SC30-3167-0
File No. S370/4300-30

Advanced Communications Function for Network Control Program and System Support Programs

Program Product

Installation and Resource Definition

Program Numbers: 5735-XX9
5735-XXA

Version 2

IBM

First Edition (April 1982)
This edition applies to Version 2 of the ACF/NCP program product (program number 5735-XX9) and the System Support Programs for ACF/NCP program product (program number 5735-XXA) and to all subsequent releases and modifications unless otherwise indicated in new editions or Technical Newsletters. Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, refer to *IBM System/370 Bibliography,* GC20-0001, for the editions that are applicable and current.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

This publication has been written by the IBM System Communications Division, Publications Development, Department E02, P. O. Box 12195, Research Triangle Park, North Carolina 27709. A reader's comment form is provided at the back of this publication. If the form has been removed, comments may be sent to the above address. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

© Copyright International Business Machines Corporation 1982

Preface

This publication provides the information necessary to define and generate an Advanced Communications Function for Network Control Program (ACF/NCP) for the IBM 3705-I and 3705-II Communications Controllers.

The publication is directed to system analysts and system programmers responsible for preparing an ACF/NCP to be used in communicating with an IBM System/370 in which one or more of the following access methods are being executed: ACF/TCAM, ACF/VTAM, TCAM, and VTAM. Also in this publication is information on the partitioned emulation programming (PEP) extension to ACF/NCP which permits the 3705 to emulate the operation of an IBM 2701, 2702, or 2703 transmission control unit for specified communication lines. Stations on such lines communicate in emulation mode with application programs in the System/370 via BTAM, QTAM, TCAM, or equivalent access methods that can be used with the transmission control units mentioned.

Note: In this publication, generic terms are used for brevity, as follows. *Network control program* (or NCP) refers to ACF/NCP. *Access method* refers to whichever access method is communicating with ACF/NCP, when there is no need to distinguish between TCAM and VTAM or between their ACF and non-ACF versions. Where necessary, the text refers to *VTAM*—meaning ACF/VTAM or VTAM—and to *TCAM*—meaning ACF/TCAM or TCAM. *ACF/TCAM* and *ACF/VTAM* refer specifically to the ACF versions of these access methods.

In the context of operations in emulation mode, *access method* refers to whichever access method (BTAM, QTAM, TCAM, etc.) is communicating with those lines operating in emulation mode. Some other terms used in this publication are explained in Chapter 1.

This publication also contains the information needed by users of the IBM Airlines Control Program to specify communication lines on which airlines line control is used. (RPQ numbers 858911 and 858912 must be installed in the 3705 to which such lines are attached.) See Appendix L for information on how to specify communication lines requiring airlines line control (ALC).

The term network has at least two meanings. A *public network* is a network established and operated by common carriers or telecommunication Administrations for the specific purpose of providing circuit-switched, packet-switched, and leased-circuit services to the public. A *user application network* is a configuration of data processing products, such as processors, controllers, and terminals, established and operated by users for the purpose of data processing or information exchange, which may use transport services offered by common carriers or telecommunication Administrations.

Network, as used in this publication, refers to a user application network.

Prerequisite and Related Manuals

Prerequisite to use of this publication is a basic understanding of data communications and related access methods. You should also have a general knowledge of the purposes of the IBM 3705 Communications Controllers; this knowledge may be obtained from the publications, *Introduction to the IBM 3704 and 3705 Communications Controllers,* GA27-3057 and *IBM 3704 and 3705 Communications Controllers Principles of Operation,* GC30-3004. See the *ACF/VTAM Installation* and *ACF/TCAM Installation* manuals respectively, for VTAM and TCAM information.

Within this publication, mention is made of the following publications:

Control Panel Guide refers to:

Guide to Using the IBM 3705 Communications Controller Control Panel

Program Reference Handbook refers to:

ACF/NCP Network Control Program, Reference Summary and Data Areas (LY30-3062)

NCP Program Logic Manual refers to:

ACF/NCP Network Control Program Logic (LY30-3061)

NOSP Installation Manual refers to:

Network Operation Support Program Installation and Maintenance Manual (SC38-0279 [for DOS/VS]; SC38-0278 [for OS/VS])

TCAM Programmer's Guide refers to any of the following publications, as appropriate:

OS/VS1 TCAM Programmer's Guide (GC30-2054)
OS/VS2 TCAM Programmer's Guide (GC30-2041)
OS/VS TCAM System Programmer's Guide (GC30-2051)

Consult your IBM representative for the titles and order numbers of ACF/TCAM publications you may need. The *Teleprocessing Preinstallation Guide for IBM 3704 and 3705 Communications Controllers* (GC30-3020) and the *Teleprocessing Installation Record for IBM 3704 and 3705 Communications Controllers* (GC30-3021) may also be used as aids to defining your network control program. The *Preinstallation Guide* gives the appropriate values for many of the network configuration parameters that must be specified in the control program and gives the value(s) appropriate to each type of station and line set accommodated by the program. The *Installation Record* contains a set of formatted sheets representing the communications controller and the attached lines, with labeled spaces suitable for recording the parameter values appropriate for the network being documented. This record should be filled out soon as the equipment configuration of your network is known, and should subsequently be updated each time any changes are made to the configuration.

Please consult your IBM representative for the editions that are current and applicable.

IMPORTANT

This publication explains the functions the network control program can perform, the network configurations that are supported, and how to define and generate a program that perform the functions your installation requires. This book does not contain the restrictions and programming considerations imposed by specific types of stations (terminals) or by other program components (such as CICS) with which the network control program may communicate.

To successfully define a network control program suitable for your installation, consult the appropriate programming manuals for each of the program components and each of the types of data communication equipment (controllers, stations) that make up your installation. Consult your IBM representative to determine the applicable publications that are available.

Contents

Figures

Chapter 1. Introduction

The IBM 3705 Communications Controller can be programmed to communicate with a large variety of remote terminals, transmission control units, and computers. This programming can accommodate many different data communication applications and operational requirements.

Control programs for the communications controller are first defined in the form of macro instruction source statements, then generated by a compilation process, and finally loaded into the communications controller.

Advanced Communications Function for Network Control Program

The Advanced Communications Function for the Network Control Program (ACF/NCP) (referred to in this book as the network control program or NCP) controls the transmission of data between the host processor and the remote stations in the network connected to the communications controller.

An ACF/NCP can communicate with up to eight access methods concurrently via channel connections or data links to other network control programs. In a 3705-I it can communicate with one or two access methods over channel connections to one or two host processors. In a 3705-II the NCP can communicate with up to four access methods over channel connections to the host processors. The NCP may also communicate with access methods that are channel-attached to adjacent network control programs, via communication links. Terminology pertaining to relationships between network control programs and access methods in a network is discussed in this chapter under the heading "Subareas and Network Resources."

The functions of an NCP are divided into two major categories: network control functions and emulation functions.

Network Control Functions

Network control functions include the wide range of capabilities for which the 3705-I and 3705-II controllers are principally designed.

The NCP recognizes and fulfills requests by the communication access method to transmit data to and receive data from the network. In doing so, the program performs whatever operations are needed to establish communications with stations; including polling, addressing, dialing, and answering, as appropriate for the type of station and type of communication line linking the station and the controller. Then the program receives message data into buffers, inserts and deletes transmission control characters as required, and translates message data from processing code (EBCDIC) into transmission code, and vice versa. Finally, the NCP transmits the data from the buffers to an access method or to the stations in the network.

The NCP also governs many aspects of communication between the network and the host processor, such as the amount of data to be accepted from a station once connection is established, the number of devices on a multipoint line with which the access method can communicate concurrently, and exchange of identification sequences with stations on switched lines.

Network control functions also include: (1) automatic error recovery and statistical recording, (2) diagnosis of controller, line, and station malfunctions, and (3) changes to operating parameters during program execution upon request from an access method.

Emulation Functions

Emulation functions comprise a more restricted range of functions, equivalent to those provided by the IBM 2701, 2702, and 2703 (collectively referred to in this book as *transmission control units).* These emulation functions permit most existing user application programs to operate unchanged when a 3705 replaces one or more transmission control units.

You may generate a program that performs only network control functions, only emulation mode functions, or both network control and emulation functions, according to the needs of your installation. (For Emulation Program functions only, refer to the *IBM 3704 and 3705 Emulation Program Generation and Utilities Guide,* GC30-3002.) A program capable of both network control and emulation functions is called a network control program with the partitioned emulation programming (PEP) extension. When generating a program having the PEP extension, you specify, for each communication line in the network, whether that line is to operate in network control mode or emulation mode, or both (alternately). Operation in network control mode means that all of the network control functions apply to data transmission over that line. Operation in emulation mode means that only the emulation functions are performed for that line. These are equivalent to the functions performed by the 2701, 2702, or 2703 to which the line was formerly attached. Because the functions performed by the three types of transmission control units differ in some respects, you specify which of these units is to be emulated for each line.

If you specify operation in both modes, operation can be changed whenever desired from one to the other by command from the access method that communicates with the NCP.

The emulation mode of the NCP with the partitioned emulation programming (PEP) extension allows many programs written to be used with the IBM 2701, 2702, and 2703 transmission control units to operate with the IBM 3705 controllers with no modification. These programs include IBM Type I access methods for the 2701, 2702, and 2703, as well as IBM Type II and Type III programs and user-written programs that interface with the 2701, 2702, and 2703 in a manner equivalent to Type I access method programs. Programs that involve timing dependencies and support of certain special and custom features may, however, require modification.

The emulation mode of operation requires that a type 1 or type 4 channel adapter be installed in the controller for attachment to a byte-multiplexer channel of the System/370.

Emulation functions, in conjunction with the type 1 or type 4 channel adapter, permit the use of the same control sequences and data transfers as do the 2701, 2702, and 2703. They also provide most of the standard functions of these control units. Not supported are the parallel data adapter, synchronous data adapter type 1 programmable two-processor switch, direct attachment of the IBM 1032 Digital Time Unit, the IBM 2712 Remote Multiplexer attachment features, and the reverse channel feature. ASCII transparency can be used only for a communication line serviced by a type 3 communication scanner. *(Exception:* Programming RPQs [PRPQ] are available that permit the type 1 and type 2 scanners to accommodate ASCII transparency and 6-bit transcode.) The ADPREP and SEARCH commands are not supported on lines attached to a type 3 scanner.

Subareas and Network Resources

The term *network control program* (or NCP) appears frequently throughout this publication. When the network configuration includes several network control programs, confusion about which program is being referred to can be avoided by construing the unqualified term, network control program, as meaning that program which is presently under consideration (being planned, coded, generated, executed, etc.). For clarity, this program is sometimes referred to as "the network control program you are defining," or "the present network control program." Where necessary to refer to some other network control program in the network, the text uses an appropriate qualifying word or phrase such as "the distant NCP" or "an adjacent NCP." Similarly, access methods with which the present network control program communicates may be referred to with qualifiers such as "adjacent."

Figures 1-1 through 1-4 show elements of the network as they relate to the present network control program and indicate the terms by which these elements are referred to throughout this publication.

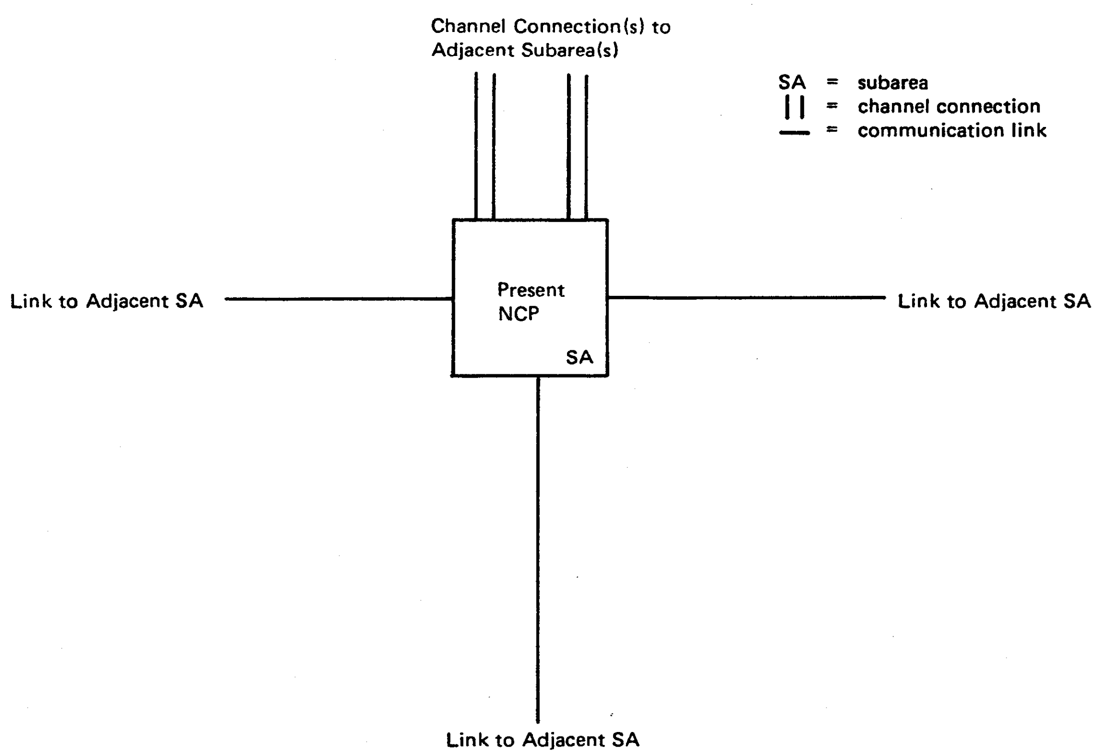


Figure 1-1. Network Control Program and Links to Adjacent Subareas

As shown in Figure 1-1, the present NCP is a subarea of the network and communicates with other subareas (adjacent subareas) over either a channel connection or a communication link. In this publication, the term *subarea* refers to either a network control program or an access method. (Groups of locally attached devices controlled directly by VTAM are also subareas; see the *ACF/VTAM Installation* manual for information on this subject.) Any subarea with which the present network control program communicates directly (that is, without passing through an intervening subarea) is an adjacent subarea.

Figure 1-2 shows three adjacent network control programs to which the present network control program is attached. The link between any two network control programs is called a subarea link. These links always employ synchronous data link control (SDLC) procedures. Figure 1-2 also shows two access methods adjacent to the present NCP. (This figure and the two figures following are intended only to show relationships between the elements of a network as they relate to the present network control program and should not be interpreted as indicating how many of each kind of element may be associated with the NCP.)

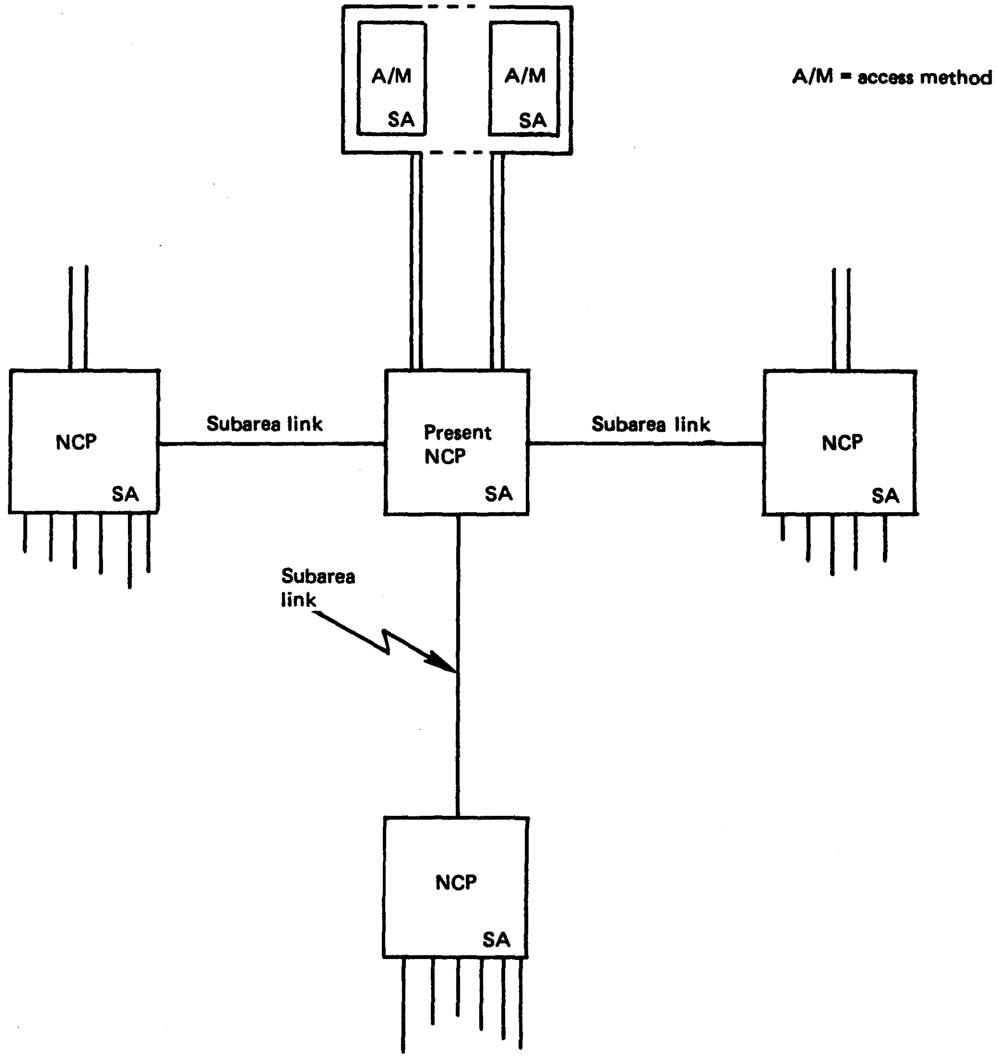


Figure 1-2. Relationship of Network Control Program to Adjacent Subareas

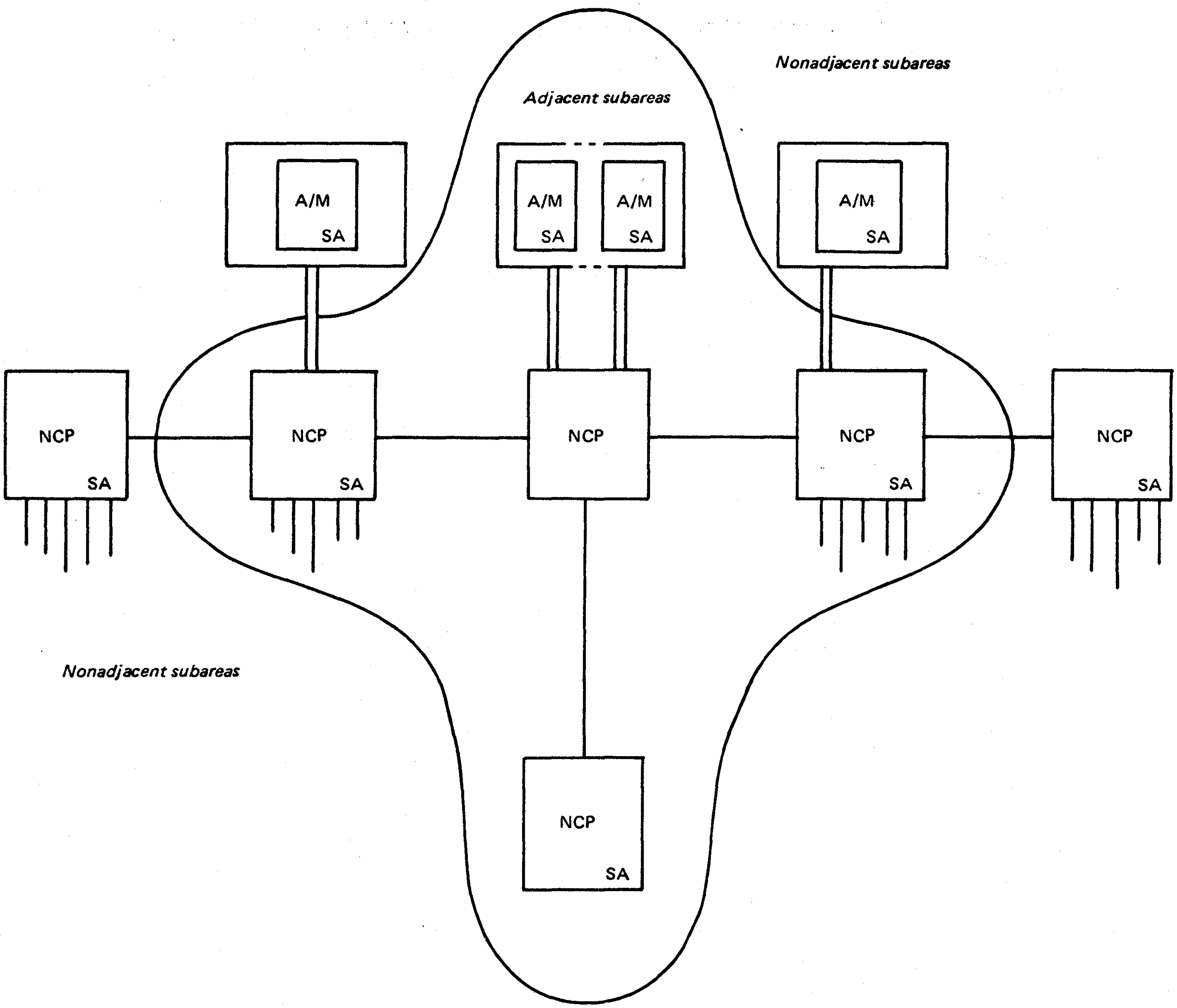


Figure 1-3. Relationship of Network Control Program to Nonadjacent Subareas

The present NCP can communicate with subareas that are separated from it by more than one link or channel connection, as shown in Figure 1-3. These are referred to as nonadjacent subareas and, like adjacent subareas, may be access methods or other network control programs. Nonadjacent subareas may only be communicated with via adjacent subareas.

Each subarea in a network has a unique subarea address by which it is known to other subareas in the network for the purpose of routing communications between them. The source statements for each program specify that program's own subarea address and the subarea addresses of each other subarea in the network with which it is to communicate. (These addresses are specified in the SUBAREA operands of the BUILD, HOST, PATH, and PU macros, described later in this publication.) The maximum subarea address that can appear anywhere in the network must also be specified in each NCP (in the BUILD macro) and in each access method in the network.

Figure 1-4 shows a simple configuration involving one access method and one NCP. In this case there are but two subareas, and the subarea addresses assigned might be 1 and 2, as shown. Figure 1-5 shows three access methods and three network control programs, with addresses 1 to 6 assigned to them. Figure 1-6 shows a larger network containing twelve subareas.

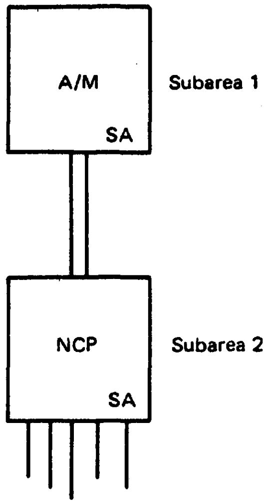


Figure 1-4. A Network with Two Subareas

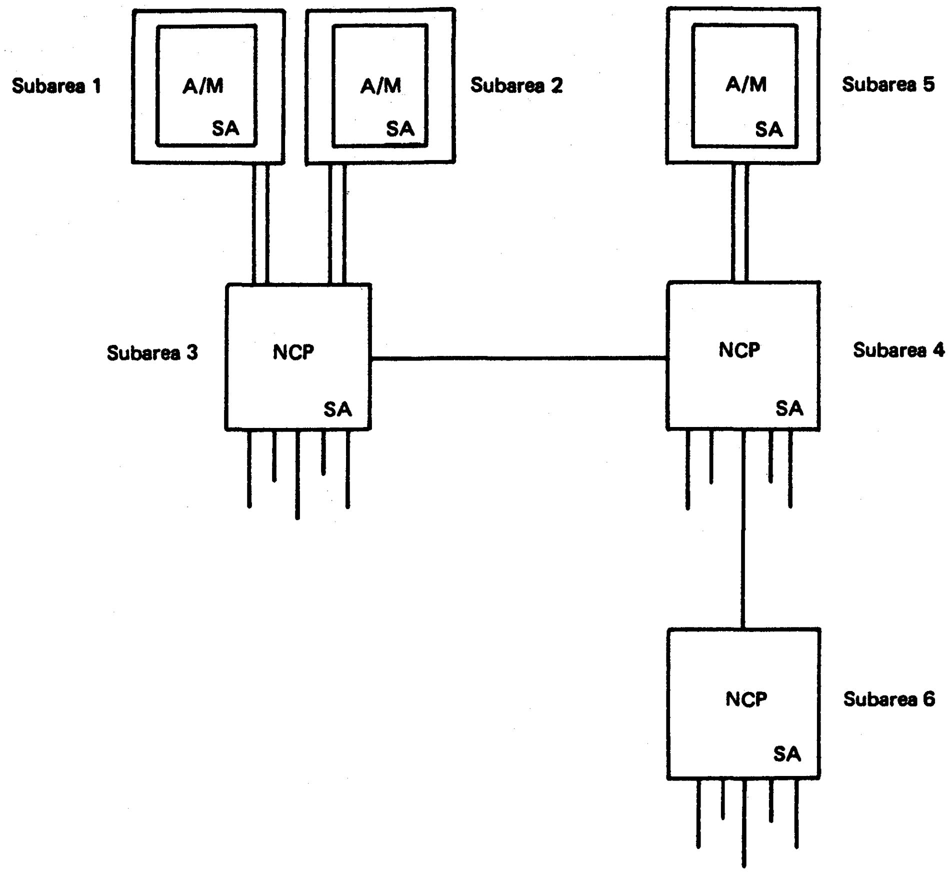


Figure 1-5. A Network with Six Subareas

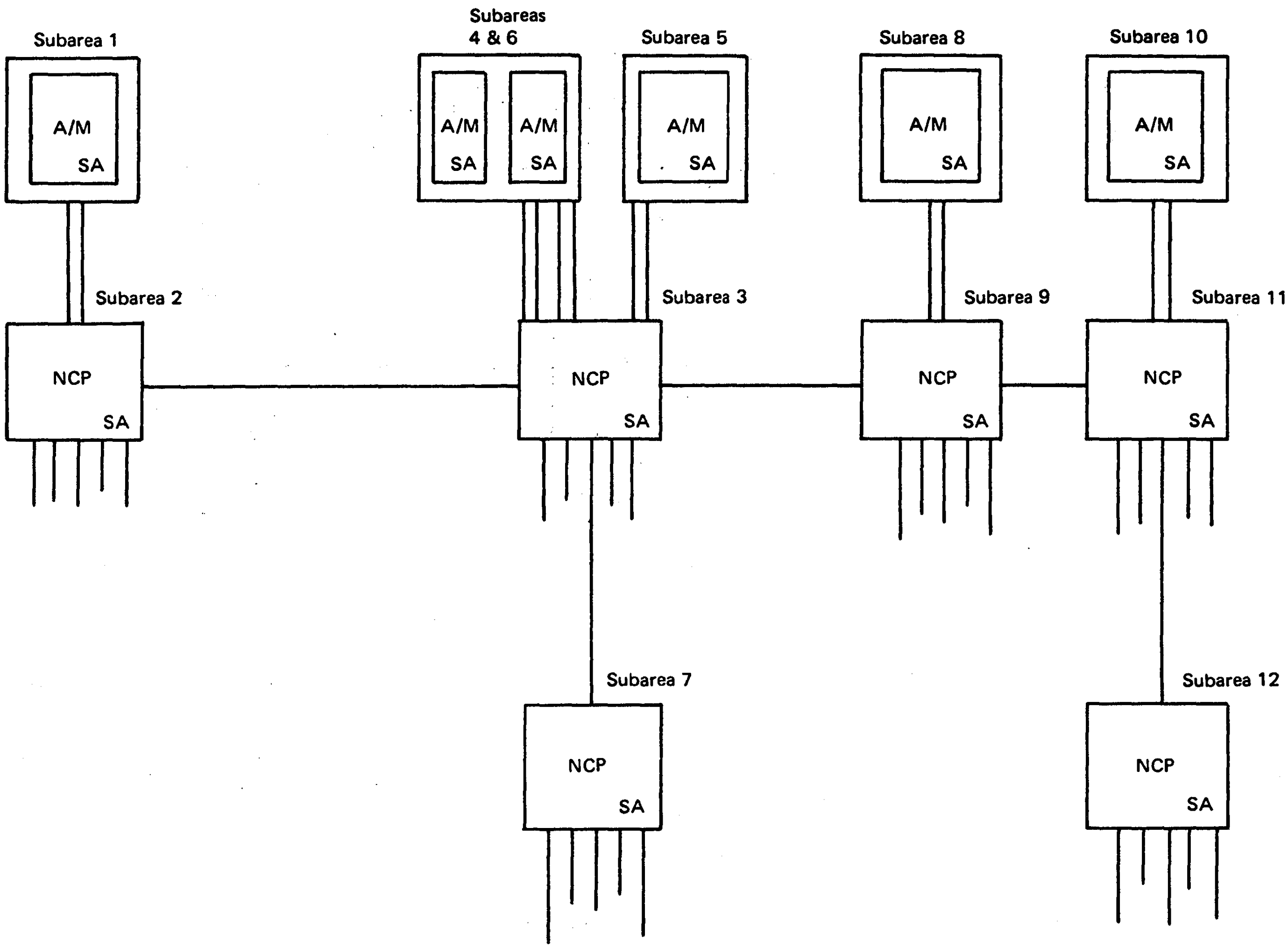


Figure 1-6. A Network with Twelve Subareas

The illustrations thus far have shown only the access method and NCP elements of the network. These elements are kinds of *network resources.* Each communication link, SDLC (SNA) physical unit, and each logical unit, as well as each non-SDLC (non-SNA) station (terminal, transmission control unit, or computer) in the network is also a network resource. *(Exception:* stations communicating with the network control program over lines operating in emulation mode are not referred to as network resources.) Figure 1-7 shows the various kinds of resources in a network. The resources (lines and stations) attached to a network control program constitute a part of that program's subarea, as depicted by the dashed lines in the figure.

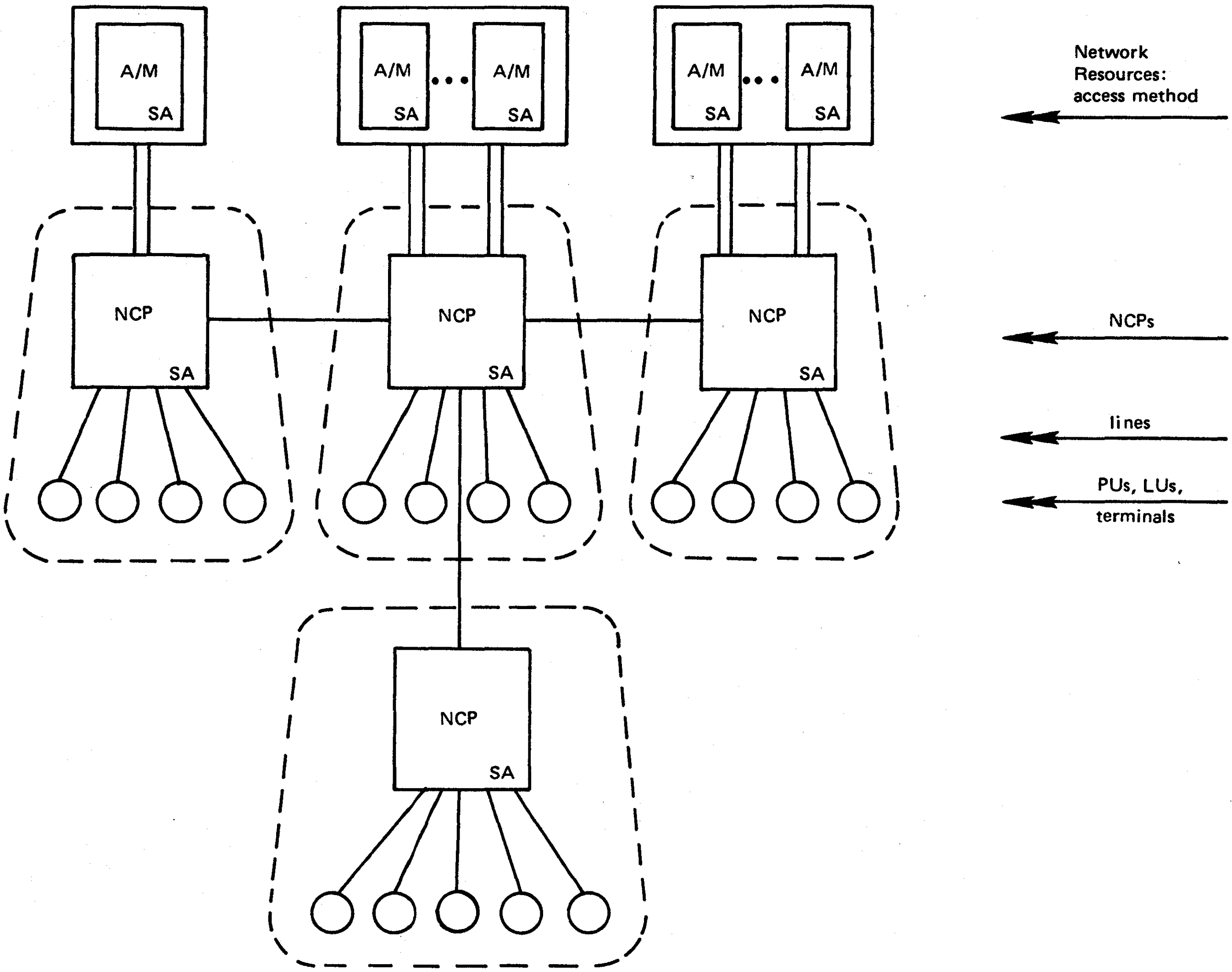


Figure 1-7. Network Resources

Each resource in a network must be assigned a symbolic name that identifies it to the subarea (NCP) that controls it. This name is specified in the name field of the NCP macro that represents the resource. The program generation procedure relates each resource name and its associated subarea address with a numeric value. This value is the *network address* of the resource. The access methods and network control programs in the network use these addresses to route messages to their destinations.

Resources are controlled at the access method level by a part of the SSCP (system services control point) that resides within the access method: the domain resource manager. A domain is the set of resources (links, physical units, logical units, and non-SDLC stations) associated with (owned by) an access method.

A single network may be divided into two or more domains, each of which is controlled by its own domain resource manager. Conversely, two or more independent networks may be combined, by the addition of appropriate communication links, into a single network. Each of the networks may constitute a domain in the new network; or, new associations may be formed between the combined resources and the access methods to yield new domains.

Some resources can be "owned" by more than one domain resource manager; such resources therefore belong to more than one domain. Other resources must be owned by a single domain resource manager and therefore cannot be shared between domains.

In using this publication, it is important to be aware that you establish multiple domains and arrange for shared ownership of network resources at the access method level. See the *ACF/VTAM* and/or *ACF/TCAM Installation* manuals listed in the Preface for information on these subjects.

Communication between Controller and Host Processor

The network control or emulation mode interacts with one or more access methods executing in the host processor or in each of several host processors. The access method and any associated application programs must be designed to interact with the control program in the manner appropriate to the mode in which the lines served by the access method are to operate—that is, in network control mode or emulation mode, or both.

Communication in network control mode between the NCP and the access method generally consists of an exchange of requests issued by the access method and responses returned by the NCP. Each request and each response contains the control information necessary to identify the network resource to which it applies, the operation required (for example, read or write), and status information pertaining to that operation. Requests and responses also contain the text of messages to be passed between the access method and the network. Some responses from the network control program are unsolicited; that is, not returned in response to a request. Unsolicited responses report error conditions and status information that may develop during operation of the controller.

All requests and all responses for lines operating in network control mode pass between controller and host processor over the network control subchannel. This subchannel is represented by a subchannel address of the host channel to which the controller is attached. There is always one network control subchannel regardless of how many lines are operated in network control mode. If the controller is to perform only network control functions, this is the only subchannel required. Over the network control subchannel also pass load module data sent by the access method loader or independent loader and the contents of controller storage during the dumping process.

Communication in emulation mode between the access method and the NCP is essentially the same as between the access method and the transmission control unit being emulated. As is the case for transmission control units, each communication line operated in emulation mode requires its own host subchannel address. The subchannels associated with lines operated in this mode are called emulation subchannels. In defining a program that performs emulation functions, you associate each communication line with an emulation subchannel address. Use of the multi-subchannel line access (MSLA) feature of the program allows more than one subchannel address to be associated with a line, but communication is possible over only one subchannel at a time. The converse, however, is not true: more than one line cannot be associated with a single subchannel address. (An exception is a program option for emulation of a 2701 that is equivalent to the dual communications interface feature of the 2701. This option allows either of two lines to be associated alternately with a single subchannel.)

For either the MSLA or dual communications interface feature, the line with which a subchannel is to communicate is established by command from the access method. Selection is possible only among the line/subchannel associations established when the program was defined. The source program must be modified and reassembled to change these associations.

A program that performs both network control and emulation functions requires (1) a single network control subchannel for each channel adapter that the NCP is to operate and (2) one or more emulation subchannels for each line to be operated in emulation mode (even if such a line will sometimes operate in network control mode).

Defining the Network Control Program

Achieving an operating network control program is a three-step process; (1) defining the program, (2) generating the program, and (3) loading the program. The first step, defining the program, is the most involved. Many different variables and options must be considered in preparing a program that meets the requirements of a particular network configuration and application.

A network control program is defined in the form of a source program consisting entirely of macro instructions called *control program generation macro instructions*. These include configuration macros for specifying the elements of the network and are similar to those used in some data communication access methods (for example, the LINE and TERMINAL macros). The source program, when punched into cards and preceded by the appropriate job control statements, forms the input to the generation procedure.

Chapter 2 describes characteristics of the network to be considered when defining a program that is to perform network control functions for SDLC stations only. Chapter 3 describes additional functions that must be considered when BSC or start-stop stations are included.

Once you understand the characteristics and functions explained in Chapters 2 and 3, you should be able to determine the requirements and options best suited for your network control program. Chapter 5 describes the macro instructions needed to define the control program.

Generating the Network Control Program

After the network control program is defined in the form of a source program containing control program generation macro instructions, it is ready to be generated. Generating the program is a compilation procedure consisting of assembly and link-editing steps. The procedure may be executed in the host processor or in any other central processing unit that (1) can fulfill the operating system assembly and link-editing requirements and (2) has access to the IBM-supplied NCP module libraries. These libraries, supplied by the IBM Program Information Department, must be added to the operating system before any network control programs can be generated. Documentation provided with the libraries explains how to add them to the operating system.

The primary output of the generation procedure is a control program load module, ready for loading into the communications controller. Chapter 4 describes the generation procedure under OS/VS and VSE.

VTAM Initialization Process

The VTAM initialization process requires as input the same control program source statements used for NCP generation, supplemented by several other source statements meaningful only to VTAM. These VTAM-only statements, which consist of separate macros and additional operands of existing NCP macros, may be placed in the network control program source deck either before or after the NCP is generated. Placing the VTAM-only statements in the deck before generating the network control program is recommended. Adding them to the deck after generating the program can introduce inadvertent errors such as misspelled operands or transposed cards. This causes the information given to the VTAM initialization procedure to differ from the generated NCP.

Chapter 5 lists these VTAM-only macros and operands, and indicates where they must appear in the NCP source deck used for VTAM initialization. However, their use is not explained; the *ACF/VTAM Installation* manual tells how to use these VTAM-only source statements. The program generation procedure checks only the keyword part of VTAM-only operands for proper spelling. The procedure does not check the accuracy of the parameters specified and does not verify the appropriateness of the operands coded.

Caution: Because the VTAM initialization does no validity checking of NCP parameters, it is imperative that the NCP source statements be entirely free of errors before being given to the VTAM initialization procedure. Therefore, before VTAM initialization, the network control program must be assembled, via stage 1 of the generation procedure, and reassembled if necessary until the stage 1 output listing shows no MNOTE statements having severity codes of 4 or 8.

Loading the Network Control Program

The final step in achieving an operating network control program is loading the program load module into the communications controller. For a local NCP, this loading procedure requires that a loader utility program be executed in the host processor, with the controller on-line to the processor. For a remote NCP, loading requires that a loader utility program be executed in the host processor and that a network control program be executed in the local controller to which the remote unit is connected. Apart from transferring load module data between the host processor and the controller being loaded, the local NCP does not participate in the loading process.

The loader utility program executed in the host processor may be an access method (VTAM or TCAM) facility or an independent utility program provided by IBM as part of the system support programs. The independent utility, use of which is explained in the *ACF/NCP Utilities* manual or the TCAM facility may be used only for loading a local controller. The VTAM loader facility *must* be used for loading a link-attached NCP or PEP and *may* be used for loading a channel-attached NCP or PEP. See the appropriate VTAM or TCAM publications, of those listed in the Preface, for information on the access method loading facility.

Obtaining the Contents of Controller Storage

A utility called the dump program allows a selected portion or all of the contents of the controller storage to be transferred from the controller to the host processor, which then prints the contents in hexadecimal format. The dump program, like the loader program, may be an access method facility or an independent program supplied as part of the system support programs. Either dump program has two modules, one of which the host processor transfers to the controller before the dumping process begins. The two modules then interact to transfer the contents of controller storage to the host processor; the host processor module then formats and prints the storage contents.

Executing the dump utility stops operation of the NCP. After the dumping process is completed, a control program must be reloaded into the controller before operations can resume. An alternate means of obtaining the storage contents that does not require stopping the control program is the "Dynamic Dump Utility." This utility is available only in a network control program with the PEP extension.

Dynamic Dump Utility

The dynamic dump utility is an optional Emulation Program or PEP utility program that allows the contents of controller storage to be transferred from the controller to the host processor without interrupting operation of the control program. A full storage dump or a dump of the trace tables for lines in emulation mode can be obtained. In addition, portions of storage can be displayed on the operator's console at the host processor. The utility can also activate or deactivate the emulation mode line trace function, which allows the selection of two program levels to be traced.

Chapter 2. Functions for SDLC Resources

This chapter describes the functions of a network with SDLC resources that you must identify to tailor an NCP to your particular installation.

The chapter is divided into seven major sections. The first five define:

- Network characteristics
- Communication controller hardware configuration
- Data transfer between the communications controller and the host processor
- Communication between Network Control Programs
- Procedural options governing message traffic between the controller and the network

The remaining two major sections explain (1) the optional diagnostic and service aid facilities that may be included in the NCP, and (2) the program generation options and data sets (files) that the generation procedure uses in creating an NCP load module. The description of each function and option is not exhaustive; it is intended to provide sufficient information to enable you to select the appropriate parameters when coding the program generation macro instructions.

For many functions, especially those relating to the equipment configuration, the decisions about what to code in the macro instructions have been made by the *system designer.* (This is the individual who determines the data communication equipment, network configuration, and communication services that constitute your network.) You need only determine what these functions are and code the appropriate macros and operands accordingly.

Other functions relate to resource, such as the size of the buffers in the buffer pool, or to procedural options, such as pacing. These affect the message-handling capacity and throughput of the communication system and require careful consideration before specifying the parameters.

Once you are familiar with the functions that apply to your configuration and applications, you are ready to code the program generation macro instructions to define your network control program.

Network Characteristics for SDLC Resources

This section applies only to SDLC operation. See "Network Characteristics for BSC and Start-Stop Resources" in Chapter 3 for information on BSC and start-stop operation.

Physical and Logical Units

In this book, *SDLC station* refers to the devices the NCP can communicate with over an SDLC link, such as the IBM 3601 and 3650 Control Units (called cluster controllers), IBM 3767 and 3770-series terminals, and IBM 3705 Communications Controllers.

To the network control program each SDLC station appears as a *physical unit.* A physical unit is a specific set of defined functions performed by programming or hardware. Each physical unit is represented by a PU macro that identifies it as a type 1, type 2, or type 4 physical unit. These type designations are used instead of numeric machine types. The type designations reflect the degree of program function performed by the physical unit. Functionally equivalent physical units have the same type designation. For example, IBM 3270 Models 11 and 12 and 3767 SDLC terminals are both type 1 physical units.

Other parameters that may be associated with the physical unit are the station address, the subarea address assigned to the unit, and procedural options that govern communication between the network control program and the physical unit.

Associated with type 1 and type 2 physical units may be one or more *logical units*. A logical unit on a nonswitched SDLC link is represented by an LU macro. An LU macro creates a logical unit control block (LUB) that defines the logical unit in the NCP.

For a logical unit on a switched SDLC link, the NCP allocates control blocks from a pool of logical unit control blocks. The pool is defined by an LUPOOL or LUDRPOOL macro. In this case, the logical unit parameters are defined and maintained in the host rather than in the NCP. After establishing a session with an SDLC station, the host passes the appropriate logical unit parameters to the NCP, which inserts them in the logical unit control block (LUBs) allocated for that session. These LUBs represent the logical unit for the duration of the session. When the session ends, the NCP returns the LUBs to the LU pool for reuse in other sessions.

The decision as to whether to use the LUPOOL macro or the LUDRPOOL macro depends on the access methods with which the NCP will communicate. One LUPOOL macro is required for each access method that *does not* support the Request Network Address Assignment (RNAA) command and will communicate concurrently with type 1 or type 2 physical units over switched lines. If one or more access methods *does* support RNAA, one LUDPROOL macro can be coded to include all of the logical units in one pool.

Communication Line Characteristics

A *communication line* as used in this book includes the entire transmission link between a station and the communications controller, including modems, physical conductors, microwave links, satellite links, etc. Communication lines over which synchronous data link control procedures are used are called *SDLC links.*

SDLC links are further defined in two categories; subarea links and peripheral links. A *subarea link* is an SDLC link that physically connects two adjacent type 4 PUs (communication controllers). A *peripheral link* is an SDLC link that connects a type 4 PU to a type 1 or type 2 PU.

Parallel links are multiple subarea links operating concurrently between two adjacent subareas. They provide the capability of simultaneous data flow over more than one link and the flexibility of intermixing links that have different line characteristics. It also increases availability of a link, thereby improving the reliability of communication. The limitation on the total number of links is determined by the maximum number of physical links allowed by the controller hardware.

Transmission groups are used to simplify parallel link routing definitions in the network. A *transmission group* represents one connection or logical link between adjacent subareas. As shown in Figure 2-1, parallel links can consist of one or more transmission groups. Notice that a single host channel is considered a transmission group. A single link can also be considered a transmission group.

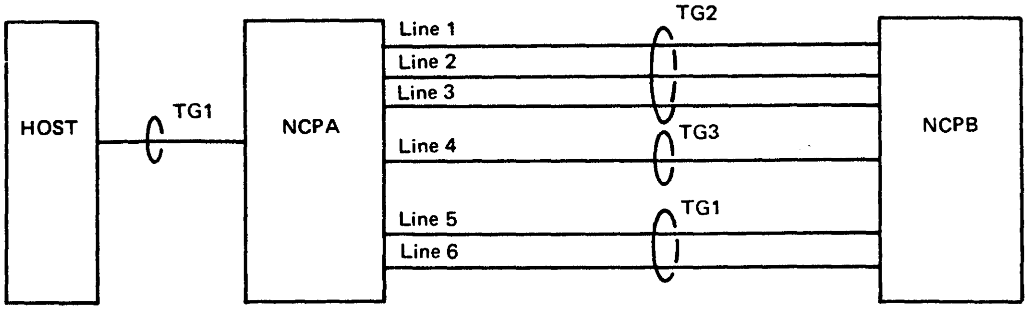


Figure 2-1. Transmission Groups

A transmission group (TG) is a common reference that can be defined consistently throughout the network for the purpose of defining network routing. Within a particular subarea, a TG must be defined for each possible path to an adjacent subarea.

A *transmission group number* (TGN) is associated with each TG for identification and to allow physical/logical independence of the lines and transmission groups. This number is associated with each link station (type 4 PU) that represents an adjacent subarea. The TGN is an arbitrary number used to express the relationship between this subarea and other subareas in the network. The transmission group number is specified in the TGN operand of the PU macros defining the adjacent subarea type 4 PUS.

A *link station* is a logical component in the NCP that represents a physical unit attached to the NCP by an SDLC link. The link station is generated for each PU by coding the PU macro.

Line characteristics refer to the functional attributes of the communication line (for example, whether the communication facility is half-duplex or duplex) and to related aspects of the line such as the physical address.

SDLC stations may communicate with a communications controller over a nonswitched point-to-point link, a nonswitched multipoint link, or a switched link. Each SDLC link must be represented within the network control program by a LINE macro which specifies certain characteristics of the link.

All SDLC links are treated as multipoint lines by the communications controller, regardless of the number of stations on the link. The controller contacts a specific station by sending a control character (physical address) that is recognized only by that station. The station receiving that character responds appropriately; the other stations ignore the transmission unit.

A service order table, defined by a SERVICE macro,is required for all non-switched SDLC peripheral links with more than one station on the link. Code a SERVICE macro directly following each LINE macro that represents an SDLC link.

Half-Duplex versus Duplex Links

The network control program must know whether a communication facility is half-duplex or duplex. This is specified in the DUPLEX operand of the LINE macro. This operand represents the characteristics of the entire communication path including common-carrier lines and equipment and the modems at both ends of the path. The operand does *not* specify the mode of data transfer over the line. It is important not to assume that a two-wire modem is necessarily a half-duplex modem. If the clear-to-send signal lead in the modem is continuously energized, the modem is duplex, regardless of whether it is a two-wire or four-wire modem. If in doubt, consult the installer or supplier of the modem.

Line Speeds and Clocking

The SPEED operand of each LINE macro specifies the data rate at which the line is to operate. This is the rate at which the station, controller, and modems are designed to transmit data over the link between the station and the controller.

If the modem that connects the line to the communications controller has two possible data rates, the DATRATE operand of the LINE macro specifies whether the line is to operate at the higher or lower of the two rates.

The CLOCKNG operand of the LINE macro, specifies whether internal (business-machine) clocking or external (modem) clocking is used for the communication line. Internal clocking is provided by the communication scanner that services the line. External clocking is provided by the modem, whether the modem is a separate unit or built into the controller.

Each communication scanner may have from one to four oscillators. The bit rates for each oscillator must be specified in the SPEED operand of the corresponding CSB macro.

Line Addresses

Each SDLC link attached to the communications controller is identified to the NCP by one or two physical line addresses. If a single line address is used for both transmitting and receiving over the SDLC link, specify that address in the ADDRESS operand of the LINE macro representing the link. If separate line addresses are used for transmitting and receiving, specify both addresses in the ADDRESS operand.

Modem New Sync Feature

Certain types of synchronous modems are equipped with a feature called new sync. This feature reduces the amount of line-turnaround time that is normally expended each time the direction of transmission on the line is reversed. The NEWSYNC operand of the LINE macro specifies whether this feature is to be used.

NEWSYNC=YES is valid only if the modem at the controller has the new sync feature *and* if the communications controller is the multipoint master station for a duplex line that uses multipoint line control.

Consult your IBM representative or the modem installer or supplier to determine whether the modem has the new sync feature.

NRZI versus NRZ Bit Stream Encoding

The network control program transmits data over an SDLC link in either non-return-to-zero (NRZ) mode or non-return-to-zero-inverted (NRZI) mode. The mode to be used is specified in the NRZI operand of the LINE macro that represents the SDLC link. The choice is determined by the type of modems serving the link.

The modems at each end of the link must maintain synchronism with each other for the entire duration of message transmission. Some modems require bit transitions (0 to 1 or 1 to 0) at intervals in the data stream in order to maintain synchronism. (Such modems are said to be sensitive to transitionless bit streams.) When operating in NRZI mode, the data terminal equipment at the ends of the link manipulate the bit stream transferred to the modems in such a way that transitions are introduced into the bit stream even when the message data being transmitted is transitionless. (Such binary sequences are likely to occur in messages containing storage dump data or IPL data being sent to programmable controllers or terminals.) The transitions thus introduced ensure that the modems remain synchronized. The terminal equipment that receives the altered bit stream reconverts it to its original form.

In NRZI mode, sequences of zeros (000000...) are converted to alternating ones and zeros (101010...), thus satisfying those modems sensitive to transitionless bit streams. Certain other modems, however, are sensitive to just this alternating pattern (101010...) rather than to transitionless bit streams; they will lose synchronism if subjected to 101010 patterns of sufficient length. When the SDLC link is equipped with this type of modems, the terminal equipment must operate in NRZ mode rather than in NRZI mode.

The rules for selecting NRZI versus NRZ operation are:

- If internal (business-machine) clocking is used on an SDLC link, NRZI operation is required; specify NRZI=YES in the LINE macro representing the link.
- If external (modem) clocking is used on the SDLC link, specify NRZI=YES *unless* the modems on the link are sensitive to repeated 101010 patterns, in which case specify NRZI=NO in the LINE macro. Consult your IBM representative or the modem supplier or installer to determine whether the modems are sensitive to repeated 101010 bit patterns.

Caution: All stations on the same SDLC link must use the same encoding scheme. Mixing of modes on the same SDLC link will result in total lack of communication between stations on the link. Where the stations on the SDLC link are remote 3705 controllers, the setting of the NRZI bit in the IPL configuration data set of the remote program loader must correspond to the program-specified option (NRZI=YES or NRZI=NO).

Automatic Network Shutdown

Part or all of the network attached to a communications controller and currently operating in network control mode is shut down automatically, in an orderly manner, under any of several conditions. (Any lines currently operating in emulation mode are unaffected by shutdown of lines in network control mode.) This procedure is called *automatic network shutdown* (ANS).

Automatic network shutdown is performed for network resources on behalf of the SSCP (system services control point) that currently owns the resources when the NCP loses its ability to communicate with that SSCP. The network control program may detect the loss of the SSCP if the SSCP is adjacent (residing in a host processor channel-connected to the NCP), or an adjacent

NCP in the path to the affected SSCP may notify the present network control program of the loss of the SSCP.

Automatic network shutdown occurs for all or any part of the network under the following conditions.

Network controlled by channel-attached NCP:

- An adjacent access method fails to respond to the NCP within a specified interval after the NCP has presented an attention signal to the channel by which it communicates with that access method. This interval is specified in the TIMEOUT operand of the HOST macro that represents the access method.
- An adjacent network control program notifies the present NCP that it has lost contact with a subarea in the network.
- The SSCP ends its session with the network control program (by sending a Deactivate PU command).
- The network control program receives, from an SSCP with which it is currently in session, an unexpected request to establish a session with that SSCP. Such receipt of a second request may indicate that communication between the SSCP and the NCP was interrupted (possibly with resultant loss of data) without awareness by the NCP that the interruption occurred. (This can happen if, at the moment of interruption, the NCP did not have an attention signal present on the channel.)
- A shutdown request is entered at the control panel of the communications controller.

Network controlled by link-attached NCP:

- The link-attached NCP detects a lapse in communication activity over the subarea link it is presently using to communicate with the adjacent network control program. The lapse may occur through outright failure of the link or through badly degraded performance of the link as indicated by exhaustion of error-recovery procedures performed by the adjacent NCP. The lapse interval is determined by the value you specify in the ACTIVTO operand of the GROUP macro representing the SDLC links to the channel-attached network control program. This interval must be sufficiently long for the adjacent NCP to complete its error-recovery procedures for the link.
- The SSCP ends its session with the network control program (by sending a Deactivate PU command).
- The network control program receives, from an SSCP with which it is currently in session, an unexpected request to establish a session with that SSCP. Such receipt of a second request may indicate that communication between the SSCP and the NCP was interrupted (possibly with resultant loss of data) without awareness by the NCP that the interruption occurred.
- The adjacent network control program, upon entering automatic network shutdown for the link to the this controller, signals the this NCP to shut down the network controlled by the owning SSCP.

The NCP takes the following action for each kind of line and station undergoing shutdown:

For SDLC links, the network control program:

- Dissociates the link from the owning SSCP with which communication has been lost.
- Disables the link, if it is a switched link, so that it cannot answer calls from stations.

- Cancels the line trace or online test operation if the link is currently being traced or is undergoing online testing.
- Breaks the switched connection, if any, to the station.

Dynamic Reconfiguration

Dynamic reconfiguration gives the host access method the ability to add and delete type 1 and type 2 physical units and logical units on a nonswitched line to the network configuration without going through an NCP generation. The dynamic reconfiguration function is included in the NCP by coding the PUDRPOOL macro in the program generation.

Note: Dynamic reconfiguration is only for the addition or deletion of physical and logical units on an existing link. Links cannot be added or deleted with this function.

The PUDRPOOL and the LUDRPOOL macros create pools of null physical and logical unit control blocks. When a PU or an LU is dynamically added to the network, a physical or logical unit control block is created in place of one of the dummy control blocks in the pool. The information in the control block is supplied by parameters entered through the access method at the time the unit is added. When a unit is deleted, the control block is returned to the pool for future use. Only one PUDRPOOL macro and one LUDRPOOL macro may be included in the NCP generation. The LUDRPOOL macro creates two logical unit control block pools: one is for logical units added to a type 1 PU and the other is for logical units added to a type 2 PU.

If a generation-defined resource is deleted, in order for that resource to be reused, the assigned network address must be replaced by a network address from the resource vector table (RVT) extension. The RESOEXT operand of the BUILD macro defines the size of the RVT extension and should specify the anticipated number of generation-defined resources that may be deleted and then added at a later time via dynamic reconfiguration. If the RVT extension is depleted, generation-defined resources can still be deleted but not reused. However, unlike the addresses defined during the NCP generation, the addresses in the RVT extension can be reused when they are deleted.

The MAXPU operand of the LINE macro and the MAXLU operand of the PU macro limit the number of PUs and LUs that can be added to a particular line. These two operands must reflect more than the generated number of physical or logical units if dynamic reconfiguration is to be used.

Physical and logical units that were defined in the NCP generation can be deleted with the dynamic reconfiguration function if the PUDR or LUDR operand was coded YES in their respective PU or LU macro. The dynamic reconfiguration function will not delete any unit unless PUDR=YES or LUDR=YES is coded.

When a physical unit is added to a line, an entry is placed in the service order table for that line. Therefore, the MAXLIST operand of the SERVICE macro must reflect a number greater than the number of entries in the ORDER operand. The difference between these two numbers is the number of physical units that can be added to that line.

The IBM 3270 Models 11 and 12 SDLC terminals need additional programming support over other SDLC terminals. When these 3270 terminals are not included in the initial NCP generation, but it is anticipated that they will be added through dynamic reconfiguration, you must code the DR3270 operand. This operand of the BUILD macro ensures that the proper 3270 support is generated into the NCP. If 3270s are included in the network when the program is generated, no additional support is necessary for dynamic reconfiguration, and the DR3270 operand is ignored.

The default for DR3270 is YES; therefore, when dynamic reconfiguration is specified and no 3270s are anticipated, this operand should be coded NO.

To dynamically reconfigure the NCP, the access method must define a dynamic reconfiguration data set (DRDS) consisting of ADD statements, DELETE statements, or both, and their associated PU and LU macros. The reconfiguration information in this DRDS is then used to modify NCP control blocks to reflect the changed configuration. Each time the NCP configuration is changed, a new DRDS must be defined that contains only the current modifications. For example, do not add and delete the same PU or LU to the network in one DRDS. If a device that was added through dynamic reconfiguration is to be deleted, define a new DRDS to delete it; do not put the DELETE statement in the same DRDS as the ADD statement.

The ADD statement directs the access method to ADD a PU or LU to an existing link or PU, respectively. The format of this statement is:

```
        ADD        TO=symbol
```

where *symbol* is the name of the link or PU defined in the NCP generation that the new device will be added to.

The DELETE statement tells the access method to delete a PU or LU from an existing link or PU. The format of this statement is:

```
        DELETE     FROM=symbol
```

where *symbol* is the name of the link or PU that the device will be deleted from. If a PU is deleted from a link, all LUs associated with that PU are also deleted.

If an ADD or DELETE statement specifies a link in the TO or FROM operand, it must be followed by a PU macro. The PU macro may then be followed by one or more LU macros.

If the ADD or DELETE statement specifies a PU in the TO or FROM operand, it must be followed by one or more LU macros.

For example, if PU6 and its associated LUs (LU1 and LU2) are to be physically moved from LINE6 to LINE7, the following statements must be placed in the DRDS:

```
name    DELETE      FROM=LINE6
PU6     PU          operands (optional)
name    ADD         TO=LINE7
PU6     PU          operands
LU1     LU          operands
LU2     LU          operands
```

As another example, if only one LU (LU1) is to be moved from PU3 to PU4, the DDRS must contain:

```
name    DELETE    FROM=PU3
LU1     LU        operands (optional)
name    ADD       TO=PU4
LU1     LU        operands
```

The *name* on the ADD and DELETE statement is optional.

When deleting a PU or LU from the configuration, only the *symbol* on the respective PU or LU macro needs to be coded. If any operands are coded, they will be ignored.

When adding a physical unit to a link, the PU macro must be coded as specified in the macro definition in Chapter 5. However, for dynamic reconfiguration, the DATMODE, MAXLU, and SUBAREA operands are ignored. Also, if the MAXDATA operand is omitted, it defaults to 266; and the RETRIES operand is always set to 7 for both the *t* and *n* parameters.

When adding a logical unit to a physical unit, the LU macro must be coded as specified in the macro definition in Chapter 5. The only exception is that the *m* parameter of the PACING operand must be 1; the *n* parameter remains as coded.

In defining the dynamic reconfiguration data set for an ACF/VTAM environment, the first statement must be:

```
VBUILD    TYPE=DR
```

This statement must precede the ADD and DELETE statements. ACF/VTAM stores the DRDS in the system definition library.

For ACF/TCAM, the DRDS must be assembled against the NCP macro library to create the delta resource resolution table (DRRT). The DRRT is then link-edited into the ACF/TCAM partitioned data set that contains the resource resolution tables for NCP load modules.

For more information on defining and using the DRDS with ACF/TCAM, see the *ACF/TCAM Installation Guide,* or for ACF/VTAM, see the *ACF/VTAM Installation* manual.

Communications Controller Hardware Configuration

Several characteristics that must be identified to the NCP reflect the system designer's choice of hardware options for the communication controller. These are (1) the size of storage installed in the controller; (2) the type and number of channel adapters that join the communication controller to one or more host processors; (3) the type, number, and oscillator bit rates of the communication scanners installed; and (4) the interrupt priority to be used for each line serviced by a scanner.

Specify the storage size in the MEMSIZE operand of the BUILD macro. Also specify the type and number of channel adapters in the CA operand of the BUILD macro.

A communication controller can be equipped with from one to four communication scanners. The IBM 3705 models A1, A2, and E1-E8 always have a single scanner. Models B1-B4 can have one or two scanners; models C1-C6, up to three scanners; and models D1-D8, up to four scanners. Models F1-F8 of the 3705-II have one or two scanners; models G1-G8 have three scanners; and models H1-H8 have four scanners. The number of lines serviced by each scanner depends upon the data rates (line speeds) at which the lines operate. Each scanner may be equipped with from one to four oscillators or internal clocks and can therefore provide internal clocking for up to four different speeds of lines. In addition, the scanner may service lines for which external modems (including integrated modems within the 3705) are used, without restriction as to the number of different external clock speeds used for those lines. To service a line that is externally clocked, however, a scanner must be equipped with an oscillator that operates at less than one-half of the data rate of that line. (This oscillator may be the same one that furnishes clocking for one or more of the internally clocked lines.) A scanner equipped with 600 bps and 1200 bps oscillators, for example, could service lines operating at these speeds, using *internal* clocking, and also service lines using *external* clocking at speeds exceeding 1200 bps; for instance, 2000 and 7200 bps. This scanner could not, however, service externally clocked lines of 1200 bps or less because, in this example, there is no oscillator that operates at less than one-half of 1200 bps.

For each scanner, you must specify to the network control program (1) the type of scanner, (2) the machine module in which it is installed, and (3) the bit rates of the oscillators with which each scanner is equipped. This information, like the storage size and channel information, should be obtained from the system designer before you code the program generation macro instructions. Specify the details of the scanners in the TYPE, MOD, and SPEED operands of a CSB macro—one macro for each scanner in the controller.

The NCP is interrupted by the line interface hardware of the controller each time a data character or a data buffer (depending on the type of scanner) is to be sent over or received from a communication line. To avoid character overrun or underrun, lines having a high data rate require service from the program more frequently than lines having lower data rates. Each line serviced by a given communication scanner is therefore assigned an interrupt priority relative to other lines serviced by the same scanner. If all lines on the scanner have the same data rate, the priority may be equal. If the lines have differing rates, however, those with high rates should be assigned higher priority than those with lower rates.

The priority may be 0, 1, 2, or 3 (3 is the highest priority). These priority values are specified in the INTPRI operand of the LINE macro.

Appendix E gives a method for determining the interrupt priority for each line in the network.

Communication Between Controller and Host Processor

Information on both the buffers within the access method and buffers within the NCP must be specified in the HOST macro to properly coordinate data transfers to and from the access method. Each access method with which the network control program concurrently communicates is represented by a separate HOST macro.

Data Transfer from Host Processor to Controller

The amount of data conveyed from the host processor to the communications controller during a single data transfer over the network control subchannel may vary over a wide range, depending on the number of requests and the amount of accompanying message data to be transferred. Efficient operation of the NCP requires that the program preallocate a suitable number of buffers for incoming data transfers, rather than allot buffers one at a time. Once the set of buffers is allocated, data transfer from the access method can proceed without further attention by the network control program's supervisory routine until the data transfer ends or all the preallocated buffers are filled. If the amount of data received during one transfer is insufficient to fill all of the preallocated buffers, the remaining buffers are used for subsequent data transfers until all are filled, at which point the program again allocates the same number of buffers.

The INBFRS operand of the HOST macro associated with an access method specifies the number of buffers the NCP is to allocate for data transfers over the network control subchannel from that access method. You should consider two factors when estimating a value for INBFRS.

If the size of a data transfer consistently exceeds the preallocated buffer space, the NCP supervisory routine is frequently interrupted to provide more buffers for the excess data. The time the program must spend in processing the interrupts reduces the time it can devote to servicing communication lines.

On the other hand, preallocating an excessive quantity of buffers for receiving messages from the access method may deplete the buffer pool to the point that insufficient buffers are available for receiving messages over the communication lines. Buffer depletion is especially likely when (1) the buffer pool is relatively small and (2) a low message rate over the channel from the access method causes the preallocated buffers to be filled slowly, thus unduly delaying return of these buffers to the pool.

In choosing a value for INBFRS, a reasonable balance between degraded program efficiency and unnecessary over-allocation of buffers must be achieved.

Data Transfer from Controller to Host Processor

There is a limit to the amount of data the access method can receive from the NCP during a single data transfer over the network control subchannel. This limit must be specified when defining the NCP so that the program does not attempt to send more data than the access method can accept. You specify this limit with the MAXBFRU and UNITSZ operands of the HOST macro. MAXBFRU designates the number of buffer units the access method allocates for a data transfer, and UNITSZ indicates the size of each unit in bytes. The size of the buffer used to receive a data transfer is the product of the two values. (A buffer unit is the smallest amount of contiguous storage area handled as buffer space; a buffer may consist of one or more units.)

In sending a series of response (or request) blocks to the host processor, the NCP causes the access method to begin receiving each successive block in a new buffer.

In some applications, the access method inserts prefixes in buffers ahead of the message data. An NCP option allows each new block sent to the host processor to be offset from the beginning of the access method buffer by enough space to allow the access method to insert the prefix. The amount of offset is specified in the BFRPAD operand of the HOST macro. OS/VS VTAM requires 28 bytes for the buffer pads; VSE VTAM requires 15; ACF/VTAM requires 0; OS/VS TCAM requires a minimum of 17 bytes; ACF/TCAM requires 17-28 bytes.

Communication Between Network Control Programs

An NCP can have up to eight routes for sessions between any two subareas. These routes provide multiple and alternate paths between the subareas and can be used simultaneously by different sessions. There are two types of routes for NCP session control; explicit routes and virtual routes.

At the physical level, an *explicit route* is a sequence of subareas and transmission groups by which two subareas communicate. Data can flow over an explicit route in either direction. Explicit routes are defined by the PATH macros within each NCP.

A *virtual route* is a logical connection between a pair of subareas. Virtual route definition takes place in and is the responsibility of the host access method. Each virtual route in the network is assigned to an explicit route and takes on the physical characteristics of that route. Many virtual routes may be assigned to the same explicit route.

Each virtual route in a network is assigned one of three route transmission priorities. If many virtual routes share the same explicit route, messages flowing on virtual routes having a higher transmission priority will be transmitted ahead of messages flowing on virtual routes having a lower transmission priority.

Explicit Routes

An explicit route is an ordered set of network elements that a PIU travels in the network from the origin subarea to the destination subarea. Therefore, an explicit route is a particular chain of transmission groups and subareas.

Up to eight explicit routes can be defined between any two subareas. These explicit routes are defined by one or more PATH macros and assigned an explicit route number from ER0 through ER7. Data to be transmitted from one subarea to another is assigned an ER number at the origin subarea. This ER number must be the same throughout the entire path between the subareas. That is, all intermediate subareas must also define that same ER number. A return path must also be defined between the two subareas. The ER numbers for the two paths do not have to be the same, but they must pass through the same set of transmission groups for the route to be valid. This arrangement makes possible relatively simple procedures for dealing with a route failure caused by a transmission group failure. There must always be a return pathway from the point of failure so that notification of the failure can be sent to the origin.

Alternate Routing

Multiple explicit routes between subareas allow the assignment of an alternate virtual route to sessions when an operational virtual route is lost. A virtual route may be lost because one or more of the physical elements in the explicit route to which it is assigned failed or was deactivated. In this event, all active sessions using the virtual route are terminated and appropriate session partners are notified. If an alternate virtual route between the same two subareas is active, SSCPs and LUs can reinitiate the interrupted sessions over the alternate route.

Configurable Stations

To increase the flexibility of the entire network, an NCP that is link-attached to another NCP is defined as a configurable station. This means that at the time the NCP is generated, it is neither a primary or a secondary station on the subarea link. The primary and secondary ends of the link are determined dynamically during the contact procedure. The NCP with the highest subarea number becomes the primary station.

To define a subarea link, you must code the SDLCST operand in the GROUP macro for the link with the names (symbols) of two SDLCST macros. One of the SDLCST macros is coded with the MODE=PRIMARY operand for operation when the station is configured as primary and the other SDLCST macro specifies MODE=SECONDARY for secondary station operation.

Procedural Options

Several procedural options characterize the operation of SDLC links. These options include (1) the manner in which the NCP starts up and shuts down the network, (2) the amount of data to be transferred at one time between stations and the controller, and (3) the amount of data to be accumulated from a station before passing the data to the access method. By careful selection of these options you can customize a network control program to best meet the requirements of your data communication applications.

Some procedural options require no more than a simple yes/no choice as to whether the option is to be included. Other options require you to choose from a range of values, such as the size of NCP buffers or the maximum amount of data to be transferred at one time between the NCP and stations in the network.

Buffer Size

The NCP contains one buffer pool of fixed-size buffers. Buffers from this pool are used for all message data. Specify the size of the buffers to be in the BFRS operand of the BUILD macro. The minimum is 72 (76 if on-line testing is supported); the maximum is 248. The buffer size is always a multiple of 4 bytes.

Buffer initialization occurs immediately after the NCP is loaded into the communications controller. During initialization, the NCP formats buffers in all controller storage remaining after the program is loaded. You can determine the number of buffers in the pool by dividing the buffer area by the buffer size plus 12 bytes (for buffer chaining fields).

Path Information Units

The basic unit of transmission in the network is the *path information unit* (PIU), which consists of network control and routing information and accompanying message text (optional). A PIU either requests a particular data communication operation (request PIU) or indicates the result of an operation (response PIU). Path information units associated with SDLC physical units can originate at either the host processor or the physical unit.

Under access method control, the NCP establishes physical and logical connections between the access method and the SDLC stations. While a connection is established, the NCP automatically controls the operation of each SDLC link in response to the data transfer and requested control operations.

Several parameters govern the amount of data the network control program sends to a station on an SDLC link.

The size of the buffer within an SDLC physical unit must exceed by at least 5 bytes (for a type 1 physical unit) or 9 bytes (for a type 2 physical unit) the size of a NCP buffer. In the MAXDATA operand of the PU macro representing a physical unit (type 1 or 2), you specify the maximum amount of data, in bytes, that the physical unit can accept.

The MAXOUT and PASSLIM operands of the LINE macro determine how PIUs are sent to an SDLC station. The MAXOUT operand allows you to specify how many PIUs or PIU segments (up to seven) can be outstanding at any given time (that is, the number that can be sent to the SDLC station before an acknowledgment is required from the station).

The maximum number of PIUs or PIU segments that can be outstanding is seven because of the sequence-numbering scheme used in SDLC to identify PIUs. Unless you specify a different value in the MAXOUT operand of the PU macro that represents the station, only one PIU is sent to each station before an acknowledgment is required. The higher the value of this parameter, the greater the utilization of the link. However, a high maximum-outstanding value also results in more PIU retransmission when an error occurs. Thus for links on which high error rates are experienced, you may wish to select a maximum-outstanding value lower than that used for links having lower error rates.

You may wish to cause some stations on an SDLC link to be serviced more frequently than others by representing them more times in the service order table for the link. By thus allowing more opportunities for service with a particular station, relative to others, the total amount of data transferred to the station may be similarly increased. You may counter this effect, however, by using the PASSLIM operand to restrict the amount of data (number of PIUs) exchanged with the station for any one appearance of the station in the service order table.

Only one PIU per pass is sent or received unless you specify a larger value in the PASSLIM operand of the PU macro representing the station.

Caution: When choosing values for MAXOUT and PASSLIM, observe any restrictions imposed on these operands by specific types of SDLC stations. See the appropriate programming publications for the types of stations used in your network for such restrictions.

Pacing

The outbound pacing option limits the number of path information units sent to a logical unit before an acknowledgment is required. This option can be used to prevent needless transmission of PIUs to a logical unit that is momentarily unable to accept them. The PACING operand of the LU macro allows you to specify how many PIUs the NCP can send in each transmission.

When the specified number of PIUs has been transmitted, the NCP waits for a pacing response from the LU before sending any more PIUs. See Appendix G for an example of pacing.

A similar function, called inbound pacing, applies to message flow from the NCP to the access method. This function is controlled by the access method and does not require any parameters in the NCP source statements.

The NCP also allows primary and secondary LUs to negotiate certain session parameters through the Bind session command. Therefore, in some cases, pacing parameters may be changed for a particular session in order to better utilize the communication link. At the end of these negotiated sessions, all values return to those specified in the generation parameters.

When choosing a value for the PACING operand, restrictions may be imposed on this operand by specific types of SDLC stations. See the appropriate programming publications for pacing restrictions for the types of stations used in your network.

Half-Duplex versus Duplex Data Mode

Subarea links may operate in either half-duplex or duplex data mode. If the SDLC link has two paths, as indicated by use of separate transmit and receive addresses, simultaneous sending and receiving on the link is possible. This is called operation in duplex data mode. If, on the other hand, the same address is used for both transmitting and receiving, these two functions must alternate (half-duplex data mode). If separate transmitting and receiving paths are available, the channel-attached NCP will operate the subarea link to an adjacent network control program in duplex data mode, unless you specify DATMODE=HALF in the PU macro representing the link-attached NCP.

Network Slowdown

The NCP can receive message data from the access method and from the SDLC links only as long as it has buffers available for the data. The program normally receives and sends data at the same average rate, although momentary overloads can occur. When the program receives more data than it sends during a given time interval, it can exhaust its supply of buffers. To prevent this, the network control program continuously monitors its supply of buffers and, when the supply falls to a specified level, automatically enters *slowdown mode.* The level is specified as a percentage of the total number of buffers in the program.

When in slowdown mode, the program reduces the amount of data it receives from SDLC links and from the access methods but continues to send at the normal rate. Since the rate at which buffers are released exceeds the rate at which new buffers are obtained for receiving data, the result is a net gain in the number of available buffers. When the buffer supply is sufficiently replenished, the program automatically resumes normal operation.

The SLODOWN operand of the BUILD macro allows you to specify 12, 25, or 50 percent as the minimum percentage of available buffers. When this percentage is reached, program enters slowdown mode. However, during initialization, the NCP may dynamically change the percentage you specify if the following requirements cannot be met. (1) The number of buffers at slowdown must equal or exceed the value of the CWALL operand plus 2, and (2) the buffer pool minus the number of slowdown buffers must equal or exceed 30.

Monitor Mode

Monitor mode consists of SDLC Monitor Mode and Channel Monitor Mode. SDLC monitor mode provides the capability for an NCP to attempt to contact adjacent NCPs without the aid of an SSCP or operator. Subarea links between two NCPs can be activated automatically if both ends of each link are defined to use monitor mode in the LINE or GROUP macro. SDLC monitor mode is invoked when the NCP is initialized and when all SSCP sessions to the NCP are inactive. The monitor mode issues the commands necessary to establish a session with the NCPs physical unit. Termination of the SDLC monitor mode takes place when the NCP receives an Activate Physical Unit command from an external SSCP.

The SDLC monitor mode function is included on an individual line basis by specifying the MONLINK operand on the LINE macro defining the link to be monitored.

Channel monitor mode allows the NCP to monitor any type 1 or type 4 channel adapter for an enabled condition after the channel adapter has been initialized. This function provides the capability for a channel-attached NCP to accept a Contact from a host access method and allows a nondisruptive host reconnection to an active NCP. Channel monitor mode is terminated when all of the generation-defined active channel adapters have been enabled.

Data Flow Control

The primary purpose of data flow control is to control the amount of data entering the network, thereby protecting the network resources from overload and deadlock situations. An operand (AVGPB) in the PU macro is used to inform the NCP of the average number of bytes each SDLC station is expected to send. (The AVGPB operand on the LINE macro is used for BSC and start-stop stations.) Then, before polling a particular station, the NCP can determine if it has enough buffer space available to handle the incoming data. If buffer space is not sufficient, the NCP will delay the poll until it can ensure that all the data from the station can be handled without overrun.

The CWALL operand on the BUILD macro is used to specify a specific number of buffers to be reserved in the NCP to complete the movement of data out of the controller. When the available buffers reach this threshold, the NCP stops input channel and subarea link operations and will not accept any more PIUs. Operations that send PIUs out of the NCP continue. When the number of buffers raises above the threshold, the input channel is again reopened and data transfer proceeds normally in both directions.

Error Recovery and Recording

Transient noise on the communications line and intermittent hardware malfunctions are the most common causes of errors affecting communication networks. By implementing the appropriate error recovery procedures, most of these errors can be recovered from and go unnoticed. When an error is detected in received data, the NCP signals the station to retransmit the data. On the other hand, when the station detects an error, it informs the NCP and the NCP then retransmits data to the station.

The maximum number of retransmissions may be specified for each SDLC station in the network. If error-free transmission is not achieved before the retransmission limit is reached, the network control program indicates the fact in its response to the access method.

If the error is the kind that inherently cannot be recovered from (such as a modem error), the NCP does not attempt error recovery. In this case, the NCP records the error status and transmits the indication to the host processor.

The number of error recovery attempts for errors affecting message data is determined by the RETRIES operand of the LINE and PU macros representing the SDLC link and station. For errors occurring when receiving from a station, there can be one retry sequence. For errors occurring when transmitting to a station, there can be one or more sequences of retry attempts, with a pause between successive sequences. Specifying a pause of several seconds between sequences allows time for transient noise conditions on the link, which may be responsible for the repeated errors, to subside.

The number of sequences and the pause are specified in either the PU macro for the station or the LINE macro for the SDLC link. The number of retries per sequence is specified in the LINE macro.

Normally error recording is done only for permanent errors. The initial error status causing the error recovery to begin and the ending error status is recorded for transmission to the host. Operator commands can be issued to cause the NCP to implement *intensive-mode* error recording. This mode records all temporary errors as they occur. The expanded statistics made available by this function will often preclude the need to run specific link tests and thus provide more efficient, timely problem determination.

Note: Intensive-mode error recording puts an added burden on the buffer pool and should be used with caution.

Diagnostic and Service Aids

The network control program diagnoses difficulties in network operations by means of several diagnostic and service aids. These aids are useful in identifying malfunctions within the network and the NCP. Some aids are standard (always present in the program); others are optional. Inclusion of all service aids is recommended.

Online Link Testing

Online link testing is a diagnostic aid by which a terminal or console may request a variety of tests to be performed upon a communication link. The terminal operator requests the test by entering a test-request message having a defined format. The requested test is performed, and the results are printed at the terminal or console. This diagnostic aid, important in problem determination and online maintenance of communication links, is included in a NCP unless you exclude it via the OLT operand of the BUILD macro.

The NCP is only an intermediary on online test operations. It recognizes test-request messages, routes them to the access method, recognizes interpretive commands from the host processor, and executes data communication operations accordingly. Recognizing the message as a test request, the program sends it unchanged to the access method.

The access method detects that the message requests the online test function and interprets the parameters within the message to determine the kind of test to perform. The access method then selects the appropriate test modules and sends a series of interpretive commands to the NCP that indicate what data communication operations to perform. The NCP executes these operations and returns responses as necessary to the access method. Upon analyzing the responses, the access method determines what further operations to perform and sends the NCP the appropriate interpretive commands.

Online link test operations require buffer space to hold the interpretive commands and an online test control block. These buffers, which the program obtains from the same pool as for normal operations, are required only for the duration of the test operation.

The link test does not require disruption of traffic involving any PU but the one participating in the test. Further, more than one link test can be run at the same time, each involving a different PU on the link. Once each link test is completed, the result of the test is sent to the requestor of the test.

Link testing is important whenever degradation or failure of a communication line is suspected. The nondisruptive nature of the enhanced capability means that link test will more likely be run as soon as the problem is suspected rather than be deferred until normal operations are at a minimum.

Note: Diagnostic programs that communicate with the network control program via VTAM or TCAM (for example, TOTE) may impose restrictions on the NCP buffer size (specified in the BFRS operand of the BUILD macro). See the appropriate manuals for such diagnostic programs for restrictions that may apply.

The NCP can execute online testing operations concurrently for any number of communication lines operating in network control mode. Normal message traffic on lines not undergoing testing can continue as usual.

Address Trace

Address trace is a service aid that records the contents of selected areas of controller storage and controller external registers when an interrupt occurs. Certain types of interrupts, or all interrupts, are designated to control the trace. The NCP records the trace data in a trace table within controller storage. When the desired data has been recorded, the contents of the trace table can be displayed on the 3705 control panel. The contents of controller storage can be transferred to the host processor by the dump program and the trace table examined in the listing of the dump.

The TRACE operand of the BUILD macro specifies whether the address trace option is to be included in the NCP and specifies the size of the trace table. The address trace function is performed only on lines operating in network control mode.

Line Trace

The line trace is a service aid that permits detailed analysis of the operation of any communication line operating in network control mode. (All SDLC links operate only in network control mode.) This facility records operating parameters of a line each time a level 2 interrupt occurs for that line. The trace information is placed in buffers obtained as required from the buffer pool and is transmitted to the host processor. The host processor should accumulate these line trace records in a data set (file) to be printed for analysis.

A line trace can be initiated or ended at any time by a request from the host processor as long as the line is in network control mode.

The line trace activity does not interfere with normal operation of the communication line. Performance may diminish somewhat because of the additional processing needed each time a character-service or buffer-service interrupt occurs for the line being traced. The amount of decrease in performance depends upon how heavily the communications controller is currently loaded. Buffer requirements are increased times three for lines on a type 3 scanner and times five for a line on a type 2 scanner. The line trace facility has no effect on performance or the number of buffers used, except when a line is actually being traced.

The NCP accepts and fulfills requests for tracing up to eight lines concurrently, as specified by the LTRACE operand of the BUILD macro. Any Activate Line Trace requests received when the specified limit has been reached are rejected. Each leg of a duplex line may be traced, but the traces are independent of each other.

The line trace facility for SDLC links is always present in the network control program.

Transmission Group PIU Trace

The transmission group PIU trace allows SNA dependent data such as SNA headers, SNA requests, and SNA responses to be traced as they enter and exit the NCP on subarea links within a transmission group. The trace is activated and deactivated by a request from the host to one of the links within the transmission group to be traced. When activated, the trace records all PIUs on all the links in the transmission group.

Channel Adapter Trace

Channel adapter trace is an optional service aid that stores in a trace table certain information about the channel adapters. Any combination of up to four channels can be traced.

An entry is placed in the table for each level 3 interrupt. After the last entry in the table is used, succeeding entries overlay previous entries, beginning with the first.

The CATRACE operand of the BUILD macro specifies whether the channel adapter trace option is to be included in the NCP, and specifies the size of the trace table. This trace can be activated or deactivated from the communications controller control panel. The channel adapter trace facility does not interfere with normal operation of the controller. Performance may diminish somewhat because of the additional processing needed. The amount of decrease in performance depends on how heavily the controller is currently loaded.

Abnormal End (ABEND)

Programming errors detected during execution of nonsupervisory portions of the NCP (level 5 dispatched) cause abnormal termination of program execution. Examination of abend codes in a storage dump can help in locating the error.

The optional abend service aid extends detection of programming errors to the NCP *supervisor* (levels 1-4), thus causing termination of the program before a supervisor error can be propagated into nonsupervisory portions of the program. The abend code appearing in the storage dump, therefore, gives a better indication of the location of a supervisor error, if one should occur, than an error code reflecting a resultant error in the nonsupervisory portion would give. Including the abend option (by the ABEND operand of the BUILD macro) is recommended when you first begin using a network control program. Later, as experience demonstrates that your network operates routinely without abnormal termination, the abend option may be deleted from the program.

Panel Tests

Some testing of communication lines can be done from the control panel of the communications controller. These tests (called panel-initiated line tests or panel tests) are explained in the *Control Panel Guide* (see Preface). Using the test routines, the operator at the controller can perform many of the communication functions (such as polling, addressing, and data transfer) normally executed by the controller and its control program upon command from the access method.

The panel test function is always present in the NCP.

Network Performance Analyzer

The Network Performance Analyzer (NPA) is a facility that collects performance statistics on programmed resources, SDLC resources, and BSC 3270 devices. this function consists of two parts; an NCP portion to accumulate the resource information, and a host program to format and display the statistics.

The host program is an IBM field developed program that starts and stops data collection and retrieves, logs, and displays the information at the host. For more information about the host portion, refer to the *Network Performance Analyzer Program Description and Operations* manual.

The NCP portion of NPA is included in the control program by coding the NPA= operand on the BUILD macro. One group of programmed resources must also be defined. This group must contain one each of the GROUP, LINE, PU, and LU macros. The LINE and PU macros for the NPA group do not have any operands. The following is an example for coding the NPA group.

```
symbol    GROUP   NPARSC=YES,VIRTUAL=YES,LNCTL=SDLC
          LINE
          PU
symbol    LU      MAXCOLL=nn
```

A GROUP macro must be coded as shown to define a virtual SDLC group as the NPA resource. All other operands on this macro will be ignored. One LINE, one PU, and one LU macro must be coded, and with the exception of MAXCOLL= and NPACOLL= on the LU macro, all other operands will be ignored and should not be specified. The MAXCOLL= operand specifies the maximum number of resources on which data can be collected and may be specified on either the GROUP, LINE, PU, or LU macro. MAXCOLL= is optional and the value defaults to 10 plus the number of lines defined in the NCP.

Specifying NAP=YES on the BUILD macro and defining an NPA group, causes the NPACOLL= operand to be valid for every resource on which data can be collected. NPACOLL= allows each individual resource to be designated as eligible or ineligible for NPA data collection. Specifying NPACOLL=YES on a invalid resource will cause an error indication. For all SDLC resources normally eligible for collection, the default is NPACOLL=YES when NPA is included in the NCP, and NPACOLL=NO when it is not. For all BSC resources (whether eligible for data collection or not) the default is NPACOLL=NO.

If NPACOLL=NO is specified, its effect will cascade to lower resources. That is, if NPACOLL=NO is specified on a LINE macro, no data can be collected for any resource on that line; specifying NPACOLL=YES for any of these resources will cause an error and the operand will be ignored. If data collection is allowed on an LU, it must also be allowed on the associated PU and LINE macros. However, a line may have NPACOLL=YES specified, but have NPACOLL=NO specified for some of the PUs and LUs under it.

Program Generation Options and Data Sets (Files)

All of the options described thus far in this chapter have related to the operational characteristics of the network. Described in this section are several options affecting the generation procedure and the program data sets (files) used in the procedure.

Program Generation Options

Program generation options pertain to the type of communications controller (channel or link attached), complete versus partial generation procedure, and several assembly and link-editing options. All program generation options are specified in the BUILD macro.

Type of Program to be Generated

As explained in Chapter 1, when defining the control program for a channel-attached communications controller, you must decide whether the program is to perform network control functions only or both network control and PEP functions. Specify the choice in the TYPGEN operand of the BUILD macro. Also specify whether the program is to be a channel-attached NCP (TYPGEN=NCP) or a link-attached NCP (TYPGEN=NCP-R).

Model of Controller

The same NCP can be executed in an IBM 3705-I or 3705-II Communications Controller. However, differences in the addressing requirements between controller models require that you specify type of controller in the MODEL operand. Changing the value in this operand is the only modification required to allow a network control program originally defined for one type of controller to be executed in the other type, *provided* that the network and controller configurations are identical.

Partial Generation

Assembling and link-editing the modules making up an NCP requires substantial processing time. Once you have generated a complete network control program, however, modifications resulting from changes in network configuration and procedural options can be done in significantly less time via *partial generation.*

In a partial generation, only selected modules are reassembled. They are then link-edited with the modules that require no changes to produce the modified program.

To perform a partial generation, code PARTIAL=YES and specify the names of the modules requiring reassembly in the CONDASM operand. The modules requiring reassembly for each of various changes in the program functions are listed in Appendix B.

In a VSE system, it is important that you retain all of the stage 1 and stage 2 assembly listings and the object library produced by the complete generation procedure. The VSE object library is an essential part of the partial generation.

Other Options

Other program generation options and their operands of the BUILD macro are:

- Whether stage 2 of the generation consists of a single- or multiple-step job or a separate job for each step, and whether a job card is required (JOBCARD).
- If a qualifier is added to conditional assemblies so that NCP generations with different names can run concurrently (OBJQUAL).
- Whether the JCL normally produced by the generation procedure is replaced by cataloged procedures. Procedures may be specified for assembly steps, post-assembly steps, and the linkage editor (OUTPUT).
- The region size for stage 2 linkage-editor job steps (LESIZE).
- Whether the generation procedure is to produce cross-reference listings for stage 2 assemblies (ASMXREF).
- Whether the macro expansion or NCP tables and NCP conditional assemblies are printed (PRTGEN).

- The value of the TIME parameter in stage 2 assembly EXEC statements (TIME).
- The type or class of devices used for utility data sets during stage 2 (UNIT).

Note: Some of these options apply only to OS/VS systems. See the macro descriptions in Chapter 5 for individual requirements.

Data Sets (Files) Used in the Generation Procedure

The names of various program data sets to be used in the generation procedure when generating under OS/VS are specified by the LOADLIB, MACLIB, OBJLIB, QUALIFY, UT1, UT2, UT3, and USERLIB operands of the BUILD macro. The NEWNAME operand specifies the name to be given to the generated NCP load module.

The only file name required when generating under VSE is NEWNAME.

Chapter 3. Functions for BSC and Start-Stop Resources

Described in this chapter are the many aspects of a network that you must identify to the network control program to support BSC and start stop stations.

You should read this chapter if you wish to define a program capable of operating BSC lines, start-stop lines, or both in network control mode only or in both network control mode and emulation mode. (Upon request from the access method, the NCP can change the operation of a line from network control mode to emulation mode, and vice versa, if start-stop or BSC stations are attached to the line.)

Many of the functions of the network covered in this chapter apply only to operation in network control mode. Others apply to both network control and emulation modes. In most cases, the functions are specified in exactly the same way for both modes. For example, the type of line control—start-stop or BSC—is specified in the same operand (LNCTL) of the same macro (GROUP) whether the lines in the group are to be operated in network control mode or in emulation mode, or in both modes alternately.

If the program you are defining is to operate all lines in network control mode, ignore any references to emulation mode in this chapter.

If the program you are defining is to operate some lines in network control mode and others in emulation mode or the same lines in both modes alternately, observe the references to emulation mode as well as those to network control mode.

This chapter explains the characteristics of the network with respect to:

- The stations and lines of the network
- The communications controller hardware configuration
- Data transfer between the communications controller and the host processor
- Procedural options governing message traffic between the controller and the network in network control mode
- Procedural options governing message traffic between the controller and the network in emulation mode
- Optional message processing within the controller (network control mode only)
- Diagnostic and service aid facilities
- Program generation options and data sets (files)

The description of each characteristic and option is not exhaustive; it is intended to provide sufficient information to enable you to select the appropriate parameters when coding the program generation macro instructions given in Chapter 5.

For many characteristics, especially those relating to the equipment configuration, the decisions about what to code in the macro instructions have been made by the *system designer*. (This is the individual who determines the data communication equipment, network configuration, and communication services that constitute your communication system.) You need only determine what these characteristics are and code the appropriate macros and operands accordingly.

Other characteristics relate to resources, such as the size of the buffers in the buffer pool, or to procedural options, such as the number of buffers of message data to be accumulated from a start-stop or BSC station before forwarding them to the host processor. Such characteristics, which affect the message-handling capacity and throughput of the communication system, require careful consideration before specifying the corresponding parameters in the program generation macro instructions.

Once you are familiar with those characteristics that apply to your equipment configuration and applications, you are ready to code the program generation macro instructions that define the NCP. At that point you should go on to Chapter 5, which provides detailed information on coding the macro instructions.

Network Characteristics for BSC and Start-Stop Resources

This section applies only to start-stop and BSC resources. (See Chapter 2 for information on SDLC resources.) The descriptions of the network characteristics give the names of the applicable macro instructions and operands. Unless otherwise indicated, the macros and operands named apply to both network control mode and emulation mode. For your convenience, the operand mentioned can often be specified in a macro instruction different from the one named, as explained in Chapter 5. The description of the operand always appears under the lowest level macro.

Station Characteristics

In this book, *station* refers to any equipment, regardless of type, that can transmit data onto, or receive data from, a communication line connected to the communications controller. For line operations in network control mode, this definition includes (1) computers; (2) communications control units such as the IBM 2701, 2703, and 3704; (3) other 3705 controllers; (4) the input/output units (keyboards, printers, tape and card readers, punches, and display screens) usually referred to as *terminals* and (5) control units (such as IBM 3270 cluster control units) with input/output units attached.

Each start-stop or BSC station that communicates with the 3705 Communications Controller in network control mode is represented by a TERMINAL macro instruction. (Stations with which the controller communicates only in emulation mode are represented by characteristics defined in the LINE macro, not the TERMINAL or CLUSTER macros.)

Type of Station

Type of station means the numerical designation by which the station is known, or an abbreviation thereof (for example, 1050, 2780, SYS3 "System/3"). Appendix A lists the types of stations the communications controller can communicate with in network control mode and in emulation mode. For network control mode, type of station is specified in the TERM operand of the TERMINAL macro or, for certain types of stations, in the CUTYPE operand of the CLUSTER macro. For operation in emulation mode, type of station is specified in the TERM or CUTYPE operand of the LINE macro.

Terminal Features

For some types of terminals and control units, the presence or absence of certain features with which the terminal or control unit may be equipped must be known to the NCP. The features that must be specified differ for network control mode and emulation mode. If the communications controller is to communicate with a terminal in network control mode, specify the appropriate features from the list of features for network control mode, and similarly, for emulation mode. If the controller is to communicate with the terminals in both modes, specify the appropriate features from both lists.

Features for Operation in Network Control Mode
The presence of some of the features below is specified in the FEATURE operand of the TERMINAL macro. For these, the suboperand that specifies the presence or absence of the feature appears in parentheses after the description. Other features are specified in the macros and operands indicated.

Transmit Interrupt (IBM 1050, 2741, 3767 in 2741 mode): If the terminal has this feature, the communications controller can interrupt a transmission from the terminal by sending the break signal. (BREAK or NOBREAK)

Buffered Receive (IBM 2740 Model 2, 2770, 2972 Models 8 and 11, 3270, 3780): If the terminal has this function, the NCP allows a time interval to elapse between successive transmissions to the terminal. During the interval, the NCP can communicate with other terminals on the same multipoint line. The presence of the feature and the interval are specified by the BFRDLAY operand. See also the discussion of buffered terminals under *BSC and Start-Stop Operation* in this chapter.

Conversational Mode (IBM 1050, 2740 Models 1 and 2 with Record Checking feature, 2770 with Conversational Mode feature, and all IBM BSC stations except 2715 and 2780): A station equipped with this feature can receive message data, instead of the usual positive acknowledgment, in response to a message block sent by the station. The message block the station receives in reply serves as the positive acknowledgment. Exchanging message blocks in this way improves line utilization because the time normally spent in re-addressing (re-selecting) the station is eliminated. Conversational mode is specified in the CONV operand.

If you specify the conversational mode feature, the NCP automatically replies to a message block from the station with the next block it currently holds for sending to that station. If the program has no data to send, it replies with a positive acknowledgment.

Accelerated Carrier Return (IBM 1050): If your network includes IBM 1050 terminals having the accelerated carrier return feature, you should specify this in the FEATURE operand of the TERMINAL macro for each terminal so equipped. The communications controller then sends a fewer number of idle characters than if the terminal did not have the feature, thus saving a small amount of transmission time whenever the new line (NL) character occurs in message data. (ACR or NOACR)

Record Checking, Station Control, Transmit Control (IBM 2740): The command sequence the NCP uses to communicate with the IBM 2740 differs for each of these features, or combinations thereof. (CHECK or NOCHECK, SCTL or NOSCTL, XCTL or NOXCTL)

Interrupt (IBM 2741, 3767 in 2741 mode), Receive Interrupt (IBM 1050): If the terminal has this feature, it can interrupt the NCP while the program is sending to the terminal. (ATTN or NOATTN)

Features for Operation in Emulation Mode

The presence of the following features is specified in the FEATURE operand of each LINE macro.

Record Checking: Some start-stop stations have the record checking capability (also called longitudinal redundancy checking), and other do not. For each line operated in emulation mode, you must specify whether the terminals on that line have the record checking capability. If the terminal is an IBM 1050, 1060, 2260, 2265, 2845, 2848, or System/7—specify LRC in the FEATURE operand. Also specify LRC for an IBM 2740 (Model 1 or 2) if it is equipped with the Record Checking feature. For other types of start-stop terminals, specify NOLRC.

Downshifting on Space Characters: Some AT & T 83B3, Western Union 115A, and World Trade teletypewriter (teleprinter) terminals, upon sending or receiving a space character, automatically downshift so that subsequent message text is in lowercase, or downshifted, mode. Automatic downshifting avoids the need to send a LTRS character to effect downshifting. Specify SPACE in the FEATURE operand if the terminals have this feature.

Immediate End: Upon receiving an end-of-transmission character from a start-stop terminal (in emulation mode), the NCP normally delays ending the receive operation for several character times (the time required for the transmission of one character) until the line becomes electrically "quiet". The absence of further characters following the EOT verifies that the EOT character is valid and not a data character converted by line noise to a false EOT. Checking for false EOTs in this manner is appropriate for many applications. On some applications, however, the terminal continues to send data immediately after sending the EOT (as when the terminal is transmitting from a paper tape with data interspersed with EOTs). If the end of the receive operation were in this case delayed, the program would not recognize the EOT because of the immediately following data characters. In this instance, it is necessary to specify IMEND in the FEATURE operand; this causes the program to end the receive operation immediately upon detecting the EOT, without waiting to detect the presence or absence of any following characters.

Dual Code: Either of two transmission codes (EBCDIC and USASCII) can be transmitted on a binary synchronous communication line attached to an IBM 2701 Data Adapter Unit equipped with the Dual Code feature for that line. The code used is changed from one to the other by command from the access method. The same function can be performed when the IBM 3704 or 3705 is installed in place of the 2701. Specify DUALCODE in the FEATURE operand if the Dual Code feature was used for that line when the line was attached to the 2701. Otherwise, specify NODUALCD or omit the parameter. (In addition to EBCDIC and USASCII, transparent USASCII is supported as a dual code option for a line serviced by a type 3 scanner.)

End-of-Transmission Character

You may specify that the EOB character is to signify end of transmission for messages from terminals equipped to send EOB EOT ending sequences. If specified, the terminal operator can signal the end of each transmission by pressing only the EOB key rather than both the EOB and EOT keys. (In this case, each transmission from the terminal consists of a single block.) This option, which applies only to operation in network control mode, is specified in the ENDTRNS operand of the TERMINAL or COMP macro.

Printer Line Length and Carriage Return Rate

The network control program recognizes each carriage return (CR) character and horizontal tab (HT) character in text being sent to a nonbuffered start-stop terminal in network control mode. Upon detecting either character, the program sends a sequence of idle characters immediately following the CR or HT character. A sufficient number of idle characters delays further printing on a terminal printer until the movable printing mechanism (carriage, or carrier) has had sufficient time to reach the next printing position. The next position is one of the tab locations, in the case of the HT character, or the left margin on the next printing line, in the case of the CR character.

The appropriate number of idle characters to send is determined from (1) the maximum length of the line of print—that is, the number of character positions between the left margin and the rightmost printing position; and (2) the rate at which the printing mechanism moves, expressed as the number of character positions traversed by the mechanism for each idle character. From these two values, which you specify for each start-stop line in the network by the LINESIZ and CRRATE operands of the LINE macro, the generation procedure calculates the required number of idle characters. The values should be carefully selected to suit the type of terminal connected to the line. Too few idle characters sent following each CR or HT character will allow insufficient time for the mechanism to reach the next printing position, resulting in random printing of text characters on the print line. Too many idle characters, on the other hand, will cause excessive delay in the resumption of printing, resulting in wasted time on the communication line.

Thus, for example, if the terminals attached to a given line have a printer line length of 60, and a carrier return rate of 10 printing positions per idle character, you would specify the values 60 and 10, respectively , in the LINESIZ and CRRATE operands. If you do not specify the line size or return rate, the network control program uses the default values given under the description of these operands.

If the printers attached to the line have differing line lengths, specify in LINESIZ the maximum length used by any terminal on the line.

Communication Line Characteristics

Nonswitched Multipoint Line

Multipoint line control is typically used for a nonswitched line with several stations attached. The controller contacts a specific station by sending a polling character or addressing character assigned to and recognized only by that station. The one station recognizing that character responds appropriately; the other stations ignore the character.

Multipoint line control must also be used for a line with only one station attached, if that station must be polled or addressed by the controller before sending or receiving data. A multipoint line is therefore one on which a multipoint discipline must be used, regardless of the number of stations—several or only one—the controller communicates with over that line.

To specify a line as requiring a multipoint discipline, specify POLLED=YES in the LINE macro if the line is to be operated in network control mode. If it is to be operated only in emulation mode, you need not specify it as a multipoint line. (The access method is responsible for properly controlling multipoint lines in emulation mode.)

The network control program requires a service order table for each nonswitched start-stop or BSC communication line that requires a multipoint discipline and that is to be operated in network control mode. This table contains one or more entries representing each station and each component of a station with which the program can establish a session upon request from the access method. (Sessions are described later in this chapter under *BSC and*

Start-Stop Operation.) The program attempts to establish sessions with stations and components in the same sequence as their respective entries appear in the service order table.

Directly following a LINE macro for a start-stop or BSC multipoint line (or a nonswitched point-to-point line that requires a multipoint discipline), code a SERVICE macro that defines the service order table to be used for that line.

Nonswitched Point-to-Point Line

To designate the communications controller as the secondary station on a BSC point-to-point line operated in network control mode, code YIELD=YES in the LINE macro (or omit the operand). To designate it as the primary station, code YIELD=NO.

Whichever choice you make, the station at the other end of the line must be prepared to assume the complementary role (that is, primary or secondary).

Except for the YIELD operand, you need code no other operands to designate the type of line as nonswitched point-to-point. The line is assumed to be of this type unless you explicitly code operands that specify another type. A nonswitched point-to-point line is not identified as such if it is to be operated only in emulation mode.

Switched Point-to-Point Line

For each switched point-to-point line connection, or "port," over which the communications controller may call stations or receive calls from stations, you must code a LINE macro. In the GROUP macro that precedes the LINE macros for the switched line, code DIAL=YES. (The GROUP macro, rather than the LINE macro, indicates that the lines are switched lines. If any lines within the group are switched, all must be.) This operand is applicable whether the line is to be operated in network control mode or in emulation mode, or both.

A switched line port can receive calls from either BSC stations or start-stop terminals, but not from both. Any type of BSC station can call the controller over a line designated for use by BSC stations (provided that all use the same transmission code), as the line control discipline for all such stations is similar. On the other hand, a line designated for use by start-stop terminals can receive calls from only a single type of terminal except when multiple-terminal-access operation is specified for that line. (See *Multiple Terminal Access Facility* later in this chapter.)

If the line is to be operated in network control mode, you may designate in the CALL operand of the LINE macro, whether the line is to be used for receiving calls from stations (CALL=IN), for making calls to stations (CALL=OUT), or both (CALL=INOUT). Each line used for outgoing calls must be included in a dial set by means of the DIALSET macro. Dial sets are explained in *Switched Network Operation* under *BSC and Start-Stop Operation*

Half-Duplex versus Duplex Lines

Half-duplex data transfer is always used for any start-stop or BSC station with which the controller can communicate.

Transmission Codes

The transmission code to be used for communicating with each station must be identified to the NCP. The program translates outgoing data characters from its internal processing code, EBCDIC, to the specified transmission code, and vice versa, for incoming data characters. (The transmission code used on a multipoint line must be the same for all stations attached to that line.)

Specify the required transmission code in the CODE operand of the LINE macro representing the communication line. (For BSC stations, the code you specify in the LINE macro also informs the network control program which line control scheme is to be used; the transmission code and line control scheme are related.)

Line and Subchannel Addresses

Each communication line attached to the communications controller is identified to the NCP by a line address. Whether the line is to be operated in network control mode or in emulation mode, specify this address in the ADDRESS operand of the corresponding LINE macro. If the line is to be operated in emulation mode, also specify the host processor subchannel address(es) corresponding to the line address. (Each line operated in emulation mode requires its own subchannel address(es) in the host processor. The multi-subchannel line access [MSLA] facility of the network control program with PEP extension permits two or more emulation subchannels to communicate alternately with the same communication line. The address of each subchannel to be associated with a line must be specified in the ADDRESS operand.)

Automatic Calling Units

Any switched network line that the NCP uses for calling stations may be equipped with an automatic calling unit (ACU). If a line is equipped with an ACU (whether the line is to be operated in network control mode or in emulation mode), specify the ACU address in the AUTO operand of the corresponding LINE macro.

The (ACU) address is determined by the physical location of the ACU hardware on the line interface base (LIB). A 1E or 9A line set is required for the ACU.

Determine the ACU addresses from the *Teleprocessing Installation Record* or the system designer.

Ring Indicator Mode (not applicable in U. S. and Canada)

Certain European modems may require that their ring indicator signal line be energized (signifying that the modem is being called by a station) before the communications controller indicates its readiness to receive by energizing the modem's data terminal ready signal line. (These and other signal lines constitute the interface between the communications controller and the modem.) If this requirement applies to a modem in your network, code RING=YES in the LINE macro for the communication line attached to the modem, whether that line is to be operated in network control mode or in emulation mode. Most modems do not have this requirement, and for these you would specify RING=NO in (or omit the RING operand from) the LINE macro. Specifying RING=YES for a modem that does not have this requirement can result in unnecessary delay in establishing the connection.

Communication between Controller and Host Processor

For the NCP to operate lines in network control mode, information on both the buffers within the access method and buffers within the network control program must be specified. This information is specified in the HOST macro. Each access method the NCP concurrently communicates with in network control mode is represented by a separate HOST macro. Buffers for network control operation are allocated from a single pool of buffers used for all line and channel data transfers in network control mode.

Associated with each communication line specified as operable in emulation mode and serviced by a type 3 scanner is a pair of buffers contained within the control blocks related to the line. The size of each buffer in the pair is user specified as 4, 8, 16, 32, 64, 96, 128, 160, 192, or 224 bytes.

For a given amount of data passing over the line, use of larger buffers affords more protection against possible overruns than does use of smaller buffers. (Overruns can result from temporary slowdowns of channel operation or from momentary peaks in data traffic through the network.) Use of larger buffers also results in less interrupt-processing overhead for line operations and—up to 32 bytes—less interrupt-processing overhead for channel operations. The amount of data transferred across the channel is equal to *n,* up to 32 bytes. For values of *n* exceeding 32, the amount of data transferred over the channel is 32 bytes.

The size of the emulation mode buffers for a line serviced by a type 3 scanner is specified in the BUFSIZE operand of the LINE macro. If you do not specify a size, 32-byte buffers are provided for lines operating at speeds of 9600 bps or less, and 64-byte buffers are provided for lines operating at higher speeds (as specified in the SPEED operand of the LINE macros).

Transfer of data in emulation mode between the host processor and the line occurs in a manner equivalent to that provided by the IBM 2701, 2702, or 2703 being emulated. In the CU operand of the LINE macro, specify the type of transmission control unit to be emulated for that line—2701, 2702, or 2703.

Procedural Options for Operation in Network Control Mode

A number of procedural options characterize the operation of lines in network control mode. These options include (1) the manner in which the program starts up and shuts down the network, (2) the amount of data to be transferred at one time between stations and the controller, (3) the amount of data to be accumulated from a station before passing it to the access method, and (4) the number of sessions to be conducted concurrently on a start-stop or BSC multipoint line. (NCP sessions are described later in this chapter under *BSC and Start-Stop Operation.)* By careful selection of these options, you can customize a network control program to best meet the requirements of your data communication applications.

Logical Connection Stations

Each start-stop and BSC station connected to a communication line operated in network control mode is generally represented within the network control program by its own set of control blocks, each defined by a separate TERMINAL macro. However, for a switched line over which the controller receives calls from stations, known as a *call-in* switched line. The program maintains one set of control blocks for all stations that call in over that line, regardless of the number of terminals that may do so. The control blocks represent a dummy station called a *call-in logical-connection* station. The control blocks represent whichever station has called over that line at any given moment. A single TERMINAL macro in which CTERM=YES is specified represents a call-in logical-connection station.

BSC and Start-Stop Operation

This section describes those procedural options that apply in general to communication lines operated in network control mode, and not uniquely to binary synchronous stations or start-stop (asynchronous) terminals. Options specific to one or the other, but not both, of these categories of stations appear in subsequent sections. (Operation of SDLC links is covered in Chapter 2.)

Sessions

The ability of the NCP to conduct multiple sessions on the same start-stop or BSC multipoint line in network control mode depends on the fact that data transfer does not occur continuously for the duration of the session. For example, in interactive applications such as inquiry-response, the elapsed time between receiving a response from the host processor and entering the next inquiry typically exceeds the time required for transmission of the inquiry and response. The terminal operator typically needs 5 seconds or more think time after seeing the response to prepare his next inquiry, whereas transmission time for the inquiry and response together often consumes but 1 or 2 seconds. The interval during which the terminal is not using the line can profitably be used to service other terminals on the same line.

Buffered terminals are another example. With such terminals, the operator keys his message data into a buffer instead of directly on the communication line; the line is not needed for transmission until the entire message or block of a message has been accumulated in the buffer. The IBM 2740 Model 2 is an example of such a terminal. A terminal may also receive data from the line into a buffer rather than sending it directly to the printer or other output device. The 2740 Model 2 with the buffered receive feature works in this manner.

While its operator is keying message data into the buffer, on the one hand, and while the terminal is printing the contents of the buffer, on the other, the terminal has no need of the communication line. Since data transmission to and from a buffered terminal usually is much faster than the data entry or printing operations, the terminal requires the line for a relatively small proportion of the session period. Again, the line can be used for servicing other terminals in the interim.

Interleaving transmissions with several stations maximizes the utilization of a multipoint communication line, thus permitting more stations to share the same line than if only one session were possible. A direct result is reduced cost of communication lines and line attachment hardware within the communications controller.

The number of concurrent sessions to be conducted on a line depends on several factors. Among these are (1) the relative amount of time when a terminal is in use that it does not need the communication line and (2) the permissible delay between readiness to use the terminal and availability of the communication line.

The number of concurrent sessions is called the *session limit* and is specified in the SESSION operand of the LINE macro. The NCP not only limits the number of sessions to this value, but it also tries to maintain that many sessions in order to get maximum line utilization. The number of sessions in progress will be less than the session limit whenever the NCP has requests for fewer devices than allowed by the session limit or when fewer devices are ready to communicate with the host processor.

In the case of clustered BSC stations such as the IBM 3270, the session limit is not applicable if general polling is used to solicit input from the attached terminals. The NCP cannot control the number of terminals that may respond to a general poll of the clustered station, and a separate session is established with each terminal that responds. However, for 3270 terminals, the session limit should equal the sum of the number of cluster controllers and the number of terminals attached to the line.

Service Order Table

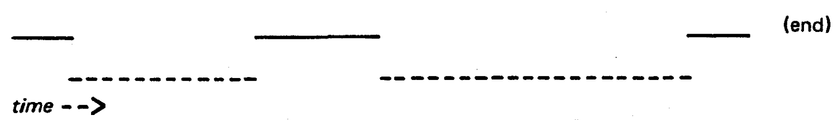
The sequence in which the NCP attempts to establish sessions on a multipoint line operated in network control mode is determined by a *service order table* associated with the line. This table is defined by the SERVICE macro you code directly following the LINE macro representing the nonswitched multipoint line. Each station with which the host processor may request a session must be represented by at least one entry in the table. If the station consists of a control unit with one or more individually pollable or addressable components (for example, the IBM 3270 and 1050 terminals), each terminal or component must be represented in the service order table whether it is polled, addressed, or both. In addition, a cluster control unit must be represented in the table if general polling is to be used.

The same device may be represented by more than one entry in the service order table. Multiple entries are of value if you wish the program to attempt to begin sessions more frequently with some devices than with others.

The contents of the service order table for a BSC or start-stop line can be changed during program execution by a control request from the host processor. Control requests can cause the program to add or delete devices or change the order or frequency with which the devices are serviced. Thus the NCP can be kept responsive to application requirements if these should change from time to time.

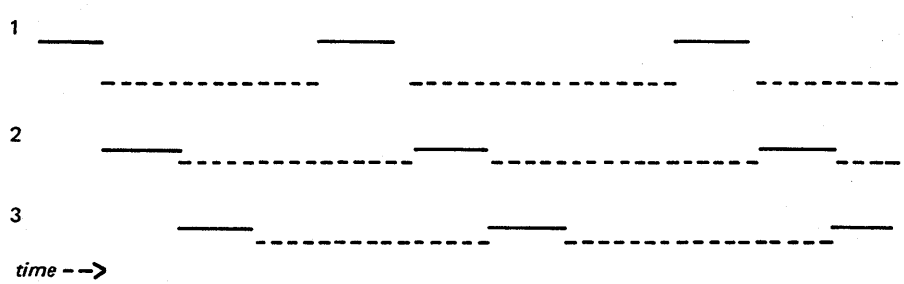
Logical Connections

A session is said to be *active* when the NCP is communicating with, or is ready to communicate with, the associated device. The rest of the time the session is *suspended* (or *inactive*). The active and inactive portions of a session may be represented thus:



The solid lines represent the session in its active state; the broken lines represent the suspended state.

Three concurrent sessions may be represented thus:



In most applications it is necessary to limit the amount of time a session is permitted to be active to prevent a device, once in session, from monopolizing the communication line to the exclusion of other devices. The period during which a session is active is called a *logical connection.* A logical connection exists when the NCP and the device in session are engaged in data transfer from one to the other. The length of a logical connection is limited by the *transmission limit,* which is the maximum number of transmissions that may be transferred in either direction between the NCP and the device during the logical connection. When the limit is reached, the NCP breaks the logical connection, thus suspending the session for the moment. The program is then free to service the next session or attempt to establish a new session if the session limit has not been reached.

Although the intent of the transmission limit is to restrict the time a session may be active, it does not represent a fixed amount of time or number of data characters. The actual length of a logical connection, in time, is determined by the number of transmissions, the number of data characters in each, and the speed of transmission.

The transmission limit will not always be reached, as the device in session may run out of data to send or become unable to receive or the NCP may run out of data transfer requests for the device before the transmission limit is reached.

The transmission limit can be individually specified for each device with which the host processor will establish sessions. This capability allows you to base the limit on the application requirements for the device. The limit is specified in the XMITLIM operand of the TERMINAL or COMP macro representing the device. If the application warrants, you may specify that unlimited transmissions are to be allowed during an active session. This choice should be carefully considered, however, as it can result in monopolization of the communication line by a single device for lengthy periods, thus negating the benefits of having multiple sessions.

Once a session has been established, the network control program re-polls the device for each subsequent transmission to be solicited from the device. The logical connection is maintained during the polling operation. Unless you specify otherwise, the program polls a device only once to solicit the next transmission. If ready to transmit, the device will respond positively to this poll.

You may have the program repeat the polling operation one or more times if you wish to allow the device more time in which to respond with its next transmission. Specify the required number of polls in the POLIMIT operand of the LINE macro for the line. The network control program will then accept an equivalent number of negative responses to polling before breaking the logical connection. The value specified in the POLIMIT operand is accordingly called the *negative response limit.*

Allowing the program to poll the device more than once is often appropriate for interactive applications in which the terminal operator needs several seconds of think time in which to prepare his next transmission. Consider, however, that (1) no message data is communicated during the polling operation and (2) no other session can be serviced until the negative polling limit is reached, even if the terminal has no more data to transmit.

Once the polling limit is reached, the network control program can proceed to the next session. Unless you specify otherwise, the program breaks the logical connection and cancels the read request that caused polling. The program then goes on to service the next suspended session (or to resume service seeking).

There are two alternatives to this course of action, however. You may specify that the NCP (1) break the logical connection or (2) maintain the connection and notify the host processor that the negative polling limit was reached. In the latter case, the program also flags all subsequent requests that may be on the queue for the device to prevent them from being executed. It is then the responsibility of the host processor either to signal the NCP to go ahead with execution of the remaining requests or to cancel those requests and send other ones in their place. These options, like the negative polling limit, are specified in the POLIMIT operand of the LINE macro.

Sessions may be suspended in one other way. Most types of input/output errors that occur during an active session cause suspension of that session.

Session Servicing and Service Seeking

Establishing a new session on a start-stop or BSC line in network control mode is called *service seeking.* The NCP performs service seeking whenever the number of existing (suspended) sessions is less than the session limit established for the line. Thus, the program always tries to conduct the specified number of sessions.

Servicing existing sessions is called *session servicing.* Servicing a session consists of establishing a logical connection, then sending or receiving data. The logical connection ends when (1) a request from the host processor ends the session, (2) the transmission limit is reached, (3) the negative polling limit is reached, or (4) an input/output error occurs.

Session servicing and service seeking alternate in a sequence of operations called a *service cycle.*

A service cycle consists of both session servicing and service seeking whenever there is at least one session but the total number of sessions is less than the session limit. If no sessions exist at the moment, session servicing does not take place—the service cycle consists only of service seeking. Conversely, if the number of existing sessions equals the session limit, there is no need for service seeking—the service cycle consists exclusively of session servicing.

When session servicing, the NCP makes a single logical connection for each existing session, in the same sequence as the devices appear in the service order table.

When service seeking, the program attempts to establish a new session with one or more of the devices for which no session currently exists and for which the network control program currently contains a request to begin a session.

As in session servicing, the sequence in which the network control program attempts to establish new sessions corresponds to the order in which the device entries appear in the service order table. Each service-seeking operation begins with the entry following the last entry handled in the previous service-seeking operation. The service order table is a "wraparound" table; that is, service seeking does not stop at the end of the table but resumes automatically with the first entry in the table.

The maximum number of devices with which the program attempts to establish a session during each service-seeking operation is called the *service limit.* If the device were always ready to engage in a session, only one service order table entry would require checking in each service-seeking operation because a new session would always be established with the device. This is not normally the case, however, and the network control program usually must make an attempt for each of several devices before successfully establishing a session.

Unless you specify otherwise, the network control program uses as the service limit one-half of the devices represented in the service order table.

You may instead designate in the SERVLIM operand of the LINE macro a specific maximum number of inactive devices for which the program is to attempt service seeking.

Several factors influence the distribution of the service cycle between session servicing and service seeking.

One major factor is the amount of data transferred between the communications controller and devices during logical connections. The longer the transmissions, the more time is spent in session servicing.

A second major factor is the value selected for the service limit. In periods when the network control program has data communication requests for few of the devices represented in the service order table, a large service limit can result in much service-seeking activity because the program will have to make numerous attempts before establishing a new session. On the other hand, in periods when the program does have data communication requests for most of the devices, it will be able to establish sessions much sooner. The value of the service limit would have less influence in this case since most often service seeking would end with establishment of a new session before the service limit was approached.

Another factor affecting the relative time spent in session servicing and service seeking is the *service priority*. This factor is effective only when the session limit exceeds the number of existing sessions by more than one. Unless you specify differently, the network control program returns to servicing existing sessions after one service-seeking operation—that is, after one attempt, successful or not, to establish a new session. This is referred to as giving priority to old sessions.

The alternative is to have the program perform the service-seeking operation more than once, the total number of operations equaling the difference between the number of existing sessions and the session limit. This is called giving priority to new sessions and is specified by coding SERVPRI=NEW in the LINE macro.

Assume, for example, that two sessions currently exist and that the session limit is 5. If priority is given to old sessions (SERVPRI=OLD), the NCP will perform a single service-seeking operation after servicing the two existing sessions. Then it will return to servicing the two sessions once again. But if new sessions have priority (SERVPRI=NEW), the program performs three service-seeking operations in succession, the value of 3 being the difference between the session limit and the number of existing sessions.

It can be seen that the larger the difference between the number of old sessions and the session limit, the more your choice of service priority affects the relative time spent in session servicing and service seeking. When the session limit exceeds the number of existing sessions by only one, the value in SERVPRI has no effect since in either case only one service-seeking operation will be performed.

A final factor influencing the distribution of the service cycle between session servicing and service seeking is the negative polling limit specified in the POLIMIT operand of the LINE macro. The higher the limit and the more often devices fail to respond promptly to polling once a logical connection has been established, the more time will be spent in session servicing.

During periods of low communication-line activity, there may be intervals when no sessions currently exist on a BSC or start-stop line operating in network control mode. The service cycle accordingly consists only of service seeking (provided the NCP currently contains at least one request to begin a session). Nonproductive polling and the resultant processing overhead can be minimized by specifying a *service-seeking pause* of from several seconds to many minutes. This pause, which you specify in the PAUSE operand of the LINE macro, is in effect at the end of each service-seeking operation. When at least one new session is established, the pause is inoperative, since to observe it would delay session servicing as well as minimize nonproductive polling. Since it is not in effect when the service cycle consists of both session servicing and service seeking, the service-seeking pause is not a factor in the distribution of the service cycle between these two activities.

During periods when no sessions currently exist and the program currently has no requests to begin a session, no service cycle exists; the line is idle. The service cycle resumes when the program receives from the host processor a new request to establish a session.

As indicated by the foregoing discussion, numerous factors influence the handling of sessions on a multipoint communication line. When specifying operations over start-stop and BSC lines, consider all of these factors in terms of their effect on your data communication applications.

Sessions on Point-to-Point Lines

The concept of a session as a defined sequence of data interchanges between host processor and device is valid for point-to-point lines as for multipoint lines.

However, the advantage of multiple concurrent sessions is not available for a point-to-point line; since the network control program communicates with only one station over the line, only one session can exist.

As mentioned earlier, the access method requests that the communications controller establish sessions on lines in network control mode without regard for the way in which the network control program will conduct them. The type of communication line—point-to-point or multipoint—over which the network control program will conduct the session, therefore, does not concern the host processor.

Because only one session at a time can exist, the session limit, service limit, service priority, service-seeking pause, transmission limit, and polling limit parameters are not applicable for a point-to-point line.

Switched Network Operation

The switched network facilities of the network control program are designed to permit a high degree of utilization of the switched network connections, or ports, available to the communications controller. Maximum utilization of these ports reduces the number required to support a given number of terminals, with attendant savings in line and controller hardware costs.

The network control program's switched network facilities (applicable only to communication in network control mode with BSC and start-stop stations) accommodate both *call-out* operation, in which the controller calls remote stations upon request from the host processor, and *call-in* operation, in which the controller answers calls from stations. Switched network connections (ports) may be designated for use in fulfilling call-out requests, call-in requests, or both.

Call-Out Operation

The network control program maximizes the utilization of call-out lines by dynamically allocating them to handle the call-out requests with a minimum of delay. To enable the NCP to fulfill call-out requests, you must (1) define a dial set consisting of switched lines having similar characteristics and each line designated as a call-out line and (2) specify the stations with which the NCP can communicate using lines in the dial set. (See "The Multiple Terminal Access Facility" later in this chapter for another method of increasing switched-line utilization.) Because the program allocates the lines dynamically, any line in the set may be used to communicate with any of the designated stations. You might, for example, establish a dial set of three lines to accommodate call-out requests for 20 stations. The program fulfills call-out requests in the order they are received from the host processor. If at the moment a request is received there have been no other requests received before it and a line is available, the program fulfills the request immediately. Otherwise, the program places the request on the queue for the dial set to be handled in its turn.

Lines to be used for handling call-out requests must be designated as CALL=OUT in the corresponding LINE macros; alternatively, they may be designated as CALL=INOUT if they are to be used interchangeably for both kinds of requests.

A station associated with the dial set is represented by its own TERMINAL macro, which represents that station and no other; the telephone number by which the program calls the station is contained within the program. The number is specified when defining the program via the DIALNO operand of the TERMINAL macro. It can be changed later, during program execution, by means of the dynamic control facility.

You may improve line utilization by assigning an *alternate dial set* to the original dial set, which is accordingly called the *primary dial set.* The alternate dial set in effect helps to handle the load of call-out requests for the primary set when the alternate set is not fully occupied with its own call-out requests. The alternate dial set is in fact a primary dial set for its own group of stations.

Thus, for example, the switched network facilities for an installation might consist of three dial sets--*A, B, and C*--each of which services call-out requests for a group of 20 stations. Each is the primary dial set for its designated group. If B were also defined as the alternate dial set for A and C as the alternate for B, B could help with A's overloads, while C could help with B's overloads.

Dial sets are defined with the DIALSET macro. In addition to specifying the list of communication lines to make up the set, you may, in the DIALALT operand, specify the name of an alternate dial set.

Two other DIALSET macro operands—QLIMIT and QLOAD—determine the network control program's action when the program cannot immediately service a call-out request because all lines in the dial set are busy. Unless you specify a different value in QLIMIT, the program will place only one call-out request on the queue for the dial set. When this *queue limit*--whether it is 1 or a greater value that you specify--is reached, the program rejects any further call-out requests for that dial set. Or you may specify a queue limit of 0, in which case the program rejects *any* call-out request it cannot immediately service.

The other operand, QLOAD, specifies the number of unfulfilled requests the program will permit to accumulate on the queue for the dial set before using a line from the *alternate* dial set (if any) to service requests. The number must not exceed that specified in QLIMIT, for then the number of unfulfilled requests cannot reach the value that would cause the program to use the alternate dial set.

Call-In Operation

A switched line used for call-in (but not call-out) operation is not included in a dial set.

Since the host processor does not know which station will be calling from moment to moment, it directs its call-in requests to a logical-connection station associated with each line. Once a station has called, the logical-connection station represents that station until the logical connection ends.

To specify that a line is available for call-in requests, specify CALL=IN in the LINE macro. Alternately, you may designate the line as available for servicing either call-in or call-out requests (CALL=INOUT).

Using Lines in a Dial Set for Both Call-Out and Call-In Operations

If you want lines in a dial set to service both call-out and call-in requests. This is another way to maximize line utilization because the lines that might otherwise be idle in periods when the program receives few call-out requests can instead be used to service call-in requests.

To specify a line in a dial set as available for handling either incoming or outgoing calls, code CALL=INOUT in the corresponding LINE macro. A dial set may include any combination of lines specified as available for call-out operation or both call-out and call-in operation.

If all lines in a dial set are designated as available for both incoming and outgoing calls, the possibility exists that all may become occupied with outgoing calls, thus preventing any station from being able to call the controller. To prevent this, you may specify, in the RESERVE operand of the DIALSET macro, a minimum number of lines the program must hold in reserve for accommodating incoming calls.

Switched Network Backup

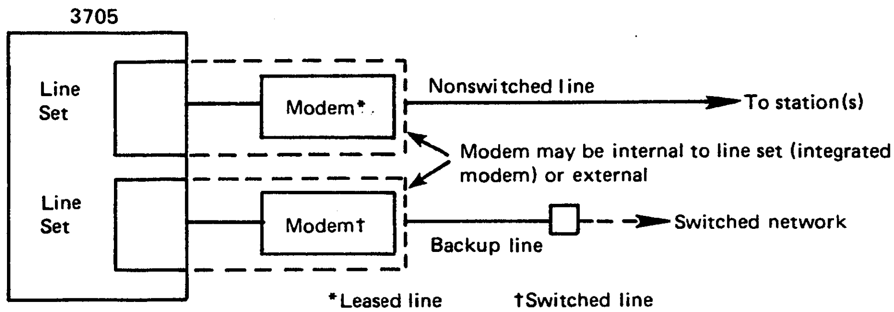
The switched network backup facility of the network control program permits communication between a communications controller and a BSC or SDLC station over a temporary, switched communication path provided as an alternate, or backup, line to the usual (principal) nonswitched point-to-point or multipoint line. Provision of a backup line permits communication between controller and station to continue despite failure of the principal line.

The availability of a backup line requires the installation of appropriate equipment at the controller and stations.

Two forms of switched network backup are available: same-port backup and alternate-port backup. The same-port backup technique can be used for either SDLC or BSC stations. (See Chapter 2 for details on same-port backup.) Alternate-port backup is available only for BSC stations.

Alternate-Port Backup

The alternate-port backup technique, available only for IBM 3270 (BSC or SDLC) and BSC stations, employs a switched-network connection (port), equipped with its own, separate, switched network modem, thus:



The principal and backup lines are represented to the NCP by separate LINE macros. The modem (3872, 3874, or 3875) at the remote station must be equipped with the switched network backup feature (or a separate switched network modem provided in addition to the regular modem, with provision for switching the station from one to the other).

If more than one station is attached to a modem equipped with the switched network backup and fanout features, all attached stations (up to three) can operate concurrently over the backup connection provided that all are switched to backup operation.

Only terminal components represented to the NCP and the access method by TERMINAL macros can participate in backup operation; components represented by COMP macros cannot participate.

In the case of a multipoint principal line, the NCP may continue to communicate normally with any stations on that line not affected by the line failure while concurrently communicating in backup mode with another station over the backup connection. Further, the program can communicate in backup mode with several stations affected by the failure of the principal line provided that sufficient backup ports are available.

Establishing the switched backup connection using the alternate-port technique requires that the operator at the host processor enter the appropriate access method commands. Either the operator or the NCP can select the particular switched port to be used. The operator at the remote station to be contacted must switch the station's modem to switched network operation (and back to normal operation when the principal line is restored to service); thus, voice contact is required between the host processor and the remote station when the backup connection is made or terminated.

Two alternatives are available for dialing the backup connection to a station: automatic calling and program-assisted manual dialing. (A backup connection can be made only from the host processor, not from a remote station.)

Automatic calling —that is, automatic dialing of the telephone number—is possible if the backup port is equipped with an automatic calling unit (ACU), the line address of the ACU is specified in the AUTO operand of the LINE macro for the backup port, and the dial digits are specified in the DIALNO operand of the TERMINAL macro representing the station.

Manual dialing is required if the backup port is not equipped with an ACU. The operator at the host processor must manually dial the telephone number of the station to be reached. If the number is specified in the DIALNO operand of the TERMINAL macro for the station, the access method will inform the operator of the telephone number and the specific line on which to make the call when he enters the command to switch to the backup line. If the number is not specified in the DIALNO operand, the access method tells him only which line is to be used for the call; he must consult a list of stations and their backup telephone numbers to determine which number to dial.

When communication with the remote station(s) is to be restored to the principal line, the operator at the host processor enters a command to switch the station(s) to the principal line. The NCP breaks the backup connection when all stations (if more than one are attached to the line via a fanout modem) have been switched back to the principal line.

Note: If the telephone number of a remote station is changed after the network control program is generated and the automatic calling technique is to be used, the new number *must* be specified in the DIALNO operand and the NCP tables reassembled via partial program generation in which CONDASM=TABLE is specified in the BUILD macro (or via complete program generation).

See Appendix J for an example of how alternate-port backup operation is specified in the network control program. Refer to the *ACF/TCAM Operation* manual for the commands required for backup operation.

Beginning Network Operation

After the NCP is loaded into the communication controller and begins execution, the communications access method must send control requests that logically activate the communication lines. After a line is activated, the NCP accepts requests for any devices connected to that line.

Type of Line Control

All types of stations with which the communications controllers can communicate use one of three line control schemes: binary synchronous (BSC), start-stop (or asynchronous), and synchronous data link control (SDLC). Each communication line attached to the controller uses one of these schemes; the same line never uses more than one. (Some stations can use two line-controls; however, only one type of line control can be defined for these stations in a network control program.)

When defining the NCP; specify which type of line control the program is to use for each line. All lines in a line group use the same line control; therefore the type is specified in the GROUP macro. The operand is LNCTL.

Data Flow Control

The primary purpose of data flow control is to control the amount of data entering the network, thereby protecting the network resources from overload and deadlock situations. The AVGPB operand of the LINE macro is used to inform the NCP of the average number of bytes the BSC or start-stop stations are expected to send. Then, before polling a particular station, the NCP can determine if it has enough buffer space available to handle the incoming data. If buffer space is not sufficient, the NCP will delay the poll until it can ensure that all the data from the station can be handled without overrun.

The CWALL operand on the BUILD macro is used to specify a specific number of buffers to be reserved in the NCP to complete the movement of data out of the controller. When the available buffers reach this threshold, the NCP stops input channel operations and will not accept any more PIUs. Operations that send PIUs out of the NCP continue. When the number of buffers raises above the threshold, the input channel is reopened and data transfer proceeds normally in both directions.

Network Slowdown

The NCP can receive message data from the access method (via the network control subchannel) and from lines in network control mode only as long as it has buffers available for the data. The program normally receives and sends data at the same average rate, although momentary overloads can occur when the program receives more data than it sends during a given time interval. Should the overload be protracted, however, the NCP can exhaust its supply of buffers. To prevent this condition, the NCP continuously monitors its supply of buffers and, when the supply falls to a specified level, automatically enters *slowdown mode.* The level is specified as a percentage of the total number of buffers in the program.

When in slowdown mode, the program reduces the amount of data it receives from lines in network control mode and from the network control subchannel(s), but it continues to send at the normal rate. Since the rate at which buffers are released after transmission of their contents exceeds the rate at which new buffers are obtained for receiving data, a net gain in the number of available buffers results. When the buffer supply is sufficiently replenished, the program automatically resumes normal operation.

The SLODOWN operand of the BUILD macro allows you to specify 12, 25, or 50 percent as the minimum percentage of available buffers below which the program enters slowdown mode. However, during initialization, the NCP dynamically increases the percentage you specified if the minimum NCP buffer requirements cannot be met. The minimum number of buffers that the program must contain for each percentage value is 80 buffers, for 12 percent; 40 buffers, for 25 percent; and 20 buffers, for 50 percent.

Terminal Time-Outs

The NCP normally observes for each communication line two *timeout* intervals of several seconds duration. One of these intervals is the *reply timeout,* which limits the amount of time the program will await a station's response to polling or response to message data sent to the station. The other interval is the *text timeout,* which limits the time that may elapse between receipt of successive message characters from the station after message transmission has begun. If the timeout expires before the response or the next message character is received, the program ends the read operation for that station and notifies the

..ost processor of a timeout error. These timeouts apply to each line in th network whether the line is operating in network control mode or in emul mode.

By observing these two timeout intervals, the NCP prevents a comm..icat line from being idled indefinitely because of excessive delay in entering successive message characters at a terminal or because a malfunction or po failure at the station interrupts its transmission to the communications controller.

Unless you specify different values in the REPLYTO and TEXTTO operand of the GROUP macro, the NCP uses the timeout intervals indicated in the descriptions of these two operands for all lines in the group represented by th macro. Some applications may justify allowing unlimited intervals, that is, no timeout at all. This also may be specified in the REPLYTO or TEXTTO operands.

Conversational Response

Some BSC and start-stop stations equipped with parity checking accept message data as a positive response to a block of text the station has transmitted. The IBM 1050, 2740 Models 1 2 with Record Checking, 2770, 2972 without the Batched Message Input feature, and all BSC stations except the 2715 and 2780 are such devices.

Transmitting message data instead of the normal positive response eliminates the line turnaround time incurred when sending a positive response character followed by an addressing character and then receiving a response to addressing before sending message data to the station. The benefit of this conversational write operation is improved line utilization.

For each station or component capable of accepting message data as a positive response to text, you may specify conversational operation in the CONV operand of the TERMINAL or COMP macro representing the device. The NCP then withholds sending a positive response after executing a read (or invite) operation if the next request directed to the same device is a write request. Instead, the program sends the message data conveyed by the write request. (For BSC stations, the NCP sends message data as a response only if the data received in the preceding read or invite operation was a complete message "ended by an ETX character") This option applies only to stations on lines operating in network control mode.

Polling and Addressing Characters

Certain types of start-stop and BSC stations must be polled or addressed by the communications controller in order for them to transmit to or receive from the controller. To receive data from the station, the controller sends a polling character (or sequence) assigned to and recognized by that station. Receipt of the polling character causes the station to unlock the keyboard, allowing the operator to enter data or to activate an input device such as a tape reader if he has previously readied the device to transmit data. Similarly, the controller sends a specific addressing character (or sequence) to signal the station to be ready to accept data from the controller.

If the station is a terminal having more than one input component, such as a card reader and/or a tape reader as well as a keyboard, a polling sequence may be assigned to each component. This allows the controller to solicit data from individual components. Or the polling sequence may specify activation of any input component that the operator has made ready to transmit data.

If the terminal has more than one output component—such as a display, a card punch and/or tape punch, as well as a printer—each may have its own addressing sequence, which allows the communications controller to send data to a specific unit.

Some BSC stations, such as the IBM 2770, recognize component selection characters within message data the station receives over the line. The output component to which data is sent may thus be changed as receipt of the message progresses.

Stations activated by polling and addressing characters are used in network configurations requiring attachment of more than one station to a communication line. Such a line is called a multipoint line, and each station is assigned polling and addressing characters different from those assigned to any other stations on that line. This allows a particular station to be activated while all others remain idle. In addition to the individual characters, a group addressing character (or sequences) may be assigned to certain stations to permit simultaneous transmission of data to all stations in the group. Or a broadcast address character may be assigned to all stations on the line to permit addressing all stations simultaneously.

Stations using polling and addressing are not limited to use on multipoint lines. The IBM 1050, for example, always must be polled and addressed, even if only one 1050 is attached to a line and regardless of whether that line is nonswitched or switched.

The NCP performs the polling and addressing functions for all stations connected to lines operating in network control mode. The access method performs these functions for all stations connected to lines operating in emulation mode. Therefore, in defining the NCP, you must specify the polling and addressing characters for each station with which the program communicates over a line operating in network control mode, but not for those stations on lines that will operate only in emulation mode. Specify the polling and addressing characters for lines in network control mode as follows.

In the POLL operand of each TERMINAL macro representing a station that must be polled, specify the polling character or characters recognized by that station. Similarly, in the ADDR operand of the same macro, specify the addressing character or characters recognized by that station. If a station is used for output only, you would specify only the ADDR operand; conversely, you would specify only the POLL operand for a station to be used for input only.

If a station has more than one individually addressable output component, the access method may establish individual sessions with each, provided that all components with which sessions are to be established are represented by a TERMINAL or COMP macro, the ADDR operands of which specify the addressing characters required. Likewise, if you wish to establish independent sessions with each of two or more input components, each must be represented by a TERMINAL or COMP macro, the POLL operands of which specify the appropriate polling characters. If a station has but one input and one output component, only the TERMINAL macro is required. Each additional component requires a COMP macro.

A terminal having multiple input and output devices requires only a TERMINAL macro if that the terminal has the common polling and addres capability. (An example is the IBM 1050, for which the common polling character is 0 and the common addressing character is 9.) The common polling and addressing characters must be specified in the POLL and ADDR operands of the TERMINAL macro.

If your network includes start-stop terminals that permit group or broadcast addressing and you wish to use the facility (in network control mode), code an additional COMP or TERMINAL macro. This macro's ADDR operand specifies either the broadcast or group address.

If your network includes multiple component BSC stations connected to point-to-point lines, you may wish to specify selection sequences for each of the output components. Selection characters for one output component may be specified in the ADDR operand of the TERMINAL macro. Specify the selection sequences for any additional output components in separate COMP macros.

Station Telephone Numbers

To call a station (in network control mode) that is connected to the switched telephone network, the network control program must be given the telephone number of that station. You specify the dial digits in the DIALNO operand of the TERMINAL macro representing the station.

If necessary, you may change the number to a different number via the dynamic control facility.

When specifying a telephone number in the DIALNO operand, you may include a dialing pause, in multiples of one second, between successive digits. A dialing pause can allow time for receiving a secondary dial tone, as when an outside-line code or direct-distance-dialing network access code must be dialed before the station's telephone number. End-of-number and separator characters may be included in the sequence of dial digits specified in the DIALNO operand if the modem is designed to use these characters. (The use of these characters is explained in the description of the DIALNO operand.)

Number of Attempts to Dial a Station

Unless you specify otherwise, the NCP, on receiving a request to call a station in network control mode, automatically dials the station's telephone number up to four times in succession. If the last attempt is unsuccessful, the network control program returns to the host processor a response indicating the failure. To specify a different number of attempts to dial a station, code the desired value (up to 255) in the REDIAL operand of the LINE macro. (A value of 255 indicates that the NCP will redial the station indefinitely until the station answers or the access method resets the request.)

Preventing a Monopoly of NCP Buffers

The NCP fills all requests for buffers from a single buffer pool, and no BSC or start-stop station should monopolize the supply of buffers to the extent that other stations are prevented from communicating with the controller in network control mode. Such excessive buffer monopolization could occur if the network control program were to accumulate too much data from a station before forwarding the data to the host processor.

You can prevent buffer monopolization with the TRANSFR and CUTOFF operands of the LINE macro. The TRANSFR operand prevents excessive accumulation by the controller of message data *from* a station. The CUTOFF operand sets a maximum limit on the data received during a single logical connection.

Normally, the NCP routine that receives data from a station accumulates an entire block of a message before passing the data to the host processor. This is desirable because message processing routines within the host processor can then examine an entire block at once.

If the station were to send an excessively long block of data, as could happen if a terminal operator entered thousands of characters without sending an end-of-block character, an unreasonably large number of NCP buffers could be filled by the arriving data. To prevent all of these buffers from being tied up until the block is complete, the NCP restricts the number of buffers that can be filled with arriving data before passing their contents to the host processor. Reception of data from the station is not interrupted when this happens; the NCP continues to allocate buffers for the remaining data.

Each partial block the program passes to the access method is called a *sub-block,* and the NCP is said to be operating in *sub-blocking mode* with respect to the line over which the data is being received. The response header that precedes each partial block indicates to the access method that the data that follows is a sub-block, not a complete block.

Aside from the consideration of buffer monopolization, there is a limit to the amount of data the access method can receive from the NCP during a single data transfer operation. The program accordingly restricts a sub-block to only as much data as it can deliver to the host processor in a single data transfer operation over the network subchannel.

In the TRANSFR operand of the LINE macro, you may specify the size of a sub-block in terms of the number of buffers to be filled before forwarding to the access method. Specifying this parameter in the LINE macro allows you to set a different limit for each of various lines.

If you omit the TRANSFR operand or if you specify a number of buffers greater than the NCP can pass at one time to the access method, the NCP uses as the sub-block size the maximum number of buffers it can deliver to the access method.

Limiting the size of a sub-block in this manner usually prevents undue buffer usage by any one station. This assumes, however, that the NCP promptly transfers the contents of the sub-block buffers to the access method and then releases the buffers to the buffer pool. If for any reason the NCP cannot promptly transfer the sub-blocks to the access method (as when the host processor channel is slow to respond to the controller's signal for service), the data accumulating from the station can still cause monopolization.

For this reason, you may also set a limit on the number of sub-blocks the network control program will accept from a start-stop or BSC station during a single logical connection. If the station sends enough data to cause the limit to be reached, the program breaks the logical connection for that station and is then free to service another station on the same line, if a multipoint line. Limiting the number of sub-blocks also serves to prevent a station from monopolizing a multipoint line so as to delay servicing of other stations on the same line. In the CUTOFF operand of the LINE macro, you specify the number of sub-blocks to be accepted from a station. If you omit the CUTOFF operand, the NCP continues to accept sub-blocks from a station until the entire block is received. If any sub-block contains an error (indicated by a bit in the response header for the sub-block), the application program should normally discard all of the sub-blocks in the block, not only the one in error. There is no provision for requesting retransmission of sub-blocks in error.

Use of Buffer Delay for Buffered Terminals

Some types of IBM terminals receive incoming data into buffers at high speed and then print (or otherwise display) the data at a much slower rate. If the NCP has multiple data blocks to send to the same terminal, it must wait after sending each block for the terminal to print the contents of the terminal buffer. Then the program sends the next block. If the line is a multipoint line operating in network control mode, the NCP can use the time the line would otherwise be idle for communicating with other terminals. That is, at any given moment, the program can be sending to one of several terminals while the others are printing data received earlier.

For each start-stop or BSC terminal attached to a line operating in network control mode, specify in the BFRDLAY operand of the TERMINAL macro the delay in seconds that the NCP should wait after sending each block before sending the next one. The value you specify should equal the length of time the terminal needs to print or otherwise display the contents of its buffer.

The terminals for which specifying a buffer delay is appropriate are:

IBM 2740 Model 2 with Buffered Receive feature
IBM 2770
IBM 2980
IBM 3275
IBM 3277
IBM 3284
IBM 3286
IBM 3780

Error Conditions and Recovery Procedures

Communication between the communications controller and stations in the network is subject to *input/output* (I/O) errors, usually caused by the transient noise conditions to which communication lines are sometimes susceptible or by hardware malfunctions. The NCP takes no action for errors on lines operating in emulation mode. Any error-recovery procedures desired must be executed by the access method.

For lines operating in network control mode, if the error is of a kind that can be recovered from, the NCP makes the appropriate recovery efforts. For example, upon detecting a parity error in received data, the program signals the station to retransmit the data if the station is of the kind that can automatically retransmit, as from a buffer, or of the kind that can visually inform its operator to re-enter the same data. Conversely, upon being informed by the station that it has received data in error, the program retransmits the data.

The maximum number of retransmissions may be specified for each line operated in network control mode. If error-free transmission is not achieved before the retransmission limit is reached, the NCP indicates the fact in its response to the access method.

If the I/O error is of the kind that inherently cannot be recovered from (such as a modem error), the NCP makes no error-recovery attempt but immediately indicates in its response to the access method what kind of error occurred.

Once the NCP notifies the access method that it is unable to clear the error condition, it makes no further attempts at error recovery. Nor does the program attempt to execute any further requests for the device affected. Instead, it places the device in an *error-lock* status. The access method can then take appropriate action. For example, the access method can send a control request that causes the NCP to remove all unfulfilled requests from its queue for the device and return them to the access method, which can then modify the sequence of requests or build a new sequence and send it to the NCP. Or the access method can ignore the error condition and send a control request that reinstates the queued requests, thus removing the device from error-lock status.

Input/output errors can occur either during transmission of message data or while the NCP is performing a control function preparatory to or following message transmission. I/O errors are consequently divided into *text-mode* errors and *control-mode* errors.

The number of error-recovery attempts for text-mode errors is determined by the RETRIES operand of the LINE macro. The number you specify applies to all stations with which the NCP communicates over the line represented by the LINE macro. The number of error-recovery attempts for control-mode errors is set by the CRETRY operand of the GROUP macro. Thus, the same maximum number of recovery attempts applies to all stations on all lines making up the line group.

Text-mode errors may be further classified as *text-read* errors and *text-write* errors. Text-read errors are those occurring when receiving from a station; text-write errors occur when sending to a station.

For text-read errors you may specify a single sequence of retransmissions, from 1 to 254. Or you may specify unlimited retransmissions.

For text-write errors you may specify a single retransmission sequence or multiple sequences. When you specify multiple sequences, the program executes the first sequence immediately after detecting the error condition. If transmission is still unsuccessful at the end of the sequence, the program pauses and then executes the next sequence. This activity continues until successful transmission is accomplished or the limit on retransmissions is reached. Specifying a pause between retransmission sequences allows time for transient noise conditions on the line to subside. A problem that may be responsible for the repeated errors.

Unless you specify otherwise in the RETRIES or CRETRY operands, the NCP will perform a maximum of two recovery attempts for start-stop stations and seven attempts for BSC stations for text-read, text-write, and control-mode errors. As stated above, error recovery is attempted only for lines operating in network control mode.

Normally error recording is done only for permanent errors. The initial error status causing the error recovery to begin and the ending error status is recorded for transmission to the host. Operator commands can be issued to cause the NCP to implement *intensive-mode* error recording. This mode records all temporary errors as they occur. The expanded statistics made available by this function will often preclude the need to run specific link tests and thus provide more efficient, timely problem determination.

Note: Intensive-mode error recording puts an added burden on the buffer pool and should be used with caution.

Protecting against Failure of Modem or Automatic Calling Unit

Various malfunctions occurring within a modem or automatic calling unit (ACU), if not protected against, could cause the communication line attached to the modem or ACU to be tied up indefinitely without the knowledge of the NCP, thus effectively rendering the line unusable. To prevent such an occurrence, the NCP has three timeout values for lines operating in network control mode: *enable timeout, disable timeout,* and *dial timeout,*

If a switched line is not enabled (that is, the data set ready signal line in the modem is not turned on) within the enable timeout period, the enable operation is terminated abnormally. Similarly, if the switched line is not disabled (data set ready signal line is not turned off) within the disable timeout period, the disable operation terminates abnormally.

Unless you specify different values, the network control program uses an enable timeout of 2.2 seconds and a disable timeout of 3.0 seconds. These values are generally appropriate, but you may specify different ones in the ENABLTO and DSABLTO operands of the BUILD macro. The timeouts specified apply to all communication lines operating in network control mode.

The third timeout, dial timeout, protects against the failure of the automatic calling unit for a switched line to reach a called station within a reasonable period. This condition may result from failure of the called station to answer, from failure of the common-carrier equipment to complete the connection path, or from a malfunction within the ACU. Expiration of the dial timeout for any of these reasons causes abnormal termination of the dialing operation.

A dial timeout of 60 seconds is normally appropriate, and this is the value the NCP uses for lines in network control mode unless you specify a different one in the DIALTO operand of the BUILD macro.

Direction of Transmission

In most network configurations, the stations are capable of both sending and receiving message data. However, for each BSC or start-stop station attached to a line operating in network control mode, you may specify that the station be used for input only via the DIRECTN operand of the TERMINAL macro representing the station. If you specify input only, the NCP rejects any requests from the host processor to send to the station. Conversely, specifying output only causes the program to reject any requests to accept data from the station.

Erasing Critical Data in Buffers

Normally, for BSC and start-stop stations, the NCP releases buffers to the buffer pool after use without first clearing the buffer contents. As buffers are reallocated for subsequent operations, the old contents are overlaid with new message data.

For maximum assurance that security-sensitive data remaining in buffers returned to the buffer pool is not subsequently transmitted to an unintended destination, you may specify that the network control program clear buffers before returning them to the pool as follows. Specifying ERASE=YES in the BUILD macro causes the NCP to: (1) erase all data buffers received from the host processor but rejected because the network control program has entered slowdown mode; (2) erase all buffers containing data blocks received from the host processor in which the NCP detected a transmission error; and (3) erase data buffers containing PIUs sent to the host processor, after the host processor has indicated successful receipt of the data.

In addition, you may specify CDATA=YES in individual CLUSTER, TERMINAL, and COMP macros to cause the program to erase buffers that have been used for receiving from or sending to the corresponding start-stop or BSC stations. This erase function is not applicable for data received from or sent to SDLC stations.

Automatic Network Shutdown

For BSC and start-stop lines, the network control program:

- Cancels the command currently being executed for the line
- Breaks the switched connection if the line is a switched line
- Cancels the line trace or online test operation if the line is currently being traced or is undergoing online testing
- Dissociates the line from the owning SSCP with which communication has been lost

For BSC and start-stop stations, the network control program:

- Stops general polling of clustered stations
- Cancels any commands currently pending for the station
- Sends a predefined message to stations for which CRITSIT=YES is specified in the TERMINAL macro (see "Critical Situation Notification" in the following paragraph)
- Cancels any sessions in which the station is currently active
- Resets the station from monitor mode if that mode is currently in effect

Critical Situation Notification

During automatic network shutdown of lines in network control mode, the program can automatically send a predefined message to each active start-stop and BSC station that alerts the station to the impending shutdown. This critical situation message is sent to each active station in whose TERMINAL macro you have specified CRITSIT=YES, provided that the line to that station is currently operating in network control mode. Define the content of the message in the CSMSG and CSMSGC operands of the BUILD macro. A message header, if required for IBM 3270 terminals, may be specified in the CSMHDR and CSMHDRC operands of the BUILD macro.

The critical situation message always begins with the date and time (the latter is in 24-hour format) and ends with the text you specify in the CSMSG operand.

Example:

```
05/14/78 19.27.05 NETWORK SHUTDOWN IN PROGRESS-NO FURTHER
TRANSMISSIONS UNTIL NOTIFIED.
```

The NCP does not automatically send a message to notify stations when the network is again operational; this is the responsibility of the access method.

BSC Operation

The options described in this section apply only to binary synchronous (BSC) stations. Options common to both BSC and start-stop stations are described under the heading, *BSC and Start-Stop Operation.*

Transmission in Transparent Mode

The NCP is capable of sending and receiving data over a BSC line in *transparent mode.* Transparent mode allows transmission of message data containing any bit patterns, including those that the sending and receiving station normally recognize and act upon as line control characters.

For transmission to a station on a line operating in network control mode, the data communication request specifies whether the NCP is to send the message data in normal or in transparent mode. The program accordingly transmits, at the appropriate time, either the nontransparent control characters—STX, ETB, and ETX—or the transparent control sequences—DLE STX, DLE ETB, DLE ETX.

For transmission in network control mode from a station to the communications controller, the NCP automatically deletes the line control characters it detects in the received message data before forwarding the data to the host processor. The control information accompanying the data indicates to the host processor whether the NCP received the data in nontransparent mode or in transparent mode. If a station is on a line operating in emulation mode, the NCP does not insert and delete the required transparent control sequences. The access method must include all required control sequences in message data it delivers to the NCP for forwarding to a station. Conversely, the NCP delivers to the access method unchanged, transparent message data it receives from lines in emulation mode.

Intermediate Block Checking Mode

When receiving from a station in network control mode, the NCP automatically examines the block-check characters (BCC) that follow each intermediate-transmission-block (ITB) character, if any, in the received data. In the ITBMODE operand of the TERMINAL macro for the station, you specify whether the NCP is to send error information bytes (EIB) to the host processor following each ITB character. If you specify insertion of EIBs, the application program in the host processor can scan the received data for ITB—the error information byte—to determine whether an error occurred in the intermediate block.

When receiving from the host processor message data to be sent to a station in network control mode, the NCP can automatically remove EIBs, if any, before transmitting the message data. If you specify deletion, the NCP deletes the first character following each ITB it detects within the data to be sent. You should specify deletion only if each first-following character is an EIB; otherwise the receiving station will not receive the first data character of each intermediate block.

Intermediate Block Checking Mode for Transparent Text

If you specify XITB=YES in the BUILD macro, the NCP is capable of inserting ITB sequences and error information blocks in transparent text as well as in nontransparent text for BSC stations on lines operating in network control mode. Insertion of ITB sequences and error information blocks occurs only for those BSC stations whose TERMINAL macros specify, in the ITBMODE operand, use of the intermediate block-checking facility, as follows:

If the BUILD macro specifies XITB=YES and the *first* parameter of the ITBMODE operand specifies intermediate block checking, the NCP substitutes an error information block for each DLE ITB sequence in transparent text received from the station.

If the BUILD macro specifies XITB=YES and the *second* parameter of the ITBMODE operand specifies intermediate block checking, the NCP inserts DLE ITB sequences into transparent text being sent to the station. Special 2-byte fields within the text received from the host processor determine the intervals at which the DLE ITB sequences are inserted.

If the BUILD macro specifies XITB=NO (or you omit the XITB operand), no insertion of ITBs and EIBs is performed for any BSC stations.

This function is performed only if the communication line to a station is currently operating in network control mode.

ID Exchange and Verification

The NCP can receive an identification (ID) sequence from any BSC station that calls the communications controller over a switched line operating in network control mode and can either check that sequence against a list of valid sequences within the program or pass the sequence to the access method for checking. (If the access method is operating under VSE, sequences *must* be passed to the access method for checking.) Upon failing to recognize the sequence as valid, the NCP does not proceed with message transmission. Instead it either breaks the line connection or maintains the connection but forwards the unrecognized sequence to the host processor for checking against a list kept there. In the latter case, the host processor can signal the NCP to proceed with message transmission or break the line connection.

After either the NCP or the access method checks the received sequence, the NCP can send an ID sequence in reply.

Advantages of NCP Verification versus Access Method Verification

If the NCP verifies a received ID sequence, message transmission can begin sooner than if the ID must be forwarded to the access method for checking. But the storage space needed within the NCP to maintain a list of ID sequences can be considerable if there are many sequences. Conversely, if the access method maintains the list, storage requirements within the controller are minimized, but ID sequence checking by the access method may take longer.

A compromise, if OS/VS VTAM or TCAM is being used, is to keep within the network control program the ID sequences for the stations that call most often, and to keep within the host processor those sequences representing stations that call less frequently. (If VSE VTAM is the access method used, all sequences received from BSC stations *must* be passed to VTAM.)

The various ID verification options for checking by the NCP are specified with the IDSEQ operand of the TERMINAL macro and the NOMATCH operand of the IDLIST macro. ID verification by VTAM is specified by the (VTAM-only) VIDLIST macro. See the *ACF/TCAM Installation* manual for information on ID verification by TCAM.

Defining a Controller ID Sequence

If you choose to define an ID sequence within the controller to be sent in response to ID sequences received from BSC stations, specify the sequence in the CUID operand of the BUILD macro. The maximum length is 20 characters. However,
because different types of stations may expect ID sequences of different lengths, you must specify the required length in the CUIDLEN operand of the TERMINAL macro for each station to which the sequence is to be sent. If you omit the CUIDLEN operand, no controller ID sequence is transmitted.

Controller ID sequences are not sent to stations on lines currently operating in emulation mode.

Sending and Receiving WACK Sequences

When receiving message data from a BSC station in network control mode, the NCP may need to temporarily defer further input from that station. This can happen, for example, when the NCP has no further read requests from the host processor for receiving additional message data. When this occurs, the program responds to the block just received with a WACK (wait-acknowledgment) sequence instead of the usual positive acknowledgment (ACK-0 or ACK-1). The WACK sequence informs the sending station that the NCP is deferring the positive acknowledgment until it is again able to receive from the station. Upon receiving the WACK sequence, the station replies with an ENQ character.

Exchanging of WACK and ENQ sequences can continue for as long as the NCP needs to defer input from the station or until the station breaks the connection. When the program is ready to resume receiving from the station, it sends the deferred positive acknowledgment (ACK-0 or ACK-1).

There is no limit to the number of WACK sequences the NCP can send. The program does, however, limit the number of WACK sequences it accepts from the station unless you wish to allow the program to receive them without limit.

Unless you specify otherwise, the NCP accepts up to 15 consecutive WACK sequences from a station. You may specify a different number, or specify unlimited acceptance of WACK sequences, in the WACKCNT operand of the GROUP macro.

Sending and Receiving Temporary Text Delay Sequences

When the NCP must temporarily suspend *sending* to a station in network control mode, it need not break the logical connection. Instead it can transmit a *temporary text delay* (TTD) sequence in lieu of the next message block. The TTD sequence informs the receiving station that the communications controller will continue sending after a short pause. The station replies to the TTD sequence with a NAK character. Exchange of TTD and NAK can continue as long as the program needs to defer transmission or until the station breaks the logical connection.

There is no limit on the number of TTD sequences the NCP will send. The program does, however, limit the number of TTD sequences it will accept from a station unless you wish to allow unlimited acceptance. When the limit is reached, the program breaks the logical connection.

Unless you specify otherwise, the program will accept up to 15 consecutive TTD sequences from a station. You may specify a different limit or specify that the program is to accept them without limit in the TTDCNT operand of the GROUP macro.

Frequency of Transmission of Synchronous Idle (SYN) Characters

In binary synchronous communications, a *synchronous idle* (SYN) character must be transmitted on the communication line at regular intervals to maintain the sending and receiving stations in synchronism. Binary synchronous stations transmit these characters periodically when sending message data. (A sequence of SYN characters is also transmitted when the line is otherwise idle.)

The NCP conforms to normal BSC practice by sending the SYN characters once each second. In rare circumstances it may be appropriate to change this interval; this can be done with the SYNDLAY operand of the GROUP macro. (The change will be effective only for lines serviced by a type 2 scanner. The type 3 scanner hardware sends SYN characters at one-second intervals.) This function applies only to lines operating in network control mode.

Start-Stop (Asynchronous) Operation

The options described in this section apply only to start-stop terminals (also called asynchronous terminals). Options common to both start-stop and BSC stations are described earlier in this chapter under the heading, *BSC and Start-Stop Operation.*

Multiple Terminal Access Facility

A major feature of the NCP is its ability to communicate in network control mode with a variety of dissimilar, commonly used start-stop terminals over the same switched network connection, or "port." This feature, called the *multiple terminal access* (MTA) facility, makes it unnecessary to reserve a separate port for each type of terminal as has often been the case in networks. This facility therefore serves to minimize the number of communication lines and their attendant modems and line-attachment hardware, resulting in lowered communication costs. At the same time, greater utilization of the remaining lines is achieved.

These types of terminals are accommodated by the multiple terminal access facility:

IBM 1050
IBM 2740 (basic)
IBM 2740 with Record Checking
IBM 2740 with Transmit Control
IBM 2740 with Transmit Control and Checking
IBM 2741 or IBM 3767 (in 2741 compatibility mode)
Western Union TWX

Note: In an MTA environment, the 2741 and the 3767 (in 2741 mode) are mutually exclusive.

The multiple terminal access facility allows the NCP either to call any MTA terminal over a line defined as a call-out line or to receive calls from MTA terminals over a line defined as a call-in line. The same line may be used for both call-out and call-in MTA operation. Lines used for call-out operation are included in dial sets, as for non-MTA operation.

MTA support for TWX terminals provides for discrimination of 110 baud and 300 baud terminals and can associate one of three translation tables (ASCII, DIC1, or DIC3) with a TWX terminal according to the first character entered at the terminal.

When answering calls over a line defined as an MTA line (specified by the MTALIST operand of the LINE macro), the program automatically determines the type of terminal in terms of its line control discipline and transmission code employed. Once these have been determined, the NCP carries on sessions and logical connections in the usual way.

In addition, by analyzing a code entered by the terminal operator when calling the controller, the NCP can distinguish among terminals that, while of the same type, require differing terminal or line operating parameters or procedural options.

These parameters and options are:

- The carriage return rate
- The presence or absence of the accelerated carrier return (ACR) feature for 1050 terminals
- The length of the print line used by the terminal printer
- The line speed, interrupt priority, modem data rate, and bit clocking options
- The maximum number of sub-blocks to be accepted from the terminal during a single logical connection
- The maximum number of attempts to recover from text-mode errors

To use the multiple terminal access facility, you (1) define the types of terminals to be handled as MTA terminals (these may be any combination of the terminals listed above) and (2) specify the lines with which the NCP will communicate with each of the types of terminals. All lines used for MTA operation must operate in network control mode.

Your requirements for the MTA facility are specified in the MTALCST, MTALIST, MTATABL, and MTAPOLL macros; the MTALIST operand of the LINE macro; the LCST and TERM operands of the TERMINAL macro; and the MTARTO and MTARTRY operands of the BUILD macro. An example of the use of MTA facilities appears in Appendix J.

Transmission of Attention Signals

The NCP, when transmitting to a terminal in network control mode, can respond to attention signals received from the terminal in either of two ways. (1) The program can interrupt its transmission to the terminal and immediately notify the host processor that the terminal has sent the attention signal. The program halts any remaining requests for that terminal. It is then up to the host processor to determine the next operation for the terminal. (2) The NCP can ignore the interrupt and continue sending to the terminal.

In the FEATURE operand of the TERMINAL macro for each terminal equipped to send attention signals (IBM 1050, 2741; AT&T 83B3; WU 115A, TWX), specify the ATTN parameter. If you wish the NCP to interrupt its transmission upon receiving the attention signal, specify ATTN=ENABLED in the TERMINAL macro. If you wish the program to ignore the signal, specify ATTN=DISABLED (or omit the ATTN operand). The foregoing applies to attention signals received while the NCP is transmitting to a terminal. The program can also monitor a communication line operating in network control mode for an attention signal or a disconnect condition detected while the program is momentarily executing no read or write commands for the line, provided that an active session is in progress with that terminal. The program notifies the access method that it has detected an attention signal or a disconnect condition.

Specify whether this function is required in the MONITOR operand of the LINE macro. The option is required if terminals on the line will communicate with TSO applications. (The NCP performs the monitoring function only if directed to do so by command from the access method. TSO causes the access method to send the required command.)

Logical Keyboard Lock for TWX Terminals

The keyboard of a TWX terminal, unlike those of other start-stop terminals, cannot be locked by the NCP when no read or write operation is in effect for the terminal. (Locking the keyboard prevents data from being entered when the NCP is not ready to receive from the terminal.)

The NCP may be requested to send TWX terminals a character sequence that "jogs" the printing mechanism when the program is not ready to receive. This serves as a signal to the terminal operator not to enter data.

The character used to jog the TWX printer mechanism is specified in the KBDLOCK operand of the GROUP macro. Any character specified should be a non-printing, non-spacing character that jogs the printer mechanism.

Note: The control program also provides an internal safeguard against loss of data due to the unlocked keyboard by automatically executing a Read command when no other command is executing and not in the monitor mode.

Carriage Return Delay

A terminal operator may press the return (carriage return) key of the terminal at the end of a message block he is entering from the keyboard. If the NCP sends message data to the terminal immediately after receiving the block the terminal has just sent, the first several characters of the data the program sends may be printed randomly during the return motion of the terminal's printing mechanism. To prevent this from happening, you may specify that the program pause momentarily after completing a read operation before starting the next write operation. This allows time for the printing mechanism to return to the left margin. Specify this function in the CRDLAY operand of the TERMINAL macro that represents the terminal printer. The program pauses between the read and write operations only if the message block received from the terminal ended with a carriage return (new line) character or, for an IBM 1050 equipped with the automatic EOB feature, an EOB character.

This carriage return delay function is performed only for IBM 1050, 2740 Model 1, 2741 terminals, and TWX terminals (as specified in the TERM operand of the TERMINAL macro) and any multiple-terminal-access terminal (TERM=MTA).

Downshifting on Space Characters

Some AT&T 83B3, Western Union 115A, and World Trade teletypewriter terminals, upon sending or receiving a space character, automatically downshift so that subsequent message text is in lowercase mode. Automatic downshifting avoids the need to send a LTRS character to effect downshifting. In the LINE macro for each teletypewriter line, indicate whether the terminals are equipped with the downshift function. Specify this function in the SPSHIFT operand if the line is to operate in network control mode and in the FEATURE operand (SPACE parameter) if the line is to operate in emulation mode.

Deleting FIGS and LTRS Characters

Message data received from 83B3, 115A, and World Trade teletypewriter terminals contains the 2 case-shifting characters FIGS and LTRS. If the lines to which such terminals are attached operate in control mode, the NCP removes FIGS and LTRS characters from the data it transfers to the host processor unless you specify, in the FGSLTRS operand of the LINE macro for such terminals, that the program is to leave these characters in the data. (The characters are not deleted from message data received over lines operating in emulation mode.)

TWX ID Exchange and Verification

The NCP can recognize an identification (ID) sequence from any TWX terminal that calls or is called by the communication controller over the switched telephone network; the NCP can either check that sequence against a list of valid sequences within the program or pass the sequence to the access method for checking. (If the access method is VSE VTAM, sequences must be passed to VTAM for checking.) Upon failing to recognize a sequence as valid, the NCP does not proceed with message transmission. Instead it either breaks the line connection, or it maintains the connection but forwards the unrecognized sequence to the access method for checking. In the latter case, the access method can signal the NCP to proceed with message transmission or to break the line connection.

After either the network control program or the access method checks the received sequence, the NCP can send an ID answerback sequence to the terminal before receiving text from the terminal.

Defining an Answerback Sequence

In the TWXID operand of the BUILD macro, you may specify the answerback sequence to be sent to TWX terminals. Two different sequences may be specified: one to be sent when a terminal calls the controller and the other to be sent when the controller calls a terminal. The maximum length of either sequence is 20 characters. In the CUIDLEN operand of the TERMINAL macro for each TWX terminal, you must specify the length of the answerback sequence to be sent. If you omit the CUIDLEN operand, the program does not send the answerback sequence to that TWX terminal.

Answerback sequences are not sent to TWX terminals on lines operating in emulation mode or nonswitched lines operating in network control mode.

Options for World Trade Teletypewriter Terminals

In addition to the downshift-on-space character and FIGS/LTRS options mentioned above, there are two other procedural options for World Trade teletypewriter terminals.

Pad Characters

Some World Trade teletypewriter terminals have a motor that runs continuously whether or not the terminal is sending or receiving data. Others have motors that run only during data transmission; the motor stops automatically after about 10 seconds have elapsed since the terminal sent or received a character. Terminals of the latter type must receive several pad, or idle, characters before receiving message characters to allow sufficient time for the motor to reach operating speed. The number of characters required depends on the data rate on the communication line.

For lines running in emulation mode, the access method is responsible for including the appropriate number of idle characters in message data it sends to the terminal.

For lines running in network control mode, the NCP automatically sends the idle (pad) characters if you specify the required number in the PADCNT operand of the GROUP macro for the group of lines to which such terminals are attached.

EOB and EOT Sequences

You may specify the character sequence the NCP is to recognize as the end-of-block (EOB) and end-of-transmission (EOT) sequences when receiving from a terminal.

The EOB sequence may be either FIGS *x* or *nnnn*. *x* and *n* may be any code combination except a combination representing the FIGS or LTRS character. If the terminal is equipped to send who-are-you (WRU) sequences, *x* also may not be the letter D.

The EOT sequence may be FIGS *y* LTRS; *y* may be any code combination except one representing FIGS, LTRS, or the same *x* character used in the EOB sequence, FIGS *x*.

Specify the required EOB and EOT sequences in the EOB and EOT operands of the GROUP macro if any of the lines in the group are to operate in emulation mode and in the WTTYEOB and WTTYEOT operands of the GROUP macro if any of the lines are to operate in network control mode.

Procedural Options for Operations in Emulation Mode

There are three procedural options when defining a line that always operates in emulation mode: (1) the type of line control discipline to be used for each line; (2) the terminal timeouts required, and (3)—for World Trade teletypewriters only—the end-of-block and end-of-transmission sequences to be recognized by the program.

Type of Line Control

All types of stations with which the communications controller can communicate in emulation mode use one of two line control disciplines: binary synchronous (BSC) and start-stop (or asynchronous). Each line attached to the controller uses either BSC or start-stop line control; the same line never uses both types.

The type of line control discipline used is specified in the LNCTL operand of the GROUP macro. (All lines in a group must use the same line control discipline.)

Terminal Timeouts

The NCP normally observes for each communication line two *timeout* intervals of several seconds' duration. One of these intervals is the *reply timeout,* which limits the amount of time the program will await a station's response to polling or response to message data sent to the station. The other interval is the *text timeout,* which limits the time that may elapse between receipt of successive message characters from the station after message transmission has begun. If the timeout expires before the response or the next message character is received, the program ends the read operation for that station and notifies the access method of a timeout error. These timeouts apply to each line in the network and prevent a communication line from being idled indefinitely.

Unless you specify different values in the REPLYTO and TEXTTO operands of the GROUP macro, the program uses the timeout intervals indicated in the descriptions of these two operands for all lines in the group represented by that macro. Some applications may justify unlimited intervals, that is, no timeout at all. No timeout may also be specified in the REPLYTO or TEXTTO operands.

EOB and EOT Sequences for World Trade Teletypewriter Terminals

You may specify the character sequence the NCP is to recognize as the end-of-block (EOB) and end-of-transmission (EOT) sequences when receiving from a terminal.

The EOB sequence may be either FIGS *x* or *nnnn*. *x* and *n* may be any applicable telegraph code combination except a combination representing the FIGS or LTRS character. (If the terminal is equipped to send who-are-you (WRU) sequence, *x* also may not be the letter D.)

The EOT sequence may be FIGS *y* LTRS; *y* may be any applicable telegraph code combination except one representing FIGS, LTRS, or the same *x* character used in the EOB sequence, FIGS *x*.

Specify the required sequences in the EOB and EOT operands of each GROUP macro representing a World Trade teletypewriter (teleprinter) line group.

Note: Appendix H lists the transmission code bit patterns for the ITA2 and ZSC3 codes.

Multi-Subchannel Line Access Facility

The multi-subchannel line access (MSLA) facility of the PEP extension allows the program to communicate in emulation mode over two type 4 channel adapters concurrently. The channel adapters may both be attached to the same host processor or may be attached to separate processors. The MSLA facility further allows two or more host subchannels (on the same or different channels) to communicate alternately with the same communication line. In operation, a command issued over one of the subchannels seizes the line for use of that subchannel and the access method using that subchannel. The access method retains use of the line via that subchannel until it issues a Disable command, thus releasing the line for use by another subchannel. (Alternately, the 3705 control panel can be used to release a line from control of one subchannel in order to switch it to another subchannel. This action is required if the access method using the line does not issue Disable commands.)

Subchannel-to-line associations are established during program definition and can be changed only by regenerating the program.

The physical characteristics of the line (such as type of line control, line speed, etc.) remain constant regardless of which subchannel is currently using the line. The use of the line by each subchannel must be consistent with the line characteristics. Violation of this requirement will cause unpredictable results when the access method communicates with the line.

The MSLA facility can be used in the following ways:

- Load balancing—communication lines can be switched from one host processor to the other during high-traffic periods to balance the load on the processors.
- Host processor backup—communication lines can be switched to a backup host processor if the original host processor, channel, or access method fails. Execution of the control program does not end, and the program need not be reloaded into the communications controller.
- Line sharing—two access methods in the same or different host processors can share the same communication line alternately. The same line can thus be assigned to different applications at different times of day.

The description of the ADDRESS operand of the LINE macro explains how to associate subchannels with a line.

Block-Handling Options

Block handling refers to the optional message processing of message data within the communications controller. The NCP can process either message data from the host processor before sending it to a start-stop or BSC station or message data received from a station before sending it to the access method. Processing is possible only when the message data between controller and station is transmitted in network control mode.

The IBM-supplied network control program modules provide two standard message-processing functions. Each is performed by a block-handling routine invoked by a particular program generation macro instruction. In addition, user-coded block handling routines may be added to the network control program during the generation procedure. A program generation macro, UBHR, allows you to invoke the user block-handling routines in the same way as IBM-provided block-handling routines. (The IBM-supplied block-handling routines cannot be included in a user block-handling routine.)

Two optional IBM-supplied block-handling routines allow for insertion of date and time-of-day into messages and automatic correction of incorrect message text.

Insertion of Date and Time

The NCP can insert the current date, or time of day, or both, into message blocks it receives from the access method over the network control subchannel or from a station over a line in network control mode.

The date may be in any of four formats: (1) month/day/year, for example, 10/21/78; (2) year followed by day of year, for example, 78. 294 (October 21, 1978); (3) year/month/day (78/10/21): or (4) day/month/year (21/10/78).

The time of day is in the format hh. mm.ss. (hours, minutes, seconds). The continental (24-hour) form is used. For example, 09. 17.25 and 21. 17.25 represent 9:17:25 a. m. and 9:17:25 p. m., respectively. (Each format is preceded by an EBCDIC blank character.)

The date and time may be placed in the first block of each message or transmission, or in every block of the message or transmission.

Date and time insertion is specified with the DATETIME macro.

Automatic Text Correction

Automatic text correction is an editing function by which the NCP replaces text incorrectly entered from a terminal keyboard with the corrected characters the terminal operator subsequently sends. The program does this by scanning each block for predefined characters called text canceling characters. The NCP deletes from the block each such character it finds, plus 1 preceding text character. For example, if the program finds a sequence of 3 canceling characters, it deletes the 3 characters plus the 3 immediately preceding characters.

A keyboard operator may, for instance, enter COMMUNCIATE and, seeing that he has misspelled it, enter 5 backspace characters to "back up" to the first erroneous character. Then he re-enters the corrected characters, thus:

COMMUNCIATE bksp bksp bksp bksp bksp ICATE

If you have specified the text correction option and designated *backspace* as the text-canceling character, the text correction block-handling routine deletes the five backspace characters and C I A T E. The remaining characters form the correctly spelled word COMMUNICATE.

The text-canceling character need not be a backspace character. Any other character (except a line control character) is adequate if it is not used in any other way within message text. For example if / is the character chosen and a keyboard operator enters ATLANITC///TIC, the text correction block-handling routine corrects the word to ATLANTIC.

The EDIT macro specifies the text correction function.

User Block-Handling Routines

Any block-handling routine you provide is referred to as a *user* block-handling routine. You code a user block-handling routine using the communications controller assembler language (similar to the operating system assembler language), assemble it using the controller assembler, and then place the routine in a data set available to the NCP generation procedure. Then you include in the program generation source statements a UBHR macro instruction that specifies the name of your routine and the point at which the network control program is to execute it.

Guidelines for Writing User Block-Handling Routines

User block-handling routines permit you to add certain data-handling functions to the network control program. These routines typically examine and manipulate incoming or outgoing data contained in NCP buffers. If you have a good general understanding of the network control program and the access method, you may add such routines to the program with little likelihood of disrupting the NCP code. On the other hand, routines that perform more complex functions, such as leasing and releasing buffers or scheduling input/output operations, require that you have an intimate understanding of the internal operation of the network control program and the access method. Adding such routines must be approached with caution to avoid disrupting the network control program logic.

Coding user block-handling routines requires knowledge of the information in these publications: *IBM 3704 and 3705 Communications Controllers Principles of Operation, IBM 3704 and 3705 Communications Controller Assembler Language, ACF/NCP Logic Manual.*

The assembled object modules containing the user block handling routines must be placed in the data set specified by the USERLIB operand of the BUILD macro. During stage 2 of the program generation procedure, user block-handling routines that you have specified in the appropriate UBHR macros are included in the generated NCP load module.

Specific rules and guidelines to be observed in coding user block-handling routines are as follows:

- All registers may be used in a user block-handling routine. Before passing control to a user routine, the network control program saves all registers; when receiving control from the routine, the NCP restores all registers.
- At entry to a user block-handling routine, register 2 points to the queue on which the block being handled is enqueued.
- A POINT (2), (3) macro instruction causes the address of the block at the head of the queue to be returned in register 3.
- You may use a SCAN macro instruction to scan the text in chained NCP buffers containing the block being processed.
- You may use the DEQUE, ENQUE, and INSERT macros to dequeue, enqueue, or insert the block whose address was returned by the POINT macro instruction.
- You may use the LEASE macro to obtain NCP buffers you must release any buffers thus obtained with a RELEASE macro.
- Use a SYSXIT macro to return control from the user block-handling routine to the network control program.
- If a user block-handling routine is to be executed for more than one BSC or start-stop station or line, code it such that it is serially reusable.
- The ACF/NCP Logic manual describes the macro operand formats and gives details for use of supervisory macros.
- If the user routine changes the amount of message text accompanying a BTU (basic transmission unit), the routine must accordingly update the BCUTLEN field of the BTU and the data count fields of the buffer prefix areas.
- User block-handling routines should not modify any part of the first 34 bytes (the BTU) of a header buffer or the first 4 bytes of any other buffer.
- Logic errors encoutered in user block-handling routines can cause the NCP to end abnormally (abend).

For many kinds of logic errors, a dump listing of the NCP will reveal: (1) The level 5 instruction address register (IAR—register 0) will point within the user block-handling routine. (2) Bit 4 of storage location X'685' (indicating block handlers in execution) will be on. (3) The abend code will appear at location X'760'. (Abend codes are described in the *ACF/NCP Program Reference Summary).*

Associating Block-Handling Routines with Stations

The requirements of the application determine how the NCP is to process messages before sending them to the network or the access method. The requirements may vary for different stations or for different components of a station. You may wish, for example, to provide the text-correction function for messages entered from a terminal keyboard but not for messages received from a paper tape reader. Or, you may wish to insert time and date information in messages received from station A but not in those received from station B.

Network control program generation macro instructions provide a means of grouping individual block-handling routines into block handlers and for combining block handlers into block-handler sets. Block-handler sets can then be associated with individual stations or station components.

Each block handler within a set can be executed at a different logical point in the flow of message data through the NCP. For instance, one block handler in the set can be executed immediately upon arrival of a message from the host processor before the NCP has obtained the use of a communication line for transmitting the message to the station. Another block handler in the same set can include routines that process message data from the host processor *after* the program has obtained use of a line. This block handler may also include routines that process message data from a station before the NCP releases the line over which it received the data.

A third block handler in the set may be assigned to process message data received from a station after the program has released the line for use in communicating with another station.

The network control program generation macro instructions for grouping block handling routines into block handlers are STARTBH and ENDBH. A third macro, BHSET, combines block handlers into sets.

To assign block handler sets to stations or station components, you code the name of the set in the BHSET operand of the appropriate TERMINAL, COMP, or CLUSTER macro instruction. In the BHEXEC operands of the same macros you specify which block handlers within the set are to executed at the logical points in the message flow.

Diagnostic and Service Aids for Emulation Mode

Two additional service aids are available with the PEP extension: (1) line trace for emulation lines and (2) dynamic dymp facility. The dynamic dump facility is optional; however, you should include it in the network control program.

Line Trace Facility for Emulation Mode

The emulation mode line trace facility of the program is a service aid that permits detailed analysis of the operation of any communication line currently operating in emulation mode. This facility records operating parameters of a line each time a level 2 interrupt (except bit-service interrupt) or level 3 interrupt occurs for that line. (Level 2 is the program level at which bit service or character service for the communication line is performed. Level 3 is the program level at which the servicing of channel interrupts is performed.) The program accumulates this information within controller storage. The emulation mode line trace, unlike the line trace for network control mode, does not accumulate the trace information in buffers and does not automatically transfer the buffer contents to the host processor. The contents of the controller storage must be dumped to make the line trace records available or dynamically dumped using the dynamic dump utility.

The line trace facility does not interfere with normal operation of the communication line. Performance may diminish somewhat because of the additional processing needed each time a character service or level 3 interrupt occurs for the line or lines being traced. The amount of decrease in performance depends upon how heavily the communications controller is currently loaded and how many lines are being concurrently traced. The line trace facility has no effect on performance except when a line is actually being traced.

Line traces of lines in emulation mode are initiated by the controller operator at the control panel of the communications controller or via the dynamic dump utility. Any number of lines may be traced concurrently.

The line trace facility for emulation mode is always present in a network control program with the PEP extension. The number of lines to be traced and the size of the trace table are specified in the LINETRC operand of the BUILD macro.

Dynamic Dump Facility

The dynamic dump facility is a service aid that transmits communications controller storage contents to the host processor over an emulation subchannel without stopping execution of the NCP. A full storage dump or a dump of the trace table can be obtained. Additionally, the emulation mode line trace can be activated, deactivated, or modified. Portions of controller storage can also be displayed on the operator's console at the host processor.

The DYNADMP operand of the BUILD macro specifies whether the dynamic dump option is to be included in the NCP and specifies the emulation subchannel addresses over which the controller storage contents are to be dumped. Each channel adapter in the controller can have one host subchannel address assigned for this purpose; the assigned subchannels cannot be used for communicating with any line in the network.

The dynamic dump facility can be used only with a network control program that includes the PEP extension.

Chapter 4. Generation and Catalog Procedures

Program Generation under OS/VS

The control program generation under OS/VS is a two-stage process. The procedure creates a load module that executes in the 3705 according to your particular configuration needs. NCP generation can be done on any host processor. It does not have to be done on the processor that will control the network.

Stage 1 is an assembly job using either the communications controller assembler (CWAX00) or an OS/VS assembler. Due to the difference in assembly time, it is recommended that you use the OS/VS assembler for the stage 1 assembly. This stage uses the generation macro instructions that you have coded to prepare a job stream for input to stage 2. The output may be placed on cards, tape, or a direct-access device. The stage 1 output data set contains (1) data constants, (2) macros that cause stage 2 to generate the control tables and conditionally assemble the required program modules, (3) job control statements for stage 2, and (4) linkage editor control statements.

Stage 2 first uses the communications controller assembler (CWAX00) to assemble the control tables and program modules that require conditional assembly. The resulting object modules are placed on the object library. Stage 2 then link-edits these modules, along with other preassembled modules (located in SYS1.OBJ3705), into an NCP load module and places it on the load library. From this library, either the access method loader or the independent loader provided in the system support programs can load the control program into the communications controller. The object library and the load library used in these steps are the libraries specified in the OBJLIB and LOADLIB operands in the BUILD macro.

Stage 2 also produces a *resource resolution table* load module and, if you have coded any block-handling routines, a *block handler set resolution table* load module. Stage 2 then places them on the library specified in the LOADLIB operand. These load modules contain information required by the access method.

Note: If user-written code is included in the NCP via the INCHI operand of the GENEND macro, only two load modules are generated. The block-handler set modules and the user-written modules become part of the NCP load module.

Operator intervention is required between the two stages of program generation. Diagnostic messages produced at the end of stage 1 indicate any errors that may have occurred. If these are serious errors, no job stream is produced. The source statements must be corrected, and stage 1 must be re-executed. If no serious errors occur in stage 1, the operator initiates the second stage, using the stage 1 output as input.

Caution: Because the VTAM initialization does no validity checking of parameters, it is imperative that the NCP source statements be entirely free of errors before being given to the VTAM initialization procedure. Therefore, the NCP must be reassembled, if necessary, until the stage 1 output listing shows no MNOTE statements having severity codes of 4 or 8.

Figures 4-1 and 4-2 show the contents of the stage 1 input job stream and stage 1 output (stage 2 input) job stream using the communications controller assembler.

The partial generation procedure is the same as for a complete generation. The only difference is that fewer modules are conditionally assembled in a partial generation; in some cases only the control tables are reassembled.

The same source deck used to generate a given control program may be modified and used to generate a different program. Care must be exercised in specifying program and library names associated with the subsequent program. If the new program is to replace the original one and the original object modules in the object library will not be needed, the NEWNAME, LOADLIB, and OBJLIB operands of the BUILD macro in the source deck need not be changed. The subsequent control program load module will be cataloged in place of the original. If both the original and the subsequent load modules will be needed, either change the NEWNAME operand to specify a different name or change the LOADLIB operand to specify a different library.

The object modules associated with the original load module should be saved before submitting the modified source deck to the generation procedure to create a new load module. Failure to save the old object modules will cause them to be replaced in the object library with the modified object modules; this would prevent your later regenerating the original program should you wish to do so. To save the object modules, specify a different data set name in the OBJLIB operand of the BUILD macro before submitting the revised source deck to the subsequent generation procedure.

An alternate method of saving the object modules is by using the OBJQUAL operand of the BUILD macro. This operand allows you to specify a unique name to the object modules of each generation. (See the OBJQUAL operand of the BUILD macro.)

Providing User Job Cards

The format of the stage 2 job cards produced when you specify JOBCARD=YES or JOBCARD=MULTI in the BUILD macro is:

```
//NCPGENnn    JOB    1,'NCPSYSGEN',MSGLEVEL=1
```

where *nn* is a sequential identification number provided by the program generation procedure. The job card may be changed by using the OS/VS IEBUPDTE utility program before executing the stage 1 assembly. (Refer to the *OS/VS Utilities* manual, GC35-0005, for information on the IEBUPDTE program.)

When you modify the job statement, the *name* parameter must be *jobname* &SNOA. *jobname* consists of 1 to 6 alphanumeric characters (including @, #). The first character must be alphabetic, @, or #. &SNOA is a counter that is incremented by the program generation procedure to provide unique job names. (See the *OS/VS Job Control Language* manual, GC28-0618, for information on the job statement.)

The following example illustrates how the job statement may be changed:

```
//CHNGJOB  JOB     (G40,060,SG,-,2),name,MSGLEVEL=1
//UPDATE   EXEC    PGM=IEBUPDTE
//SYSPRINT DD      SYSOUT=A
//SYSUT1   DD      DSN=SYS1.GEN3705,DISP=OLD
//SYSUT2   DD      DSN=SYS1.GEN3705,DISP=OLD
//SYSIN    DD      *
./         REPL    NAME=JOBCARD,LIST=ALL
./         NUMBER  NEW1=100,INCR=100
           PUNCH   '//mygen&SNOA  JOB (81,3,B62),myname,'
           PUNCH   '// MSGLEVEL=(1,1),CLASS=A'
./         ENDUP
/*
```

```
//STAGE1        JOB             MSGLEVEL=1
//STEP1         EXEC            PGM=CWAX00
//SYSLIB        DD              DSN=SYS1.GEN3705
         .
         .
    (JCL statements for assembler)
         .
         .
//SYSIN         DD              *
         .
         .
                BUILD           ...
                                ...
         .                                  Control
         .                                  Program
        (etc.)                              Generation
         .                                  Macro
         .                                  Statements
                GENEND          ...
                END
/*
```

Figure 4-1. OS/VS Generation Stage 1 Input

```
//NCPGEN1  JOB    MSGLEVEL=1,...  //S1EXECPGM=CWAX00,...
(Communications controller
                                  assembler) //SYSLIBDD
DSN=SYS1.MAC3705
                                    Two assembly job steps for
                                    control tables plus   (JCL
                                    statements for assembler)one
                                    assembly job step for each
                                    program module requiring
                                    conditional //SYSINDD*
                                    assembly (JOB card is
                                    provided for each assembly
                                    job step only if   (Data for
                                    conditional assembly)
                                    JOBCARD=MULTI is coded in
                                    BUILD macro

/* //S2    EXEC   PGM=CWAX00,...  //SYSLIBDDDSN=SYS1.MAC3705

(JCL statements for assembler)

//SYSIN    DD     *

(Data for conditional assembly) /*

//Sn       EXEC   PGM=IEWL,...  (First linkage editor job step)

(JCL statements for OS/VS linkage editor)

//SYSIN    DD     *

(INCLUDE statements for linkage editor)

/* //

 (INCLUDE statements specify object modules obtained from
 SYS1.OBJLIB and object modules obtained from the library
 specified in the OBJLIB operand of the BUILD macro.  The load
 module is placed on the library specified in the LOADLIB
 operand.)
```

Figure 4-2. OS/VS Generation Stage 1 Output (Stage 2 Input)

Program Generation Under VSE

The control program generation under VSE is a three-stage process. The procedure creates a load module that executes in the 3705 according to your particular configuration needs. NCP generation can be done on any host processor. It does not have to be done on the processor that will control the network.

Stage 1 of the generation procedure is a series of assembly jobs using the communications controller assembler (IFZASM) to prepare a job stream (sequential file) for input to stage 2. The file may be placed on cards, tape, or a direct-access device. The stage 1 output (stage 2 input) contains (1) data constants, (2) macros that cause stage 2 to generate the control tables and conditionally assemble the required program modules, (3) job control statements for stage 2, and (4) an assembly step that punches stage 2 statements.

Stage 2 assembles the control tables and those program modules that require conditional assembly; it then punches job control and linkage editor control statements.

Stage 3 catalogs the tables and modules assembled in stage 2 and link-edits them into a load module. This module is placed on the core image library. From there the CSERV utility must be used to move it to a user-defined file.

In addition to the load module produced by the linkage editor, unresolved external references may also be produced. These references will not cause program execution problems. See the section "Unresolved External References" in this chapter for additional information.

The VTAM loader facility or the independent loader utility can now obtain the load module from the file and load it into the communications controller.

Stage 3 also produces a *resource resolution table* load module and, if you have coded any block-handling routines, a *block handler set resolution table* load module. These load modules contain information VTAM needs for its initialization process; VTAM must obtain them from the library where they were placed in stage 3.

Note: If user-written code is included in the NCP via the INCHI operand of the GENEND macro, only two load modules are generated. The block-handler set modules and the user-written modules become part of the NCP load module.

Operator intervention is required between the stages of program generation. Diagnostic messages produced at the end of each stage indicate any errors that may have occurred. If these are serious errors, no job stream or partial job stream is produced. The source statements must be corrected, and the stage must be re-executed. If no serious errors occur, the operator initiates the next stage, using the output of the previous stage as input.

Caution: Because the VTAM initialization does no validity checking of NCP parameters, the NCP source statements must be entirely free of errors before being given to the VTAM initialization procedure. Therefore, stage 1 of the network control program must be reassembled, if necessary, until the output listing shows no MNOTE statements having severity codes of 4 or 8.

Figures 4-3, 4-4, and 4-5 show the contents of each input job stream using the communications controller assembler.

The partial generation procedure is the same as for a complete generation. The only difference is that fewer modules are conditionally assembled in a partial generation; in some cases, only the control tables are reassembled.

Partial generation is possible only if the relocatable library previously used during the complete generation procedure is available. You should, therefore, always save this library and the stage 1, stage 2, and stage 3 assembly listings produced by the complete generation.

Providing User Job Cards

The format of the stage 2 job cards produced when you specify JOBCARD=YES or JOBCARD=MULTI in the BUILD macro is:

```
// JOB  module name
```

If you specify JOBCARD=MULTI, you may provide different job cards before initiating the program generation procedure by submitting the following job:

```
// JOB         CHGJOBCD
// EXEC        ESERV
   GENEND
   DSPCH       F.ASMJCL
)  COL 73, 4
)  REP 1228
   PUNCH       '// JOB name'
)  END
/&
```

This job creates a new ASMJCL macro. You must then assemble the macro with IFZASM and catalog it.

Only the second and any subsequent job cards may be changed; the format of the first job card is unchanged.

```
//  JOB       jobname
//  PAUSE                (Before executing stage 1, assign appropriate
                         source statements and private relocatable libraries)
//  EXEC      IFZASM

    BUILD

    •                    NCP generation
    •                    macro
    •                    statements
    GENEND
    END
/&
```

Figure 4-3. VSE Generation Stage 1 Input

This input job stream is produced automatically by stage 1 of the generation procedure.

```
//  JOB      jobname
//  PAUSE                 (Before executing stage 2,
                           assign appropriate libraries)
//  OPTION   DECK
//  EXEC     IFZASM
    PUNCH    '// JOB    jobname'
    PUNCH    '// EXEC   MAINT'
    PUNCH    ' CATALR   module name'
      (Source code for conditionally assembled modules)
    END
/*
//  EXEC     IFZASM
    PUNCH    ' CATALR module name'
      (Source code for conditionally assembled modules)
    END

/*
    •
    • (Other conditional assemblies as in above step)
    •

//  EXEC     IFZASM
    PUNCH    ' CATALR   INITINCS'
    PUNCH    ' INCLUDE  ...' (INCLUDE statements for UBHR
                               modules, if specified)
    PUNCH    ' INCLUDE  ...' (INCLUDE statements for initial-
                               ization routines and tables)
    PUNCH    ' CATALR   LOADINCS'
    PUNCH    ' ACTION   MAP,NOAUTO'
    PUNCH    ' PHASE    phasename,+0'
    PUNCH    ' INCLUDE  ...' (INCLUDE statements for remainder
    •                   •      of program modules)
    •                   •
    •                   •
    PUNCH    '/*'
    PUNCH    '// OPTION CATAL' (If program includes UBHRs,
                                 the option is LINK)
    PUNCH    ' INCLUDE  LOADINCS'
    PUNCH    ' INCLUDE  INITINCS' (Omit this statement if
                                   program contains UBHRs)
    PUNCH    '// EXEC   LNKEDT'
    PUNCH    '/&'
    END
/&
```

Figure 4-4. VSE Generation Stage 2 Input

This input job stream is produced automatically by stage 2 of the generation procedure.

```
//  JOB         jobname
//  PAUSE       (Before executing stage 3,
                 assign appropriate libraries)
//  EXEC        MAINT
    CATALR      module name
       .
       . (Object code)
       .
    CATALR      module name
       .
       . (Object code)
       .
      (etc.)
       .
       .
       .
    CATALR      INITINCS
    INCLUDE     ...(INCLUDE statements for UBHR modules, if
                    present)
    INCLUDE     ...(INCLUDE statements for initialization
                    routines and tables)
    CATALR      LOADINCS
    ACTION      MAP,NOAUTO
    PHASE       phasename,+0  (INCLUDE statements for
    INCLUDE     module-1       remainder of
    INCLUDE     module-2       object modules)
       .
       .
       .
    INCLUDE     module-n
/*
//  OPTION      CATAL  (If program includes UBHRs, the option
                        is LINK)
    INCLUDE     LOADINCS
    INCLUDE     INITINCS (This statement omitted if program
                          contains UBHRs)
//  EXEC        LNKEDT
/&
```

Figure 4-5. VSE Generation Stage 3 Input

Including User-Written Modules

To include user-written code in the NCP load module, the code must be pre-assembled and placed on the appropriate object library. If you are operating under VSE, the object modules must be placed on the relocatable library. If your system is OS/VS, the object modules must be placed on the NCP object library (SYS1.OBJ3705) or a user object library. Modules on SYS1.OBJ3705 are in load module format. (See the USERLIB operand in the BUILD macro.)

An INCLUDE statement must be coded for each object module. These INCLUDE statements are then separated into the following categories:

1. Level 2 and 3 code that must reside in the lower 64K of controller storage.
2. Level 2 and 3 code that may reside anywhere in storage.
3. Code that is not level 2 or 3 and must reside in the lower 64K of controller storage.
4. Code that is not level 2 or 3 and may reside anywhere in storage.
5. Initialization code that is overload by the buffer pool when initialization is complete.

Place the INCLUDE statements on the macro library that contains the user control block source modules. You may place all of the INCLUDE statements within one category in one member of the macro library.

On an OS/VS system, each CSECT name in the user-written modules must be identified on a linkage editor ORDER statement. These ORDER statements must be placed on the macro library in the same manner as the INCLUDE statements. That is, you may place all of the ORDER statements within one of the previously named categories in one member of the macro library. Stage 1 of the generation procedure creates an INCLUDE statement for each member that contains an ORDER statement and inserts it into the proper place in the linkage editor job stream.

Any CSECT not specified on a linkage editor ORDER statement will be placed at the end of the load module and will be overlaid by the buffer pool once the initialization procedure is complete.

The names of the members that contain the INCLUDE and ORDER statements in the macro library must then be specified in the appropriate operand in the GENEND macro. (See the GENEND macro in Chapter 5.)

Including user-named control blocks in the NCP stage 2 generation causes an exposure to a duplicate label problem. The situation
pertains only to the control block assembly, not to the assembly of individual user modules.

It exists because the NCP was not designed with this inclusion in mind, and prefixes were not reserved for label used in the assembly. In order to eliminate this exposure, you should not use any of the following prefixes:

$	ACB	ATB	BCB	BCH
BCO	BCT	BCU	BOQ	CCB
CM	CRP	CTB	CX	DAE
DRS	DVB	DVI	DVQ	ICW
IOB	IRN	LCB	LCS	LCW
LGT	LKB	LXB	MDR	OLL
PAD	PCB	RCV	RG	RH-digit
RU-digit	SCB	SYS	STQ	TH
TVS	UIB	UNASGN	UI	X
CY	RN-number	R-number	ATP	PIU

The most effective solution would be to establish your own unique prefix for all of your control blocks.

Special Considerations for VSE Link-Edits

When link-editing under VSE with user-written code included in the NCP load module, it is necessary to perform one or two temporary link-edits before the final link-edit in order to obtain a load module for the core image library. The first link-edit is performed using modules that must reside in the low 64K of storage. The resulting load module from the first link-edit resides in the relocatable library. If the user has incorporated, into the first link edit, user written "low core" CSECTS then, the ALIGN macro needs to be assembled using the size of the linked module as the value of the HICORE operand. The ALIGN macro creates a dummy CSECT which causes a link-edited module to be padded to the next 2K boundary. If user code was included in the first link edit, the use of the ALIGN macro becomes necessary, since there is no way of determining the size of the user's CSECTs. A first or subsequent link edit must be performed, using as input the "low core" modules, and the first pad module, if applicable. The output from this link edit resides in the relocatable library. The ALIGN macro is assembled using the size of the link edited module as the HICORE operand value. The dummy CSECT, which is produced by the assembly of the ALIGN macro, is created so that user "high core" modules will begin on a 2K boundary. The modules in "high core" include initialization plus user block handler routines. The final link edit is performed, using as input the "low core" modules, the first pad CSECT, if applicable, the SAT, the second pad CSECT and the "high core" modules. The load module obtained from the final link edit is cataloged in the core image library. An explanation of the link edit process and the names of the pad modules can be found in the link edit portion of the STAGE-1 listing. The format of the ALIGN macro is:

```
ALIGN     HICORE=value
```

where *value* is the hexadecimal *hicore* value found in CXTEND of the stage 3 link edit. The following job stream is used to create the dummy object module.

```
//     JOB name
//     OPTION DECK
//     EXEC IFZASM
       ALIGN HICORE=value
       END
/&
```

Unresolved External References

Due to the many options available when generating a network control program with PEP, unresolved external references can appear in the linkage editor listings. These references are the result of certain options not being selected. The following list shows external references that may appear in your listing as unresolved. If any of these references do appear and you have not selected the option they apply to, the reference should be ignored.

	Unresolved references:	
For initial test and phase 1 loader:	SEL01	OLT03
	SEL2	OLT04
	SELFF	OLT05
	OLT00	OLT06
	OLT01	

	Unresolved references for:	
	Type 1 Channel Adapter	*Type 4 Channel Adapter*
If trace option is not specified:	CYATRCEI	CYATRCEI CYETRCRS CYETRCSP
If trace option is not specified and dynamic dump option is specified:	CYASETRC CYATABLE	CYASETRC CYATABLE
If trace option is not specified and BSC terminals are specified:		CYATRCL2 CYETRCL2
If dynamic dump option is specified:	$DSCCB	
If dynamic dump option is not specified:	CYADSTRT	CYADSTRT
If panel test option is not specified:	CYAPANLT	CYAPANLT
If panel test option is specified and no EBCDIC lines are serviced by a type 2 scanner:		CYAEBCDT
If panel test option is specified and no ASCII lines are serviced by a type 2 scanner:		CYAASCDT
If panel test option is specified and no ASCII lines are serviced by a type 2 scanner:		CYAASCDT
If only one type 4 channel adapter is specified:		CYECHCB2 CYECHVT2
If panel test option is specified and no BSC terminals are specified:		CYAEBCDT CYAASCDT
If panel test option is specified and no start-stop terminals are specified:		CYAXTABL

	Unresolved references for:	
	Type 1 Channel Adapter	*Type 4 Channel Adapter*
If start-stop terminals are specified but no display terminals are specified:	CYAATDA4	CYAATDA4
	CYAB28CL	CYAATDA5
	CYAB2848	CYAB28CL
		CYAB2848
If start-stop terminals are specified but no teletypewriter (83B3, 115A) or TWX terminals are specified:	CYATDONE	CYATSTYE
		CYASRCH
If start-stop terminals are specified but no DELAY or QUIETCT operand is specified:	CYADAT1	CYADAT1
	CYADAT2	CYADAT2
If no start-stop terminals are specified:	CYABARP1	CYABARP1
	CYABTDA0	CYABTDA0
	CYACBKPL	CYACBKPL
	CYACBRES	CYACBRES
	CYACPOLS	CYACPOLS
	CYACPRES	CYACPRES
	CYACRDCL	CYACRDCL
	CYACREAS	CYACREAS
	CYACSEAS	CYACSEAS
	CYACWRIS	CYACWRIS
	CYAMTBFR	CYAMTBFR
	CYAQUIET	CYAQUIET
	CYASTPER	CYASTPER
	CYATRN	CYATDONE
		CYATRN
		CYATSTYE
If BSC terminals are specified:		
If no type 2 scanner is specified:		CYARARS0
		CYATAPD0
		CYATAX10
		CYATBSWR
		CYATSTMW
		CYATXDA0
If no type 3 scanner is specified:		CYEABRTW
		CYEPRPRC
		CYERCVN
		CYERCVN1
		CYERCVP
		CYERCVPS

	Unresolved references for:	
	Type 1 Channel Adapter	*Type 4 Channel Adapter*
		CYERCVP1
		CYETXEND
		CYEXITB
		CYEXMITN
		CYEXMSYN
		CYEXMTEN
		CYEXPOLL
		CYEXTEND
If no BSC terminals are specified:		
	CYABSHIO	CYABSHIO
	CYABSTOP	CYABSTOP
	CYACADPB	CYACADPB
	CYACPOLB	CYACPOLB
	CYACPREB	CYACPREB
	CYACREAB	CYACREAB
	CYACSEAB	CYACSEAB
	CYACSETB	CYACSETB
	CYACWRIB	CYACWRIB
	CYARARS0	CYARARS0
	CYATAPD0	CYATAPD0
	CYATBSPL	CYATBSPL
	CYATBSPR	CYATBSPR
	*CYATBSRD	CYATBSRD
	CYATBSSM	CYATBSSM
	CYATBSWR	CYATBSWR
	CYATSTMW	CYATSTMW
		CYEPSUDO
		CYERCVPS
		CYERPLTO
		CYETMPMS
		CYETOTRN
		CYETXEND
		CYETXWTO
If no ALC lines (LNCTL=ALC) are specified:		CYETMALP
		CYETMALR
		CYETMALU

Note 1: External reference TM598 is defined in the CYASL210/310/320 modules.

Note 2: External reference CYAATDA5 is defined in the CYADSP10/20 modules.

Coding Sequence for Generation Macros

This section shows the required sequence of NCP generation macro instruction statements in the stage 1 input job stream. (See Appendix I for examples of several sample network programs.)

There are five distinct groups of program generation macro instructions. These groups must be submitted in the following sequence:

1. System Macro Instructions
2. Configuration Definition Macro Instructions
3. Network Configuration Macro Instructions
4. Block Handling Macro Instructions
5. Generation Delimiter Macro Instructions

The block handling macro instruction group is valid only if start-stop and/or BSC lines are included in the network configuration macro instructions.

The following charts show the names of all macros within a group and shows:

- Whether the macro is required or optional, and the number that must or can be coded.
- The operands are always required.
- The position of the macro within the group, when a specific sequence is required.

Only those operands that must always be coded are shown; the configuration and characteristics of the data communications network and the procedural options needed determine which other operands are needed. Refer to Chapters 2, and 3 for explanations of the characteristics and options; refer to Chapter 5 for descriptions of each of the macro instructions and operands.

(VTAM Users Only): The PCCU, VIDLIST, and VTERM macros are VTAM-only macros. They provide information only to the VTAM initialization process. Their presence in the network control program generation deck is not required (but is permissible) for the generation procedure, but must be included when the same deck is used as input to the VTAM initialization process. PCCU is always required; whether VIDLIST and VTERM are required depends upon VTAM application requirements. See the VTAM *System Programmer's Guide* for details of the meaning and use of these macros.

1-SYSTEM MACRO INSTRUCTIONS

Usage	*Macro*	*Required Operands*	*Macro Position*
Required for VTAM initialization: 1	PCCU (VTAM Users Only)	(See VTAM System Programmer's Guide for required operands)	VTAM-only macro: must precede BUILD macro for VTAM initialization
Required: 1	BUILD (for TYPGEN=NCP or TYPGEN=PEP)	LOADLIB= (OS/VS only) MAXSUBA= MEMSIZE= OBJLIB= (OS/VS only) SUBAREA= TYPGEN=	first macro in NCP source statements
Required :1	SYSCNTRL	OPTIONS=	directly following BUILD macro
Optional: 1 for each Network Addressable Unit	NCPNAU	NAUFVT=	directly following SYSCNTRL macro

Usage	*Macro*	*Required Operands*	*Macro Position*
Required: 1 for each access method to communicate with local NCP over channel. Not required for remote NCP.	HOST	INBFRS= MAXBFRU= UNITSZ=	macros in this group (2) may appear in any sequence except as indicated for MTALCST and MTALIST macros
Required: 1 for each scanner in controller	CSB	SPEED=	
Optional: 1 for each list of ID sequences to be checked by NCP	IDLIST	IDSEQ=	
Optional: 1 for each list of ID sequences to be checked by VTAM (required only for VTAM initialization)	VIDLIST (VTAM Users Only)	(See VTAM System Programmer's Guide for required operands)	
Optional: 1 for each LCST entry to be defined	MTALCST	GROUP= SPEED=	must appear in one sequence
Optional: 1 for each list of line control types for MTA lines	MTALIST	LCTYPE=	must appear in one sequence
Optional: 1 for each unique combination of line control types and transmission codes for MTA lines	MTATABL	LCST=	
Optional: 1 only, for all 1050s that may call controller over any MTA line	MTAPOLL	POLL=	
Optional: 1 for each dial set to be difined	DIALSET	LINES	
Optional: 1 or more for all logical units to be associated with SDLC stations on switched links	LUPOOL	NUMBER=	

Optional: 1 each for Dynamic Reconfiguration	LUDRPOOL PUDRPOOL		
Required: 1 for each path from NCP being defined to a non-adjacent subarea.	PATH	DESTSA=	must precede first GROUP macro
Optional: 1 for each SDLC selection table needed.	SDLCST	GROUP=	must precede first GROUP macro

3-NETWORK CONFIGURATION MACRO INSTRUCTIONS

Usage	*Macro*	*Required Operands*	*Macro Position*
Required: 1 for each physical line group	GROUP		at beginning of line group definition
Required: 1 for each line within group	LINE	ADDRESS= SPEED=	directly following GROUP macro or another line definition
Required: at least 1 if line uses multipoint discipline; more as needed to accommodate all stations (omit if LINE macro is coded in remote NCP and represents SDLC link to remote controller)	SERVICE	ORDER=	directly following LINE macro

For all start-stop and all BSC stations except IBM 2972, 3271, 3275:

Required: 1 for each station to be identified to NCP; represents first or only input component and/or first or only output component	TERMINAL	TERM=	directly following SERVICE macro (if present) or LINE macro
Optional: Used only for call-in MTA terminals to be associated with specific VTAM application programs.	VTERM (VTAM Users Only)	(see VTAM System Programmer's Guide for required operands)	directly following TERMINAL macro in which TERM=MTA, CTERM=YES are coded.
Required: 1 for each additional input or output component (one COMP macro can specify both one input and one output component)	COMP		directly following the TERMINAL macro or another COMP macro

OR
For IBM 2972, 3271, 3275:

Required: 1 for each BSC cluster-type station (2972, 3271 [BSC], 3275 [BSC])	CLUSTER		directly following SERVICE macro (if present) or LINE macro; or following another station definition
Required: 1 for each terminal address on cluster control unit	TERMINAL	TERM=	directly following CLUSTER macro or another station definition

OR
For SDLC stations:

Required: 1 for each SDLC station (e.g., 3270 [SDLC] 3600, 3650, 3660, 3767, 3770])	PU	PUTYPE=	directly following SERVICE macro or another PU macro
Required: 1 for each logical unit associated with physical unit, if physical unit is on nonswitched link	LU	LOCADDR=	directly following PU macro or another LU macro associated with same cluster

OR
For 3705:

Required: 1 for each 3705	PU	PUTYPE=	directly following SERVICE macro, when coded in primary NCP for link; directly following LINE macro, when coded in secondary NCP for link

4-BLOCK HANDLING MACRO INSTRUCTIONS (OPTIONAL)

Usage	*Macro*	*Required Operands*	*Macro Position*
Required: 1 for each block handler to be defined	STARTBH		at beginning of block handler
Optional: 1 allowed in each block handler	EDIT		preceding DATETIME, if DATETIME present
Optional: 1 allowed in each block handler	DATETIME		following EDIT, if EDIT present
Optional: no limit on number allowed	UBHR	NAME=	anywhere between STARTBH and ENDBH: may be inter-mixed with EDIT and DATETIME macros
Required: 1 for each block handler	ENDBH		at end of block handler
Optional: 1 for each block handler set needed (limit 255)	BHSET		following all block handler macros

5-GENERATION DELIMITER MACRO INSTRUCTION

Usage	*Macro*	*Required Operands*	*Macro Position*
Required: 1	GENEND		last macro in NCP source statements

Chapter 5. NCP Generation Macro Instructions

This chapter gives detailed descriptions of the macro instructions used to define a network control program. The order of presentation is arranged in a logical order of progression for defining a complete program. These coded macro instructions become the input to stage 1 of the NCP generation procedure. Some macros may be coded only once, while others may require more than one. See Chapter 4 for the sequence requirements and conventions.

Figure 5-1 lists all the macro instructions in the order that they appear in this chapter. The figure may be used as a quick reference to determine which macros apply to your network configuration.

Macro Instruction Coding Conventions

The following conventions are used in the descriptions of the macro instructions.

- Capital letters represent values you code directly, without change.
- Small letters represent parameters for which you must supply a value.
- Brackets [ ] enclose operands or symbols that are either *optional* or *conditional.*

An optional operand is one that you may code or omit, independent of other operands. Depending on the operand, omitting it may cause a default value to be given. The assumed value is always given.

A conditional operand is one that you may need to code or to omit, depending on how you code (or omit) other operands in the same or other macros.

- Braces { } indicate that you must choose from the enclosed items.
- An underlined value represents the default value of the operand that is, the network control program will use that value if you omit the operand.
- Parentheses ( ) must enclose a sequence of values coded in one operand.
- Quotes must be used to frame a character string if it can be confused with a keyword value for an operand. This is to avoid preventing your use of certain names as symbols.

Symbols coded in the name field of a macro instruction must not begin with a $ character.

Within the macro instruction formats and descriptions, operands that are always required appear first, in alphabetical order. Then, operands that are conditional or optional follow in alphabetical order. These are enclosed in brackets—[].

Data set (file) names must begin with an alphabetic character or $, @, or #.

				Switched		VTAM	
Macro	*SDLC*	*BSC*	*S/S*	*Yes*	*No*	*Only*	*Comments*
PCCU						•	
BUILD	R	R	R	R	R		
SYSCNTRL	R	R	R	R	R		
NCPNAU	•			•	•		Programmed Resources only
HOST	R	R	R	R	R		Channel-attached only
CSB	R	R	R	R	R		One for each scanner
IDLIST		•	•	•			OS/VS only
VIDLIST		•	•	•		•	
LUPOOL	•			•			
LUDRPOOL	•			•	•		
PUDRPOOL	•				•		Dynamic reconfiguration only
PATH	R				•		
SDLCST	•				•		Subarea links
DIALSET		•	•	•			
MTALCST			•	•			Multiple-terminal-access only
MTALIST			•	•			Multiple-terminal-access only
MTAPOLL			•	•			Multiple-terminal-access only
MTATABL			•	•			Multiple-terminal-access only
GROUP	R	R	R	R	R		
LINE	R	R	R	R	R		
SERVICE	•	•	•		•		
PU	•			•	•		
LU	•				•		
CLUSTER		•		•	•		
TERMINAL		•	•	•			
VTERM		•	•	•		•	
COMP		•	•	•	•		
STARTBH		•	•	•	•		Block handling routines only
BHSET		•	•	•	•		Block handling routines only
DATETIME		•	•	•	•		Block handling routines only
EDIT		•	•	•	•		Block handling routines only
UBHR		•	•	•	•		Block handling routines only
ENDBH		•	•	•	•		Block handling routines only
GENEND	R	R	R	R	R	R	

R = Macro required for NCP Generation
• = Macro applicable to specified line

Figure 5-1. NCP Generation Macros

System Definition Macro Instructions

The system definition macro instructions—PCCU, BUILD and SYSCNTRL—to be used in defining a network control program. (PCCU is a VTAM-only macro instruction.)

PCCU Macro Instruction (VTAM Only)

The PCCU macro instruction identifies the 3705 communications controller in which the NCP being defined is to be loaded and executed. The macro must precede the BUILD macro at the beginning of the NCP generation input deck before the deck is provided to the VTAM initialization process. You may include one or more PCCU macros in the deck provided to the NCP generation procedure, but this is not required.

See the *ACF/VTAM Installation* manual for a complete description of the macro and its operands.

The format of the PCCU macro is:

```
[symbol]        PCCU          [operands]
```

Operands

```
AUTODMP=
AUTOIPL=
AUTOSYN=
BACKUP=
CHANCON=
CONFGDS=
CONFGPW=
CUADDR=
DUMPDS=
DUMPSTA=
INITEST=
LOADSTA=
MAXDATA=
NCPLUB=
OWNER=
RNAME=
SUBAREA=
VFYLM=
```

The first macro instruction in the program source statements BUII macro specifies:

- The type of controller (3705-I or 3705-II) that is to execute the N(whether the program is to control the 3705 as a channel-attached o. link-attached controller.
- The controller storage size.
- The size of buffers in the buffer pool.
- The name that is to be assigned to the network control program and r resolution table load modules.
- The type, number, and status (active or inactive) of channel adapters i communications controller.
- The subarea address to be assigned to the NCP being defined.
- The upper limit of the range of subarea addresses assigned to subareas i network.
- Certain optional facilities that may be included in the NCP.
- Certain program generation options that may be desired.
- The names of program data sets used in the generation process.
- Whether a complete or a partial program generation is to be performed.

The format of the BUILD macro is:

```
[symbol]        BUILD        operands[,operands]
```

Operands	LINE TYPE NCP	EP	LNCTL= SDLC	BSC	S/S	DIAL= YES	NO
LOADLIB=dsname,	R	R	R	R	R	R	R
MAXSUBA=n,	R	R	R	R	R	R	R
MEMSIZ=n,	R	R	R	R	R	R	R
OBJLIB=dsname,	R	R	R	R	R	R	R
SUBAREA=n,	R	R	R	R	R	R	R
{NCP } TYPGEN={NCP-R} {PEP }	R	R	R	R	R	R	R
[,ABEND={YES}] {NO }	•		•	•	•	•	•
{SHORT} [,ASMXREF={NO }] {YES }	•	•	•	•	•	•	•
[,BFRS={88 }] {size}	•	•	•	•	•	•	•
[,CA=(adapter0[,adapter1[,adapter2 [,adapter3]]])]	•	•	•	•	•	•	•
[,CATRACE={(YES[,{count}])}] { {10 } } {NO }	•	•	•	•	•	•	•
[,CONDASM={TABLE }] {(value1,...)}	•		•	•	•	•	•
[,CSMHDR=chars]	•			•		•	•
[,CSMHDRC=chars]	•			•		•	•
[,CSMSG=chars]	•			•	•	•	•
[,CSMSGC=chars]	•			•	•	•	•
[,CUID=chars]	•			•	•	•	
[,CWALL={8}] {n}	•	•	•	•	•	•	•
[,DELAY=([ca0cnt],[ca1cnt],[ca2cnt],[ca3cnt])]	•	•	•	•	•	•	•
[,DIALTO={60.0 }] {count}	•		•	•	•	•	
[,DR3270={YES}] {NO }	•		•			•	•

Operands	LINE TYPE NCP	EP	LNCTL= SDLC	BSC	S/S	DIAL= YES	NO
[,DSABLTO={<u>3.0</u> }] {count}	•		•	•	•	•	•
{addr1} {addr2} {(YES,[{NSC }][{NSC }])} [,DYNADMP={ {NONE } {NONE } }] {NO }		•		•	•	•	•
[,ENABLTO={<u>2.2</u> }] {count}	•		•	•	•	•	•
[,ERASE={<u>NO</u> }] {YES}	•	•		•	•	•	•
[,HICHAN=([addr1][,addr2])]		R		•	•	•	•
[,ITEXTTO={<u>YES</u> }] {<u>NONE</u> } {count}	•			•	•	•	•
[,JOBCARD={NO }] {MULTI}	•	•	•	•	•	•	•
[,LESIZE=size]	•	•	•	•	•	•	•
[,LINETRC=([YES][,lines][,entries]])]		•		•	•	•	•
[,LOCHAN=([addr1P[,addr2])]		R		•	•	•	•
[,LTRACE={<u>2</u> }] {count}	•		•	•	•	•	•
[,MACLIB=(dsname1[,dsname2]...[,dsname5])]		•		•	•	•	•
[,MAXSSCP=count]	•		•	•	•	•	•
[,MODEL={<u>3705</u> }] {3705-2}	•	•	•	•	•	•	•
[,MTARTO={<u>1.0</u> }] {count}	•				•	•	
[,MTARTRY={<u>0</u> }] {count}	•				•	•	
[,NCPCA=([status0],[status1],[status2],[status3])]	•		•	•	•	•	•
{symbol} [,NEWNAME={NCP001}] {PEP001}	•	•	•	•	•	•	•
[,NPA={<u>NO</u> }] {YES}							

Operands	*LINE TYPE* *NCP*	*EP*	*LNCTL=* *SDLC*	*BSC*	*S/S*	*DIAL=* *YES*	*NO*
[,NUMHSAS=n]	•		•				
[,OBJQUAL={CG }] {symbol}	•		•	•	•	•	•
[,OLT={YES}] {NO }	•		•	•	•	•	•
[,OPCSB2={NO }] {YES}		•		•	•	•	•
[,OUTPUT=([asm],[post-asm],[link])]	•	•	•	•	•	•	•
[,PARTIAL={NO }] {YES}	•		•	•	•	•	•
[,PRTGEN=({NOGEN} [,{NOGEN}])] {GEN } {GEN }	•	•	•	•	•	•	•
[,PWROFF={YES}] {YES }	•		•				•
{symbol} [,QUALIFY={SYS1 }] {NONE }	•	•	•	•	•	•	•
[,REMLOAD={NO }] {YES}	•		•				•
[,RESOEXT={0 }] {count}	•		•				•
{12} [,SLODOWN={25}] {50}	•		•	•	•	•	•
[,TIME=integer]	•	•	•	•	•	•	•
[,TIMEOUT=([ca0cnt],[ca1cnt],[ca2cnt],[ca3cnt])]	•	•	•	•	•	•	•
{YES } [,TRACE={(NO [,{10 }])}] {size}	•		•	•	•	•	•
[,TRANSFR=count]	•		•	•	•	•	•
[,TWXID=([inchars][,outchars])]	•				•	•	
[,TYPSYS={OS }] {DOS}	•	•	•	•	•	•	•
[,UCHAN={NO }] {YES }							

Operands	*LINE TYPE*		*LNCTL=*			*DIAL=*	
	NCP	*EP*	*SDLC*	*BSC*	*S/S*	*YES*	*NO*
[,UNIT=unit type]	•	•	•	•	•	•	•
[,USERLIB=dsname]	•		•	•	•	•	•
[,UT1=dsname]	•	•	•	•	•	•	•
[,UT2=dsname]	•	•	•	•	•	•	•
[,UT3=dsname]	•	•	•	•	•	•	•
[,VRPOOL=n]	•		•				
[,XBREAK={integer}] {NONE }	•				•	•	•
[,XITB={NO }] {YES}	•			•		•	•

[symbol]

Provides a name for the macro. *symbol* may be any valid assembler-language symbol. The generation procedure does not check the symbol for validity.

LOADLIB=dsname

(required for OS/VS only)

Specifies the name of a partitioned OS/VS data set that will contain the NCP load module and resource resolution table module. (The data set name may be qualified, as determined by the QUALIFY operand of this macro.) The unqualified name may be up to 8 characters long; the first character must be alphabetic or $, #, or @. This data set must be cataloged.

The LOADLIB operand is required for generation under OS/VS and is not applicable for generation under VSE.

MAXSUBA=n

(required)

Specifies the upper limit of the range of subarea addresses used within the *network* you are defining. The same value is required for all MAXSUBA operands in all network control programs that may be active at the same time.

VTAM Note: This value must be the same as specified in the MAXSUBA VTAM start parameter.

The maximum subarea address value is always a power of 2 minus 1, within the range of 3 through 255 (3,7,15,31,63,127,255). If you specify a value that is not one of those listed, the generation procedure rounds the specified value to the next higher value. (For example, any value between 16 and 30 is rounded to 31.)

The total number of resources that can be associated with any subarea address depends on the value of MAXSUBA, as follows:

Value of MAXSUBA	*Maximum Number of Resources Possible*
3	16382
7	8190
15	4094
31	2046
63	1022
127	510
255	254

Note: Specifying an unnecessarily high value for *n* will waste NCP storage. (Storage is assigned for all subarea addresses whether used or not.)

For example, assume that the network includes one host access method, assigned subarea address 1; two channel-attached communication controllers, assigned subarea addresses 2 and 3; and four link-attached controllers, assigned addresses 4 through 7. The highest address being 7, you would specify an upper limit of at least 7. If you wish to allow for adding more access methods or controllers to the network, you would specify a value greater than 7. A value of 31 would allow up to 30 access methods and controllers (addresses 2 through 31) to be included in the network.

This operand is required.

MEMSIZE=n

(required)

Specifies the storage size of the controller in K (1,024) bytes.

The value of *n* must be one of the following:

For 3705-I (MODEL=3705): 144, 176, 208, or 240
For 3705-II (MODEL=3705-2): 128, 160, 192, 224, 256, 320, 384, 448, or 512

This operand is required.

OBJLIB=dsname

(required for OS/VS only)

Specifies the name of a partitioned OS/VS data set that will contain the output from all assemblies during stage 2 of the generation procedure. (The data set name may be qualified, as determined by the QUALIFY operand of this macro.) The unqualified name may be up to 8 characters long; the first character must be alphabetic or $, #, or @. This data set must be cataloged.

Note: OBJLIB=OBJ3705 must not be used if QUALIFY=SYS1 or if the QUALIFY operand is omitted.

This operand is required for generation under OS/VS and is not applicable for generation under VSE.

SUBAREA=n

(required)

Specifies the subarea address assigned to the network control program you are defining.

The minimum subarea address is SUBAREA=1. The maximum address is the value specified in the MAXSUBA operand of this macro (not to exceed 255).

This operand is required.

```
        {NCP  }
TYPGEN= {NCP-R}
        {PEP  }
```

(required)

Specifies (1) whether the program you are defining is to operate the 3705 as a channel-attached or as a link-attached communication controller; and (2) whether the program will include PEP functions in addition to network control functions. Select the appropriate parameter from the following:

Parameter	*The program: will include emulation functions*	*Will operate the controller as:*
NCP	No	Channel-attached and link-attached
NCP-R	No	Link-attached
PEP	Yes	Channel-attached

Note: This operand is required for an NCP or PEP generation. If neither NCP nor PEP is specified, stage 1 of the assembly generates only the emulation program and ignores all NCP-only macros and operands. If your network contains only EP resources, see the *IBM 3704 and 3705 Emulation Program Generation and Utilities Guide and Reference Manual,* (GC30-3002).

```
[ABEND={YES} ]
       {NO }
```

Specifies whether the abend facility is to be included in the network control program. This facility terminates program execution when programming errors are detected in the NCP supervisor (levels 1-4). This prevents propagation of an error into nonsupervisory portions of the program.

```
          {YES  }
[ASMXREF= {NO   } ]
          {SHORT}
```

Specifies whether you want the generation procedure to produce cross-reference listings for the stage 2 program assemblies.

(OS/VS Only): Specify ASMXREF=SHORT if you want a cross-reference table of all symbols that are referred to in the assembly. Any symbols defined, but not referred to, are not included in the table.

Specify ASMXREF=YES if you want a cross-reference table of all symbols used in the assembly. This includes symbols that are defined but never referred to.

Specify ASMXREF=NO if you do not want cross-reference listings produced.

```
[BFRS={88  } ]
      {size}
```

Specifies the size, in bytes, of the buffers in the network control program buffer pool.

Specify *size* as a multiple of 4 bytes. The minimum size is 72 bytes. If you include the on-line testing facility in the NCP (OLT=YES) the minimum size is 76 bytes. The maximum is 240. The generation procedure rounds the size specified to the next higher multiple of 4 bytes if the value is not a multiple of 4.

The generation procedure adds 12 bytes to each buffer for control use. These 12 bytes are never used to contain message data. If you omit the BFRS operand, the buffer length is 100 (88+12 bytes for control use).

Note: Diagnostic programs that communicate with the network control program via VTAM or TCAM (for example, TOTE) may impose restrictions on the values specified for the BFRS operand. See the appropriate manuals for such diagnostic programs for restrictions that may apply.

`[CA=(adapter0[,adapter1][,adapter2][,adapter3])]` *(channel-attached NCP only)*

Specifies the types of channel adapters installed in the communication controller in which the program being defined will be executed.

`adapter0`

Specifies the type of adapter in channel adapter position 0. *adapter0* may be specified as TYPE1, TYPE2, TYPE3, or TYPE4.

Channel adapter position 0 is located in the base module of the controller.

`adapter1`

Specifies the type of adapter in channel adapter position 1 and indicates whether the adapter is installed in the base module or the first expansion module of the controller.

A type 2 or type 4 channel adapter in position 1 can be installed in either the base module or the first expansion module; a type 3 adapter in position 1 is always installed in the first expansion module.

TYPE2-0 or TYPE4-0 specifies that the adapter is installed in the base module; TYPE2-1 or TYPE4-1 specifies that it is installed in the first expansion module. TYPE2 or TYPE4 specifies the type of adapter but does not indicate the module in which it is installed. If two adapters are installed [CA=(adapter0,adapter1)], the generation procedure assumes that the second adapter is installed in the first expansion module; if three or four adapters are installed [CA=(adapter0,adapter1,adapter2) or CA=(adapter0,adapter1, adapter2,adapter3)], the generation procedure assumes that the second adapter is in the base module.

`adapter2`

Specifies the type of adapter in channel adapter position 2. *adapter2* may only be specified as TYPE4. Code this parameter only if the first two channel adapter positions contain type 4 adapters.

`adapter3`

Specifies the type of adapter in channel adapter position 3. *adapter3* may only be specified as TYPE4. Code this parameter only if the first three channel adapter positions also contain type 4 adapters.

If you omit this operand and the NCPCA operand is omitted, CA=TYPE2 is assumed.

To specify that a link-attached 3705-II is equipped with one or more channel adapters, you must code the CA operand. Omitting the operand causes the program generation procedure to assume that the link-attached 3705-II has no channel adapters.

This operand is invalid for a link-attached network control program (TYPGEN=NCP-R) to be executed in a 3705-I controller.

Figure 5-2 shows how the CA and NCPCA operands may be coded for each possible channel adapter configuration.

```
[CATRACE={(YES[,{count}])}]
         {      {10    }  }
         { NO             }
```

(channel-attached NCP only)

Specifies whether the network control program is to include the channel adapter trace facility.

count specifies the number of 32-byte entries the channel adapter trace table is to include. The minimum is 1; the maximum is 255.

If you omit this operand or specify CATRACE=YES without the *count* parameter, the channel adapter trace option is included and the trace table will contain ten 32-byte entries.

Channel Adapter Configuration				CA and NCPCA Operands
Base Module		Expansion Module		Note: In the BUILD macro, code the CA operand that represents the installed channel adapter configuration and the NCPCA operand that is appropriate.
Channel Adapter Type:				
1	–	–	–	CA=TYPE1 NCPCA=ACTIVE
2	–	–	–	CA=TYPE2 NCPCA=ACTIVE
3	–	–	–	CA=TYPE3 NCPCA=ACTIVE
4	–	–	–	CA=TYPE4 NCPCA=ACTIVE
1	–	2	–	CA=(TYPE1,TYPE2) NCPCA=(ACTIVE,ACTIVE) NCPCA=(ACTIVE,INACTIVE)
1	–	3	–	CA=(TYPE1,TYPE3) NCPCA=(ACTIVE,ACTIVE) NCPCA=(ACTIVE,INACTIVE)
2	–	2	–	CA=(TYPE2,TYPE2) NCPCA=(ACTIVE,ACTIVE) NCPCA=(ACTIVE,INACTIVE)
2	–	3	–	CA=(TYPE2,TYPE3) NCPCA=(ACTIVE,ACTIVE) NCPCA=(ACTIVE,INACTIVE) NCPCA=(INACTIVE,ACTIVE)
3	–	2	–	CA=(TYPE3,TYPE2) NCPCA=(ACTIVE,ACTIVE) NCPCA=(ACTIVE,INACTIVE) NCPCA=(INACTIVE,ACTIVE)
3	–	3	–	CA=(TYPE3,TYPE3) NCPCA=(ACTIVE,ACTIVE) NCPCA=(ACTIVE,INACTIVE) NCPCA=(INACTIVE,ACTIVE)

Figure 5-2 (Part 1 of 2). CA and NCPCA Operands for 3705 Channel Adapter Configurations

Channel Adapter Configuration				CA and NCPCA Operands
Base Module		Expansion Module		
4	2	–	–	*CA=(TYPE4,TYPE2-0) NCPCA=(ACTIVE,ACTIVE) NCPCA=(ACTIVE,INACTIVE) NCPCA=(INACTIVE,ACTIVE)
4	–	2	–	CA=(TYPE4,TYPE2-1) NCPCA=(ACTIVE,ACTIVE) NCPCA=(ACTIVE,INACTIVE) NCPCA=(INACTIVE,ACTIVE)
4	–	3	–	CA=(TYPE4,TYPE3) NCPCA=(ACTIVE,INACTIVE) NCPCA=(INACTIVE,ACTIVE)
4	4	–	–	*CA=(TYPE4,TYPE4-0) NCPCA=(ACTIVE,ACTIVE) NCPCA=(ACTIVE,INACTIVE) NCPCA=(INACTIVE,ACTIVE)
4	–	4	–	CA=(TYPE4,TYPE4-1) NCPCA=(ACTIVE,ACTIVE) NCPCA=(ACTIVE,ACTIVE) NCPCA=(INACTIVE,ACTIVE)
4	4	–	4	*CA=(TYPE4,TYPE4-0,TYPE4) NCPCA=(ACTIVE,ACTIVE,INACTIVE) NCPCA=(ACTIVE,INACTIVE,ACTIVE) NCPCA=(INACTIVE,ACTIVE,ACTIVE) NCPCA=(ACTIVE,ACTIVE,ACTIVE)
4	–	4	4	CA=(TYPE4,TYPE4-1,TYPE4) NCPCA=(ACTIVE,ACTIVE,INACTIVE) NCPCA=(ACTIVE,INACTIVE,ACTIVE) NCPCA=(INACTIVE,ACTIVE,ACTIVE) NCPCA=(ACTIVE,ACTIVE,ACTIVE)
4	4	4	4	*CA=(TYPE4,TYPE4-0,TYPE4,TYPE4) NCPCA=(ACTIVE,INACTIVE,INACTIVE,INACTIVE) NCPCA=(ACTIVE,INACTIVE,ACTIVE,INACTIVE) NCPCA=(ACTIVE,INACTIVE,INACTIVE,ACTIVE) NCPCA=(INACTIVE,ACTIVE,ACTIVE,INACTIVE) NCPCA=(INACTIVE,ACTIVE,INACTIVE,ACTIVE) NCPCA=(INACTIVE,INACTIVE,ACTIVE,ACTIVE) NCPCA=(ACTIVE,ACTIVE,ACTIVE,INACTIVE) NCPCA=(ACTIVE,ACTIVE,INACTIVEACTIVE) NCPCA=(ACTIVE,INACTIVE,ACTIVE,ACTIVE) NCPCA=(ACTIVE,ACTIVE,ACTIVE,ACTIVE)

Note: In the NCPCA operand, a comma may be substituted for each ACTIVE parameter thus for example, NCPCA=(,INACTIVE,,INACTIVE) is equivalent to NCPCA=(ACTIVE,INACTIVE,ACTIVE,INACTIVE).

* This configuration is invalid if a 3705-II is equipped with both a channel adapter and remote program loader in the base module.

Figure 5-2 (Part 2 of 2). CA NCPCA Operands for 3705 Channel Adapter Configurations

```
[CONDASM={TABLE        }]
         {(value1,...)}
```

Specifies conditionally assembled network control program modules to be reassembled during a partial generation.

This operand is valid only if you specify PARTIAL=YES in the BUILD macro.

TABLE

Specifies that only the modules containing network control program tables are to be reassembled.

`value1,...`

Identifies specific modules to be reassembled. value1,... represents a sequence of 2-digit numbers corresponding to the last two digits of the names of the modules to be assembled.

For example, to reassemble modules SYSCG007 and SYSCG00A, you would code CONDASM=(07,0A).

The modules that may be individually reassembled, and the corresponding values to be coded in the CONDASM operand, are:

SYSCG000	00	SYSCG009	09
SYSCG001	01	SYSCG00A	0A
SYSCG002	02	SYSCG00B	0B
SYSCG003	03	SYSCG00C	0C
SYSCG006	06	SYSCG00D	0D
SYSCG007	07	SYSCG00E	0E
SYSCG008	08	SYSCG010	10
		SYSCG011	11

The NCP tables are always assembled, regardless of which specific modules you specify in CONDASM=(value1,...).

Appendix B lists the modules that must be reassembled for various changes in program functions.

`[CSMHDR=chars]`

(BSC 3270 only)

Specifies the header of the critical situation message to be sent to any IBM 3270 terminals in the network controlled by this NCP. A header must be specified for these terminals if the program is to send critical situation messages. The header must contain the appropriate device control characters and may also include any other valid characters desired. (See the IBM *3270 Component Description* manual, GA27-2749, for the required device control characters.) The header specified by this operand is sent only to 3270 terminals.

Code *chars* as the hexadecimal representation of the EBCDIC characters to be sent.

You may specify up to 127 EBCDIC characters in this operand. If the header required exceeds this length, code a CSMHDRC operand for the remaining characters, up to a combined total of 238 characters.

Note: Although CSMHDR and CSMHDRC together may specify up to 238 characters of header and CSMSG and CSMSGC together may specify up to 238 characters of text, the combined total of header and text specified in these four operands may not exceed 238 characters.

[CSMHDRC=chars]

(BSC 3270 only)

Specifies up to 111 additional characters of header for the critical situation message specified by the CSMHDR operand. This operand is valid only if CSMHDR is specified. See the note on total header and text length under the CSMHDR operand.

[CSMSG=chars]

(start-stop and BSC stations only)

Specifies the text of the critical situation message to be sent to the active start-stop and BSC stations on lines in network control mode before automatic network shutdown occurs. The message will be sent to each active station whose TERMINAL macro specifies CRITSIT=YES. Code *chars* as the hexadecimal representation of the EBCDIC characters desired.

If this operand is omitted, the NCP will not notify stations before automatic network shutdown occurs.

You may specify up to 127 EBCDIC characters in this operand. If the message required exceeds this length, code a CSMSGC operand of the remaining characters, up to combined total of 238 characters.

Note: Although CSMSG and CSGMSGC together may specify up to 238 characters of text and CSMHDR and CSMHDRC together may specify up to 238 characters of header, the combined total of header and text specified in these four operands may not exceed 238 characters.

[CSMSGC=chars]

(start-stop and BSC stations only)

Specifies up to 111 additional characters of text for the critical situation message specified by the CSMSG operand. This operand is valid only if CSMSG is specified. See the note on total header and text length under the CSMSG operand.

[CUID=chars]

(switched BSC stations only)

Specifies the identification characters the NCP sends to BSC stations on switched lines operating in network control mode. Code *chars* as the hexadecimal representation of the EBCDIC characters to be sent. You may specify a maximum of 20 EBCDIC characters. The NCP sends these characters to each station with the CUIDLEN operand coded in the TERMINAL or COMP macro. The characters are sent each time the NCP calls the station or answers a call from the station.

If this operand is omitted, the network control program can verify station ID sequences it receives, but will not send a controller ID sequence in return.

[CWALL={8}]
{n}

Specifies a number of buffers which the NCP reserves for data flow control to complete the movement of data out of the NCP toward its destination. When the NCP buffers are depleted to this threshold, no more data is accepted by the program until additional buffers are available. The range of this operand is from 8 to 255. The default value is 8.

[DELAY=([ca0cnt],[ca1cnt],[ca2cnt],[ca3cnt])]

Specifies the interval that the NCP delays between the time data is available for the host and the time the attention signal is presented to the channel. The delay intervals are coded to the nearest tenth of a second between 0 and 6553.5. The default values are 0 and specify that the attention signal is to be presented across the corresponding channel adapter as soon as data is available.

The parameters correspond to the active channel adapters and are positional. Therefore, the commas are required even if all parameters are not specified.

Note: If the amount of data is sufficient to fill the buffers allocated by the host, the attention signal presented before the delay count is reached.

[DIALTO={60.0 }]
{count}

(switched lines only)

Specifies the timeout used by the NCP in detecting the failure of an automatic calling unit's "abandon call and retry" (ACR) signal. Specify the timeout either as an integral number of seconds (DIALTO=30) or to tenths of a second (DIALTO=40.4).

The default value (60 seconds) is recommended unless the system designer recommends a different one.

The maximum timeout you may specify is 1632 seconds.

Note: See the section "Restriction on Number of Time Intervals" following the description of the GROUP macro.

[DR3270={YES}]
{NO }

(dynamic reconfiguration only)

Specifies whether the NCP is to include SDLC 3270 Model 11 and 12 terminal support when dynamic reconfiguration is supported and no 3270s are being defined during system generation. This operand should be coded YES if 3270 terminals may be added at a later time through the dynamic reconfiguration function. The default for this operand is YES if a PUDRPOOL macro is coded.

This operand is effective only if dynamic reconfiguration is included in the NCP. If it is not included, the operand is ignored. However, if BNNSUP=3270 is coded in the PU macro, 3270 support is always included regardless of the value coded for this operand.

[DSABLTO={3.0 }]
{count}

(network control mode only)

Specifies the timeout used by the NCP in detecting failure of the "data set ready" signal line of the modem to be turned off when the line attached to the modem is disabled. Specify the timeout either as an integral number of seconds (DSABLTO=5) or to tenths of a second (DSABLTO=7.5).

The maximum timeout you may specify is 1632 seconds.

The line remains disabled for the period specified, regardless of whether the "data set ready" signal line is turned off within the period.

Note: See the section "Restriction on Number of Time Intervals" following the description of the GROUP macro.

```
                {addr1}    {addr2}
[DYNADMP={(YES[,{NSC  }][,{NSC  }])}]
         {      {NONE }    {NONE }  }
         {NO                        }
```

(emulation mode only)

Specifies whether the dynamic dump facility is included in the network control program. The dynamic dump allows the storage contents of the communications controller to be transferred to the host processor without interrupting execution of the program. This operand also specifies one or more subchannels to be available for the transfer.

addr1 is the address of an emulation subchannel in the channel adapter in adapter position 0 over which dynamic dump data can be transferred to the host processor.

addr2 is the address of an emulation subchannel in the channel adapter in adapter position 1 over which dynamic dump data can be transferred to the host processor.

NSC specifies that the native subchannel of the channel adapter is to be used for dump data transfer. NSC in the first and second address positions refers to the native subchannel of channel adapters positions 0 and 1, respectively.

NONE specifies that no subchannel of the channel adapter is to be used for dump data transfer. NONE in the first and second address positions refers to channel adapters positions 0 and 1, respectively. kp off

Examples:

1. If you wish to allow dynamic dump data to be transferred over the emulation subchannel of a single type 4 channel adapter, code DYNADMP=(YES,*addr*).
2. If the controller has a type 4 channel adapter in adapter position 0 and a type 2 or type 3 adapter in adapter position 1, code DYNADMP=(YES,NSC) to allow dump data transfer over the native subchannel of the type 4 channel adapter; code DYNADMP=(YES,*addr1*) to allow dump data transfer over an emulation subchannel of the type 4 adapter.
3. If the controller has two type 4 channel adapters, code (DYNADMP=(YES,*addr1,addr2*) to allow dump data transfer over a specified emulation subchannel of each of the channel adapters; code DYNADMP=(YES,NONE,*addr2*) to allow transfer over only the specified subchannel of the second channel adapter.

The following rules govern which type of subchannel (native or emulation) can be used to transfer dynamic dump data.

For type 1 channel adapter: (1) If the native subchannel of the adapter is used for network control operations, an emulation subchannel of this adapter can be used for dynamic dump data transfer; the native subchannel cannot be used for this purpose. (2) If the native subchannel of the adapter is *not* for network control operations, it can be used for dynamic dump data transfer.

For type 2 and type 3 channel adapter: These types of channel adapters cannot be used for dynamic dump data transfer.

For type 4 channel adapter: (1) An emulation subchannel can be used for dynamic dump data transfer regardless of whether the native subchannel is used for network control operations. (2) The native subchannel can be used for dynamic dump data transfer if it is not used for network control operations.

Figure 5-3 shows how the DYNADMP operand can specify subchannels used for transfer of dynamic dump data to the host processor.

If CA=	*Then to allow dynamic dump data transfer over these subchannels*:*	*Code DYNADMP=*
TYPE1	ESC_1	(YES,*addr1*)
(TYPE1,TYPE2)	ESC_1	(YES,*addr1*)
	NSC_1	(YES,NSC)
(TYPE1,TYPE3)	ESC_1	(YES,*addr1*)
	NSC_1	(YES,NSC)
TYPE4	ESC_1	(YES,*addr1*)
(TYPE4,TYPE4)	ESC_1	(YES,*addr1*)
or (TYPE4,TYPE4-0)	ESC_2	(YES,NONE,*addr2*)
or (TYPE4,TYPE4-1)	ESC_1 and ESC_2	(YES,*addr1,addr2*)
(TYPE4,TYPE2)	ESC_1	(YES,*addr1*)
or (TYPE4,TYPE2-0)	NSC_1	(YES,NSC)
or (TYPE4,TYPE2-1)	ESC_1	(YES,*addr1*)
or (TYPE4,TYPE3)	NSC_1	(YES,NSC)

* ESC - Emulation subchannel

NSC - Native subchannel

Subscripts indicate channel adapter position:
$_1$Adapter position 0
$_2$Adapter position 1.

Figure 5-3. Subchannel Address Specification for Dynamic Dump Data Transfer

```
[ENABLTO={2.2  } ]
         {count}
```

(network control mode only)

Specifies the timeout used by the NCP in detecting the failure of the "data set ready" signal line of the modem to be turned on when the communication line attached to the modem is enabled (for nonswitched lines) or when a dialing operation is completed (the automatic calling/answering unit has signalled a connection for switched lines). Specify the timeout either as an integral number of seconds (ENABLTO=3) or to tenths of a second (ENABLTO=3.2).

The maximum timeout you may specify is 1632 seconds.

This timeout is also used to monitor for "clear-to-send" (CTS) after raising "request-to-send" when initiating transmit operations. This interval may take longer for switched lines (consult your modem specification).

For a nonswitched line or a switched line where calls are made by an automatic calling unit (ACU), the default value of 2.2 seconds is usually appropriate. Generally, 2.2 seconds is also appropriate for switched lines with automatic answer, however, it may take longer for "data-set-ready" to turn on for lines with the ring indicator mode option (consult your modem specification). It is not appropriate if the channel-attached NCP being defined includes any switched backup SDLC links to a link-attached controller. Such a link requires an enable timeout long enough to allow the system operator to dial the telephone number, receive an answer, and place the modem (data set) in data mode. This process may typically take from 30 seconds to more than a minute. On the other hand, the timeout value should be no greater than necessary; otherwise it will needlessly extend the time required to shut down the NCP if shutdown is initiated while a line is being activated by command from the access method. (The NCP does not shut down until all lines are deactivated.)

Note: See the section "Restriction on Number of Time Intervals" following the description of the GROUP macro.

```
[ERASE={NO }]
       {NO }
```

(start-stop and BSC stations only)

Specifies whether the NCP is to include the buffer erase function.

ERASE=YES is required if you specify CDATA=YES in any CLUSTER, TERMINAL, or COMP macro within the program.

```
[HICHAN=( [addr1] [,addr2] )]
```

(required for emulation mode only)

Specifies the highest subchannel address on each channel adapter associated with any line operating in emulation mode (or the address of the subchannel used for the dynamic dump operation). The address must equal or exceed the highest emulation subchannel address specified in the ADDRESS operand of any LINE macro (or the address specified in the DYNADMP operand of the BUILD macro).

If the controller has a single type 1 or type 4 channel adapter, specify *HICHAN=addr1*. For example, HICHAN=2B. If the controller has two type 4 adapters, specify HICHAN=(*addr1,addr2*), where *addr1* is the highest subchannel address associated with the first type 4 channel adapter and *addr2* is the highest subchannel address associated with the second type 4 adapter. The value of *addr1* and *addr2* must be one of the following hexadecimal addresses:

03	07	0B	0F	83	87	8B	8F
13	17	1B	1F	93	97	9B	9F
23	27	2B	2F	A3	A7	AB	AF
33	37	3B	3F	B3	B7	BB	BF
43	47	4B	4F	C3	C7	CB	CF
53	57	5B	5F	D3	D7	DB	DF
63	67	6B	6F	E3	E7	EB	EF
73	77	7B	7F	F3	F7	FB	FF

The range of subchannel addresses specified by HICHAN and LOCHAN must not include any addresses associated with shared UCWs (unit control words) in the host processor.

This operand is required if the program includes emulation functions.

The subchannel address is the same address specified in the unit channel block (UCB) for OS/VS or in the physical channel block (PUB) for VSE.

Note: See the description of the LOCHAN operand.

```
[ITEXTTO={NONE }]
         {count}
```

(start-stop and BSC stations in network control mode only)

Specifies the text timeout interval used by the NCP for any terminal or component for which INHIBIT=TEXTTO is specified in the TERMINAL or COMP macro.

The maximum timeout you may specify is 1632 seconds.

ITEXTTO=NONE specifies that no timeout is to occur.

Note: See the section "Restriction of Number of Time Intervals" following the description of the GROUP macro.

```
         {NO   }
[JOBCARD={YES  }]
         {MULTI}
```

Specifies whether the program generation procedure is to provide a job card for the stage 2 input stream and specifies whether the input stream will consist of more than one job.

The job card provided is in the form:

```
//NCPGENnn JOB 1,'NCP GENERATION',MSGLEVEL=1 (OS/VS)
// JOB ASSEMBLE MAINT JCL FOR STAGE 3 (VSE)
```

If you specify JOBCARD=YES or omit the operand, a single job card is provided and the program generation input stream consists of a single, multiple-step job. The job card label is // NCPGEN00.

If you specify JOBCARD=MULTI, a job card is provided for each step and the input stream consists of multiple jobs. The job card labels are //NCPGENnn, where *nn* is a sequential identification number provided by the generation procedure.

Note: If you code JOBCARD=YES or JOBCARD=MULTI, you may specify a job card different from the one shown by using the IEBUPDTE utility program to change the job statement information in the stage 1 macro library. See Chapter 4 for information on this procedure.

Note: If you code JOBCARD=MULTI, you may specify job cards different from the one shown by using the ESERV utility to change the job statement information. (Only the second and any subsequent job cards may be changed; the first job card has the format shown above.) See Chapter 4 for information on this procedure.

```
[LESIZE=size]
```

(OS/VS only)

Specifies the OS/VS region size, in K (1024) bytes, to be used by all linkage editor job steps during stage 2 of program generation. The number you specify is reduced by 10 and used as *value1* of the linkage editor SIZE parameter.

value2 of the SIZE parameter is always 48(K), regardless of what you specify in the LESIZE operand.

size must exceed 10 and be less than 16 384 (16 384K bytes).

If you omit the LESIZE operand, the EXEC card for the linkage editor job steps will have a REGION parameter of 384K and PARM parameter values of 374 (for *value1*) and 48 (for *value2*).

`[LINETRC=( [YES] [[,lines] [,entries]] )]` *(emulation mode only)*

Specifies the maximum number of lines in emulation mode that can be traced concurrently and the number of trace table entries provided.
The line trace functions may be initiated from the control panel of the communications controller. The *Control Panel Guide* explains the use of the line trace option.

`lines`

Specifies the maximum number of lines that are to be traced concurrently. If you omit this parameter, all lines currently operating in emulation mode can be traced at once. The minimum value of *lines* is 1; the maximum is 352.

`entries`

Specifies the number of 8-byte entries in the trace table. The minimum is 12 entries; the maximum is 23,680.

If you omit this parameter, the trace table will contain 200 eight-byte entries.

`[LOCHAN=( [addr1] [,addr2] )]` *(required for emulation mode only)*

Specifies the low end of the range of subchannel addresses associated with the channel adapters installed in the communication controller.

If the controller has a single type 1 or type 4 channel adapter, specify LOCHAN=*addr1*. If the controller has two adapters, specify LOCHAN=(*addr1,addr2*), where *addr1* is the lowest subchannel address associated with the first channel adapter and *addr2* is the lowest subchannel address associated with the second adapter. The value of *addr1* and *addr2* must be one of the following hexadecimal addresses.

00	10	20	30
40	50	60	70
80	90	A0	B0
C0	D0	E0	F0

Note: Specifying an address that is not listed causes an MNOTE warning message to appear in the assembly listing.

The range of subchannel addresses specified by HICHAN and LOCHAN must not include any addresses associated with shared UCWs (unit control words) in the host processor. Optimum storage utilization is achieved by sequential addressing of all emulation subchannels. Each unassigned subchannel address between the values specified by the LOCHAN and HICHAN operands adds 10 bytes to the control program storage requirements.

CAUTION

1. All commands (except Sense, Test I/O, and I/O No-Op) issued to unassigned subchannels within the LOCHAN to HICHAN range are rejected. (Unassigned subchannels are those not specified in the ADDRESS operand of any LINE macro or in the DYNADMP operand of the BUILD macro.)
2. Although the channel adapter recognizes as valid any commands issued for a subchannel address that is outside the LOCHAN to HICHAN range, the emulation program does not recognize the address and, therefore, ignores any such commands received from the host channel. This causes permanently busy ("hung") subchannel.
3. If a unit control block (UCB) exists for a device associated with a subchannel outside the LOCHAN to HICHAN range but within the channel adapter's address range, initial program load (IPL) of the operating system in the host processor cannot be completed because Test I/O and Sense commands—though accepted by the controller—are ignored.

The address range specified by the LOCHAN and HICHAN operands applies only to emulation subchannels. The network control (native) subchannel address may, but need not, lie within this range.

Exception: The network control subchannel address must not be within the LOCHAN-HICHAN range if the communications controller is equipped with two type 4 channel adapters.

It is recommended that values specified by the LOCHAN and HICHAN operands correspond to the low and high subchannel addresses actually installed within the controller. However, the program generation procedure does not verify that this is the case. Failure to follow this recommendation may cause the following:

- All commands (except Sense, Test I/O, and I/O No-Op) issued to unassigned subchannels within the LOCHAN to HICHAN range will be rejected.
- All commands issued to subchannels outside the LOCHAN to HICHAN range will result in a permanently busy ("hung") subchannel.

The subchannel address is the same address specified in the unit channel block (UCB) for OS/VS or in the physical channel block (PUB) for VSE.

This operand is required if the program includes emulation functions.

```
[LTRACE={2    }]
        {count}
```

(network control mode only)

Specifies the maximum number of lines that the NCP is to trace concurrently. The minimum is two lines; the maximum is eight.

If you omit this operand, the NCP will allow only one or two lines at a time to be traced. (Line traces are requested from the host processor.)

The line trace facility is a service aid always included in the network control program.

When using line trace with 230.4 Kbps lines, a maximum of one duplex or two half duplex lines may be traced simultaneously. It may also be necessary to allow some system traffic to quiesce when tracing 230.4 Kbps lines.

Note: The performance of a line being traced may diminish somewhat because of the additional processing required each time a character-service or buffer-service interrupt occurs for the line. The performance of the network control program may be similarly affected to a lesser extent. In both cases, the amount of decrease in performance depends upon the type of scanner servicing the line and the degree to which the communications controller is currently loaded.

`[MACLIB=(dsname1[,dsname2]...[,dsname5])]` *(user code and OS/VS only)*

Specifies the name of the partitioned OS/VS data sets (up to five) to contain the user-written source code to be included in the stage 2 generation assembly. If user-written code is included in the NCP, the linkage editor INCLUDE and ORDER statements are also contained in this library. If the MACLIB operand is omitted and SRCHI or SRCLO is specified in the GENEND macro, stage 1 of the generation assumes that the user source code and the INCLUDE and ORDER statements are in the NCP macro library (SYS1.MAC3705).

The data set name may or may not be qualified, depending on the QUALIFY operand in this macro. An unqualified name may be up to 8 characters long. The first character must be alphabetic or $, , or @. This data set must be cataloged.

Note: The NCP stage 2 macro library precedes the user macro library in the generation job control statements. Therefore, if a member name in the user library is the same name as a member in the NCP library, the member in the NCP library is used.

`[MAXSSCP=count]` *(network control mode only)*

Specifies the maximum number of system service control points (SSCP) the NCP can be in session with concurrently. These sessions can be conducted over channel adapters or over SDLC communication links, or a combination of the two.

An NCP can be in session with as many as eight SSCPs concurrently. The maximum value of *count* is 8. The program can be in session with as few as one SSCP. However, the minimum value of *count* is the number of channel adapters you have specified in the NCPCA operand as active. If you do not code the NCPCA operand, the minimum value of *count* is 1.

For example, if the 3705 has three channel adapters and you specify that two channel adapters are to be currently active in network control mode, the minimum value for count is 2.

If you omit this operand, the value assumed for MAXSSCP equals the number of concurrently active channel adapters as specified in the NCPCA operand. In this case, no SSCP sessions can take place over SDLC links. The default value for this operand is 1 if TYPGEN=NCP-R.

```
[MODEL={3705  }]
       {3705-2}
```

Specifies whether the network control program is to be loaded into and executed by a 3705-II (MODEL=3705-2) or a 3705-I (MODEL=3705). (A 3705-I may be specified as either MODEL=3705 or MODEL=3705-1.)

Note: This information is needed only by the generation procedure. The network control program does not differ for the two machine types.

```
[MTARTO={1.0  }]
        {count}
```

(multiple-terminal-access lines only)

Specifies the reply timeout used when a start-stop terminal on a switched multiple-terminal-access line calls the controller. Specify the timeout either as an integral number of seconds (MTARTO=5) or to tenths of a second (MTARTO=5.5).

This operand is valid only if the network includes lines defined as multiple-terminal-access lines (see the MTALIST operand of the LINE macro.)

Note: See the section, "Restriction on Number of Time Intervals" following the description of the GROUP macro.

```
[MTARTRY={0    }]
         {count}
```

(multiple-terminal-access lines only)

Specifies the number of times the NCP is to retry the start-stop multiple-terminal-access sign-on procedure after identifying the type of MTA terminal that called this controller.

The maximum number of retries is 255.

This operand is valid only if the network includes lines defined as multiple-terminal-access lines (see the MTALIST operand of the LINE macro).

```
[NCPCA=( [status0],[status1],[status2],[status3] )]
```

Specifies the active or inactive status of each channel adapter that operates in the network control mode. *status0, status1, status2,* and *status3* are positional parameters and correspond to the channel adapter positions as specified in the CA operand. Figure 5-2 shows how NCPCA and CA operands may be coded for possible adapter configurations.

If the adapter is to be active, code ACTIVE as the corresponding *status* parameter. A comma may be substituted for each ACTIVE parameter; for example, NCPCA=(,INACTIVE,,INACTIVE) is equivalent to NCPCA=(ACTIVE,INACTIVE,ACTIVE,INACTIVE).

If the adapter is to be inactive, code INACTIVE as the corresponding *status* parameter. Channel adapters used in emulation mode only, must be coded as inactive.

Any channel adapters you specify as INACTIVE are disabled for all communication with the network control program (network control mode and emulation mode).

For CA=(TYPE4,TYPE3), NCPCA=(ACTIVE,ACTIVE) is invalid because the two adapters are not of the same or equivalent types.

Note: The network control program must be loaded into the 3705 over one of the adapters you have specified as active. Any channel adapter not connected to an operational host must be disabled.

```
           {symbol}
[NEWNAME={NCP001}]
           {PEP001}
```

Specifies the name to be given to the generated network control program load module.

Code NEWNAME=*symbol*, where *symbol* is any valid symbol that does not exceed 7 characters. The generation procedure automatically assigns the name you specify, followed by the letter R, to the resource resolution table load module that corresponds to the NCP load module. If a block handler set resolution table load module is generated, its name is the name you specified as *symbol*, followed by the letter B.

Alternatively, specify NCP001 or PEP001 in this operand, whichever is appropriate. If you omit the operand, the name assigned is NCP001 if TYPGEN=NCP or NCP-R and PEP001 if TYPGEN=PEP.

```
[NPA={YES}]
     {NO }
```

Specifies whether the Network Performance Analyzer (NPA) function is to be included in this NCP. The default for this operand is NO. If YES is specified, NPA will be included in this NCP. Also, one virtual group must be defined with NPARSC=YES in the GROUP macro. (See the NPARSC operand in the GROUP macro.)

```
[NUMHSAS=n]
```

Specifies the number of host subareas that can communicate concurrently with this NCP. The minimum value for this operand is 1; the maximum is the number specified in the MAXSUBA operand minus 1. If this operand is omitted, the default is the value of MAXSUBA divided by 2.

Note: Care should be exercised when determining the value of NUMHSAS. An excessive value could waste NCP storage.

```
[OBJQUAL={CG    }]
         {symbol}
```

Specifies a 2 character alphanumeric symbol to uniquely name conditionally assembled object modules. By giving the generated object modules different names, multiple NCP generations can be run concurrently. The value specified in this operand is used as the fourth and fifth characters of the object module names. For example, if OBJQUAL=X1, then the NCP conditional assemblies will be SYSX1001, SYSX1002, etc.

symbol may be any valid alphabetic or numeric character, including $,   , and @.

An object module qualifier eliminates the necessity of having multiple object libraries for the NCP conditionally assembled object modules. When an object qualifier is used for a partial generation, OBJQUAL must be the same as was specified in the complete generation that is being used as a base.

```
[OLT={YES}]
     {NO }
```

(network control mode only)

Specifies whether the optional online terminal test and online line test facilities (for lines in network control mode) are to be included in the NCP. Code OLT=NO to omit the facilities; code OLT=YES (or omit the operand) to include them.

```
[OPCSB2={YES}]
        {NO }
```

(emulation mode only)

Specifies that a 20-byte data buffer is to be provided for communication lines (1) that are serviced by a type 2 communication scanner, (2) which you have specified as operable in emulation mode, and (3) for which you have specified CHNPRI=HIGH in the LINE macro. These buffers are permanently assigned to the line and provide extra protection against overruns that can result from temporary slowdowns in channel operation or temporary peaks in data traffic in the network. Lines serviced by a type 2 scanner where OPCSB2=YES is not specified, have two 4-byte buffers.

The 20-byte buffer is used only when the line is operating in emulation mode.

If you omit this operand, OPCSB2=NO is assumed for lines associated with subchannels on a type 1 channel adapter; OPCSB2=YES is assumed for lines associated with subchannels on a type 4 channel adapter for which CHNPTRI=HIGH is specified.

If you specify OPCSB2=YES, do not specify *both* CHNPRI=HIGH and TADDR=*address* in the same LINE macro. (Either, alone, may be specified.)

```
[OUTPUT=([asm],[post-asm],[link])]
```

(OS/VS only)

Specifies the names of cataloged procedures to be used in place of the normally generated JCL when doing an NCP generation. The parameters of this operand are positional; therefore, if all of the parameters are not specified, the commas are still required.

`asm`

Specifies the name of a cataloged procedure to be used in the assembly steps of the generation. A cataloged procedure allows you to direct the assembly output to a media other than print. When asm is specified, the cataloged procedure should use the symbolic parameter &MOD. &STEP may be used to conditionally execute the assembly steps. When the assembly output is directed to tape, &FILE must be used as the symbolic parameter for the tape label operand in the procedure.

The cataloged procedure should provide the data set name for the object library but not specify the SYSIN DD statement. The SYSIN DD statement is generated in the JCL job stream. The OBJLIB operand of the BUILD macro also should not be coded.

An example of the JCL statement generated when *asm* is specified would be:

```
//S1 EXEC ASMPROC,MOD=NCP001,STEP=S1,FILE=1
```

STEP is the step number of the procedure.
FILE is the number of the print file generated the by assembly step.
MOD is the name of the load module.

`post-asm`

Specifies the name of the cataloged procedure to be inserted in the JCL job stream after the assembly step. This provides the ability to retrieve the assembly output as specified in the cataloged procedure. For example, you may want to print the assembly output only under certain error conditions. An example of the JCL statement generated when *post-asm* is specified would be:

```
//S2 EXEC POSTPROC,STEP=S1,FILE=1
```

STEP is the step number of the procedure.
FILE is the number of the print file generated by the assembly step.

`link`

Specifies the name of a cataloged procedure to be used in the linkage editor step of the generation. The OBJLIB operand of the BUILD macro should not be coded, but the data set name of the object library should be provided by the cataloged procedure. The LOADLIB operand should be coded and the data set name specified in the procedure. The SYSLIN DD statement is generated in the job stream and therefore should not be specified in the cataloged procedures.

An example of the JCL statement generated when *link* is specified would be:

```
//S3 EXEC LINKPROC,STEP=S3,NAME=LOADLIB
```

STEP is the step number of the procedure.

Figure 5-4 shows an example of cataloged procedures that may be modified to match the host configuration and used as a result of the OUTPUT operand. The ASMPROC procedure causes the assembly output to be written on a nonlabeled tape. The POSTPROC procedure checks the condition code setting of the assembly step. If the condition code of the assembly is 0 or 4, the POSTPROC procedure does not retrieve and print the assembly output produced by the ASMPROC procedure. If the condition code setting of the assembly step is 8 or greater, then the assembly output is retrieved and printed. This gives you the ability to selectively print the assembly output.

NCP generation macro specification:

```
NCPGEN BUILD TYPSYS=OS,OUTPUT=(ASMPROC,POSTPROC,LINKPROC)
```

Cataloged Procedures:

```
ASMPROC      PROC
//ASM        EXEC     PGM=CWAX00,PARM=(DECK,XREF)
//SYSPRINT     DD     DCB=(DEN=3,LRECL=121,BLKSIZE=3025,RECFM=FB),
//                    UNIT=2400,LABEL=(&FILE,NL),DISP=(,PASS),VOL=SER=ALPHA1
//SYSUT1       DD     UNIT=SYSSQ,SPACE=(1700,(800,800)),DCB=(OPTCD=C)
//SYSUT2       DD     UNIT=SYSSQ,SPACE=(1700,(800,800)),DCB=(OPTCD=C)
//SYSUT3       DD     UNIT=SYSSQ,SPACE=(1700,(800,800)),DCB=(OPTCD=C)
//SYSLIB       DD     DSN=SYS1.MAC3705,DISP=SHR,UNIT=3330-1,VOL=SER=BETA
//SYSPUNCH     DD     DSN=SYS1.OBJ730D(&MOD),DISP=OLD
//NULLFILE     DD     DUMMY,DSN=&STEP,DISP=SHR
//SYSUDUMP     DD     SYSOUT=A
             PEND
POSTPROC     PROC
//POST       EXEC     PGM=IEBGENER,COND=(8,GT,&STEP..ASM)
//SYSPRINT     DD     SYSOUT=A
//SYSIN        DD     DUMMY
//SYSUT1       DD     DCB=(DEN=3,LRECL=121,BLKSIZE=3025,RECFM=FB),
//                    UNIT=2400,LABEL=(&FILE,NL),DISP=(,PASS),VOL=SER=PE0081
//SYSUT2       DD     SYSOUT=A
             PEND
LINKPROC     PROC
//LKED       EXEC     PGM=HEWL,REGION=450K,
//                    PARM='LIST,LET,DC,NCAL,XREF,SIZE=(440K,48K),ALIGN2'
//SYSPRINT     DD     SYSOUT=A
//SYSUT1       DD     UNIT=SYSSQ,SPACE=(1700,(800,800)),DCB=(OPTCD=C)
//NULL1        DD     DUMMY,DSN=&STEP,DISP=SHR
//NULL2        DD     DUMMY,DSN=&STEP,DISP=SHR
//SYSLMOD      DD     DSN=SYS1.LOADLIB,DISP=SHR
//OBJ3705      DD     DSN=SYS1.OBJ3705(&NAME),DISP=SHR
//SYSPUNCH     DD     DSN=SYS1.OBJ730D,DISP=SHR
             PEND
```

Figure 5-4. Example of Cataloged Procedures

```
[PARTIAL={NO }]
         {YES}
```

Specifies whether a partial program generation is to be performed.

If you code PARTIAL=YES, only the tables and conditionally assembled modules specified in the CONDASM operand are assembled and link-edited with the remaining object modules. The conditionally assembled modules not specified by the CONDASM operand are obtained from the library specified by the OBJLIB operand. The modules assembled by the partial generation procedure replace the corresponding modules from the previous generation.

Appendix B lists the modules that must be reassembled for various changes in program functions.

If you code PARTIAL=NO (or omit the operand) a complete program generation is perforᵐed.

```
[PRTGEN= {GEN  }   {GEN  }
       ( {NOGEN} [ , {NOGEN} ] )]
```

Specifies whether macro generated statements are printed for the NCP table assemblies and NCP conditional assemblies. The first parameter is for the two NCP table assemblies. The second parameter is for all of the conditional assemblies.

If GEN is specified (or the parameter is omitted) the generation procedure prints all of the generated statements produced by the stage 2 assembly.

Specifying NOGEN reduces the amount of printed output from stage 2 during a partial generation. NOGEN suppresses printing of the assembled statements with the exception of MNOTEs. MNOTEs are always printed.

```
[PWROFF={NO }]
        {YES}
```

(channel-attached NCP only; VTAM users only)

Specifies whether the program in a channel-attached communications controller will turn off the link-attached controller's power upon command from VTAM. PWROFF=YES is valid only if the link-attached controller is equipped with the remote power off feature. (Power can be turned on again only at the control panel of the link-attached controller.)

```
          {symbol}
[QUALIFY={SYS1  }]
          {NONE  }
```

(OS/VS only)

Specifies the first-level qualifier for OS/VS data sets specified by the LOADLIB, OBJLIB, USERLIB, UT1, UT2, and UT3 operands of this macro. The data set name is formed by appending the characters SYS1, or *symbol*, to the name specified *dsname* in each of the previously mentioned operands.

symbol

Specifies the qualifier as from 1 to 8 alphanumeric characters; the first character must be alphabetic (including $, @ and). (Omit the period that separates the qualifier and the data set name; the generation procedure appends the period to the qualifier you specify.)

NONE

Specifies that no qualifier is to be placed before the simple name specified by *dsname*.

SYS1

Specifies that SYS1 is to be used as the qualifier.

```
[REMLOAD={YES}]
         {NO }
```

Specifies whether the remote program loader feature is installed on an 3705-II.

REMLOAD=NO is invalid for a link-attached network control program (TYPGEN=NCP-R) that is to be executed in a 3705-II.

If you specify TYPGEN=NCP-R and MODEL=3705-2, and omit this operand, REMLOAD=YES is assumed, otherwise REMLOAD=NO is assumed.

Note: A 3705-II equipped with one or more channel adapters and a remote program loader may operate at different times as either a channel-attached or a link-attached controller. (Specify REMLOAD=YES if the controller has a remote program loader even if the program you are defining is a channel-attached network control program.)

```
[RESOEXT={0    }]
         {count}
```

(dynamic reconfiguration only)

Specifies the number of network addresses in the resource vector table (RVT) extension. The size of the RVT extension determines the number of generation-defined resources that can be deleted and reused.

When a generation-defined physical or logical unit is deleted, the resource is returned to its respective pool. In order for that resource to be reused, the generation-defined network address must be replaced by a network address from the RVT extension. If the RVT extension is depleted (all addresses used), generation-defined resources can still be deleted but not reused. However, unlike the network addresses defined during the NCP generation the addresses in the RVT extension can be reused once they are deleted.

The maximum value of *count* is subject to the maximum number of resources that can be defined in a particular NCP. (See the MAXSUBA operand of the BUILD macro.)

```
         {25}
[SLODOWN={12}]
         {50}
```

Specifies the minimum percent of NCP buffers that are available (not in use) before the network control program enters slowdown mode. When the percent of buffers available drops below this value, the program reduces the amount of data it accepts from lines operating in network control mode and from the network control subchannel, but it continues to send data over the lines and the subchannel. This procedure reduces the number of buffers in use.

Slowdown mode is entered when fewer than one-half (SLODOWN=50), one-quarter (SLODOWN=25) or one-eighth (SLODOWN=12, or operand is omitted) of the buffers are available.

During initialization, the NCP may dynamically change the percentage you specify if the following requirements cannot be met. (1) The number of available buffers at slowdown must equal or exceed the value of the CWALL operand plus 2, and (2) the buffer pool minus the number of slowdown buffers must equal or exceed 30.

If the number of buffers contained in the NCP is less than 40, the program abends.

```
[TIME=integer]
```

(OS/VS only)

Specifies the time value, in minutes, to be used as the TIME parameter in the stage 2 assembly EXEC statements. *integer* must be greater than 0 and less than 1441.

If you omit this operand, no TIME parameter is used for the stage 2 assembly EXEC statements.

```
[TIMEOUT=( [ca0cnt],[ca1cnt],[ca2cnt],[ca3cnt] )]
```

Specifies the interval that the NCP is to wait for a response to an attention signal it has sent to the host before initiating channel discontact. The timeout intervals are coded either as an integral number of seconds (15,20) or to the nearest tenth of a second (12.5,,18.2). The minimum value is 0.2 and the maximum is 840.0. The default for each parameter is based on the NCPCA operand. If the channel adapter is inactive, the default is 0; if it is active, the default is 420.0.

The parameters correspond to the channel adapter locations in the communications controller and are positional. Therefore, the commas are required even if all parameters are not specified.

```
[TRACE={(YES[,{size}])}]
       {      {10  }  }
       { NO         }  }
```

(network control mode only)

Specifies whether the address trace option is to be included in the NCP. Code TRACE=YES to include the option; code TRACE=NO (or omit the operand) to omit the option.

If you code TRACE=YES, you may also specify the number of 16-byte entries the trace table is to contain, from 10 [TRACE=(YES,10)] to 256 [TRACE=(YES,256)]. If you omit the number or specify fewer than 10, the table will contain 10 entries.

```
[TRANSFR=count]
```

(network control mode only)

Specifies the number of NCP buffers corresponding to the maximum length PIU that the network control program is to send to a destination host processor.

The path between the NCP being defined and the destination host processor may comprise a channel connection to the attached host processor or a combination of SDLC links between network control programs and a channel connection to a distant host processor.

The principal use of this operand when used with the BUILD macro, is to limit the amount of line trace data accumulated in NCP buffers to no more than the access method in the destination host processor (and intermediate access methods, if any), can accept in a single PIU (or to no more than the data transfer limit imposed by a subsequent NCP in the path).

You may also use this operand to specify the data transfer limit for any communication line where the TRANSFR operand is omitted from the LINE macro. See the description of the TRANSFR operand in the LINE macro.

To calculate the value of *count,* proceed as follows:

1. Determine which host processors in the network will receive the line trace data.
2. Calculate the data transfer limit, in bytes, imposed by the access method in each destination host processor. To do so, multiply the values specified by the MAXBFRU and UNITSZ operands and then subtract the value of the BFRPAD operand of the HOST macro that represents the access method to the NCP.
3. Determine which of the data transfer limits thus calculated is the smallest, and divide that limit by the buffer size specified in the BFRS operand of the BUILD macro in the present network control program. The result (ignoring any fractional remainder) is the maximum value of *count* you should specify in this TRANSFR operand.

See the description of the TRANSFR operand in the LINE macro for an example showing how to calculate the data transfer limits. Also see this description for determining the *minimum* value of *count* if you use this TRANSFR operand to specify the data transfer limits for any communication line for which you omit the TRANSFR operand from the LINE macro.

It is most convenient to establish the same data transfer limit for line trace data as for message data received from communication lines by the present network control program. Because the buffers used for both purposes are of the same size, the *count* values you specify in the TRANSFR operands of the BUILD and LINE macros would be the same.

If you omit the TRANSFR operand from some LINE macros, be sure that the value you specify in this TRANSFR operand (in the BUILD macro) is appropriate for the corresponding lines.

The minimum you may specify in this operand is one buffer (TRANSFR=1); the maximum is 255 (TRANSFR=255).

If you omit this TRANSFR operand, the program generation procedure determines a value in one of the following ways:

- If one or more HOST macros appear in this program, the generation procedure computes the data transfer limits from the MAXBFRU, UNITSZ, and BFRPAD operands as described. It then divides the result (lowest result, if there is more than one HOST macro) by the NCP buffer size, rounds down to the next lower integer, and uses the resulting value (or a maximum of 255) as the maximum buffer count for the line trace data.
- If the program contains no HOST macros, the generation procedure assumes an arbitrary value of 7 buffers for the TRANSFR operand.

Upon filling the specified number of buffers with line trace data, the NCP transfers the data to the destination host processor and continues to accumulate trace data in other buffers. The cycle of obtaining buffers, filling them, and transferring their contents to the host processor continues until the line trace function is ended by the operator.

```
[TWXID=( {inchars} {,outchars} )]
        {xon     } {,xon      }
```

(TWX terminals in network control mode only)

Specifies the characters used for both the initial ID answerback and subsequent prompting sequences for all TWX terminals on switched lines in network control mode.

inchars specifies the answerback and prompting sequences to be sent to all TWX terminals when the switched connection is terminal initiated.

outchars specifies the answerback and prompting sequences to be sent to all TWX terminals when the switched connection is controller initiated.

inchars and *outchars* must be specified as the hexadecimal representation of the EBCDIC characters to be used. Each may contain up to 20 EBCDIC characters. Any EBCDIC character may be specified, including those that translate into TWX carriage return or line feed.

The last character specified (presumably an X-on or WRU) is not transmitted during the initial ID answerback or on the subsequent prompting sequences that are sent as part of each TWX terminal read operation. Instead, the NCP generates and sends an X-on character in lieu of the last character specified. Prompting is selected by the host via the set destination mode and is always bypassed if data has been received in Read-ahead mode.

Note: Since the prompting sequence is sent at the beginning of *every* terminal read operation, you may want to specify a short sequence to avoid an excessive amount of time required in transmitting and printing the prompting sequence.

This operand has no meaning if communication with TWX terminals is only in emulation mode or if the terminals are in network control mode on nonswitched lines.

```
[TYPSYS={OS }]
        {DOS}
```

Specifies whether stage 2 of the NCP generation procedure is to be run under OS/VS or VSE.

```
[UCHAN={YES}]
       {NO }
```

Specifies whether user-written channel handling code will be provided for channel adapter positions 0 and/or 1. If this operand is omitted, UCHAN=NO is assumed.

If UCHAN=YES is specified, user-written channel handling code must be assembled in stage 2 of the generation through the SRCLO operand of the GENEND macro, and:

- TYPE4 specified in position 1 of the CA operand or TYPE4 in position 0 and TYPE4, TYPE4-0, or TYPE4-1 in position 1.
- the HICHAN operand must specify the highest subchannel address that can be used by either the user-written channel code or any EP lines defined.
- the LOCHAN operand must specify the lowest subchannel address that can be used by either the user-written channel code or any EP lines defined.
- CHANLNK=YES must be specified in at least one GROUP macro.

[UNIT=unit type]

(OS/VS only)

Specifies the type of device used for the assembler and linkage editor utility data sets during stage 2 of program generation under OS/VS.
You may specify either an actual device type (for example, UNIT=3330) or the name of a class of devices (for example, UNIT=SYSDA). The maximum number of characters you may specify is 8.

If you omit this operand, SYSSQ is assumed to be the unit type for the assembly steps and SYSDA is assumed for the link-editing steps.

Note: The utility data set for the linkage editor must reside on a direct-access device.

[USERLIB=dsname]

(user code with OS/VS only)

Specifies the name of the partitioned OS/VS data set to contain user-written object code and block handler routines. If this operand is omitted and the generation instructions call for user-written object code, the generation procedure assumes that the user code is in the NCP object library (SYS1.OBJ3705).

The data set name may or may not be qualified, depending on the QUALIFY operand in this macro. An unqualified name may be up to 8 characters long. The first character must be alphabetic or $, , or @. The data set specified by *dsname* must be cataloged.

[UT1=dsname]

(OS/VS only)

Specifies the name of a sequential OS/VS data set to be used as work space for the assembly steps (SYSUT1). The data set name may or may not be qualified, depending the QUALIFY operand of this macro. An unqualified name may be up to 8 characters long; the first must be alphabetic or $, #, or @. This data set must be preallocated and cataloged.

If you omit this operand, a temporary data set will be created during each assembly step using the type of device specified by the UNIT operand. The data set space provided is equivalent to SPACE=(1700,(800,800)).

[UT2=dsname]

(OS/VS only)

Specifies the name of a sequential OS/VS data set to be used as work space for the assembly steps (SYSUT2). The data set name may or may not be qualified, depending on the QUALIFY operand of this macro. An unqualified name may be up to 8 characters long; the first must be alphabetic or $, #, or @. This data set must be preallocated and cataloged.

If you omit this operand, a temporary data set will be created during each assembly step using the type of device specified by the UNIT operand. The data set space provided is equivalent to SPACE=(1700,(800,800)).

[UT3=dsname]

(OS/VS only)

Specifies the name of a sequential OS/VS data set to be used as work space for assembly (SYSUT3) and link-edit (SYSUT1) steps. The data set name may or may not be qualified, depending on the QUALIFY operand of this macro. An unqualified name may be up to 8 characters long; the first must be alphabetic or $, #, or @. This data set must be preallocated and cataloged.

If you omit this operand, temporary data sets will be created during each assembly step and each linkage edit step using the type of device specified by the UNIT operand. The data set space provided is equivalent to SPACE=(1700,(800,800)).

[VRPOOL=n]

Specifies the number of entries in the virtual route pool for this NCP. The number of entries should be determined by adding the following:

- The number of concurrently active virtual routes ending in this subarea.
- The maximum number of virtual routes that can be deactivated and then reactivated because of an explicit route failure.
- The number of active internal virtual routes needed by user written code.

The minimum value that can be specified is 1; the maximum is 252. The default for this operand is six times the value specified in the NUMHSAS operand, but not to exceed 252.

Note: Care should be exercised when determining the value of VRPOOL. An excessive value could waste NCP storage.

```
[XBREAK={integer}]
        {NONE    }
```

(start-stop lines in network control mode only)

integer

Specifies the number of character times the NCP is to place the break signal on the line to interrupt transmission from the terminal. This is applicable only for lines in network control mode and terminals for which the LINE and TERMINAL macros specify DUPLEX=FULL and FEATURE=BREAK, respectively.

The minimum value is 3; the maximum is 255.

NONE

Specifies that the NCP will not send break characters.

If you omit the XBREAK operand, a value of 3 is assumed if the network includes any duplex lines (DUPLEX=FULL in the LINE macro) terminals having the break function are attached (FEATURE=BREAK in the TERMINAL macro).

If the network does not include such lines and terminals, XBREAK=NONE is assumed if you omit the XBREAK operand.

```
[XITB={NO }]
      {YES}
```

(BSC stations in network control mode only)

Specifies whether the NCP is to insert transparent ITB sequences and error-information blocks in transparent text sent to stations which have intermediate block checking specified.

If you specify XITB=YES and the *first* parameter of the ITBMODE operand of the TERMINAL macro for the station specifies intermediate block checking, the program substitutes an error information block for each DLE ITB sequence in transparent text received from the station.

If you specify XITB=YES and the *second* parameter of the ITBMODE operand specifies intermediate block checking, the program inserts DLE ITB sequences into transparent text being sent to the station.

If you specify XITB=NO (or omit the XITB operand), no insertion of DLE ITB sequences and EIBs is performed for any BSC station.