   ---->  SYCOM  ---->  NC   -----                          :
               \->  SMCS   :                /                           :
                           :              -----  NC  ---->  SMCS       :
                                   ......................................
                                               |
     .<XXX>   ---->  SYCOM                     |
     <XXX>    ------------------------------->  SMCS
     ?<XX>    ------------------------------->  SMCS  -->  MCP
     ?<nnn>   ------------------------------->  SMCS  -->  MCP
     *<XXX>   ---->  SMCS                      |
     *?<XX>   ---->  SMCS   -->  MCP           |
     *?<nnn>  ---->  SMCS   -->  MCP           |
                                               |
```

With the SMCS running in the Server System, a terminal user can either "sign-on" to a program in that system or initiate RSC type commands; thru the SMCS. When a virtual terminal is signed-on to a specified application program, it is "attached" to that program until the terminal user "signs-off". All non-prefixed input is sent to that application program for processing. (In the examples below, the "*" character is assumed to be the User System's SMCS "signal character" and the "$" character is assumed to be the Server System's SMCS "signal character"; they can be different.)

```
                           User System  I  Server System
----------------------------------------+-----------------------
                                        I
     SIGN ON AP --------------------------------------> SMCS
                                        I
                           .....................................
                           :                                   :
VDT  -->  NC  ---->  SYCOM ---->  NC  -----                    :
                \-> SMCS   :            /                      :
                           :           -----  NC  ----> AP     :
                           :                        \-> SMCS   :
                           .....................................
                                        I
     .<XXX>  -----> SYCOM               I
     *<XXX>  -----> SMCS                I
     *?<XX>  -----> SMCS  -->  MCP      I
     *?<nnn> -----> SMCS  -->  MCP      I
     <XXX>   -----------------------------------------> AP
     $<XXX>  -----------------------------------------> SMCS
     $?<XX>  ----------------------------->  SMCS  -->  MCP
     $?<nnn> ----------------------------->  SMCS  -->  MCP
                                        I
```

QUEUE DEPTH CONTROL

In the SYCOM operational environment, there is a MCP controlled data path which exists between SYCOM and the Network Controller. This data path is referred to as the "remote file" (or message queue) mechanism. Due to differences in data communication line transmission rates and peripheral input speeds, this remote file becomes a buffer. Without some kind of control on the message traffic thru this file, it could grow beyond reasonable limits or be jammed full.

To provide a control over the number of message blocks that can be queued up in this file at any given time, a "marker message" system is employed. As SYCOM is putting message blocks into this file, a count is maintained. When this count reaches the halfway point of the queue depth limit, a marker message is put into the file. The marker message is a specially prepared data message with a code in the message header. This special code ensures recognition by SYCOM when this message is echoed back to from the Network Controller.

Once a marker message is placed into the file, SYCOM continues putting message blocks into the file until the predetermined queue depth limit is reached, at which time peripheral input operations are suspended. When the Network Controller encounters the marker message during the transmission of the queued message blocks, it acknowledges receipt of a marker message by returning a "flag" in the corresponding "ACK" message to SYCOM. When SYCOM receives this acknowledgement, it resumes peripheral input, again places a marker message into the file and continues putting message blocks into the file until the predetermined queue depth is reached or until an end-of-file condition, whichever occurs first.

SYSTEM FILES

In the total system picture, there are many files and file types. However, as this system overview pertains only to SYCOM and its relationships, only those pertinent files which are required for SYCOM operations are presented. Following are file descriptions, by software element, that are necessary for SYCOM operations:

MASTER CONTROL PROGRAM (MCP) FILES

(SYSTEM)/USERCODE File

This file contains all valid usercode-passwords that are known to the system. It is used by the MCP to enforce file security disk files and programs/users that access them. With this file, the MCP can service any requests by programs regarding validity of specific usercode-password combinations.

NETWORK CONTROLLER FILES

Network Information (NIF) File

This file contains tables that describe the physical and logical attributes of the network. The initialization values for lines, terminals, and file tables of Network Controller are contained in this file. This file is created by the NDL compiler when it is in the process of generating a Network Controller.

SYCOM FILES

The Systems Communications Module utilizes many types of files in the management of its domain. There are, peripheral (includes disk), remote communications, and queue files. These files are briefly described here, under separate headings, with detail descriptions for major files available in the appendices.

Peripheral Files

These files are generally used to input/output RJE type data and include the card reader, card punch, line printer, magnetic tape, and disk files. Also there is a disk file which is used by SYCOM to manage the various options and settings. Following are general descriptions of the SYCOM files which fall into this category:

A. PARAMETERS File

The PARAMETERS file is a single record file containing required operational parameters for SYCOM. It is static and can only be changed by removal and recreation. This file is created by SYCOM upon the entry of appropriate parameters from either the card reader or the User SPO.

B. Card Input File

This file is used to input RJE card data files from the system's card reader. It can be either a 80-column or 96-column device.

C. Card Output File

This file is used to punch RJE card output data files to the system's card punch. It can be either a 80-column or 90-column device, as appropriate.

D. Printer File

This file is used to print all print files that are being, or have been, received from a Server System. The line printer device can by any one of the various models which can be attached to the system. The device can be either a 120-position or 132-position line printer.

Remote Files

The remote files are used in communications between SYCOM and the Network Controller and SYCOM and either the terminals or SMCS. There are two of these files which are described as follows:

A. LINK File

The "LINK" file is used for all input/output messages relating to RJE and file transfers; specifically, station types for Control, SPO, Cards, Printer, and File records.

B. TERMINALS File

The "TERMINALS" file is the communication link between SYCOM and the Network Controller for local terminal messages. This remote file is however, only used when there are local terminals; controlled by either the SMCS or SYCOM.

Queue Files

The queue files are utilized by SYCOM for buffered communications between itself, the MCP, and the Network Controller. The purpose of each of these cueue's is described as follows:

A. RJE Queue

Used to save printer/punch data received from the other system when the User System's printer or punch is busy.

B. LINK Queue

1. Used to save input messages from the other system during the "initialization" phase; else they may be lost in the shuffle.

2. Used to save output messages for the other system when "messages queued" exceeds "MAX-MESSAGES".

C. INQ Queue

This queue is the mechanism by which SYCOM sends reply messages to an application program dealing in program-to-SYCOM communications. SYCOM "writes" into it and the application program "reads" from it to obtain messages.

D. OUTQ Queue

This queue is the mechanism whereby SYCOM receives input command messages from an application program that deals in program-to-SYCOM communications. The application program "writes" into it and SYCOM "reads" from it to obtain input messages.

RESTRICTIONS

The following restrictions are valid for the system based on a logical evaluation of the system's requirements:

- Maximum message block size = 1950 bytes
- Minimum message block size = 78 bytes
- Maximum simultaneous system connections = 1
- Maximum message block queue cepth = 255
- Maximum number events queued to process = 99
- Maximum number of terminals supported = 15
- Maximum number of card readers supported = 1
- Maximum number of card punches supported = 1
- Maximum number of line printers supported = 1
- LOAD.DUMP tapes are not supported for file transfer operations.
- Between unlike systems, only file transfers of code files and 8-bit data files are allowed.
- Only "data type files" are file transferable between disk/tape and tape/disk (codes files are invalid).
- Multi-point networks are not supported by SYCOM; at this time.
- Message blocking is prohibited. (A message block already contains multiple records or data images.)
- Monitoring and controlling of programs in the Server System requires a SMCS executing in that system.
- For systems using "binary-synchronous" transmission lines, the NDL generated Network Controller requires certain changes in the area of "NAK" conversions.

These restrictions may be changed during subsequent development cycles, but are however, currently in effect.

SECTION III

==

COMMANDS

SECTION III

COMMANDS

GENERAL

The commands available with the Systems Communications Module (SYCOM) are described in this section. These commands are used to give SYCOM directions as to specific actions which must be executed in order to perform the required tasks. The commands are presented alphabetical order within two groups, those allowed from terminals and those restricted to entry from the system SPO or program. This way, the terminal user need not be encumbered with the details of SPO commands; which are not permitted.

To ensure that each command is physically separated, a unique format presentation, which is described below, has been developed for this document. This scheme of formatting presents commands in modular format which allows for expansion and change without significant disruption of existing syntax.

COMMANDS

Each statement in the Command Language consists of a command and, in some cases, a list of parameters. The parameters in the list are separated from the command and each other by blanks. In the following examples single spaces are used although the use of multiple spaces between command elements could be used.

When the SYCOM has finished processing a command, it displays a reply message either confirming successful completion of the command or indicating an error. Errors terminate the processing of a command; the entire command must be re-entered correctly.

COMMAND SYNTAX CONVENTIONS

The notation conventions that are described below are used throughout this section to describe the syrtax of each command.

Notation used in model format

First, a model of a command is presented. This gives the user an overview of the potential of the command structure.

```
--------------------------------------------------------------------
:                                                                  :
:                                                                  :
:  .COPY <file-name-1> [(USER=<usercode>)] [AS      ----------->  :
:                                           ONTO                   :
:                                                                  :
:                                                                  :
:  ---->  <file-name-2>] FROM (HOST=<id>) TO (HOST=<id>)  %       :
:                                                                  :
:                                                                  :
--------------------------------------------------------------------
```

BRACKETS: Information that is enclosed within brackets [] represents an option which may be used or omitted by the user. In the model example given above, the options USER, AS or ONTO, and "file-name-2" indicate three possible choices when using these options, or the choice of not using any option.

BROKEN BRACKETS: Left and right broken <> brackets are used to indicate that the enclosed variable within the brackets must be replaced by digits and/or letters whose value is determined by the context of the command.

UPPER CASE LETTERS: Words that appear in upper case letters indicate that the word is a literal and must be substituted verbatim, when used. In the model given above, the word .COPY would be entered first when constructing a command statement.

LOWER CASE LETTERS: Words that appear in lcwer case letters indicate that a value must be supplied. In the model above, the word "file-name-1" indicates that a character string representing the name of the file to be copied must be entered.

COMMAND STRING SYNTAX CONTINUATION: A few of the commands have a considerable number of syntactical items which overflow a printed line into two or more command string lines in the presentation. To provide a clear line of continuation for the command string, the use of line direction arrows are used. A line direction arrow appears as -----> and always flows left to right; to continue the command string. It is easy to follow a command string in this format as it is logical in nature.

END OF STATEMENT: The end of a command statement is indicated by the appearance of a % (percent sign). It is only a syntactical string terminator and is not required text for command input.

OPEN-CLOSE PARENTHESES (): When these characters appear in a command syntax statement they are required input. See the "COPY" command for a use reference.

SLASH /: When this character appears in a command syntax statement it is required input. A space or spaces on either side of this character are not allowed.

Punctuation Rules

The space, the comma, and the period are considered punctuation marks (except in literals) and are used as follows:

1. The space (blank) is a separator, required after words and literals in the absence of any other separator. A space may precede or follow any other separator, and many spaces are the same as one (except in literals).

2. A comma is a separator that is legal only where it is specifically indicated in the command syntax. The comma, where it is legal, can also serve as a terminator for words and numeric literals. The comma is never required, except to separate criteria in a command string.

3. The period is a required prefix to local SYCOM commands. It is never required within any command string syntax. Its only acceptability would be as an embedded character in a file name, system identifier, or usercode/password.

COMMAND RELATED ERROR MESSAGES

Error messages specific to the following SYCOM commands appear in Appendix A (Command Error Messages).

COMMAND SET

The command set which follows, is used to direct SYCOM operations from either the SPO, a system terminal, or an application program running in the system. Non-SYCOM commands are also permitted to direct User and Server System operations via respective SMCS's. This command set is separated into two subsets; 1) commands which are permitted from any device and 2) commards restricted to a SPO or special application program.

Terminal Commands

The following commands are permitted from terminals. They direct file transfer and remote Server System operations. Although these commands are for terminals, they are also permitted from the SPO and special application programs used for system control. However when these commands are entered from the SPO, the command string must be prefixed with the "job number" assigned to SYCOM followed by an "AX" and then the command string.

.ABORT

Function: The ABORT command is used to force SYCOM to "abort" an in-process file transfer, either receiving or transmitting. If "fetching" (receiving), the file created up to the point of aborting is either removed or purged as appropriate.

Syntax:

```
.ABORT FETCH  %
       PUT
```

Semantics: FETCH = causes the FETCH in-process to be aborted.

PUT = causes the PUT in-process to be aborted.

Default: None.

Example 1: (Kill off transmitting the current file.)

Input: --- .ABORT PUT

Reply: --- PUT OF FILE <file name> ABORTED

Example 2: (Stop receiving of incoming file transfer.)

Input: --- .ABORT FETCH

Reply: --- FETCH OF FILE <file name> ABORTED

Note: If a file transfer is not in-process when the command is entered, an appropriate error message is returned.

.BYE

Function: This command completely logs the current user off SYCOM. It is valid from all stations including the system SPO and application programs. The following sequence of events transpire when it is entered. A "BYE" text message is sent to the other system to cause a "log-off" from that system, the usercode is removed, an automatic "DISCONNECT" is initiated, and a "detach" message is sent to the local SMCS; if the terminal is not owned by SYCOM.

Syntax: .BYE %

Semantics: None.

Default: None:

Example:

```
   Input: ---   .BYE

   Reply: ---   GCODBYE: (usercode) LOGGED OFF SYCOM 11:45:32.1
```

Note: The initiating station need not be "logged-on".

.CONNECT

Function: The CONNECT command is used when a terminal user wishes to be logged-on to a Server System. Upon entry, the User System goes into "online" mode, attempting to establish communications with a Server System.

If a connection currently exists to a host Server System, the existing connection is utilized to attach the terminal to that system.

Syntax: .CONNECT [<message string>] % ?

Semantics: message string = this optional message string can be either a text character string or a valid usercode/password for a program's log-on.

Default: When no "message string" is specified, the user's usercode/password is forwarded to the other system if available; else blanks are sent.

Example: (Request attachment to the Server System.)

```
  Input: ---   .CONNECT  US MY/USER

  Reply: ---   ".CONNECT" INITIATED
```

Note: If the Server System is accessed via a "switched" line and its telephone number has been previously established by entry of the "PHONE" command, then this command initiates an automatic dial-out.

.COPY

Function: This is the command which is used to initiate "file transfers". The command syntax provides for the transfer of files "from" the initiating system to another system (a PUT) and from another system "to" the initiating system (a FETCH). The files can be in either magnetic tape or disk form and the only restrictions that apply are those identified in "Restrictions", Section II.

Syntax:

```
.COPY <file-name-1> [ (USER=<usercode-1>,  -------->

---> PASSWORD=<password-1> ) ]  ------------------>

---> [ AS   <file name-2> [ (USER=<usercode-2>, -->
       ONTO

---> PASSWORD=<password-2> ) ] ]  ---------------->

---> FROM  [<volume-1>]  ( [KIND=PACK  ,]  --------
                                 TAPE
                                 TAPE7
                                 TAPE9
                                 PETAPE

---> HOST=<system name-1> ) TO [<volume-2>]  ----->

---> ( [KIND=PACK  ,] HOST=<system name-2> )  %
            TAPE
            TAPE7
            TAPE9
            PETAPE
```

Note: An important point to remember is that the "parens" are REQUIRED syntactical elements as well as the keyword "FROM"...

```
.............
.           .
.      COPY .
.           .
.............
```

.COPY (continued)

Semantics:

file-name-1 = the actual name of the file to be transferred; including multifile identifiers, if any.

usercode-1, password-1 = the usercode/password values which identify the user for purposes of verifying access rights to the specified file.

AS = means create a file with the name specified as "file-name-2".

ONTO = means write over any existing file with the name specified as "file-name-2".

file-name-2 = a name to be given to the file that is transferred, at its destination.

usercode-2, password-2 = the usercode/password values which specify the security to be placed on the transferred file once the transfer has been completed.

volume-1 = the "pack-id" from where the file to be transferred can be found; providing the following KIND option is equal to "PACK". This option is not required for files on magnetic tape or system disk.

KIND = one of the keywords shown which is used to describe the peripheral device from where the file is to be transferred "from"; if other than a system disk.

system-id-1 = specifies the "system-id" from where the file is to be transferred from.

.COPY (continued)

Semantics: (continued)

volume-2 = the "pack-id" to where the file is transferred; providing the following KINC option is equal to "PACK". This cption is not required for files on magnetic tape or disk.

KIND = one of the keywords shown is used to describe the peripheral device to which the transferred file is to be written; if other than a system disk.

system-id-2 = the "system-id" of the system which is to receive the file.

Defaults:

The syntactical item "VOLUME" defaults to "system disk" if "KINC" is also defaulted.

The syntactical item KIND defaults to "KIND=PACK".

The default for "usercode-1" and "password-1" are those of the logged-on user.

The destination usercode/password (usercode-2 and password-2) are defaulted to those of the source usercode/password.

The destination file name (file-name-2) defaults to the original name of the file transferred and the "AS"/"ONTO" option defaults to "AS".

Options:

The syntactical abbreviatior "DISK" may be used in lieu of "KIND=PACK".

The syntactical abbreviation "USER=<usercode>/<password>" may be substitutec for the specified command syntax elements so named.

.COPY (continued)

```
Example 1:     (Transfer file "XYZ" from system A to system B.)

   Inout: ---  .COPY XYZ (USER=<usercode>/<password>) FROM
               (DISK, HOST = USER1) TO (DISK, HOST=SERVER1)

   Reply: ---  FILE XYZ START TRANSMISSION
          ---  FILE XYZ END TRANSMISSION

Example 2:     (Copy file from another system changing file name.)

   Inout: ---  .COPY DMPALL AS DUMPALL FROM (HOST = SERVER1)

   Reply: ---  FILE DMPALL START TRANSMISSION
          ---  FILE DUMPALL RECEIVED
          ---  FILE DMPALL END TRANSMISSION
```

Notes:

1. In the 7.0 implementation for file transfers from either tape or disk to tape in a receiving system, if a scratch tape is not mounted and/or ready, SYCOM will "hang" until the tape is made available; no notification is made of this condition and no input messages/commands are processed.

2. If a "usercode" is not wanted as a multifile-id to the file name in the receiving system, enter "file-name-1" prefixed with an asterisk "*".

.DISCONNECT

Function: This command must be used when the terminal user wishes to terminate a session with a host Server System. The terminal is logged-off of the Server System when a "BYE" text message is sent to the other system which is followed by a "disconnect" message. The terminal reverts back to its previous state of being signed-on to SYCOM. The terminal user is then free to enter more commands to SYCOM or sign-off (.BYE).

Syntax: .CISCONNECT %

Semantics: None.

Default: None.

Example:

```
  Input: ---   .DISCONNECT

  Reply: ---   STATION DISCONNECTED
```

Note: The terminal is logically disconnected but may or may not be physically disconrected. To the terminal user however, the net effect is the same.

.FETCH

Function: This command is the shorthand version of the "COPY" command for use in specifying explicit "FETCH" file transfer requests. The command syntax provides for transferring files from another system "to" the initiating system (a FETCH). Files can be either magnetic tape or disk (in any combination) and the only restrictions that apply are those identified in Section II, "Restrictions".

Syntax:

```
.FETCH <file-name-1> [(USER=<usercode-1>,  ------->

---> PASSWORD=<password-1>)]  ------------------->

---> [AS    <file name-2> [(USER=<usercode-2>, --->
      ONTO

---> PASSWORD=<password-2>)] ]  ----------------->

---> [FROM  [<volume-1>]  [(KIND=PACK   )]]  ------>
                                 TAPE
                                 TAPE7
                                 TAPE9
                                 PETAPE

---> [TO    [<volume-2>]  [(KIND=PACK   )]]  %
                                 TAPE
                                 TAPE7
                                 TAPE9
                                 PETAPE
```

Note: If a "usercode" is not desired as a multifile-id to the file name on the receiving system, then prefix "file-name-1" with an asterisk "*".

.FETCH (continued)

Semantics: Refer to the "COPY" command semantics.

Defaults: Refer to the "COPY" command defaults.

Options: Refer to the "COPY" command options.

Example 1: (Cbtain a copy of the "DMPALL" program from the Server System.)

```
   Input: ---   .FETCH DMPALL

   Reply: ---   FILE DMPALL START TRANSMISSION
          ---   FILE DMPALL RECEIVED
          ---   FILE DMPALL END TRANSMISSION
```

Note: In the 7.0 implementation for file transfers from either tape or disk to tape in a receiving system, if a scratch tape is not mounted and/or ready, SYCOM will "hang" until the tape is made available; no notification is made of this condition and no input messages/commands are processed.

.LIST

Function: This command's purpose is tc cisplay information about the system and its users. Command options provide lists of those users currently "signed-on", terminals that are currently "online", and file transfers that are currently "scheduled" to occur.

Syntax:

```
.LIST  USERS      %
       TERMINALS
       SCHEDULE
```

Semantics:

USERS = requests a list of all users currently "signed-on" to SYCOM.

TERMINALS = requests a list cf all terminals that are currently "cnline".

SCHEDULE = requests a list of all file transfers currently scheduled.

Default: Currently there is no default..

Example 1: (List the current signed-on users of the system.)

```
  Input: ---   .LIST USERS

  Reply: ---   [<uc index>] DENNIS ONLINE
         ---   [<uc index>] JEANNE OFFLINE
         ---   [<uc index>] CARLYE ONLINE
         ---   3 LOGGED ON
```

Example 2: (List the currently "online" terminals.)

```
  Input: ---   .LIST TERMINALS

  Reply: ---   [<rsn>] <SPO> <v.lsn> Aa aA
         ---   [<rsn>] <lsn> <v.lsn> Ab bA
```

.LIST (continued)

Example 3: (list all file transfers "scheduled" to occur.)

```
  Input: ---   .LIST SCHEDULE

  Reply: ---   SCHEDULE LIST NOT IMPLEMENTEC
               [<number>/<count>]
```

Notes:

1. The above example lists are displayed in reverse order on the SPO unless the MCP option "DIRN" is set to the value "1".

2. Example 1, the "<uc index>" is Usercode Index.

.PUT

Function: This command is a shorthand version of the "COPY" command and is used to specify explicit file "PUT" transfer requests. The command syntax provides for transferring files "from" the initiating system to another system (a PUT). The files can be either magnetic tape or disk (in any combination) and the only restrictions that apply are those identified in Section II, "Restrictions".

Syntax:

```
.PUT  <file-name-1> [(USER=<usercode-1>,  -------->

---> PASSWORD=<password-1>) ]  ------------------>

---> [AS     <file name-2> [(USER=<usercode-2>, --->
      ONTO

---> PASSWORD=<password-2>) ] ]  ---------------->

                                    DISK
---> [FROM   [<volume-1>]  [ (KIND=PACK    ) ] ]  --->
                                  TAPE
                                  TAPE7
                                  TAPE9
                                  PETAPE

---> [TO     [<volume-2>]  [ (KIND=PACK    ) ] ]  %
                                  TAPE
                                  TAPE7
                                  TAPE9
                                  PETAPE
```

Note: If a "usercode" is not desired as a multifile-id to the file name on the receiving system, then prefix "file-name-1" with an asterisk "*".

.PUT (continued)

Semantics: Refer to the "COPY" command semantics.

Defaults: Refer to the "COPY" command defaults.

Options: Refer to the "COPY" command options.

Example 1: (Send the program "TEST16" to the Server System for execution.)

```
   Input: ---   .PUT TEST16

   Reply: ---   FILE TEST16 START TRANSMISSION
          ---   FILE TEST16 END TRANSMISSION
```

Example 2: (Send a printer-backup file to the "hub" system.)

```
   Input: ---   .PUT (SPO)/#2121

   Reply: ---   FILE (SPO)/#2121 START TRANSMISSION
          ---   FILE (SPO)/#2121 END TRANSMISSION
```

Notes:

1. In the 7.0 implementation for file transfers from either tape or disk to tape in a receiving system, if a scratch tape is not mounted and/or ready, SYCOM will "hang" until the tape is made available; no notification is made of this condition and no input messages/commands are processed.

2. Referring to example 1; Upon completion of the "PUT" file transfer, the following message is displayed on the SPO of the "other" system:

```
% SYCOM =<job #> FILE TEST16 RECEIVED
```

This is also true under the reverse circumstances.

.USER

Function: This command is valid from any source. Once the statement is entered, the "usercode/password" is stored; without validation. This data is forwarded to the other system for "CONNECT" commands and is used for file transfer purposes; as the default usercode/password when none is specified.

Syntax:
```
.US[ER] <usercode>/<password>  %
```

Semantics: usercode = a predefined user identification.

password = a predefined user security code.

Default: None.

Example: (Log-on to SYCOM.)

```
  Input: ---   .US MY/USER

  Reply: ---   USER MY/USER LOGGED ON
```

Note: If a "usercode"/"password" is already logged-on, the current log-on attempt is still allowed and takes precedence.

.WHAT

Function: The .WHAT command solicits the return of a reply message indicating the current "status" of SYCOM's processing activity. The reply message is variable in length and content to reflect as many conditions as possible. If any file is in the process of "transfer", its name and the number of records thus far transferred is disolayed.

Syntax: .WHAT %

Semantics: None.

Default: None:

Example 1: (Assume at the time this command is entered, SYCOM is connected and waiting for something to do.)

```
  Input: ---   .WHAT

  Reply: ---   SYCOM ONLINE; IDLE
               THIS STATION OFFLINE, LOCAL HOST = USER1,
               REMOTE HOST = SERVER1
```

Example 2: (The command is entered durinç a file transfer.)

```
  Input: ---   .WHAT

  Reply: ---   SYCOM ONLINE; 00 FILES QUEUEC;
               SENDING: DMPALL(70/82); NOT RECEIVING
```

Note: The context of a WHAT reply is extremely fluid in that it is representative of what is happening at the time the command is entered. Because of the number of combinations of events that could be in progress at any given time, it is not feasible to attempt to example them all. However, the examples above will give the user some idea as to what can be expected.

```
.............
.           .
.      WHAT .
.           .
.............
```

*<SMCS Command String> (User System)

Function: This command syntax allows a terminal user on the User System to give commands to the system's SMCS for processing. Commands entered from a terminal that are prefixed with a sigral character of "*" are intercepted by the Network Controller and are immediately sent to the SMCS for processing. Consequently, the entire range of SMCS commands are available to a SYCOM user with this command syntax.

Syntax: `*<SMCS command> [<attributes>]  %`

Semantics:

SMCS command = any SMCS command which is valid from a User System terminal.

attributes = any data elements that are required by the command syntax.

Default: None.

Example:

Input: --- `*FILE (JEANNE)/GUIDE`

Reply: ---

```
'(JEANNE)/GUIDE' IS A PUBLIC FILE OF
188 RECORDS (90 BYTES, 20 PER BLOCK) IN
10 BLOCKS (10 PER AREA, 1 AREA) CREATED ON
04/24/78, LAST ACCESSED ON
06/21/78 (ACCESS SECURITY IN INPUT-OUTPUT)
AND CURRENTLY IN USE BY 0 USERS
```

Note: For a complete list and descriptions of the SMCS commands available, refer to the "SMCS Software Operational Guide".

*? <SMCS/MCP Command String> (User System)

Function: This command is used to give "safe" MCP commands to The User System's MCP for processing. This command string is message switched to the system's SMCS by the Network Controller which in turn "zips" the command string to the MCP for processing. The SMCS appends an appropriate usercode, password, and session number to the message "if" the user is currently "logged-on" to the SMCS; otherwise the command message string is zipped as entered.

Syntax: *? <XX> <mcp command attributes> %

Semantics:

```
XX                      = any MCP command which is
                          valid from a work station.

mcp command attributes  = any data elements required
                          by the command syntax.
```

Default: None.

Example:

```
  Input: ---   *? PD =/=

  Reply: ---   PD= "DCHCODE"
               PD= "DCHSOURCE"
               PD= "#DUMP"
               PD= "PROGRAMX"
               END PD
```

Notes:

1. The "mcp command string" is made secure, that is, in the above example, only the user's directory is returned.

2. Refer to "Appendix F, Restricted MCP Command Summary" for a list of those MCP commands which are permitted from a terminal.

```
.............
.           .
.   *? <XX> .
.           .
.............
```

<x><SMCS Commanc String> (Server System)

Function: This commanc syntax allows a terminal user on the User System to give commands to the Server System's SMCS for processing. Each command entered must be prefixed with the current signal character for that SMCS; it must be different than that of the User System's SMCS for obvious reasons. All messages with this signal character are intercepted by the Network Controller anc immediately sent to the SMCS for processing. Consequently, the entire range of SMCS commands are available to a SYCOM user. (This command is only valid when a SMCS is running in the Server System.)

Syntax: `<x><SMCS command> [<attributes>] %`

Semantics:

x = the Server System's current SMCS "signal character".

SMCS command = any SMCS command which is valid from a User System terminal.

attributes = any data elements that are required by the command syntax.

Default: None.

Example:

```
  Input: ---   $FILE (JEANNE)/GUIDE

  Reply: ---   '(JEANNE)/GUIDE' IS A PUBLIC FILE OF
               188 RECORDS (90 BYTES, 20 PER BLOCK) IN
               10 BLOCKS (10 PER AREA, 1 AREA) CREATED ON
               04/24/78, LAST ACCESSED ON
               06/21/78 (ACCESS SECURITY IN INPUT-OUTPUT)
               AND CURRENTLY IN USE BY 0 USERS
```

Note: In the above example, the "$" character is assumed to be the Server System's SMCS "signal character".

```
...............
.             .
. <x><command> .
.             .
...............
```

<x>? <SMCS/MCP Command String> (Server System)

Function: This command format allows certain MCP commands to be sent to a Server System for processing. They are permitted from either the User System's SPO or from one of its terminals. All replies to these commands are returned to the User System SYCOM which causes them to be displayed on a appropriate SPO/terminal. Certain MCP commands require a "usercode/password", prefixed to the desired MCP command, which is known to the Server System. To do these commands however, requires the SMCS operating in the Server System. See "Appendix F" for the list of safe MCP commands for a B1800/B1700 Server System. All commands that require a usercode/password are so indicated.

Syntax:

```
<x>?  [US <usercode>/<password>]  --------------->

---->  <XX> <mcp command attributes>  %
```

Semantics:

x = the Server System's SMCS signal character.

XX = any MCP command code or job # that would be valid from a terminal or SPO.

mcp command attributes = any data elements required by the command syntax.

Default: None.

Example: (Request a "PD" list for user currently logged-on.)

Input: --- $? PD =/=

Reply: ---

```
PD= "TESTER"
PD= "OPENS"
END PD
```

Note: In the above example, the "$" character is assumed to be the Server System's SMCS "signal character".

<x>? <XX>

System SPO/Program Commands

The following SYCOM commands are restricted to entry from either the system SPO or a special application program. These commands are generally of the type used by a system operator to control functions relevant only to operation of the system. They are generally functions a terminal user cannot logically direct without current knowledge as to the state cf the system and its peripherals. By enforcing this restriction, the system is assured that control is at the system level and not at a terminal level.

Program command input does not require the command string to be prefixed with SYCOM's "job number" and an "AX". Command input from a program should be in the same format as command input from terminals.

.CL

Function: The .CL command is used to "clear" a peripheral device in the event the device becomes inoperable when receiving a data file. It clears (discards) the buffers associated with the specified device and the receiving process is terminated.

Syntax: `.CL <udi> %`

Semantics: uci = the Unit Device Identifier is restricted to one of the following mnemonic codes:

LP = Line Printer
CP = Card Punch

Default: None:

Example 1: (Clear the line printer because its down.)

Input: --- `<sycom job #> AX.CL LP`

Reply: --- (no response)

Example 2: (Clear the card punch because there is no need to continue.)

Input: --- `<sycom job #> AX.CL CP`

Reply: --- (no response)

Notes: None.

.CP

Function: The purpose of this command is to inform the User System that the systems Card Punch has fallen into a "not ready" state. SYCOM ir turn informs the Server System (if such connection is established) of this hardware deficiency condition. The Server system would "suspend" transmitting any card punch data stream messages; if it was in this process.

It is a responsibility of the User System operator to inform SYCOM (via the appropriate command) when the card punch is again "ready" to receive output (see notes below).

Syntax: `.CP NR %`

Semantics: NR = Not Ready

Default: None.

Example: (Inform SYCOM that the Card Punch is not ready.)

Input: --- `<sycom job #> AX.CP NR`

Reply: --- (no response)

Notes:

1. To inform SYCOM that the "card punch" is now ready to receive output data, the ".RY CP" is provided.

2. If the card punch has beccme "inoperative", then the ".SV CP" command is more appropriate.

.DEBUG

Function: The "DEBUG" command provides the facilities for a printer trace of all data messages read or written by SYCOM. The "ON" option will initiate the DEBUG printer tracing, the "OFF" option will terminate tracing, and when no option is specified a request for the current status of the DEBUG facility is assumed.

Syntax:

```
.DEBUG <default option>  %
       ON
       OFF
```

Semantics: Self explicit.

Default: If no option is specified or the option is invalid, (not ON or OFF) a "status request" is assumed.

Example 1: (Turn printer debugging "ON" for all I/O messages.)

```
  Input: ---   <sycom job #> AX.DEBUG ON

  Reply: ---   % SYCOM =<job #> DEBUG TURNED ON
```

Example 2: (Obtain current DEBUG status.)

```
  Input: ---   <sycom job #> AX.DEBUG

  Reply: ---   % SYCOM =<job #> DEBUG ON
```

Example 3: (Turn debugging OFF.)

```
  Input: ---   <sycom job #> AX.DEBUG OFF

  Reply: ---   % SYCOM =<job #> DEBUG TURNED OFF
```

Notes: None.

.EOJ

Function: The ".EOJ" command prcvides the capability for an orderly shutdown of SYCOM. Any further commands are prevented (by rejection) from being processed. A message ("PLEASE SIGN OFF") is broadcast to all terminals currently "logged-cr" to SYCOM. A "close message" is sent to the other system's SYCOM after the completion of any in-process file transfers and when there are no longer any terminals signed-on to SYCOM. The SYCOM program then sends a "disconnect message" to the Network Contrcller and goes to "end-of-job".

Syntax: .EOJ %

Semantics: None to specify.

Default: None.

Example: (Entering the command to shutcown SYCOM.)

```
   Input: ---   <sycom job #> AX.EOJ

   Reply: ---   % SYCOM =<job #> PLEASE SIGN CFF
          ---   % SYCOM =<job #> EOJ INITIATEC
          ---   % SYCOM =<job #> SYCOM NOW CFFLINE
```

Note: All file transfers which are scheduled to occur (queued) are permitted to trarspire.

.LIST

Function: This command is valid from any terminal or the SPO. Its purpose is to display information about the system and its users. Command options provide lists of current system "parameters", users currently "signed-on", terminals that are currently "online", and file transfers currently "scheduled" to occur.

Syntax:

```
.LIST  PARAMETERS  %
       USERS
       TERMINALS
       SCHEDULE
```

Semantics:

PARAMETERS = requests a list of the parameters SYCOM is currently operating with.

USERS = requests a list of all users currently "signed-on" to SYCOM.

TERMINALS = requests a list of all terminals that are currently "online".

SCHEDULE = requests a list of all file transfers currently scheduled.

Default: Currently there is no default..

Example 1: (Display the list of SYCOM's parameters.)

Input: --- <sycom job #> AX.LIST PARAMETERS

Reply: ---

```
% SYCOM =<job #> TERMINAL RECORD SIZE = 180
% SYCOM =<job #> DEBUGGING              OFF
% SYCOM =<job #> PROGRAM INPUT          ALLOWED
% SYCOM =<job #> FILE TRANSFER          ALLOWED
% SYCOM =<job #> FILE SEPARATORS        SENT
% SYCOM =<job #> MAXIMUM BUFFER SIZE  = 400
% SYCOM =<job #> WAIT                   FALSE
% SYCOM =<job #> TIMEOUT INTERVAL     = 25
% SYCOM =<job #> SYSTEMID             = "USER1    "
```

```
............
.          .
.     LIST .
.          .
............
```

.LIST (continuec)

```
Example 2:     (List the current signed-on users of the system.)

  Input: ---   <sycom job #> AX.LIST USERS

  Reply: ---   % SYCOM =<job #> [<uc index>] DENNIS ONLINE
         ---   % SYCOM =<job #> [<uc index>] JEANNE OFFLINE
         ---   % SYCOM =<job #> [<uc index>] CARLYE ONLINE
         ---   % SYCOM =<job #> 3 LOGGED ON


Example 3:     (List the currently "online" terminals.)

  Input: ---   <sycom job #> AX.LIST TERMINALS

  Reply: ---   % SYCOM =<job #> [<rsn>] <SPO> <v.lsn> Aa aA
         ---   % SYCOM =<job #> [<rsn>] <lsn> <v.lsn> Ab bA
         ---   % SYCOM =<job #> 2 TERMINALS IN USE


Example 4:     (list all file transfers "scheduled" to occur.)

  Input: ---   <sycom job #> AX.LIST SCHEDULE

  Reply: ---   % SYCOM =<job #> SCHEDULE LIST NOT IMPLEMENTED
                                [<number>/<count>]
```

Notes:

1. The above example lists are displayed in reverse order on the SPO unless the MCP option "DIRN" is set to the value "1".

2. Example 2, the "<uc index>" is Usercode Index.

.LOG

Function: The purpose of this command is to display counts of specific data communication line errors and certain message transmission statistics. These counter are "not" zeroed out after they are displayed. The log effect is produced when these counters (identifiers included) are printed in the "DEBUG trace" listing (if the DEBUG option was "ON").

Syntax: .LOG %

Semantics: None.

Default: None.

Example: (Request a display of the status counters.)

```
  Input: ---   <sycom job #> AX.LOG

  Reply: ---   % SYCOM =<job #> TIMEOUT = <count>,
                                BREAK = <count>,
                                END-OF-BUFFER = <count>,
                                LOSS-OF-DSR = <count>,
                                LOSS-OF-CARRIER = <count>,
                                PARITY = <count>,
                                BUFFER OVERFLOW = <count>,
                                MEMORY PARITY = <count>,
                                NUMBER OF TRANSMISSIONS = <count>,
                                NUMBER OF BYTES SENT = <number>.
```

Note: Expect the above reply message to be displayed as a single message string.

.LP

Function: The purpose of this command is to inform the User System SYCOM that the line printer has become "not ready". SYCOM in turn informs the Server System of this temporary condition. The Server System then suspends transmitting of the line printer data stream messages until notified to resume by the User System's SYCOM.

It is the responsibility of the User Systems operator to inform SYCOM when the line printer is again ready to receive output data (see notes).

Syntax: .LP NR %

Semantics: NR = Not Ready

Default: None.

Example: (Inform SYCOM of a "not ready" Line Printer.)

Input: --- <sycom job #> AX.LP NR

Reply: --- (no response)

Notes:

1. To inform SYCOM when the line printer is ready to accept output data, the ".RY LP" command is provided.

2. If the line printer has become "inoperative", then the command ".SV LP" is more appropriate.

3. This command is only relevant to RJE functions. In particular, when the line printer is in use printing a RJE printer backup file.

.OFFLINE

Function: The purpose of the ".OFFLINE" command is to inform a Server System that the User System is going to "break" the existing connection and go into standby mode; waiting for the Server System to call back. This command can be used for either a "switched" or "non-switched" line connection. Since the Network Controller actually performs this action (upon SYCOM directive), it then continuously monitors the "data set ready" signal. When this signal comes "true", the Network Controller re-establishes the connection with the system ard normal operations then resume.

Syntax: .OFFLINE %

Semantics: None.

Default: None.

Example:

```
   Input: ---   <sycom job #> AX.OFFLINE

   Reply: ---   % SYCOM =<job #> OFFLINE INITIATED
          ---   % SYCOM =<job #> SYCOM NOW OFFLINE
```

Notes: None.

.ONLINE (establishment)

Function: The purpose of this command is to cause the Network Controller to attempt to re-establish the line connection to the Server System. Upon satisfactory completion of this task, a response message is displayed (see example below).

This command may be used in conjunction with the retry function. The error message "RETRIES-UP" is displayed when the current message buffer being sent to the Server System is not being received. By entering this command, the message buffer being sent is discarded and an attempt is made to re-establish the line connectior.

Syntax: .CNLINE %

Semantics: None.

Default: None.

Example:

```
  Input: ---   <sycom job #> AX.ONLINE

  Reply: ---   % SYCOM =<job #> ".ONLINE" INITIATED
         ---   % SYCOM =<job #> SYCOM NOW ONLINE to REMOTE HOST
```

Notes: None.

.PHONE

Function: The PHONE command is used to either "establish" a phone number to be used for the dial-out to the predefined Server System or to display an existing number. The automatic dial-out (with an Auto Call Unit) to a specific Server System requires a phone number, hence the need for the command.

Syntax:

```
.PHONE <phone number>  %
       <null option>
       =
```

Semantics:

```
phone number = the phone number of a target Server
               System.

null option  = no option specified.

=            = clears existing phone number.
```

Default: The omission of a valid phone number in the command string causes any existing phone number previously entered to be "displayed".

Example:1: (Establish the auto dial phone number of a specific Server System.)

```
  Input: ---  <sycom job #> AX.PHONE 1-805-964-6881

  Reply: ---  (no response)
```

Note: If an auto-dial unit is supplied, the specified phone number is used the next time dial-out is initiated to establish a connection to a Server System. (Dial-out is automatically initiated by the ".ONLINE" command.)

.QS

Function: This command allows a user to set a maximum number of message buffers that can be queued at any one time between SYCOM and the Network Controller. The value range for this parameter is 4 thru 15. (For a description on the use of this parameter see the writeup on "Queue Depth Control" in Section II.)

Syntax: .QS=<queue size> %

Semantics: queue size = a 1 or 2 digit integer representing the new Queue Depth limit. A suggested value for this parameter would be 10 message buffers.

Default: None.

Example: (Change the queue size to 10.)

Input: --- <sycom job #> AX.QS=10

Reply: --- % SYCOM =<job #> QUEUE SIZE CHANGED TO 10

Note: A specified value outside the defined value range produces an error response.

.READ

Function: The ".READ" command causes SYCOM to open an input card file and commence transmitting those card data images to the Server System; providing a connection has been established. When end-of-file is reached, the file is closed automatically.

This command is only processed when the User System is properly logged-on to a Server System which is defined as a "host". (It is possible to override this feature by using the ".RYCR" command instead.)

Syntax: .READ %

Semantics: None.

Default: Assumes the input card file is labeled "RJE/CARDS".

Example: (Start transmission of an RJE card data file.)

```
  Input: ---   <sycom job #> AX.READ

  Reply: ---   % SYCOM =<job #> CARD READER OPENED
         ---   % SYCOM =<job #> CARD READER CLOSED
```

Note: An input card file is defined to be any group of cards with the first card in the deck containing "<I> STREAM RJE/CARDS" and the last card containing "<I> TERMINATE RJE/CARDS". (The <I> symbol represents an invalid character in column 1 as required for MCP control.

.RY (ready)

Function: This command is used to inform SYCOM that the specified peripheral device is currently ready to send/receive data. SYCOM in turn takes the appropriate action to advise the Server System.

Syntax: .RY <udi> %

Semantics: udi = the Unit Device Identifier is restricted to one of the following mnemonic codes:

CR	=	Card Reader
CP	=	Card Punch
LP	=	Line Printer

Default: There are no defaults.

Example: (Ready the line printer.)

Input: --- <sycom job #> AX.RY LP

Reply: --- (no response)

Notes: None.

.SIZE

Function: This command allows the system operator to set the maximum message buffer size for transmitting and receiving purposes. SYCOM then informs the Server System of this change with a special system control type message.

Syntax: .SIZE=<buffer size> %

Semantics: buffer size = is a three digit integer with a value limited to a range of from "078" thru "400" bytes.

Default: None.

Example: (Reduce the send/receive message block buffer size to 320 characters.)

```
  Input: ---   <sycom job #> AX.SIZE=320
  Reply: ---   % SYCOM =<job #> ".SIZE" INITIATED
```

Note: This command applies only to the RJE functions of the SYCOM system.

.STOP

Function: This command provides the ability for an immediate shutdown of SYCOM. No further input messages are accepted as SYCOM goes to "end-of-job" immediately. Terminals are not logged-off, virtual terminals are not disconnected, nor is an inter-system connection gracefully disconnected.

Syntax: .STOP %

Semantics: None to specify.

Default: None.

Example: (Entering the immediate shutdown command.)

```
  Input: ---   <sycom job #> AX.STOP

  Reply: ---   % SYCOM =<job #> STOP INITIATED
         ---   % SYCOM =<job #> SYCOM NOW OFFLINE
```

```
.............
.           .
.       EOJ .
.           .
.............
```

.SV

Function: This command directs the User System to inform the Server System to cease transmitting the card punch or line printer data stream and to "save" the entire file at the Server System. The command permanently inhibits the trarsmission of any further card punch or line printer data streams until notified by the User System (see notes).

Syntax: .SV <udi> %

Semantics: udi = the Unit Device Identifier is limited to one of the following mnemonic codes:

CP = Card Purch
LP = Line Printer

Defaults: None.

Example: (Terminate the output to the Card Punch.)

Input: --- <sycom job #> AX.SV CP

Reply: --- (no response)

Notes:

1. To re-establish the card punch or line printer output data stream, the command ".RY CP" (for the card punch) or ".RY LP" (for the printer) must be used. This is used to inform the Server System to start sending any card punch or line printer data streams that are available for transmission.

2. All card punch and line printer data streams start at the beginning of the file.

	Local System Input			Remote System Process	
Command	No SMCS	SMCS	Source	No SMCS	SMCS
.ABORT	yes	yes	TSP	no	no
.BYE	yes	yes	TSP	no	no
.CL	yes	yes	SP	no	no
.CONNECT	yes	yes	TSP	no	no
.COPY	yes	yes	TSP	no	no
.CP	yes	yes	SP	no	no
.DEBUG	yes	yes	SP	no	no
.DISCONNECT	yes	yes	TSP	no	no
.EOJ	yes	yes	SP	no	no
.FETCH	yes	yes	TSP	yes	yes
.LIST	yes	yes	TSP	no	no
.LOG	yes	yes	SP	no	no
.LP	yes	yes	SP	no	no
.OFFLINE	yes	yes	SP	no	no
.ONLINE	yes	yes	SP	no	no
.PHONE	yes	yes	SP	no	no
.PUT	yes	yes	TSP	no	no
.QS	yes	yes	SP	no	no
.READ	yes	yes	SP	no	no
.RY	yes	yes	SP	no	no
.SIZE	yes	yes	SP	no	no
.STOP	yes	yes	SP	no	no
.SV	yes	yes	SP	no	no
.USER	yes	yes	TSP	no	no
.WHAT	yes	yes	TSP	no	no
*<User SMCS>	no	yes	TSP	no	no
*?<User MCP>	no	yes	TSP	no	no
<X><Server SMCS>	no	no	TSP	no	yes
<X>?<Server MCP>	no	no	TSP	no	yes

Note: The above commands are restricted to entry from the source indicated in the table above.

Legend: T = Terminal
S = SPO
P = Program

Table III-1: Command Input Summary

SECTION IV

===

OPERATING PROCEDURES

SECTION IV

OPERATING PROCEDURES

INTRODUCTION

This section provides system and terminal operating procedures used for the operation of a SYCOM system with and without local terminals; and/or in conjunction with the Supervisory Message Control System (SMCS). System operating procedures are slanted toward operation of SYCOM for the purpose of Remote Job Entry (RJE) type functions and the associated system control functions. These procedures present all aspects of SYCOM operations from initialization thru controlled shutdown.

Local terminal operational procedures are separated into two (2) categories; terminals under SYCOM control and terminals under the control of the SMCS. These procedures provide coverage of all terminal operations within the scope of SYCOM) from sign-on to sign-off. Actual input/output message text is used in the examples to provide a feel for the ease of operation and what can be expected by a user of the system.

SYSTEM OPERATING PROCEDURES

The system operational procedures presented here are intended to provide a User System operator with sufficient information for operation of an interactive system using SYCOM. Operation of the associated Network Controller is briefly discussed. In addition, suggestions on the use of certain basic SYCOM commands and their options are exemplified.

PRELIMINARY PROCEDURES

Before the SYCOM system is executed, the user needs to be aware of a few system variables that must be considered. First there are "program switches" which are used to invoke certain optional features. Second there is the "parameters" file which is used to maintain SYCOM's variable control functions and processing attributes. Each of these subjects in covered in turn starting with the program switches.

Program Switches

Program Switch 2 serves two purposes: 1) If "set" to the value of "2" at execute time, the initialization phase causes a display, on the SPO, of SYCOM's "compile time and date"; else no display. 2) If the switch is "set" after BOJ time, then the purpose is to cause any excess length I/O DEBUG trace lines to be truncated to exactly one print line to reduce printing; if the DEBUG option is "set" of course.

Program Switch 3 serves to expand the information message that is displayed at initialization time with a current "run date" (both time and date). Because this switch is used in conjunction with Program Switch 2, it must also be "set" prior to SYCOM execution time for switch 3 to be effective.

Preparation of Parameters Card File

Because SYCOM automatically uses an existing "parameters" file from disk, to enter changes to any existing file requires that the current "SYCOM/PARAMETERS" file be removed prior to program execution. This causes SYCOM to effect the process which creates a new parameters file based on; 1) a predetermined set of default parameters and 2) any parameters entered supersede the associated default parameter.

These parameters can be entered either from the card reader or the SPO. However, an important point to note, SYCOM will first attempt to accept parameters from a "card" file and failing to find this file, enters the mode whereby it accepts parameters from the system SPO. Therefore, if card parameter input is to be used, the card reader "must be readied" prior to execution.

A simple way to create a parameters file is to punch a card deck; providing the target system has a card reader in its hardware configuration. The external name of this parameters card file is "CARDS". Following is an example of a SYCOM system card file: (The <I> in the example below represents an "invalid" character as required for MCP type commands.)

```
<I> DATA CARDS
DEBUG
BUFFER = 400
RECORD = 240
SYSTEMID = USER1
PROGRAM
NOFS
NOFTP
WAIT
TIMEOUTS = 25
<I> END
```

The above example shows "all" of the available SYCOM parameters. However, in a real life situation, parameters are selected to fit the requirements of the individual system. Refer to the procedure "Entering the Parameters" of descriptions of these parameters.

If the system does not have a card reader, then the procedure "Entering The Parameters" must be utilized "after" SYCOM has been executed. When SYCOM is ready to accept parameters from the SPO it is sc indicated by the display of "ENTER SYCOM PARAMETERS". (Refer to "Execution of SYCOM" procedures.)

Setup MCP Program Name Table

In order for SYCOM's Network Controller to be automatically executed by the system, it is necessary to establish the identity of this Controller in the "MCP Program Name Table". To do this is simple, enter the following MCP command:

```
--> CM C USER/NC(etx)
       NETWORK CCNTROLLER CHANGED FROM "" TC "USER/NC"
```

The controller name "SYCOM/NC" is of course a user optional name and is therefore only representative. The actual Controller name selected by the user would be substituted as required.

To verify that the correct Network Controller I/D is in the MCP Name Table ("C" slot), the following MCP command can be entered:

```
-->  WWC(etx)
       C = "USER/NC"
```

Assigning Program Priorities

For optimum system response, it is necessary that the system's Network Controller and SYCOM run at a "high" priority. Therefore, to ensure that the MCP automatically executes these programs at their optimum priorities, the following MCP commands are used to permanently assign a priority of "15" to the Network Controller and a priority of "12" to SYCOM:

```
-->  MO USER/NC PR 15(etx)
       "USER/NC" MODIFIED


-->  MO SYCOM PR 12(etx)
       "SYCOM" MODIFIED
```

EXECUTION OF SYCOM * WITHOUT SMCS

To use the SYCOM system without the benefit of a SMCS, the system operator simply enters the execute SYCOM command as shown below. The MCP automatically executes the associated Network Controller program (providing one is in the "C" slot of the MCP Name Table) when the first remote file open is initiatec for the controller. Both SYCOM and the Network Controller shoulo be run at a priority which ensure adequate response times to the user of the system (see suggested priority values below).

It is the option of the user whether or not SYCOM is executed with a "privileged usercode". If accessing of usercode files is reouired, then it must be executed "with" a privileged usercode; otherwise execute SYCOM as shown below:

```
-->  EX SYCOM(etx)

       SYCOM =<job #> BOJ. PP=12, MP=12 TIME = 08:14:51.2
```

During the beginning of the initialization phase if SYCOM cannot locate the "SYCOM/PARAMETERS" file on disk, it attempts to obtain new parameters from the card reader ("CARDS" file). Should the card reader be "not ready" or if there is no card reader, then SYCOM will lapse into the mode of accepting new parameters from the SPO. When SYCOM is in this mood, the following messages are displayed on the SPO:

```
% SYCOM =<job #> ENTER SYCOM PARAMETERS
SYCOM =<job #> ACCEPT.
```

The system operator can then enter the appropriate parameters to alter the system's "default parameters" (refer to Appendix G, "Parameters File Attributes"). The entry procedure is presented with "Entering The Parameters" in this section. Although the "SYSTEMID" parameter is not required, in some instances, it is strongly suggested that a representative "system identifier" be entered with the parameters at this time.

Once this file is established on disk, it need not be recreated unless it is lost or requires changes. Changes to the file can made using either of the two methods identified. However, again, it is necessary to "remove" the existing parameters file from disk. Additionally, this requires SYCOM to be shutdown and re-executed if it is currently running; else all is in order.

When SYCOM opens its Remote Files, the MCP knows whether or not a Network Controller is running and takes the appropriate action. That is, if a Controller is running, then the remote file opens take place and the system becomes operational. If a Controller is "not" currently running, then if the Controller's name is in the "C" slot of the MCP Program Name Table, the MCP automatically executes the named Controller program and the following messages are displayed at the SPO:

```
EX SYCOM/NC PT;
SYCOM/NC =<job #> BOJ. PP=15, MP=15 TIME = 08:15:08.7
```

If there is no Controller name entered in the MCP Program Name Table, then the following message is displayed:

```
REMOTE FILE "MCSREMOTE" NEEDS A CONTROLLER
```

The system operator must then execute the appropriate Network Controller by entering the following command:

```
--> EX SYCOM/NC(etx)

        SYCOM/NC =<job #> BOJ. PP=15, MP=15 TIME = 08:16:01.2
```

The MCP displays the above "BOJ" message, permits the Remote File opens to occur and the system is operational.

Further on during the initialization phase, if Program Switch 2 is set to a value of "2", the following message is displayed on the system SPO:

```
        % SYCOM =<job #> COMPILE DATE = FRIDAY 24 JUL 78 AT 12:07
```

If Program Switch 3 is set to a value of "3", then the following message is also displayed in conjunction with the above message:

```
                                RUN DAY = SATURDAY 19 AUG 78 at 10:13
```

Upon completion of SYCOM's initialization phase and a successful connection established to a default Server System, the following messages will be displayed on the User SPO:

```
        % SYCOM =<job #> **** SYCOM (07/24/78 12:08) ****
        % SYCOM =<job #> SYCOM NOW ONLINE TO REMOTE HOST = SERVER1
```

EXECUTION OF SYCOM * WITH SMCS

When SYCOM runs in conjunction with the SMCS, it is automatically executed by SMCS thru its "job spawning" facility. This occurs when the "first" terminal user "SIGNS-ON" or "PASS's" to SYCOM from a local terminal. Therefore, the system operator must not execute SYCOM from the system SPO; else SMCS just zip executes another copy of SYCOM which causes a few operational problems.

If during the initialization phase, SYCOM cannot locate the file "SYCOM/PARAMETERS", it attempts to obtain new parameters from the card reader ("CARDS" file). Should this card file not be present or if there is no card reader, then SYCOM commences to accept new parameters input from the SPO. To signal the system operator of this requirement, the following messages are displayed:

```
% SYCOM =<job #> ENTER SYCOM PARAMETERS
SYCOM =<job #> ACCEPT.
```

The system operator must then enter the appropriate parameters to override the system's "default parameters" (refer to Appendix G, "Parameters File Attributes"). The parameter entry procedure is presented with "Entering The Parameters"; this section. Although the parameter "SYSTEMID" is not required, for some instances, it is strongly suggested that a representative "system identifier" be entered with the parameters at this time.

Also during the initialization phase, if Program Switch 2 is set to the value "2", than the following SPO message is displayed:

```
% SYCOM =<job #> COMPILE DATE = FRIDAY 24 JUL 78 AT 12:07
```

If Program Switch 3 is set to a value of "3", then the following message is also displayed in conjunction with the above message:

```
                        RUN DAY = SATURDAY 19 AUG 78 at 10:13
```

Upon completion of initialization and the successful connection to a default Server System, the following messages are displayed on the User SPO:

```
% SYCOM =<job #> **** SYCOM (07/24/78 12:08) ****
% SYCOM =<job #> SYCOM NOW ONLINE TC REMOTE HOST = SERVER1
```

COMMON SYSTEM OPERATIONS

Once SYCOM has been executed, from either the SPO or spawned by the SMCS, most operating procedures are common to both modes of operation. These common operating procedures are oriented to SPO input in the form of SYCOM parameters, SYCOM commands, and MCP directives. Following then are the common operating procedures, presented in the sequence they are most likely to occur; under individual headings.

SMCS Signal Characters

In order for a SYCOM systems network to furction effectively, it is a requirement that the SMCS "signal character" for each system in the network to be a different character. This can be easily accomplished by mutual agreement when the SYCOM system network is designed. Each system's SMCS should have a specific character which is made known to all system users. Obviously, if the SMCS is not utilized, this is not necessary.

If during actual system operations it is found that a SMCS signal character is undesirable for any reason, it can be changed while online. This can be accomplished by entering a "signal character change command" from either a local or remote source. For actual details, refer to the "SMCS Software Operational Guide".

Entering the Parameters

The parameter definitions listed below can be entered to SYCOM from the system SPO. To do so, they must be entered with one line using the following "Accept" process:

```
--> <sycom job #> AX <parameter keyword> [=<value>]
                     [<parameter keyword> [=<value>]....] (etx)
```

Each parameter is validated and the system operator is informed of any illegal parameters encountered during the parsing of the parameter data.

DEBUG parameter

This parameter indicates that the printer tracing mechanism is to be utilized to oroduce a record of SYCOM's I/O message activity.

Syntax: DEBUG

The default for this parameter is no "printer trace".

BUFFER parameter

This parameter defines the size, in bytes, of the used in SYCOM communications with the other system. The buffer size should be large enough to handle the largest message, and possibly achieve a blocking factor for multiples of a record.

Syntax: BUFFER = <integer>

The default buffer size is 400 and the maximum allowable size is 1950 bytes. (This parameter directly relates to file transfer buffer size and not RJE buffer size.)

RECORD parameter

This parameter defines the maximum record size. This value limits the record size of files, and the length of program-to-program messages transmitted by SYCOM to another system. Thus, if SYCOM is to be used to transfer code files only, the record size could be set to 180 bytes.

Syntax: RECORD = <integer>

The default record size is set to the value of 180. The maximum record size permitted is 1950 bytes.

SYSTEMID parameter

This parameter is used to specify the name of the User System. A system-id can be comprised of from one to seventeen characters.

```
     Syntax:   SYSTEMID = <system-id>          (e.g., USER1)
```

PROGRAM parameter

This parameter is specified when SYCOM is to perform program-to-SYCOM functions. The "record size" parameter limits the maximum length of these messages.

```
     Syntax:   PRCGRAM
```

This parameter defaults to "no" program-to-SYCOM input permitted.

TIMEOUTS parameter

This parameter is used to establish an "interval" for the display of "NUMBER OF TIMEOUTS = 25" error messages on the system SPO. A value of "0" inhibits the message display entirely. Other values cause the message to be displayed when the timeout count reaches the specified value.

```
     Syntax:   TIMEOUTS = <number>
```

WAIT parameter

This parameter is used to prevent SYCOM from going to "end-of-job when it receives a "disconnect message". SYCOM will disconnect and lapse into a "offline/idle" state.

```
     Syntax:   WAIT
```

NOFS parameter

SYCOM will not send a "FS1" or "FS2" (file separator) to the other system when sending card images (relevant to 7.0 RJE).

Syntax: NOFS

NOFTP parameter

This parameter inhibits File Transfers.

Syntax: NOFTP

ETX parameter

Entering an ETX in the parameter string terminates the parameter input process.

Syntax: <sycom job #> AX <parameter string> (etx)

SYCOM Commands from The SPO

The system operator is permitted to communicate with SYCOM using the MCP "AX" facility. By entering certair commands (examples below), the operator is also permitted to communicate with application programs in a Server System; thru the "virtual terminal" facility of SYCOM. A typical command session might go as follows; SPO input is flagged with "-->" and indented lines are output reply messages:

```
-->  <sycom job #> AX.WHAT(etx)

  % SYCOM =<job #> SYCOM ONLINE; IDLE
  % SYCOM =<job #> THIS STATION OFFLINE, LOCAL HOST = USER1
                   REMOTE HOST = SERVER1

-->  <sycom job #> AX.PUT DMPALL(etx)

  % SYCOM =<job #> FILE DMPALL START TRANSMISSION
  % SYCOM =<job #> FILE DMPALL END TRANSMISSION
```

```
    -->  <sycom job #> AX.CONNECT(etx)

      % SYCOM =<job #> ".CONNECT" INITIATED

    -->  <sycom job #> AXEX DMPALL;FI OUTP.FILE INV 2(etx)

      % SYCOM =<job #> DMPALL =<job #> BOJ. PP=9, MP=9 TIME =...
      % SYCOM =<job #> % DMPALL =<job #> ENTER SPECS
      % SYCOM =<job #> DMPALL =<job #> ACCEPT.

    -->  <sycom job #> AX<dmpall job #>AXCOPY DSKLST SMCS/TRACE
                         120 9(etx)

      % SYCOM =<job #> DMPALL =<job #> "OUTPUT" = "BACKUP.PRT/11
      % SYCOM =<job #> DMPALL =<job #> "BACKUP.PRT/11" RELEASED
      % SYCOM =<job #> % DMPALL =<job #> ENTER SPECS
      % SYCOM =<job #> DMPALL =<job #> ACCEPT.

    -->  <sycom job #> AX(etx)

      % SYCOM =<job #> DMPALL =<job #> EOJ. TIME = 12:35:33.6

    -->  <sycom job #> AX.DISCONNECT(etx)

      % SYCOM =<job #> STATION DISCONNECTED

    -->  <sycom job #> AX.FETCH BACKUP.PRT/11(etx)

      % SYCOM =<job #> FILE BACKUP.PRT/11 START TRANSMISSION
      % SYCOM =<job #> FILE BACKUP.PRT/11 RECEIVED
      % SYCOM =<job #> FILE BACKUP.PRT/11 END TRANSMISSION
```

Systematic SYCOM Shutdown

When SYCOM is running as a self-sustaining entity (without the SMCS), the ".EOJ" command provides the means by which an orderly shutdown can occur. Once this command has been entered, all subsequent commands entered are rejected. An appropriate "close message" is sent to the other SYCOM after the completion of all in-process file transfers. SYCOM then sends a "disconnect message" to the Network Controller and goes to "end-of-job". To initiate an orderly shutdown of SYCOM, the system operator must enter the following:

```
-->  <sycom job #> AX.EOJ(etx)
```

The following message is displayed immediately on the User SPO.

```
     % SYCOM =<job #> EOJ INITIATED
```

No further processing requests are accepted. Any file transfers queued to process are allowed to complete. At the point in time when there is no longer any SYCOM activity, SYCOM goes to "EOJ". The standard MCP "EOJ message" is displayed for this action:

```
     SYCOM =<job #> EOJ. TIME = 16:21:55.1
```

When SYCOM is running in a self-sustaining environment and it is necessary to shutdown the system "immediately", then the ".STOP" command provides this means for termination. Upon entry of this command, all in-process activity is aborted and all scheduled events are purged. A "close message" is then sent to the other SYCOM, a "disconnect message" sent to the Network Controller, and the program goes to "end-of-job". The system operator enters the following to initiate the immediate shutdown of SYCOM:

```
-->  <sycom job #> AX.STOP(etx)
```

The following messages are then displayed on the system SPO:

```
     % SYCOM =<job #> STOP INITIATED
     SYCOM =<job #> EOJ. TIME = 16:21:55.1
```

When SYCOM is running in conjunction with the SMCS, the "STOP" command entered to the SMCS in turn causes SYCOM to be terminated automatically. When this occurs, the following message will be displayed on the SPO:

```
     SYCOM =<job #> DEATH IN FAMILY. TIME 12:58:01.1
```

A note of importance: If there are any requests queued for processing, they are lost when SYCOM shuts down under these circumstances.

Reactionary Hints and Pointers

During the course of SYCOM operations, there are a few unusual system "conditions" which could occur. In order for the system operator to correctly respond to these conditions, there must be sufficient information available. That information is provided here in the form of problem descriptions, "hints", "pointers", and responses to these conditions.

A. If at any time SYCOM appears to be "hung-up", a "<job #> WY" command must be entered at the system SPO. If the following message is displayed:

```
SYCOM =<job #> WAITING MAGNETIC TAPE <tape label>
```

SYCOM has been automatically stopped waiting for the required magnetic tape to become available; at which time SYCOM operations will automatically resume.

This condition is resolved by the operator making a scratch magnetic tape available to the system. brought down with a MCP "QC" command. (A system includes the SMCS, Network Controller, CANDE, and all programs associated with the Network Controller.)

If the following message is displayed:

```
SYCOM =<job #> EXECUTING
```

and SYCOM has just been executed, then the problem lies in the fact that SYCOM is attempting to establish a line connection to the remote system. It will continue this cycle until the connection is made or the program is DS'd. Nc input commands will be entertained during this initial establishment cycle.

B. If the following message is displayed (by the MCP) on the system SPO during the initialization phase:

```
INSUFFICIENT DISK WHILE OPENING QUEUE-FILE "SYCOM"
```

It means that there is not sufficient space on disk to open the "SYCOM/LINK.Q" queue. Affirmative action by the system operator would be to make available a sufficient amount of disk space (approximately 2000 segments).

TERMINAL PROCEDURES

In this sub-section, basic terminal operating procedures are provided in the form of examples. For a SYCOM oriented system however, terminals may be owned by either SYCOM or the SMCS. Because of these possibilities and the associated implications, specific operating procedures are included for both "SYCOM controlled" and "SMCS controlled" terminals.

TERMINAL FUNCTIONS

Terminal functions are those software features found in a TD820, and/or TD830 type Video Display Terminal (VDT) devices. These terminal features would included such things as; tab stops, lower/upper case, scrolling, etc.. These functions are fully described in the Supervisory Message Control System (SMCS) Software Operational Guide (see documentation reference).

TERMINAL OPERATIONS (SYCOM CONTROLLED)

When a local terminal is under the direct control of SYCOM, it is also permanently "attached" to it. Therefore, a terminal user has immediate access to SYCOM functions. A terminal can however, be temporarily "attached" to either SYCOM or the SMCS in a Server System. Thru Server System's SMCS, the terminal can also be "attached" to an appropriate interactive application program in the Server System. The terminal user is not limited to just a few SMCS commands, a user may invoke any command which the SMCS has in its repertoire; with the exception of those restricted to input from the SPO.

Example Interactive Terminal Session

Terminals at this point are attached to the User System's SYCOM. In order to utilize the SYCOM capabilities, a terminal user need only enter the desired SYCOM commands. Among these SYCOM commands are ".CONNECT" and ".DISCONNECT". These are used to gain access to any appropriate interactive application program in the Server System. The examples which follow explore all possible terminal connections as detailed in "Message Path Assignments".

In the examples below, terminal input is identified by the flag "-->". All other lines are reply messages from the program to which the terminal is currently inputting to. First some basic local SYCOM commands are shown. All SYCOM commands are entered directly as examoled below:

```
-->  .WHAT                                              (User SYCOM)

       SYCOM ONLINE; IDLE
       THIS STATION OFFLINE, LOCAL HOST = USER1
       REMCTE HOST = SERVER1

-->  .PUT DMPALL                                        (User SYCOM)

       FILE CMPALL START TRANSMISSION

-->  .WHAT                                              (User SYCOM)

       SYCOM ONLINE; 00 FILES QUEUED; SENDING DMPALL(47/85)
       THIS STATION OFFLINE; LOCAL HOST = USER1,
       REMCTE HOST = SERVER1
       FILE DMPALL END TRANSMISSION

-->  .CONNECT                                           (User SYCOM)

       ".CONNECT" INITIATED

-->  .WHAT                                              (User SYCOM)

       SYCOM ONLINE; IDLE
       THIS STATION ONLINE, LOCAL HOST = USER1,
       REMOTE HOST = SERVER1

-->  JOBS                                               (Server SMCS)

       ACTIVE:
       SERVER/NC =<job #> PP=15, MP=15
       SMCS =<job #> PP=13, MP=14
       SERVER/SYCOM =<job #> PP=12, MP=12
       SCHEDULED:
       BOTH SCHEDULES EMPTY.
       MIX LIMIT=16
```

It is necessary that the terminal user be aware of what "signal character" the Server System's SMCS is currently using. To find out, the following command is entered to query that character:

```
-->  SIGNAL                                             (Server SMCS)

       SIGNAL CHARACTER='$'
```

Now the terminal user can "sign on" to an application program in the Server System; and alternately enter commands to the Server System's SMCS. A series of following interactions exemplify this concept: (The application program "SET/TABS" used in the example is assumed to "set" specified tab stops in the user's terminal.)

```
     -->  SIGN ON SET/TABS                                (Server SMCS)

            SIGNED ON TO 'SET/TABS', SIGNAL='$'

     -->  $JOBS                                           (Server SMCS)

            ACTIVE:
            SERVER/NC =<job #> PP=15, MP=15
            SERVER/SYCOM =<job #> PP=12, MP=12
            SET/TABS =<job #> PP=4, MP=4
            SMCS =<job #> PP=13, MP=13
            SCHEDULED:
            BOTH SCHEDULES EMPTY.
            MIX LIMIT=16

     -->  10,20,30,40,50,60,70                              (Server AP)

            (tab stops are now set)
            REMOTE FILE CLOSED BY 'SET/TABS', STATION SIGNED OFF

     -->  EX DMPALL;FI OUTP.FILE INV 2                    (Server SMCS)

            DMPALL =<job #> BOJ. PP=9, MP=9 TIME = 11:35:21.8
            % DMPALL =<job #> ENTER SPECS
            DMPALL =<job #> ACCEPT.

     -->  <dmpall job #> AX COPY DSKLST SMCS/TRACE 120 9

            % DMPALL =<job #> ENTER SPECS
            DMPALL =<job #> ACCEPT.

     -->  <dmpall job #> AX

            DMPALL =<job #> EOJ. TIME = 11:36:58.1
```

At the point in time when a user is finished entering commands to SYCOM and wishes to terminate the session, the following command is entered; however, the terminal is still "attached" to SYCOM.

```
     -->  .DISCONNECT                                      (User SYCOM)

            STATION DISCONNECTED
```

TERMINAL OPERATIONS (SMCS CONTROLLED)

When a terminal is permanently attached to, and owned by, the SMCS, the terminal user must gain access to SYCOM functions thru the use of explicit SMCS commands. These commands are exampled below. By gaining access to SYCOM, the SMCS will temporarily "attach" the requesting terminal to SYCOM for the duration of a session. However, a terminal user is not limited to just a few SMCS commands, a user may invoke any commanc which the SMCS has in its repertoire; with the exception of thcse restricted to input from the SPO.

Example Interactive Terminal Session

The terminal at this point is attached to the User System's SMCS. In order to gain access to those nifty SYCOM capabilities, the terminal user must make use of the SMCS "SIGN ON" and "SIGN OFF" commands. The sample session which follows, shows the terminal user just how easy it is to "get on" and "get off" SYCOM: (input from the terminal is identified by the flag "-->".)

```
-->  SIGN CN SYCOM                                      (User SMCS)

        SIGNED ON TO 'SYCOM', SIGNAL='*'
```

When the "signed-on" message is returned to the terminal, that terminal is then attached to SYCOM. Therefore, SYCOM commands are entered directly as exampled below:

```
-->  .WHAT                                              (User SYCOM)

        SYCOM ONLINE; IDLE
        THIS STATION OFFLINE, LOCAL HOST = USER1
        REMCTE HOST = SERVER1

-->  .PUT DMPALL                                        (User SYCOM)

        FILE DMPALL START TRANSMISSION

-->  .WHAT                                              (User SYCOM)

        SYCCM ONLINE; 00 FILES QUEUED; SENDING DMPALL(47/85)
        THIS STATION OFFLINE; LOCAL HOST = USER1,
        REMOTE HOST = SERVER1
        FILE DMPALL END TRANSMISSION
```

```
    -->  .CONNECT                                               (User SYCOM)

           ".CONNECT" INITIATED

    -->  .WHAT                                                  (User SYCOM)

           SYCOM ONLINE; IDLE
           THIS STATION ONLINE, LOCAL HOST = USER1,
           REMOTE HOST = SERVER1

    -->  *REPORT >                                               (User SMCS)

           >D1          (SPO) : SIGNED ON TO 'SYCOM'
           1 STATION REPORTED AT 11:42.6  08/22/78

    -->  JOBS                                                  (Server SMCS)

           ACTIVE:
           SERVER/NC =<job #> PP=15, MP=15
           SMCS =<job #> PP=13, MP=14
           SERVER/SYCOM =<job #> PP=12, MP=12
           SCHEDULED:
           BOTH SCHEDULES EMPTY.
           MIX LIMIT=16
```

It is necessary that the SMCS in a Server System have a "signal character" that is different than that of the SMCS in the User System. To ensure this, the following command is entered to query that signal character: (If it "is not" different, then change it; "SIGNAL $".)

```
    -->  SIGNAL                                                (Server SMCS)

           SIGNAL CHARACTER='$'
```

Now the terminal user can "sign on" to an application program in the Server System; and alternately enter commands to the Server System's SMCS. A series of following interactions exemplify this concept: (The application program "SET/TABS" used in the example is assumed to "set" specified tab stops in the user's terminal.)

```
    -->  SIGN ON SET/TABS                                      (Server SMCS)

           SIGNED ON TO 'SET/TABS', SIGNAL='$'
    -->  $JOBS                                              (Server SMCS)

           ACTIVE:
           SERVER/NC =<job #> PP=15, MP=15
           SERVER/SYCOM =<job #> PP=12, MP=12
           SET/TABS =<job #> PP=4, MP=4
           SMCS =<job #> PP=13, MP=13
           SCHEDULED:
           BOTH SCHEDULES EMPTY.
           MIX LIMIT=16

    -->  *REPORT >                                            (User SMCS)

           D1          (SPO) : SIGNED ON TO 'SYCOM'
           1 STATION REPORTED AT 11:44.2  08/22/78

    -->  10,20,30,40,50,60,70                                 (Server AP)

           (tab stops are now set)
           REMCTE FILE CLOSED BY 'SET/TABS', STATION SIGNED OFF

    -->  .DEBUG                                              (User SYCOM)

           DEBUG OFF

    -->  .DISCONNECT                                          (User SYCOM

           STATION DISCONNECTED
```

At the point in time when a User is finishec entering commands to SYCOM and wishes to terminate the session, the following command is entered: (This is a SMCS command.)

```
    -->  *SIGN OFF                                            (User SMCS)
```

The response from the SMCS will be as follows:

```
           DETACHED FROM 'SYCOM'
```

At this point, the terminal has been "detached" from SYCOM and is now "re-attached" to the SMCS.

SECTION V

==

GENERATING THE NETWORK CONTROLLERS

SECTION V

GENERATING THE NETWORK CONTROLLERS

GENERAL

This section is provided as a user oriented guideline to the generation of a Network Controller for the SYCOM system. The controller is generated by the Network Definition Language (NDL) compiler using a prepared set of NDL specifications. The system user is required to determine these NDL specifications according to the system configuration and environment under which the Network Controller is to function.

The following guidelines are offered for a target system that is to provide the full range of SYCOM services; including terminals. They are intended to aid the user in generating a Network Controller with correct functional attributes. For a complete description of the NDL specifications and definitions, refer to the "Network Definition Language reference manual" (associated documentation).

DETERMINING THE SPECIFICATIONS

The following sample Network Controller NCL specifications are presented in a sequence similar to that of the NDL reference manual. This allows the user a logical reference for evaluating the sample NDL specifications for application to the installation requirements. They identify suggested parameters which produce an operational Network Controller program for the SYCOM system configuration as specified.

OPERATIONAL PROCEDURES

The following orocedures provide the methods by which the user prepares the input to the NDL compilation process, generates a Network Controller, and induces modifications into the resident NDL compiler, if required.

Preparation of Input

The NDL "source" input file can be prepared in the form of either a card file or a NDL type disk file. The cisk file could be created and maintained using CANDE or another text editor of similar capabilities.

The "Sample NDL Parameters File" is furnished for the purpose of creating this file with the most amount of accuracy in the least amount of time. Any adjustments to this scurce file for number of lines, terminals, and other such attributes is the responsibility of the user which is attuned to the system configuration and its operational environment.

Compilation Process

Prior to initiating the NDL compilation prccess, the user should ensure the presence of a "7.0 NDL/LIBRARY" file on disk, of a 07/24/78 or later vintage). Then, there are several methods by which the NDL compilation (generative) process can be invoked. Following are two of these methods:

1. Card Input:

   Following is a primitive example cf a NDL source card input deck: (The <I> in the example below represents an "invalid" character as required by the MCP.)

   ```
   <I> CO USER/NC NDL LI
   <I> DATA CARDS

   NDL source images
   NDL library calls

   <I> END
   ```

2. Disk Input:

This method is used to initiate a NDL compiler process when the source input file resides on disk. Enter the following MCP command using the system SPO:

```
--> CO USER/NC NDL LI;FI CARDS NAM USERNC/SOURCE DSK DEF
```

From this point on, the compilation procedures are the same. When the MCP executes the NDL compiler, a normal "BOJ" message is displayed. Upon successful completion of the compile, the normal "EOJ" message is displayed. However, if an error message such as the following is displayed:

```
NDL : USER/NC =<job #> --EVALUATION/PROGRAM POINTER STACK
                          OVERFLOW DS OR DP
```

Then the procedure "NDL Compiler Modifications" must be utilized to fix the problem.

NDL Compiler Modifications

Should the NDL compilation fail because of a lack of sufficient internal work space, the following procedure can be invoked to expand the NDL compiler itself to overcome the problem:

```
-->  EX SDL/SSC(etx)
        SDL/SSC =<job #> BOJ. PR=4 TIME = 07:28:48.1
        % SDL/SSC =<job #> SDL/SSC, MARK VII.0 OF 10/21/77 21:19
        % SDL/SSC =<job #>
        % SDL/SSC =<job #> ENTER FILENAME OR BLANK TO TERMINATE
        SDL/SSC =<job #> ACCEPT.

-->  <sdl/ssc job #>AXNDL(etx)

        % SDL/SSC =<job #> ***** FILE NAME : NDL
        % SDL/SSC =<job #> ENTER SCRATCHPAD NUMBER
        SDL/SSC =<job #> ACCEPT.

-->  <sdl/ssc job #>AX0(etx)
```

```
        % SDL/SSC =<job #>
        % SDL/SSC =<job #> ***** SCRATCHPAD: 0
        % SDL/SSC =<job #> VALUE : 27000
        % SDL/SSC =<job #> NAME : 200
        % SDL/SSC =<job #> DISPLAY : 192
        % SDL/SSC =<job #> CONTROL : 40
        % SDL/SSC =<job #> EVALUATION : 40
        % SDL/SSC =<job #> PROGRAM POINTER : 50
        % SDL/SSC =<job #> DYNAMIC : 20000
        % SDL/SSC =<job #> ***** TOTAL CORE: 63112
        % SDL/SSC =<job #> ENTER STACK ID
        SDL/SSC =<job #> ACCEPT.
```

The value "300" used in the example below is a known acceptable integer; a smaller value may also suffice but would be dependent upon the size and complexity of the Network Controller being generated.

```
-->  <sdl/ssc job #>AXE 300(etx)

        % SDL/SSC =<job #> ENTER ' ' TO END CHANGES
        SDL/SSC =<job #> ACCEPT.

-->  <sdl/ssc job #>AX (etx)

        (a updated list of program parameters is displayed)

        SDL/SSC =<job #> EOJ. TIME = 07:29:50.4
```

If any other error messages are displayed along this vein, they should contain sufficient information to allow the modification of the associated program parameter; in the manner shown above.

Print The USER/NCNIF File

Once the compilation/generation of the Network Controller is complete, it is suggested that the "Network Information File" be printed. This is accomplished by executing the following program; and providing the correct file name (if different than the one suggested):

```
-->  EX NIF/ANALYZER; FI NIF NAM USER/NCNIF(etx)
        NIF/ANALYZER =<job #> BOJ. PR=4 TIME = 11:05:27.1
        NIF/ANALYZER =<job #> EOJ. TIME = 11:07:12.9
```

SAMPLE NDL SOURCE FILE (LOCAL USER SYSTEM)

Following is a sample NDL source file to be used as input to the NDL compiler for the generation of a Network Controller oriented to a User System. This example provides a Network Controller which services a local site of 3 terminals, 2 virtual terminals, and a inter-system connection. The user must tailor this sample Network Controller specifications to the needs and desires of the installation (e.g., more/less terminals, more/less virtual terminal positions, etc).

This sample NDL input file is comprised of both NDL statements and parameter specificaticns. The resulting Network Controller will provide the correct functional attributes necessary for the operation of the SYCOM system; in testing mode.

```
$ CONTROL
$ LIBINFO
$ NSSIZE 20                        % number of entries; for IOLOG
$ VSSIZE 150000                    % number of bits;    for IOLOG

%==============================================================%
%                                                              %
%                     Declaration Section                      %
%                                                              %
%==============================================================%

DECLARATION:
  NIF = "USER"/"NCNIF".            % NIF file name; user option
  MAX FILES = 16.
  MAX BUFFERS = 2.
  MAX MESSAGES = 20.

$ LIBRARY SYCOMDECL

%==============================================================%
%                                                              %
%                     Requests and Controls                    %
%                                                              %
%==============================================================%

$ LIBRARY SYCOMREQ
$ LIBRARY CANDEPOLTD               %**
$ LIBRARY CANDEFSLTD               %**
$ LIBRARY SYCOMCTL
$ LIBRARY CANDETDCTL               %**
```

```
%================================================================%
%                                                                %
%                    Network Configuration                       %
%                                                                %
%================================================================%

%================================================================%
%                                                                %
%                    Terminal Section                            %
%                                                                %
%================================================================%

TERMINAL DEFAULT SYCOMDEFAULT:
  REQUEST = SYCOMREQ:RECEIVE, SYCOMREQ:TRANSMIT.
  TRANSMISSION = 1.
  ADDRESS = 2.               % Number of polling address characters
  BUFFERSIZE = 3840.         % 2 x 1920 bytes
  TYPE = 44.                 % Required for virtual terminal msg's

TERMINAL SYCOMTERMINAL:
  DEFAULT = SYCOMDEFAULT.

TERMINAL DEFAULT TDFLT:          %**
  REQUEST = CANDEPOLTD:RECEIVE, %**
            CANDEFSLTD:TRANSMIT.%**
  TRANSMISSION = 0.             %**
  ADDRESS = 2.                  %**
  TYPE = 44.                    %**
  BUFFERSIZE = 2250.            %**

TERMINAL TD824:                 %**
  DEFAULT = TDFLT.              %**

%================================================================%
%                                                                %
%                      Station Section                           %
%                                                                %
%================================================================%

STATION DEFAULT SYCOMDEF:
  FREQUENCY = 250, 250.         %    MCS Controller capability??
  MYUSE = INPUT, OUTPUT.
  TERMINAL = SYCOMTERMINAL.
  RETRY = 25.

STATION DEFAULT TD8DF:          %**
  FREQUENCY = 249, 249.         %**  Non-Remote SPO
  MYUSE = INPUT, OUTPUT.        %**
  TERMINAL = TD824.             %**
  RETRY = 4.                    %**

%----------------------------------------------------------------%
```

```
                                  %    ---------
STATION CONTROLST1:               %    lsn 001 |
  DEFAULT = SYCOMDEF.             %            |
  ADDRESS = "00".                 %            |
                                  %            |
STATION SPOST2:                   %    lsn 002 |
  DEFAULT = SYCCMDEF.             %            |
  ADDRESS = "01".                 %            |   Must be grouped.
                                  %            |
STATION CARDST3:                  %    lsn 003  >  ----------------
  DEFAULT = SYCCMDEF.             %            |
  ADDRESS = "02".                 %            |   Don't have to be
                                  %            |   first stations
STATION PRINTERST4:               %    lsn 004 |   defined.
  DEFAULT = SYCOMDEF.             %            |
  ADDRESS = "03".                 %            |
                                  %            |
STATION FILEST5:                  %    lsn 005 |
  DEFAULT = SYCOMDEF.             %            |
  ADDRESS = "04".                 %            |
                                  %    ---------

%-----------------------------------------------------------------%

                                  %
STATION USERT1:                   %    lsn 006   * Virtual terminal
  DEFAULT = SYCOMDEF.             %                (SPO/terminals)
  ADDRESS = "Aa", "aA".           %
                                  %
STATION USERT2:                   %    lsn 007   * Virtual Terminal
  DEFAULT = SYCOMDEF.             %                (Program)
  ADDRESS = "Pa", "aP".           %
                                  %

%-----------------------------------------------------------------%

                                  %**  ------------
STATION D1:                       %**  lsn 008    |
  DEFAULT = TD8DF.                %**             |
  ADDRESS = "D1".                 %**             |
  FREQUENCY = 249, 250.           %**  Remote SPO |
                                  %**             |
STATION D2:                       %**  lsn 009    |
  DEFAULT = TD8DF.                %**              > Local terminal
  ADDRESS = "D2".                 %**             |
  FREQUENCY = 249, 250.           %**  Remote SPO |
                                  %**             |
STATION D3:                       %**  lsn 010    |
  DEFAULT = TD8DF.                %**             |
  ADDRESS = "D3".                 %**             |
  FREQUENCY = 249, 250.           %**  Remote SPO |
                                  %**  ------------
```

```
%==============================================================%
%                                                              %
%                         Line Section                         %
%                                                              %
%==============================================================%

LINE DIRECT01:
  CONTROL = SYCOMCTL.
  STATION = CONTROLST1, SPOST2, CARDST3, PRINTERST4, FILEST5,
            USERT1, USERT2.
  ADDRESS = 1:0:6.                %  (line, adapter, port; system)

LINE DEFAULT LINEDFLT:            %**
  CONTROL = CANDETDCTL.           %**
  AUTOPOLL = 15.                  %** 5 x max stations/lines

LINE MULTIC4:                     %** adapter 4
  DEFAULT = LINEDFLT.             %**
  STATION = D1, D2, D3.           %**
  ADDRESS = 1:0:4.                %** (line, adapter, port; system)

%--------------------------------------------------------------%

FILE LINK:                        %  for SYCOM
  FAMILY = CONTROLST1, SPOST2, CARDST3, PRINTERST4, FILEST5,
           USERT1.                %  rsn's 001, 002, 003, 004,
                                  %        005, 006

FILE RFUSER:                      %  For program-to-program I/O.
  FAMILY = USERT2.                %  rsn 001

FILE TERMINALS:                  %**  This file is named "TERMINALS"
                                 %**  when SYCOM is run without the
                                 %**  SMCS.  It must be named
                                 %**  "MCSREMOTE" when SYCOM is run
                                 %**  subordinate to the SMCS..
  FAMILY = D1, D2, D3.           %**  rsn's 001, 002, 003

%--------------------------------------------------------------%

FINI.
```

Source configuration notes:
===========================

If the User System does not have "local terminals" then all lines in the above example with comments indicated with "%**" can be omitted from the source file.

SAMPLE NDL SOURCE FILE (REMOTE SERVER SYSTEM)

Following is a sample NDL source file to be used as input to the NDL compiler for the generation of a Network Controller oriented to a Server System. This example provides a Network Controller which services one remote site of no terminals, an inter-system connection, and a SMCS. The user must tailor these sample test Network Controller specifications to the needs and desires of the installation (e.g., local terminals, more/less virtual terminal positions, etc).

This sample NDL input file is comprised of both NDL statements and parameter specifications. The resulting Network Controller will provide the correct functional attributes necessary for the operation of the SYCOM system; in a test mode.

```
$ CONTROL
$ LIBINFO
$ NSSIZE 20                    % number of entries; for IOLOG
$ VSSIZE 150000                % number of bits;    for IOLOG

%=============================================================%
%                                                             %
%                   Declaration Section                       %
%                                                             %
%=============================================================%

DECLARATION:
  NIF = "SERVER"/"NCNIF".       % NIF file name; user option
  MAX FILES = 16.
  MAX BUFFERS = 2.
  MAX MESSAGES = 20.

$ LIBRARY SYCOMDECL

%=============================================================%
%                                                             %
%                    Requests and Controls                    %
%                                                             %
%=============================================================%

$ LIBRARY SYCOMREQ
$ LIBRARY SYCOMCTL
```

```
%=================================================================%
%                                                                 %
%                       Terminal Section                          %
%                                                                 %
%=================================================================%

TERMINAL DEFAULT SYCOMDEFAULT:
  REQUEST = SYCOMREQ:RECEIVE, SYCOMREQ:TRANSMIT.
  TRANSMISSION = 1.
  ADDRESS = 2.               % Number of polling address characters
  BUFFERSIZE = 3840.         % 2 x 1920 bytes
  TYPE = 44.                 % Required for virtual terminal msg's

TERMINAL SYCOMTERMINAL:
  DEFAULT = SYCOMDEFAULT.

%=================================================================%
%                                                                 %
%                       Station Section                           %
%                                                                 %
%=================================================================%

STATION DEFAULT SYCOMDEF:
  FREQUENCY = 250, 250.          %  MCS Controller capability??
  MYUSE = INPUT, OUTPUT.
  TERMINAL = SYCOMTERMINAL.
  RETRY = 25.

%-----------------------------------------------------------------%

                                 %  ---------
STATION CONTROLST1:              %  lsn 001 !
  DEFAULT = SYCOMDEF.            %          !
  ADDRESS = "00".                %          !
                                 %          !
STATION SPOST2:                  %  lsn 002 !
  DEFAULT = SYCOMDEF.            %          !
  ADDRESS = "01".                %          !  Must be grouped.
                                 %          !
STATION CARDST3:                 %  lsn 003  >  -----------------
  DEFAULT = SYCOMDEF.            %          !
  ADDRESS = "02".                %          !  Don't have to be
                                 %          !  first stations
STATION PRINTERST4:              %  lsn 004 !  defined.
  DEFAULT = SYCOMDEF.            %          !
  ADDRESS = "03".                %          !
                                 %          !
STATION FILEST5:                 %  lsn 005 !
  DEFAULT = SYCOMDEF.            %          !
  ADDRESS = "04".                %          !
                                 %  ---------

%-----------------------------------------------------------------%
```

```
STATION SERVERT1:                 %  lsn 006    * Virtual terminal
  DEFAULT = SYCCMDEF.             %                (SPO/terminals)
  ADDRESS = "aA", "Aa".           %
                                  %
STATION SERVERT2:                 %  lsn 007    * Virtual terminal
  DEFAULT = SYCCMDEF.             %                (Program)
  ADDRESS = "aP", "Pa".           %
                                  %

%================================================================%
%                                                                %
%                        Line Section                            %
%                                                                %
%================================================================%

LINE DIRECT01:
  CONTROL = SYCOMCTL.
  STATION = CONTROLST1, SPOST2, CARDST3, PRINTERST4, FILEST5,
            SERVERT1, SERVERT2.
  ADDRESS = 1:0:6.          % (line, adapter, port; system option)

%----------------------------------------------------------------%

FILE LINK:                        %  for SYCOM
  FAMILY = CONTROLST1, SPOST2, CARDST3, PRINTERST4, FILEST5.
                                  %  rsn's 001, 002, 003, 004, 005

FILE RFSERVER:                    %  For program-to-program I/O.
  FAMILY = SERVERT2.              %  rsn 001
                                  %  Note: There is only 1 device
                                  %  defined here because the
                                  %  sample test program is a
                                  %  single user program.

FILE MCSREMOTE:                   %  This file is used for the
  FAMILY = SERVERT1.              %  virtual terminals assigned
                                  %  to Server System's SMCS.
                                  %  rsn 001

FINI.
```

Source configuration notes:
===========================

The Network Controller source file for a Server System "with local terminals" is configured by appropriately adding the NDL statements identified in the User System example with "%**" as comments.

SECTION VI

==

APPENDICES

APPENDIX A: ERROR MESSAGES

DESCRIPTION

During command processing various errors may occur; the response error messages and their meanings are shown below. There are six classes of these error messages:

- Command Error Messages
- Command Informational Messages
- System Parameter Error Messages
- System Error Messages
- System Condition Messages
- System Fatal Messages

The following error message groups describe all messages that may be displayed in each of the above identified message classes:

Command Error Messages

The following error messages are displayed as the result of the improper use of SYCOM commands. Generally, an error can occur for one of two reasons; they are:

1. Invalid command syntax

2. Invalid parameter data

Although the reason for each error message is recognizable, there has been no attempt to group the messages into one of the two categories. The following command error messages are however, grouped alphabetically, by command, to make locating easier for the user.

COMMAND MESSAGE REASON

UNRECOGNIZED COMMAND <x...x>

This simply means that the last SYCOM command entered was not a valid command.

.ABORT:

THERE IS NO 'PUT' IN PROGRESS TO ABORT

An ABORT command was entered with a specified option of of "PUT" and no file is being transmitted at this time.

THERE IS NO 'FETCH' IN PROGRESS TO ABORT

An ABORT command was entered with a specified option of of "FETCH" and no file is being received at this time.

UNRECOGNIZED OBJECT OF 'ABORT' <x...x>

The command was entered with an option keyword which was neither "PUT" or "FETCH". Try again...

Command Error Messages (continued)

COMMAND MESSAGE REASON

.CL:

UNRECOGNIZED COMMAND <CL xx>

The .CL command was entered either without an option or the option was found to be invalid.

.CONNECT:

"CONNECT FAILED": VIRTUAL TERMINAL TABLE FULL

This message displayed indicates that all allocated "virtual terminal" positions are currently in use. The terminal user must try again at a later time.

STATION ALREADY CONNECTED

This only means that a ".CONNECT" command was entered from a terminal (or SPO) that was already "connected" to a Server System; no problem.

.COPY:

INVALID USERCODE SYNTAX

The "=" equal sign delimiter is missing between the keyword "USER" and the assumed usercode character string.

CAN'T SPECIFY USERCODE TWICE

The "usercode" entered is already logged-on to SYCOM, therefore, a second user with the same identifier is not permitted.

Command Error Messages (continued)

COMMAND MESSAGE REASON

.COPY (continued):

INVALID USERCODE NAME

The "usercode" entered was rejected because it contains non-alphabetic characters.

USERCODE MAY NOT HAVE MORE THAN 17 CHARACTERS

The length of the "usercode" entered exceeded a maximum allowable length of 17 characters.

INVALID PASSWORD SYNTAX

The "=" equal sign delimiter is missing from between the keyword "PASSWORD" and the assumed password character string.

CAN'T SPECIFY PASSWORD TWICE

The "password" entered has already been specified by another user; it was rejected for this reason.

INVALID PASSWORD NAME

The "password" entered was rejected because it contains non-alphabetic characters.

PASSWORD MAY NOT HAVE MORE THAN 17 CHARACTERS

The length of the "password" entered exceeded a maximum allowable length of 17 characters.

INVALID FILE ATTRIBUTE: <token>

The command syntax expected a keyword of "US", "USER", or "PASSWORD" which was not located. However, what was entered is displayed in the message as "token".

Command Error Messages (continued)

COMMAND MESSAGE REASON

.COPY (continued):

INVALID FILE ATTRIBUTE DELIMITER: <token>

To be defined.

INVALID TERMINATION OF FILE ATTRIBUTE LIST: <token>

The command syntax calls for the "usercode-password" to be delimited by a ")" close paren which was not located in the command string. The character (or characters) that was found is displayed in the error message.

FILE NAME MISSING

The command syntax requires a "file name" specified that is the subject of the file transfer; it was not given or was not recognized.

KIND DEFINED TWICE

The "KIND" option keyword has been previously specified in the command string.

MISSING "="

The command syntax requires a "=" equal sign delimiter between the keyword "KIND" and the kind option keyword; which as missing.

INVALID KIND ATTRIBUTE

The kind option keyword was not "PACK", "PETAPE", "TAPE", "TAPE7", or "TAPE9" which makes what was entered invalid.

Command Error Messages (continued)

COMMAND MESSAGE REASON

.COPY (continued):

HOST ATTRIBUTE DEFINED TWICE

To be defined.

MISSING "=" AFTER HOST ATTRIBUTE

The required "=" equal sign delimiter used between the keyword "HOST" and the host "system name" is missing.

INVALID HOST NAME

The host system name entered was rejected because it contained non-alphabetic characters.

HOST NAME HAS MORE THAN 17 CHARACTERS

When a host system name is entered with more than 17 characters (maximum allowable), it will be rejected.

UNRECOGNIZABLE VOLUME ATTRIBUTE

This message is somewhat misleading in that it is only an indication that a keyword of "KIND" or "HOST" was expected and not found.

MISSING "," BETWEEN VOLUME ATTRIBUTES

An attribute separator character of "," is required by the command syntax to indicate the end of the attribute "KIND" and its associated option keyword. It is not there and it is wanted.

Command Error Messages (continued)

COMMAND MESSAGE REASON

.COPY (continued):

NO VOLUME ATTRIBUTES DEFINED

Neither the "KIND" or the "HOST" keyword attributes were specified in the command string and "HOST" is required as a minimum.

VOLUME HAS MORE THAN 17 CHARACTERS

The "volume name" (or pack-id) entered exceeded the maximum allowable length of 17 characters. This message could relate to either "volume-1" or "volume-2" names.

DESTINATION HOST NOT ONLINE

The Destination Host Name specified in the COPY command string is not a "local" or "remcte" system identifier. Therefore, the request cannot be accepted as is.

INVALID DESTINATION HOST NAME

The "destination" system name specified is not the same as the "local" system name. Therefore, because the name is not known to SYCOM, it cannot process the request.

SOURCE HOST NOT DEFINED OR REMOTE HOST NOT ONLINE

This message can be displayed for any one of several reasons; they are: 1) the system identifier specified as "source host" is neither the "local" system or the current "remote" system, 2) the required syntactical keyword "FROM" is missing, 3) SYCOM is not connected to the source host specified, or 4) required parens are is missing.

Command Error Messages (continued)

COMMAND MESSAGE REASON

.CCPY (continued):

INVALID SOURCE HOST NAME

The system identifier specified as a "source host" does not match the "local" system name, the "remote" system name, and is not blanks. The reouest is not processed.

CANNOT FETCH WHEN REMOTE SYCOM OFFLINE

This message is indication that an inter-system line connection does not presently exist and the "FETCH" reouest cannot be accommodated.

SOURCE AND HOST NAMES MAY NOT BE DEFINEC IN A FETCH

Use of the shorthand version (FETCH) of the COPY command implies the source and destination systems, therefore, they cannot be specified.

FILE <file name> IN USE AT DESTINATICN--PUT ABORTED

This message is displayed when SYCCM is attempting to "send" a file tc the other system and the specified file is currently in use on that system.

FILE <file name> IN USE AT DESTINATICN--FETCH ABORTED

The file in question has been received and when closing it (which removes existing namec file), finds that the file to be replaced is in use. The FETCH is aborted.

FILE <file name> NOT FOUND

The object file for a PUT file transfer was not found in the disk directory (disk) or was not online (tape or disk pack).

Command Error Messages (continued)

COMMAND MESSAGE REASON

.COPY (continued):

FILE <file name> LOCKED

The object file for a PUT file transfer was found to be "locked" in the disk directory (disk).

FILE <file name> FILE TYPE CANNOT BE TRANSMITTED

The disk file specified for the file transfer is not one of the file types which SYCOM can transfer. (No code files disk-tape or tape-disk, only data files.)

FILE <file name> SOURCE OR DESTINATION FILE NAME MISSING

This message indicates that a subject file name was missing from the command string which was entered as either a "COPY", "PUT", or "FETCH" from another system.

FILE <file name> SOURCE OR DESTINATION FILE NAME INVALID

The specified file name was found to contain characters other than those which are considered valid for a file identifier; or, a multi-file identifier was indicated ("/") but was not included.

FILE <file name> FILE NAME WAS IDENTIFIED WITH MORE THAN 10 CHARACTERS

The name of the file to be transmitted was found to contain an identifier in excess of 10 characters. Its the maximum allowable.

ONTO-PACK OR USER-CODE NOT FOUND--PUT ABORTED

This message would occur when SYCOM was attempting to setup for a disk (or pack) file to be received and the destination "disk pack" was not found to be online. After the message is displayed the transfer is aborted.

Command Error Messages (continued)

COMMAND MESSAGE REASON

.COPY (continued):

FILE <file name> ALREADY EXISTS AT DESTINATION - PUT ABORTED

When SYCOM is setting up to "receive" a disk (or pack) file, it checks to disk directory. If the specified file already exists, this message is displayed and the transfer is aborted.

FILE <file name> RECORD SIZE TOO BIG

The record size for the disk or tape file specified is too large for SYCOM to transfer with the present "Record Size" setting. The file cannot be transferred.

FILE <file name> TOO MANY RECORDS (>99999)

The disk (or pack) file specified for the file transfer contains an excessive number of records; outside SYCOM limitations. The file cannot be transferred.

FILE FAMILY NOT ALLOWED

As suspected, in the specified file name, an equal sign ("=") was found which represented either a "pack-id", "multi-file-id", or "file-id". This is illegal; the file named must be specific.

.DEBUG:

ERROR - "ON" OR "OFF" EXPECTED

The option keyword entered with the DEBUG command is not valid. Please try again.

Command Error Messages (continued)

COMMAND MESSAGE REASON

.DISCONNECT:

STATION ALREADY DISCONNECTED

This only means that a "DISCONNECT" command was entered from a terminal that was already disconnected; it was ignored.

.EOJ:

STOP OR WAIT IN PROGRESS

This means that a ".WAIT" command has been entered and SYCOM is responding to this request; the ".EOJ" command is ignored.

.FETCH:

CANNOT FETCH WHEN REMOTE SYCOM OFFLINE

This message is indication that an inter-system line connection does not presently exist and the "FETCH" request cannot be accommodated.

SOURCE AND HOST NAMES MAY NOT BE DEFINED IN A FETCH

Use of the shorthand version (FETCH) of the COPY command implies the source and destination systems, therefore, they cannot be specified.

FILE <file name> IN USE AT DESTINATION--FETCH ABORTED

The file in question was received and when attempting to close it (which removes any existing file copy), finds that the file to be replaced is currently in use. The FETCH is aborted and the received file is lost; it must be requested again.

Command Error Messages (continued)

COMMAND MESSAGE REASON

.LIST:

UNRECOGNIZED OPERAND "<token>"

This simply means that an incorrect or misspelled option keyword was entered; and rejected.

.OFFLINE:

STOP OR WAIT IN PROGRESS

This message indicates that SYCOM is in the process of servicing a ".WAIT" command which in effect produces an "offline" condition; the ".OFFLINE" command is ignored.

.ONLINE

This command does not cause an error message.

.PHONE:

INVALID PHONE NUMBER

This message is returned when the phone number entered is either more than 20 digits in length or contains alphabetic characters.

.PUT:

CANNOT PUT WHEN REMOTE SYCOM OFFLINE

This message means that an inter-system connection does not presently exist, therefore, a "PUT" cannot be done.

Command Error Messages (continued)

COMMAND MESSAGE REASON

.PUT (continued):

SOURCE OR DESTINATION HOST NAMES MAY NOT BE DEFINED IN PUT

Use of the shorthand version (PUT) of the COPY command implies the source and destination systems, therefore, they cannot be specified in the command string.

FILE <file name> IN USE AT DESTINATION--PUT ABORTED

This message occurs when SYCOM is attempting to "send" a file to the other system and the specified file is currently in use on that system. The PUT is aborted.

.QS:

ERROR: NO "=" IN ".QS" COMMAND

This message occurs when a ".QS" command is entered and the required equal sign "=" was missing. The function requested by this command was not done.

ERROR: ".QS" OPERAND NOT NUMERIC

The ".QS" command is entered with a non-numeric queue size integer field and was ignored. It is necessary to re-enter the command string in correct format.

ERROR: "QS" OPERAND MAY NOT EXCEED 15

The queue size integer entered was greater than 15 and was summarily rejected. It must be re-entered with an appropriate integer value.

ERROR: "QS" OPERAND MUST BE AT LEAST 4

The queue size integer entered was less than the value 4 and was rejected. The command must be re-entered with a legitimate integer value.

Command Error Messages (continued)

COMMAND MESSAGE REASON

.RY:

UNRECOGNIZED COMMAND <RY xx>

The command string contained an urrecognizable ootion keyword; not "CP", "LP", or "CR".

.SIZE:

ERROR: NO "=" IN ".SIZE" COMMAND

This message is clear, the command syntax "requires" an equal sign (=) prefixing the size integer.

ERROR: ".SIZE" OPERAND NOT NUMERIC

The integer entered with the command string was comprised of at least one non-digit character. The integer must be all diqits, no exceptions.

ERROR: ".SIZE" CANNOT EXCEED 400

This message is displayed when a size integer is entered which exceeds the maximum message block size established for the system; four hundred (400) bytes.

ERROR: ".SIZE" CANNOT BE LESS THAN 78

This message is displayed when the size integer entered is less than the minimum allowable message block size; for this system the minimum size is 78 bytes.

Command Error Messages (continued)

COMMAND MESSAGE REASON

.STOP:

This command cannot directly cause an error message.

.SV:

UNRECOGNIZED COMMAND <SV <xx>>

The .SV command was entered either without an option or the option entered was found to be invalid (not "LP" or "CP").

.USER:

INVALID USERCODE NAME

The usercode used to log-on with has been found to be invalid. Please log-off and log-on again with correct usercode/password combination.

INVALID PASSWORD NAME

The password which the user signed-on with has been found to be invalid. Please log-off and log-on again with a correct usercode/passworc combination.

Command Informational Messages

COMMAND MESSAGE REASON

SYCOM IS CLOSING; CANNOT ACCEPT INPUT

This message is returned in response to any new command entered "after" an ".EOJ" command is entered but prior to the actual shutdown of SYCOM. (Once shutdown has been initiated, commands are no longer accepted.)

.READ:

REQUEST IGNORED: CARD FILE STILL OPEN

This message is displayed if a ".READ" command is entered when an input card file is currently active. The entire command string is rejected and discarded.

REQUEST IGNORED: NOT LOGGED ON

This message could mean the card reader is not ready.

FILE "RJECARDS" NOT PRESENT, TRY AGAIN

To be defined.

.COPY:

CAN'T SCHEDULE NOW; TRY AGAIN LATER

To be defined.

REQUEST SCHEDULED; WILL SEND FILE <file name> ASAP

To be defined.

REQUEST SCHEDULED--REMOTE SYCOM OFFLINE

To be defined.

System Parameter Error Messages

The following error messages could occur during the parameters input process. The problem indicated applies to both SPO input as well as card input. In either case, SYCOM must be DS'd, the problem(s) corrected, and the parameters reentered as required.

MESSAGE REASON

% SYCOM =<job #> INVALID PARAMETER: <token>

This message is displayed when the parameter keyword entered did not match one of the acceptable keywords.

% SYCOM =<job #> '=' EXPECTED BUT GOT <token>

The parameters which include a value must be separated by the "=" character. In one of the parameters entered this word separator was missing and SYCOM is unforgiving in these matters.

% SYCOM =<job #> INTEGER EXPECTED BUT GOT <token>

The BUFFER and RECORD parameters require an integer which specifies the BUFFER/RECORD size in bytes. As this is not what was entered, what was found is included in the message.

% SYCOM =<job #> BUFFER SIZE EXCEEDS MAXIMUM ALLOWABLE

The maximum allowable BUFFER size is 1950 bytes. What was specified exceeded this limit.

% SYCOM =<job #> RECORD SIZE EXCEEDS MAXIMUM ALLOWABLE

The maximum allowable RECORD size is 1950 bytes. What was entered exceeded this limit.

% SYCOM =<job #> INVALID SYSTEM ID

The "system id" entered has exceeded the parameter limit of 17 bytes. Please shorten it up some and reenter it.

System Parameter Error Messages (continued)

MESSAGE REASON

% SYCOM =<job #> RECORD SIZE EXCEEDS BUFFER SIZE

The integer entered for the parameter "RECORD" exceeds the current "BUFFER" size. Do it again.

% SYCOM =<job #> INVALID PARAMETERS FILE, BEING REMOVED

This message occurs when the "PARAMETERS" file is opened, the parameters are read in, and the "maximum buffer size" was found to exceed the system maximum of 1950 bytes. When this condition is encountered, the "SYCOM/PARAMETERS" file is removed and new parameters must be entered.

System Warning Messages

Warning messages displayed on the User SPO indicate a problem at the system level. They have been rectified by SYCOM so that operations can continue without operator intervention. They are not generally considered to be major or extensive in nature and therefore, SYCOM has elected to take immediate corrective action.

MESSAGE   REASON

```
% SYCOM =<job #> REMOTE FILE OVERFLOW, MSG DISCARDED
% SYCOM =<job #> <60 or less characters of message>
```

Self explanatory.

```
% SYCOM =<job #> INQ OVERFLOW, FOLLOWING MSG DISCARDED
% SYCOM =<job #> <60 or less characters of message>
```

Self explanatory.

```
% SYCOM =<job #> TERMINALS FILE OVERFLOW, MESSAGE DISCARDED
% SYCOM =<job #> <60 or less characters of message>
```

Self explanatory.

```
% SYCOM =<job #> ERROR: MESSAGE TYPE = <type code>
```

This message indicates that the input message type received on the remote file "LINK" is incorrect and the message has been shunted off.

```
% SYCOM =<job #> ERROR: INCOMPLETE IO ON LINK FILE
```

To be defined.

```
% SYCOM =<job #> LINK QUEUE OVERFLOW, MESSAGE DISCARDED
```

To be defined.

```
% SYCOM =<job #> QUEUE RJEQ FULL, MESSAGE DISCARDED
```

To be defined.

System Warning Messages (continued)

MESSAGE REASON

% SYCOM =<job #> NUMBER OF TIMEOUTS = <number>

This message is displayed automatically when "timeouts" keep re-occurring on the inter-system line connection. However, it is only output on an incremental basis [redacted] to reduce SPO output activity.

% SYCOM =<job #> LOSS OF DATA-SET-READY

Display of this message occurs each time the indicated event is detected. A count is maintained of these occurrences and SYCOM goes into the state of "initialization". In progress file transfers are aborted and lost.

% SYCOM =<job #> HOST SYSTEM CARD READER HALTED

When this message occurs it is to inform the system operator of a condition that presently exists on the remote system. No action is required by the operator.

% SYCOM =<job #> HOST SYSTEM CARD READER ABORTED

This message is used to inform the system operator that card reader input from the remote system has been aborted. No action is required of the operator.

% SYCOM =<job #> HOST SYSTEM CARD READER READY

A message that is used to inform the local system operator that card input from the remote system has resumed.

% SYCOM =<job #> SYCOM WILL CLOSE WHEN NEXT IDLE

Self explanatory.

System Warning Message (continued)

MESSAGE REASON

% SYCOM =<job #> SYCOM CLOSING ON REQUEST FROM REMOTE SYCOM

Self explanatory.

% SYCOM =<job #> ERROR: MESSAGE TYPE = <message type>

An input message from the remote file "LINK" was found to be invalid. The message type code is included in the displayed error message for reference.

% SYCOM =<job #> REMOTE SYCOM HAS GONE DOWN AND COME BACK UP

This message is self explanatory.

System Condition Messages

Any of the following messages displayed on the User SPO indicate an abnormal system condition of some type. As a rule, when one of these conditions is encountered, the display of an appropriate message also elicits a response from the system operator.

MESSAGE REASON

```
% SYCOM =<job #> CARD PUNCH EXCEPTION
% SYCOM =<job #> PLEASE ENTER ".RY CP" WHEN DEVICE IS READY OR
                 ".SV CP" TO SAVE THE FILE
```

This message is displayed when SYCOM is attempting to punch a card and an "exception" condition is encountered. Generally this indicates that the card punch has gone "not ready" for any one of several reasons. It is necessary to respond to SYCOM with the appropriate command when action has been taken to ready the unit.

```
% SYCOM =<job #> LINE PRINTER EXCEPTION
% SYCOM =<job #> PLEASE ENTER ".RY LP" WHEN DEVICE IS READY OR
                 ".SV LP" TO SAVE THE FILE
```

This message is displayed when SYCOM is attempting to print a line and an "exception" condition is encountered. Generally this indicates that the line printer has gone "not ready" for any one of several reasons. It is necessary to respond to SYCOM with the appropriate command when action has been taken to ready the unit.

```
% SYCOM =<job #> PARITY ERROR ON READ PLEASE READ LAST CARD AGAIN
```

This message is displayed when SYCOM is attempting to read the card file and an "exception" condition is encountered. Generally this indicates that the card reader has gone "not ready" for any one of several reasons. It is necessary to respord to SYCOM with the appropriate command when action has been taken to ready the unit.

System Condition Messages (continued)

MESSAGE   REASON

% SYCOM =<job #> PUNCH ERROR

The card punch has either detected an error in the last card punched or has degraded into a "not ready" state. In either case, a "suspend" control message has been sent to the Server System.

% SYCOM =<job #> READER ERROR

The card reader has either detected an error in the last card read, gone into a "not ready" state, or has detected an invalid character in a card column; other than column 1. If there was a read error, the last card must be put back into the card hopper. The card reader is then made physically "ready"; all cases. SYCOM must then be informed of this new status condition with the entry of a ".READY" command.

% SYCOM =<job #> PRINT ERROR

The line printer has either detected an error on the last line printed or has gone into a "not ready" state. In both cases a "suspend" control message is sent to the Server System. When everything is again ready with the printer, a ".READY" command must be entered to inform SYCOM.

% SYCOM =<job #> CARD READER CLOSED

This simply means the card input has been completed and the card file closed.

% SYCOM =<job #> QC DONE

In doing a read on the remote file "LINK" an "EOF" condition was detected. This condition is caused by the MCP when a "QC" command is entered; which shuts down all programs associated with the Network Controller.

System Condition Messages (continued)

MESSAGE REASON

% SYCOM =<job #> ERROR: INCOMPLETE IO ON LINK FILE

In doing a read on the remote file "LINK" an "incomplete io" condition was detected. This message merely reports on the situation.

% SYCOM =<job #> PHONE NUMBER REJECTED BY NC <phone number>

To be defined.

% SYCOM =<job #> INVALID CONTROL MESSAGE RECEIVED

When SYCOM receives a system "control message" from a remote system and the "type code" in this message is not within the 00 - 23 number range, then the above message is displayed. The input message is discarded and no further processing is done on its behalf.

% SYCOM =<job #> BUG IN NETWORK CONTROLLER. VTT INDEX = <vtt>

A "close" message has been received from the system Network Controller for a program known to SYCOM which contained a "lsn list". This list included a Virtual Terminal Table Index (VTT INDEX) entry for a terminal not signed-on to that program, thus the reason for this message.

% SYCOM =<job #> LOST CONNECTION TO REMOTE SYSTEM SYCOM

This message is self explanatory also.

System Fatal Messages

These messages are of the type which precluce the automatic shut down of SYCOM. There are considered "fatal" in the outlook of continuing operations and therefore, SYCCM is shut down.

MESSAGE REASON

% SYCOM =<job #> FATAL ERROR AT <program sequence #>

The display of this message on the User SPO could be the result of any number of reasons. The SYCOM source sequence number displayed is the only clue to the problem. SYCOM initiates a "memory dump" at this point and automatically goes to EOJ after certain disconnect functions have been satisfied.

% SYCOM =<job #> REMOTE VERSION OF SYCOM IS INCOMPATIBLE
% SYCOM =<job #> HIS VERSION="<x>", MINE="4"

This message identifies the problem at hand. SYCOM goes to EOJ immediately at this point.

% SYCOM =<job #> ANOTHER SYCOM WITH SAME MULTI-FILE-ID IS RUNNING

The display of this message indicates that SYCOM cannot gain access to the "SYCOM/FTP.QUEUE" file (cn disk) as it is presently in use by another program.

% SYCOM =<job #> PRINT.PUNCH.QUEUE OVERFLOW: PROGRAM ABORT

To be defined.

% SYCOM =<job #> REMOVE FTP.QUEUE OR CHANGE ITS NAME IF 7.0 FILE

To be defined.

APPENDIX B: CHARACTER SET

The following printable characters define the character set acceptable to SYCCM for input messages from terminals. They are arranged by columns in internal collating sequence with "blank" low and "9" high:

```
blank  &   -   ?   a   h   o   v   C   J   Q   W   3

  [    ]   /   :   b   i   p   w   D   K   R   X   4

  .    $   ¤   #   c   j   q   x   E   L   \   Y   5

  <    *   ,   @   d   k   r   y   F   M   S   Z   6

  (    )   %   '   e   l   s   z   G   N   T   0   7

  +    ;   _   =   f   m   t   A   H   O   U   1   8

  |    ¬   >   "   g   n   u   B   I   P   V   2   9
```

It is recommended that the standard "signal character" of "*" be retained by the SYCOM user for use in message transfers to the SMCS. However, if this is not suitable for any reason, one of the following special characters are suggested:

```
[  &  $  %  ]  #  @  +
```

Special data communications characters reccgnized by SYCOM are as follows:

```
SOH  EOT  DLE  ENQ  STX  ETX  RS  FS
```

APPENDIX C: RESERVED WORDS

The reserved words recognized by SYCOM are dependent upon the position type format. Hence the following list of reserved words are divided into the following types:

- Type 1 - Commands
- Type 2 - Specifications

Words which are immediately followed by a word in parentheses are interchangeable alternates for that word.

Type 1

.ABORT
.BYE
.CL
.CONNECT
.COPY
.CP
.DEBUG
.DISCONNECT
.EOJ
.FETCH
.LIST
.LOG
.LP
.OFFLINE
.ONLINE
.PHONE
.PUT
.QS
.READ
.RY
.SIZE
.STOP
.SV
.US (USER)
.WHAT

Type 2

AS
CP
CR
DISK
FETCH
FROM
HOST
KIND
LP
NR
OFF
ON
ONTO
PACK
PARAMETERS
PASSWORD
RETAPE
PUT
SCHEDULE
TAPE
TAPE7
TAPE9
TERMINALS
TO
US
USER
USERS

APPENDIX D: COMMAND SUMMARY

The following command summary list is presented in alphabetical sequence for easy reference. Refer to the command descriptions for command syntax and option details.

COMMAND	STATION	FUNCTION
ABORT	any	Abort a PUT or FETCH file transfer.
BYE	any	Station log-off from SYCOM.
CL	SPO	Clear peripheral device.
CONNECT	any	Request establishment of a terminal to a Server System.
COPY	any	File transfers (PUT or FETCH).
CP	SPO	Inform SYCOM of "not ready" peripheral.
DEBUG	SPO	Start/stop input/output message tracing.
DISCONNECT	any	Request for terminal disconnection from the Server System.
EOJ	SPO	Cause SYCOM into an orderly shutdown.
FETCH	any	File transfers (fetch from other system)
LIST	any	Display selective system information.
LOG	SPO	Display a list of data comm error counts
LP	SPO	Inform SYCOM line printer "not ready".
OFFLINE	SPO	Cause SYCOM to break the connection to a Server System.
ONLINE	SPO	Establish a Server System connection.
PHONE	SPO	Establish or clear phone number.

COMMAND	STATION	FUNCTION
PUT	any	File transfers (put to other system).
QS	SPO	Change Network Controller queue depth.
READ	SPO	Open and transmit card input file.
RY	SPO	Inform SYCOM of available peripheral.
SIZE	SPO	Change maximum message block size.
STOP	SPO	Cause SYCOM to shutdown immediately.
SV	SPO	Stop transmitting and "save" file.
USER	any	Log a user onto SYCOM.
WHAT	any	SYCOM status inquiry.

APPENDIX E: COMMUNICATIONS CONTROL MESSAGES

GENERAL

As there are several message formats used for inter-SYCOM data communications, each format is specified in detail. These message formats are presented in a sequence dictated by the message type code. A list of message types are shown below, followed by individual message format descriptions.

MESSAGE TYPES

The message type codes are derivatives from the "DA1"/"DA2" (Device Address) field. The message type codes apply to message transmissions in either direction and are compatible between system types. Following then are the device address conventions used by SYCOM; all other values are considered invalid:

DA1	DA2	Usage
0	0	System control
0	1	SPO messages
0	2	Card reader/punch data messages
0	3	Line printer data messages
0	4	File transfer data messages
8	2	Packed card punch data messages
a	A	User System "transmit", Server System
thru		"receive", virtual terminal messages
z	Z	
A	a	User System "receive", Server System
thru		"transmit", virtual terminal messages
Z	z	

Device addresses are advisory only. The receiving system can direct a message to any appropriate peripheral device, based on its own requirements.

Note that the line printer data is allowed in both directions. When a system receives line printer data, it can print it on any available printer, independently of whether the system is termed a User System or Server System.

MESSAGE FORMATS

Basic Data Communication Messages

As per the line protocol, the below message format adheres to the criteria of standard RJE message formatting:

```
SATSDD                                EB
ORMTAA<--------- TEXT --------->TC
HM#X12                                XC
```

The individual characters in this message function as follows:

A. SOH - This is the standard ASCII character which is used to indicate data transmission as opposed to connection establishment communication.

B. ARM - The "Acknowledge Received Message" character which is used to accept or reject the last message block received.

C. TM# - The "Transmission Number" character used in the acknowledgement of a message. It represents one of two values (alternating characters) selected for implementation by SYCOM. The characters "5" and "&" are used by the User System SYCOM whereas the characters "F" and "U" are used by the Server System SYCOM.

D. STX - A standard control character used to delimit the preceding header characters from the message.

E. DA1/DA2 - Two characters representing a "Device Address" value.

F. TEXT - The message text field usually containing data (record images or a message).

G. ETX - The standard "end-of-text" control character.

H. BCC - A Block Check Character which is comprised of a longitudinal parity bit configuration that is used, by the hardware, to verify that the message has been received correctly.

System Control Message

System control messages are used to exchange control type data between systems. A complete descriptive list follows the message format which is a variation of the basic message format:

```
SATSODCC                                 EB
ORMTAADD<---- variables (text) ---->TC
HM#X1212                                 XC

  "00"
```

Here we have the addition of the "CD1"/"CD2" characters. These two characters represent the system message "type" code; which may or may not be followed by variable control data.

SPO Messages

The SPO Message format uses the basic format without alteration. There are no provisions for blocking SPO Messages, therefore, the rule is: one SPO Message per control message block. The format of this type message is as follows:

```
SATSOD                                EB
ORMTAA<--------- TEXT --------->TC
HM#X12                                XC

  "01"
```

Card Reader/Punch Message

The format for card reader/punch data images is as follows: The "PCC" character represents a "printer control character" which in this case is "0". The "RS" (Record Separator) character used between card images provides the means for blocking.

```
SATSDDP                      R                        EB
ORMTAAC<---- card image ---->S<---- card image ---->TC
HM#X12C                                                XC

   "020"
```

Line Printer Message

The following format variation is used to transmit Line Printer data lines between systems:

```
SATSDDP                      R                        EB
ORMTAAC<---- line image ---->S<---- line image ---->TC
HM#X12C                                                XC

   "03x"
```

This format is the same as the Card Reader/Punch format except that the "PCC" character is utilized. The value of this field can be one of the following codes:

PCC	Function	
0	Print, no spacing	
1	Print, single space	
2	Print, double space	
A	Skip to channel	1 (top of page)
B	"	2
C	"	3
D	"	4
E	"	5
F	"	6
G	"	7
H	"	8
I	"	9
J	"	10
K	"	11
L	"	12 (bottom of page)

File Transfer Message

The File Transfer message format is also the same as that used to transfer card images. Although the "PCC" character field is included, it is only there to maintain message conformity:

```
SATSDD                          R                      EB
ORMTAA<---- record data ---->S<----record data ---->TC
HM#X12                                                 XC

   "04"
```

Virtual Terminal Message

The message format utilized for virtual terminal messages is the basic message format, with one exception: the "device address" for a virtual terminal is comprised of twc letters (1 lower, 1 upper "transmit": 1 upper, 1 lower "receive") and will vary by terminal. The format is as follows: (Terminal and SPO messages are "not" blocked.)

```
SATSDD                                   EB
ORMTAA<---------- message text --------->TC
HM#X12                                   XC

   "xX"  (transmit)
   "Xx"  (receive)
```

File Separator Messages

This message format is utilized by SYCOM when it is necessary to transfer a "symbolic file header", indicate the "beginning of a file", or indicate the "end of a file". This applies equally to the "Card Reader/Punch", "Line Printer", arc "File Transfer" operations. To accomplish this, there are three minor variations to the File Transfer Message format which are identified with the following message formats.

Begin File Message

The format for the Begin File message is as follows:

```
SATSDDF                           EB
ORMTAAS1<---- file name ---->TC
HM#X12                            XC

   "02"
   "03"
   "04"
```

End File Message

The format for the "end file" message is as follows:

```
SATSDDF EB
ORMTAAS2TC
HM#X12  XC
```

Program-to-Program Messages

The following message format pertains to SYCOM's "program-to-Program" facility; thru a "Remote File":

```
REMOTE.KEY   (10 characters) % Message "header".

   - RSN     ( 3)            % Relative Station Number for
                             % "virtual terminal" as defined
                             % for the program-to-program
                             % remote file; usually "001".

   - LENGTH  ( 4)            % Length of message text.

   - TYPE    ( 3)            % Message type; usually "000".

MESSAGE TEXT (nn characters) % Maximum is "buffersize".
```

Program-to-SYCOM Messages

To be specified when implemented.

APPENDIX F: RESTRICTED MCP COMMAND SUMMARY

All input commands that are prefixed with a "?" are sent directly to the MCP exactly as entered. The MCP then generates appropriate responses which are sent to the terminal via the Network Controller. Security restrictions have imposed certain constraints on MCP commands from remote terminals. Also, certain commands, such as those which alter system options, purge, or clear system peripherals, have been disallowed for a remote terminal. The following list represents the MCP command subset permitted from User System terminals and the User SPO.

The following Command Subset is listed in the order as described in the B1800/B1700 Systems System Software Operational Guide (associated documentation). Any command listed below which is suffixed with "US" requires the command to be entered with a valid "usercode"/"password" combination.

Library Maintenance Instructions

- CH Change (US)
- AD Add (US)
- LO Load (US)
- DU Dump (US)
- UN Unload (US)
- RE Remove (US)

Program Control Instructions

- CO Compile (US)
- DY Dynamic
- EX Execute (US)
- MH Modify Header (US)
- MO Modify (US)

Program Control Instruction Attributes

•	AF	After
•	AN	After.Number
•	CA	Conditional
•	CG	Charge (certain restrictions apply)
•	DS	Dynamic.Spaces
•	FI	File
•	FR	Freeze
•	HO	Hold
•	IN	Interp interpreter
•	IT	Intrinsic.Name
•	ID	Intrinsic.Directory
•	ME	Memory
•	OV	Override
•	PR	Priority
•	SC	Schedule.Priority
•	SW	Switch
•	TH	Then
•	UC	Unconditional
•	UF	Unfreeze
•	UV	Unoverride
•	VI	Virtual.Disk

System Control Instructions

•	AX	Response to ACCEPT
•	BF	Display BACKUP files (US)
•	CP	Compute
•	CU	Core usage
•	DF	Date of file (US)
•	DM	Dump memory and continue
•	DP	Dump memory and discontinue
•	DS	Discontinue program
•	FN	Display internal file name
•	FR	Final reel of unlabeled tape file
•	FS	Force from schedule
•	GO	Resume stopped program
•	HS	Hold in waiting schedule
•	HW	Hold in waiting schedule until job EOJ
•	MR	Close output file with purge
•	MX	Display MIX
•	OF	Optional file response
•	OK	Continue processing
•	OL	Display peripheral status
•	PD	Display directory
•	PM	Print memory dump
•	PR	Change priority

System Control Instructions (continued)

- PS Prod schedule
- QF Query file (US)
- QP Query program
- RB Remove backup files (US)
- RM Remove duplicate disk file
- RS Remove jobs from schedule
- SP Change schedule priority
- ST Suspend processing
- SW Set switch
- TD Time and date
- TI Time interrogation
- TS Test switches
- UL Assign unlabeled file
- WD Display MCP date
- WM Display current MCP and interpreter
- WS Display schedule
- WT Display MCP time
- WY Program status interrogation

Job Spawning Control Attributes

- QU Queue command
- US Usercode command
- ZQ Zip queue command

MIX related MCP commands may refer only to jobs initiated by the same remote terminal where the initial command was input. All MCP commands must be entered in UPPER CASE.....

APPENDIX G: PARAMETERS FILE ATTRIBUTES

The PARAMETERS file is a disk file, data type, with one physical record of 180 characters. This parameters record is formatted as shown below.

Record Format

Field	Length (in bits)	Default Value	Description
Filler	105	" "	Reserved for identifier.
CURRENT.DATE	72		Run date as "dd mm yy"; (9 bytes).
STATION.ID	136	" "	System identifier; 17 bytes.
STATION.ID.LENGTH	24	0	Length of system identifier.
MAX.BUFFER.SIZE	24	400	File Transfer buffer size.
REC.SIZE	24	180	Maximum record size for RJE message block transmissions.
TIMEOUT.INTERVAL	24	0	A value for message display intervals.
AUDITING	1	0	DEBUG on/off switch. Default of "0" inhibits trace list.
QUE.P	1	0	When "set", this parameter permits "program-to-SYCOM" communications.
NOFTP.P	1	0	To be specified.
NOFS	1	0	To be specified.
WAIT.P	1	0	To shutdown or not shutdown when receiving a "disconnect message" from another SYCOM. 0 = shutdown; 1 = idle.

	Length (in bits)	Default Value	Description
Filler	3	000	Reserved for future use.
MY.NAME	240		SYCOM's name; 30 bytes.
MY.JOB.NO	32		SYCOM'S "job #"; 4 bytes.
Filler	128		Reserved for identifier.
COMPILE.DATE	72	" "	Compile date as "dd mmm yy"; (9 bytes).
Filler	128	" "	Reserved for identifier.
COMPILE.TIME	80	" "	Compile time as "hh:mm:ss.t" (10 bytes).
Filler	334		Reserved for future use....

APPENDIX H: SUMMARY OF FILES

Program	Internal Name	External Name	Device	Use	Size
SYCOM	RJECARDS	RJE/CARDS	Reader	I	80/1
	RJEPUNCH	RJE/PUNCH	Punch*	O	80/1
	RJEPRINT	RJE/PRINT	Printer*	O	132/1
	LINK	LINK	Remote	I/O	455/1
	LINE	LINE	Printer*	O	132/1
	IN	<file name>**	Disk/Tape	I	varies
	OUT	<file name>**	Disk/Tape	O	varies
	TERMINALS	TERMINALS	Remote	I/O	1980/1
	INQ	SYCOM/INQ	Queue	I/O	360/1
	OUTQ	SYCOM/OUTQ	Queue	I/O	360/1
	LINK.Q	SYCOM/LINK.Q	Queue	I/O	2000/1
	FTP.QUEUE	SYCOM/FTP.QUEUE	Disk(R)	I/O	354/1
	RJE.Q	SYCOM/RJE.Q	Queue	I/O	455/1
	FP	SYCOM/PARAMETERS	Disk	I/O	180/1
	CARDS	CARDS	Reader	I	80/1
SYCOM/NC	LINK	LINK	Remote	I/O	455/1
SMCS	MCSQUEUE	MCSQUEUE	Remote	I/O	455/1

* Note: Printer or printer backup disk file.

** Note: The actual file name as specified in the COPY command.

APPENDIX I: SAMPLE APPLICATION PROGRAMS

This program was designed basically to test "program-to-program" communication functions of SYCOM. What this sample application program provides is the mechanics of what is necessary for this communication process. The user can expand upon this sample to create a program with the desired capabilities as required. The program is written in UPL which is available for this purpose.

The only data communication type functions required for this type of application program is to "read" input messages from a Remote File and "write" reply messages into the same Remote File. This message interfacing architecture has been established for this purpose and must be, therefore, adhered to at all times.

PROGRAM-TO-PROGRAM (User System)

```
COMPILE  USER/TEST  UPL  LI;

%     This application program tests program-to-program
%     communication using SYCOM; in the "USER SYSTEM"....
%                                        -----------

& DETAIL
& SIZE
% & EXPAND_DEFINES  XREF  XMAP          % (user optional)
& PASS_END

FILE     RFUSER  (DEVICE = REMOTE, NUMBER_OF_STATIONS = 1,
                  REMOTE_KEY, RECORDS = 80/1, BUFFERS = 2);

DEFINE   CH                         AS #CHARACTER#;

DECLARE  TEXT                       CH (80),
         REPLY                      CH (80);
```

```
DECLARE  1 RMT_KEY                    CH (10),
           2 RMT_RSN                    CH ( 3),
           2 RMT_LENGTH                 CH ( 4),
           2 RMT_TYPE                   CH ( 3),

         1 READ_KEY                   CH (10),
           2 READ_RSN                   CH ( 3),
           2 READ_LENGTH                CH ( 4),
           2 READ_TYPE                  CH ( 3);

   RMT_RSN := "001";           % First station in "family".
   RMT_LENGTH := "0080";       % Standard message length.
   RMT_TYPE := "000";          % MCS type code.

   OPEN RFUSER INPUT, OUTPUT;

   DO  TEST_LOOP FOREVER;

       ACCEPT TEXT;

%  Could check for special input message for EOJ stuff.

       WRITE RFUSER  [RMT_KEY] (TEXT);     % Outputs messages
                                           % directly to the
                                           % Network Controller
                                           % for delivery to the
                                           % other system.

       READ  RFUSER  [READ_KEY] (REPLY);   % Obtains input directly
                                           % for the Network
                                           % Controller.

       ON EOF STOP;                        % In case of "QC"

       DISPLAY REPLY;

   END TEST_LOOP;

% The above program loop will never end; it must be DS'd.
FINI;
```

==

PROGRAM-TO-PROGRAM (Server System)
==

```
COMPILE  SERVER/TEST  UPL  LI;

%     This sample application program tests program-to-program
%     communication using SYCOM; in the "SERVER SYSTEM"....
%                                        -------------

& DETAIL
& SIZE
& PASS_END

FILE     RFSERVER  (DEVICE = REMOTE, NUMBER_OF_STATIONS = 1,
                    REMOTE_KEY, RECORDS = 80/1, BUFFERS = 2);

DEFINE   CH                        AS #CHARACTER#;

DECLARE  TEXT                      CH (80),

         1 RMT_KEY                 CH (10),
           2 RMT_RSN                 CH ( 3),
           2 RMT_LENGTH              CH ( 4),
           2 RMT_TYPE                CH ( 3),

         1 READ_KEY                CH (10),
           2 READ_RSN                CH ( 3),
           2 READ_LENGTH             CH ( 4),
           2 READ_TYPE               CH ( 3);

   RMT_RSN := "001";           % First station in "family"
   RMT_LENGTH := "0080";       % Standard message length
   RMT_TYPE := "000";          % MCS type code

   OPEN RFSERVER INPUT, OUTPUT;

   DO  TEST_LOOP FOREVER;
       READ   RFSERVER  [READ_KEY] (TEXT); % Input from User AP
       ON EOF STOP;                        % In case of "QC"..
       WRITE RFSERVER  [RMT_KEY] (TEXT);   % Output to User AP
   END TEST_LOOP;

% The above program loop will never end; it must be DS'd.
FINI;
```

==

SECTION VII

==

GLOSSARY

SECTION VII

GLOSSARY OF TERMS

This glossary defines SYCOM terms and acronyms used throughout this software operational guide. For ease of use, the glossary is separated into two sections; Terms and Acronyms. Each section is sequenced alphabetically for quick reference.

TERMS

TERM — DEFINITION

ACTIVE STATION: A work station (terminal) that is currently eligible for entering or accepting messages.

AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII): This code, established by the American Standards Associaticn, defines codes for a set of characters to be used in the interchange between business equipment over telephone and telegraph circuits. The code consists of 128 graphic and control characters.

APPLICATION PROGRAM: A program that processes data and is unique to one type cf application. It is sometimes referred tc as a "user program".

TERM DEFINITION

AUTO CALL UNIT: A device furnished by the communications common carrier which allows a User System to automatically establish a dialed link to a Server System over a communications network.

BEGINNING OF JOB (BOJ): The execution of a single program unit to be performed by the system.

BINARY CARD: A punched card with multiple punches in any one or more columns which represent a non-ASCII character.

BIT: The smallest unit of information in a binary system.

BITS PER SECOND (bps): The expression bits-per-second is used to express the speed at which bits of data are transmitted over a communication channel or internally moved within a computer system.

BLOCK: A group of characters/bits transmitted as a single unit of data. Also, see Block Check Character.

BLOCK CHECK CHARACTER (BCC): The Block Check Character is a character appended to data blocks transmitted for the purpose of error detection and parity control.

BLOCKING FACTOR: The maximum number of characters allowed per single transmission.

BROADCAST: The simultaneous dissemination of information to a number of work stations (terminals).

TERM DEFINITION

BUFFER: This is temporary storage that is used during the transfer of data from one device to another. It is also used to compensate for variations in bit rates or timing of events within the system.

CARRIAGE CONTROL: Controls the vertical paper motion for a line printer.

CHANNEL: A device that directs the flow of data between a computer's main memory and input/output devices.

CHARACTER: A digit, letter, or special character . (Letters can be upper or lower case.)

COMMON CARRIER: A company, such as Bell Telephone, that provides communication lines as a public service.

COMMON CARRIER EXCHANGE: A defined area, served by a communications carrier, within which the carrier provides service.

COMMUNICATION LINE: Any medium, such as a wire or a telephone circuit, that connects a terminal with the computer.

CONDITIONED LINES: These are private and leased transmission lines that are conditioned to reduce data distortion while providing transmission of data at lower error rates.

CONNECTION: An established data transmission path between two or more computer systems. It is considered either a temporary or permanent connection, depending on whether it is established with/without switching facilities, respectively.

TERM DEFINITION

CONTROL CARD: A punched card which contains information which identifies a job or describes its requirements to the MCP. The first column must contain an invalid character in 80 column cards, a question mark "?" in 96 column cards, or a rull character @00@ for remote jobs.

CONTROL STATE: The term "control state", for this system, refers to a program that can assume control of the system's processor with privileged operands. As a rule, the type of control state program suggested here infers an "operating system" or MCP.

CONVERSATIONAL MODE: An operational mode in which message data (full or partial messages) is used in lieu or, or in addition to, control characters as replies for message information.

DATA: Data is any representation of information, such as characters, to which meaning might be assigned.

DATA COMMUNICATIONS: Data communication is a method of transferring encoded data by means of electrical transmission between two or terminals and/or systems.

DATA COMMUNICATIONS SYSTEM: A data communications system consists of two or more computer systems that are linked by means of a communication line(s) for the purpose of data exchange.

DATA LINK: A composite of computer systems and the interconnecting network, together with control procedures, operating in a particular mode that permits information to be exchanged between these systems.

DATA SET: A device that provides an interface between a data communications system and a data communicatiors line. Most data communication lines use a modulated carrier for data transfer. The data set provides the logic tc modify the data from being received or transmitted between terminals and the system.

DATA SET READY: A signal provided by the Data Set.

DATA TRANSMISSION: Data transmission consists of transmitting data between two or more computer systems in a system network.

DISCONNECT (DEOT): The term (and mnemcnic) for the "DLE EOT" communication contrcl sequence which is used to signal that a "disconnect" of a switched circuit must be initiated.

DIAL-OUT: The use of an Auto Call Unit to establish a "switched" connection between two (2) computers.

DIAL-UP: The use of a dial or touchtone data set to establish a "switched" connection between two (2) computers.

DISK CARTRIDGE: A secondary data storage device much the same as a "disk pack" but usually smaller in size. It can be moved on-line or off-line.

DISK DIRECTORY: A disk-resident table that contains the name and type of file, together with a pointer to the disk file header or sub-directory for all permanent files which reside on disk.

DISK PACK: A secondary data storage device that can be moved on-line or off-line.

TERM DEFINITION

DISK-PACK-IDENTIFIER dp-id): An identifier which is the name assigned to a disk pack or disk cartridge.

EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE (EBCDIC): A code consisting of 8-bits which is used to represent 256 unique letters, numbers, and special characters.

ERROR: Any received character or sequence of characters that does not conform to those transmitted are considered to be in error.

ERROR RECOVERY PROCEDURES: These are data communication control procedures used to restore normal operations to a data link after an error has occurred.

FAMILY NAME: A identifier used as a file name, or the name assigned to identify a main file with subdirectory entries.

FILE SECURITY: The procedures or special devices used to prevent access to or use of data or programs without authorization.

FILE IDENTIFIER: All disk file-identifiers used on the system must be unique to prevent duplicate file-names. A file identifier can be comprised of any combination of the following file-identifier options:

- file-identifier
- family-name/file-identifier
- dp-id/family-name/file-identifier

HALF-DUPLEX CHANNEL: This is a communication channel in which the signaling may be in either direction, but not in both directions simultaneously.

TERM DEFINITION

HALF-DUPLEX TRANSMISSION: This is a type of data transmission where information is sent in only one direction at a time. It cannot send data in both directions simultarecusly.

HARD COPY: Generally, the printed copy of some material in a readable form is considered to be "hard copy". This would include computer orinted media such as Reports, Listings, and Documents like this.

IDENTIFIER: A word consisting cf from one to seventeen alphabetic, numeric, or special characters in combination.

INTEGER: A whole number as opposed to a fraction; that is, a number that contains the unit of (one) an exact number of times.

INTER-SYSTEM COMMUNICATIONS: The ability for a terminal or program in one system to communicate with a program in another system.

INTRA-SYSTEM COMMUNICATIONS: The restricted ability for a terminal or program in a system to communicate with another program in the same system.

JOB: A "job" to this system is any program that is executable under the control of the MCP; SYCOM is a job.

JOB NUMBER: A unique number assicned by the systems MCP to each job that is executed under its control.

TERM DEFINITION

JOB SPAWNING: The ability for a user to execute jobs under program control and maintain an adequate level of control over these jobs during their execution.

KEYBOARD: A device for the encoding of data by key depression, which causes the generation of the selected code element.

LEASED LINE: A data transmission line reserved for the exclusive use of a customer (data center).

LIKE SYSTEM: A "like system" is a system with physical characteristics similar to those of a target system. For a B1800/B1700 system, a "like" system would be either a B1800 or B1700.

LINE: A data communications line linking two (2) computers or a computer and its associated terminals.

LINE ADAPTER: A line adapter consists hardware and/or software logic used to interface a data set or communications line into a multi-line or single-line I/O control unit.

LINE SWITCHING: The technique of temporarily connecting two communications lines together so that two computers and/or a computer and its associated terminals can exchange data.

LOG-OFF: The procedure by which a user ends a data exchange session either between two computers or between a computer and a terminal.

LOG-ON: The procedure by which a user establishes a data exchange session between two computers or a computer and a terminal.

LOGICAL STATION NUMBER (LSN): The identification number assigned by the Network Controller to a given Station (Terminal). The Logical Station Number has no relationship to file names, but is determined by the croer in which a station is defined in the Network Controller.

MASTER CONTROL PROGRAM (MCP): The Master Control Program is the master "control state" program (operating system) used in the B1800/B1700 series computers for system control.

MESSAGE: A sequence of characters arranged in a form suitable for the purpose of conveying information. A message contains the data to be conveyed (the text) and may, in addition, contain communication characters to aid in the routinc or handling of the message.

MESSAGE FORMAT: A message format defines the placement of message elements within the message text. These elements would be such items as; header, control characters, and text data.

MESSAGE HEADER: Every message moved between a terminal and the SMCS, or an application program for that matter, is prefixed with a "header". This header contains station attributes necessary to identify the station when the time comes to returr a reoly message.

MESSAGE NUMBERING: Message numbering is the sequencing of messages transmitted by the communications system to ensure all messages sent are in turn received.

MODEM: Same as Data Set.

TERM DEFINITION

MULTIFILE IDENTIFIER (MF-ID): A generalization for a group of files with the same "family name" but with different file names.

MULTI-LINE CONTROL (MLC): The (MLC) is an I/O control device that provides the functioral control between computer hardware and the line adapters. The control is an irtegral part of a multi-line communications sub-system.

NETWORK: A number of data communication lines connecting the User System with remote work stations (terminals).

NETWORK CONTROLLER: The NDL generated data communication program used by the SYCOM to interface with the work stations (terminals) and other systems.

NETWORK DEFINITION LANGUAGE (NDL): A descriptive, free-form language, for defining and implementing a data communications network. The NDL compiler analyzes the input statements and generates a custom Network Controller.

NORMAL STATE: The term "normal state", for this system, generally refers to a program which has no facilities to seize control of a system's processor; as opposed to "control state". (e.g., SYCOM is a normal state program.)

OFFLINE: There are two facets to this term for the SYCOM system; 1) a system is "offline" when no communications link currently exists between it and a host system, and 2) a systems' terminal, SPO, or program is "offline" when it is not logically connected to an application program in a host system.

TERM DEFINITION

ONLINE: There are two meanings for this term in a SYCOM system; 1) a system is "online" when a communications link currently exists between it and a host system that provides processing services, and 2) a terminal, SPO, or program in a User System is "online" when it is logically connected to an application program in a host system. (The system must be online in order for a terminal, SPO, or program to be online.)

PARTICIPATE: This term refers to a function in the Supervisory Message Control System (SMCS) which allows it to "participate" in all input/output message traffic between the Network Controller and an application program. In reference to a SYCOM system, the SMCS "is not permitted" to participate in SYCOM's message traffic.

PASSWORD: A unique string of characters that a program, computer operator, or user must supply to meet security requirements before gaining access to data/files. The password is confidential as opposed to the "usercode".

POINT-TO-POINT CONNECTION: This is a configuration in which a network connection is established between two, and only two, computers. The connection may include a switching facility.

PRIVATE LINE: This would be a data communications line which has been furnished to the user for exclusive use.

PSEUDO CARD READER: In a B1800/B1700 system, a pseudo card reader is in fact a cisk file. The disk file may be created via cards, program output, or via a terminal. When a program requires an input card file, the pseudo card reader can be substituted; as that file can be appropriately labeled.

TERM DEFINITION

QUEUE: A logical collection of messages awaiting transmission or processing.

REMOTE JOB: Submission of control statements and data from a remote station (terminal/computer), causing the job described to be scheduled and executed at another system.

REMOTE JOB ENTRY(RJE): The Remote Job Entry program is a method be which a remote computer executes or compiles programs sent to it by other computers; which are considered terminal devices.

REMOTE SUPERVISORY CONTROL STATION (RSC): This is an input/output device located on a User System (remote to a Server System) to provide the operator with the ability to control jobs and to request information from the Server System.

RJE/TERMINAL: The official name for the software system from which SYCOM derived its basic capabilities.

RJE STATION: For this system, the terms "RJE Station" and "User System" are equivalent. The new term is used to denote a system with many additional capabilities and potential, therefore, the term "RJE Station" is no longer appropriate.

RELATIVE STATION NUMBER (RSN): Is the position within the Remote File family; assigned at generation time.

SEMANTICS: The relationship between symbols and their meanings.

SERVER SYSTEM: A Server System is a passive system which provides computing services to the "User System"; locally or remotely.

TERM DEFINITION

SESSION: The period of time during which a user engages in dialog between a terminal and a computer or between two computers. (The elapsed time between when the user logs-on the system and when the log-off occurs.

SESSION NUMBER: A unique number assigned by a originating system to each session.

SINGLE LINE CONTROL (SLC): This is an I/O control device that provides the functional control between a computers main frame and a line adapter. This control is an integral part of a single line communications sub-system.

SITE ID: A unique 17-character identifier sent in a special message to a server system prior to being logged-on to provide a positive means of identification, independent of any other system security.

SPO: The accepted term for the B1800/B1700 systems "supervisory console" (VDT or Teletype). This term has been derived from the obsolete acronym for "Supervisory Printer Output". Usage in this document is for an abbreviation for User Spo.

SPO LOG: A disk file that contains all input/output messages sent to and received from the User SPO.

SPOOLING: The reading and writing of input/output data streams on secondary storage devices, in a format convenient for later output operations and/or processing.

STATION: Same as "TERMINAL".

TERM DEFINITION

SWITCHED LINE: A communications line in which a temporary connection between a User System and a Server System is established by dialing.

SWITCHING: This is an operation involving the interconnecting of circuits in order to establish a temporary communication link between two or more systems.

SYNCHRONOUS TRANSMISSION: A data transmission process in which the sending and receiving devices are operating continuously at substantially the same frequency and are maintained, by means of correction, in a desired phase relationship.

SYSTEM DISK: A disk pack or disk cartridge that is initialized as a system pack. A system pack is under control of the MCP and one or more must be present on the system for the MCP to function. Head-per-track disk is always considered system disk.

TERMINAL: This is an input/output device designed to both receive and send source data in an environment associated with the job to be performed. Further, it is capable of sending entries to, and obtaining output from, the system of which it is a part.

TEXT: A sequence of characters forming to information to be conveyed.

TRANSMISSION: Same as Data Transmission.

TRANSMISSION BLOCK: A transmission block may contain all or part of a message block. A transmission block is independent of the message(s) from which its characters are taken.

TERM DEFINITION

TRANSPARENT MODE: An operational mode in which all coded combinations of eight-bit characters are allowed as message text characters.

UNIT DEVICE IDENTIFIER (udi): The mnemonic identifier assigned to each peripheral device attached to the system.

USER SPO: The User System's Supervisory Control station (VDT).

USER SYSTEM: A User System is the active participant system providing local service to the user wishing to utilize another computing resource; locally or remotely.

USERCODE: A symbol which consists of from one to eight characters to identify the user.

VIRTUAL TERMINAL: A logical communications capability which permits a terminal, SPO, or program in one system to "interact" with an application program in another system.

VDT: Video Display Terminal. A video type work station (terminal) such as the TD820 and TD830 terminal devices.

WAITING SCHEDULE: Contains the names ard run-time attributes of jobs that are waiting to be placed into the Active Schedule.

ZIP: A method by which a program such as SYCOM, communicates with the MCP for certain specific MCP functiors.

ACRONYMS

ACU	-	Auto Call Unit
AP	-	Application Program
ASCII	-	American Standard Code for Information Interchange
BCC	-	Block Check Character
BOJ	-	Beginning Of Job
bpi	-	Bits Per Second
D/S	-	Data Set
DDD	-	Direct Distance Dialing
dp-id	-	Disk Pack IDentifier
EOJ	-	End Of Job
I/O	-	Input/Output
lsn	-	Logical Station Number
MCP	-	Master Control Program
NC	-	Network Controller
NDL	-	Network Definition Language
NIF	-	Network Information File
RJE	-	Remote Job Entry
RSN	-	Relative Station Number
SMCS	-	Supervisory Message Control System
SOG	-	Software Operational Guide
SPO	-	Supervisory Printer Output
SYCOM	-	Systems Communications Module
udi	-	Unit Device Identifier
VDT	-	Video Display Terminal

SANTA BARBARA PLANT

GOLETA, CALIFORNIA

==

Address Comments Concerning

This Document To:

Dennis F. Ehrenberg

Data Communications Group

Computer Systems Division

6300 Hollister Rd

Goleta, California 93017

805/964-6881

10/2~~8~~7/78

See Rev A

DOCUMENTATION ON ENHANCEMENT TO SYCOM

1) When SLAVE/BACKUP is running ~~and AP-1~~ all backup files will be sent to the HUB and printed automatically. These backup files will then be removed from the sending system. Thus to print and remove backup files no operator/user intervention is required.

If you do not want a backup file printed be sure to set the "NO ATP" attribute for the file.

Example:

```
CO HUB LSDL LI: FI LINE NO ATP; FI CARDS NAME HUB/SOURCE DSK;
```

2) The PUT, FETCH and COPY syntax has been extended as follows:

```
[.PUT  ] <file name> [ (LIST, REMOVE)     ]
[.FETCH]             [      or            ]
[.COPY ]             [ (NO LIST, SAVE)    ]
                     [      or            ]
                     [ (NO LIST, REMOVE)  ]
                     [      or            ]
                     [ (LIST, SAVE)       ]
```

The attribute LIST implies that the file will be printed at the receiving end. This attribute is applicable to Backup files only.

The attribute REMOVE will cause the file to be removed from the sending system.

The default attributes for Backup files are (LIST, REMOVE). The default attribute for non Backup file is SAVE. The LIST attribute does not apply to non Backup files.

The PUT, COPY, FETCH systax is unchanged except for the above additions.

Examples:

```
.PUT  (SPO)/#4170 (SAVE) ONTO BACKUP.PRT/4170
.PUT HUB (REMOVE) ONTO SYCOM3
.PUT HUB (US=SPO/X, REMOVE) AS XYZ
```

3) Program to SYCOM communication has been fixed so that a program can initiate .CONNECT.

4) Many changes have been made to the SYCOM request section to prevent garbage messages and lost messages.

5) Compile date

SLAVE/BACKUP	10/~~26~~ 27/78	
SYCOM	10/25/78	or later
Network controller	10/24/78	or later.

6) A listing of SLAVE-BACKUP will be kept on ~~the~~ ~~system~~ SLAVE3. Documentation on SLAVE/BACKUP options is in the listing.