IBM

Advanced Communications Function NCP Rel. 2 Data Flow

Student Text

IBM

Advanced Communications Function NCP Rel. 2 Data Flow

Student Text

Major Revision September 1979
This is a major revision of and obsoletes the previous edition, SR20-4621-0.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. Address comments concerning the contents of this publication to IBM Corporation, DPD Education - Publishing/Media Support, Education Center, South Road, Poughkeepsie, New York 12602

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.

Preface

This publication is a student text on data flow in the IBM 3705 Communications Controllers Advanced Communications Function (ACF) Network Control Program (NCP) Release 2.

Prerequisite knowledge of the IBM 3705 Communications Controllers is required to understand this material. The prerequisite information may be obtained in the following:

IBM 3704 and 3705 Communications Controllers Hardware (SR20-4544)

ACF/NCP Programming (SR20-4620-1)

Advanced Function NCP and Related Host Traces (SR20-4510-4)

ACF/NCP/VS Network Control Program System Support Programs Installation (SC30-3142)

If you require additional information, please refer to:

IBM 3705 NCP Instructions and Supervisor Macros (SR20-4512-2)

ACF/NCP/VS Network Control Program - Program Reference Summary (LY30-3043)

IBM 3704 and 3705 Communications Controllers Principles of Operation (GC30-3004)

ACF/NCP/VS Network Control Program Logic (LY30-3041).

List of Illustrations

Chapter 1:

Hardware and Programming Structure

Review of Hardware Facilities

Before going on to the components of the ACF/NCP network, this section reviews the hardware facilities which are used in the programming design, as well as dispatching code and techniques. In later sections the modules are related to an interrupt level or to a dispatched module. In either case a knowledge of the hardware and dispatcher is required.

Levels of Programming

Because the communications controller is an interrupt-driven unit, the NCP directing the operation of that unit is made up of smaller programs or levels. Interrupts can be caused by the channel, the communication lines, or the program itself.

The controller has five program levels. Program level 1 has the highest priority; program level 5 (referred to as the background level) has the lowest priority. Because level 5 has the lowest priority, level 5 code runs when levels 1 through 4 are not executing. For a complete description of the five levels of the controller and the interrupt facility, refer to *IBM 3704 and 3705 Communications Controllers Principles of Operation* (GC30-3004), Chapter 2: System Structure.

Figure 1.1 is a chart of the programming levels indicating the operations performed at each level, the starting address, and the means by which the level gets control. Note that when an attempt is made to execute an instruction at location X'0000', the NCP detects a 'branch to zero', regardless of the program level. A 'branch to zero' abnormally ends program execution.

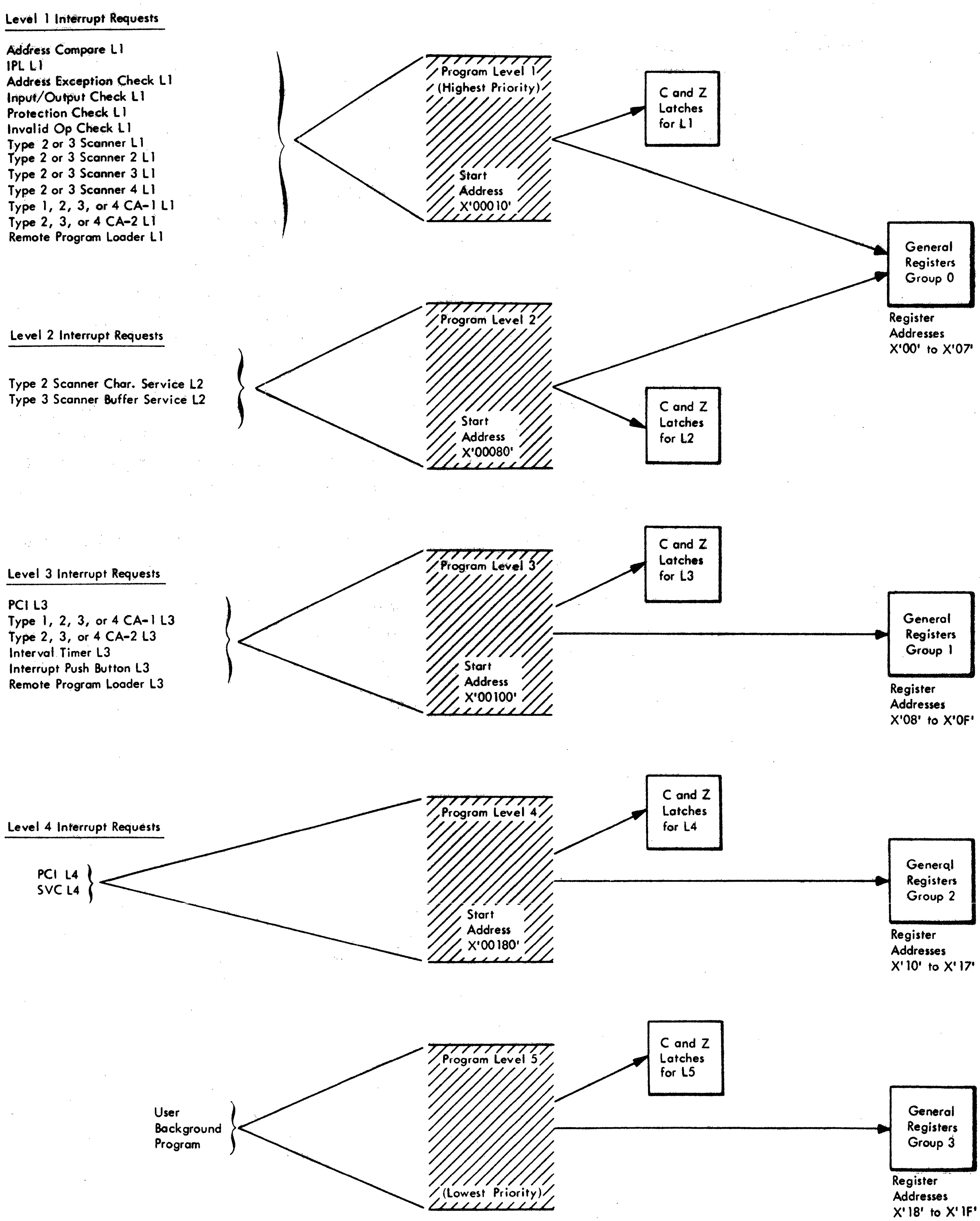


Figure 1.1 Program Levels

Level 1, Address X'0010'

When a level 1 interrupt occurs, control is given to the level 1 router, which is located at address X'0010'. By examining the contents of external registers, the router determines the cause of the interrupt and passes control to one of the following handlers: the program exception check-handler, the address trace module, the channel adapter check-router, the communications adapter check-handler, or the abend module.

Level 2, Address X'0080'

When a level 2 interrupt occurs, control is given to address location X'0080'. The level 2 router determines if the interrupt was a normal character service request (type 2 scanner) or normal end of buffer, block, or message (type 3 scanner). The address of the router is located in the CCB. The level 2 router itself processes hardware error and exceptional conditions.

Level 3, Address X'0100'

When a level 3 interrupt occurs, control is given to address location X'0100'. By examining the external registers, the level 3 router determines the cause of the interrupt, then passes control to one of the following interrupt handlers: the channel adapter input/output supervisor, the communications-line timer service, the communications control program queue-handler (signaled by a PCI), or the panel support module.

Level 4, Address X'0180'

When a level 4 interrupt occurs, control is given to address location X'0180', the level 4 interrupt handler. A level 4 interrupt is requested by (1) a level 5 supervisor call (SVC) or (2) a level 1 or level 3 program-controlled interrupt (PCI).

An SVC interrupt occurs when a supervisor macro is issued in program level 5. The program issuing the macro specifies certain parameters. After decoding the SVC code, the supervisor nucleus loads these parameters into registers and calls the appropriate supervisor SVC routine to process the request.

If the interrupt is a program-controlled interrupt (PCI), the interrupt handler branches to the address in the PCI vector table to process the request. Level 1 or level 3 place queue control blocks (QCBs) on level 4 dispatching queues. The level 4 dispatching queues represent tasks which must be dispatched in level 5. Level 4 PCI requests the supervisor to search the dispatching queues for a level 5 task.

Level 5

All level 5 tasks are dispatched by the level 4 task dispatcher. The entry point of each task is provided as a field in the queue control block (QCB), which is scheduled by placing the QCB in one of the supervisor dispatching queues. The dispatching of level 5 tasks is covered later in the supervisor section.

Interrupt Scheduling

Each programming level, except level 5, has an 'interrupt pending' latch and an 'interrupt entered' latch. An 'interrupt pending' latch is set for levels 1, 2, 3, or 4 by hardware service requirements. If a program check occurs, the level 1 'interrupt pending' latch is set. If a line requires service, the level 2 'interrupt pending' latch is set. Channel service requires the level 3 'interrupt pending' latch to be set. The level 3 latch is set for service by the channel adapter, but service is initiated by a PCI (OUT X'7C') from the level 4 supervisor. Level 4 is initiated in levels 1 and 3 by a PCI (OUT X'7D') or by supervisor calls (SVC) from level 5.

Interrupt levels may be masked off to prevent interrupts. Levels 2 through 5 may be totally suppressed. Level 1 may be masked to ignore channel adapter and scanner interrupts for test purposes. If the level is not masked off and an interrupt is pending, the interrupt is not allowed if any of the following conditions exist:

- A higher-priority interrupt request is present.
- The program level to be interrupted is already entered ('interrupt entered' latch is on).
- The program level to be interrupted is masked.
- A type 3 communication scanner cycle-steal request exists.
- A type 2, 3 or 4 channel adapter cycle-steal request exists.

At the time an interrupt is honored, the 'interrupt entered' latch for that program level is turned on. The 'interrupt entered' latch is a hardware latch which signals the controller that the associated program level has been entered. As long as this latch is on, no other interrupts to this program level are honored. The general registers and condition latches for this level are safe from change by another interrupt. The 'interrupt entered' latch is turned off either by an EXIT instruction executed at this level or by a reset condition to the entire controller.

After each instruction is executed, the controller tests for priority conditions before executing the next instruction. The type 3 communications scanner and type 2, 3 or 4 channel adapter cycle-steal requests occur between instructions. In addition, a higher-priority program level may need control. If level 3 code is executing ('interrupt entered' latch on) before executing each additional instruction the controller checks, in sequence, the 'level 1 entered' latch, 'level 1 pending' latch, 'level 2 entered' latch, 'level 2 pending' latch, and 'level 3 entered' latch. This sequence returns control to level 3 for another instruction execution.

If a second level 3 interrupt was pending, it is not checked in the sequence because the 'interrupt entered' latch is tested first. If the 'level 2 pending' latch was set, as in the previous example, level 2 code starts executing. The 'level 2 interrupt entered' latch is turned 'on' and level 2 executes until an EXIT instruction turns off the 'interrupt entered' latch. When the between-instruction check is made after the level 2 EXIT instruction, the level 2 interrupt entered latch is off, so the 'level 2 interrupt pending' latch is checked. If that latch is on, the level 2 code executes again with the 'interrupted entered' latch turned on a second time. If the 'level 2 pending latch is not on, the check returns control to level 3 where the 'interrupt entered' latch is still on. The level 3 code continues, unaware of the interrupt.

Hardware and Programming Structure Summary

The IBM 3705 Communications Controllers provide hardware support for five programming levels. The first four levels are interrupt-driven code, each having an absolute hardware address to begin instruction execution. The fifth level is dispatched under the control of the level 4 supervisor.

Chapter 2:

Network Control Program Overview and Data Flow

Identifying the Major Components

This section identifies the major components of the network control program and the program level in which the components operate. This material serves as the foundation upon which the detail of future sections is built. The major components of the network are covered in the order of subsequent topics.

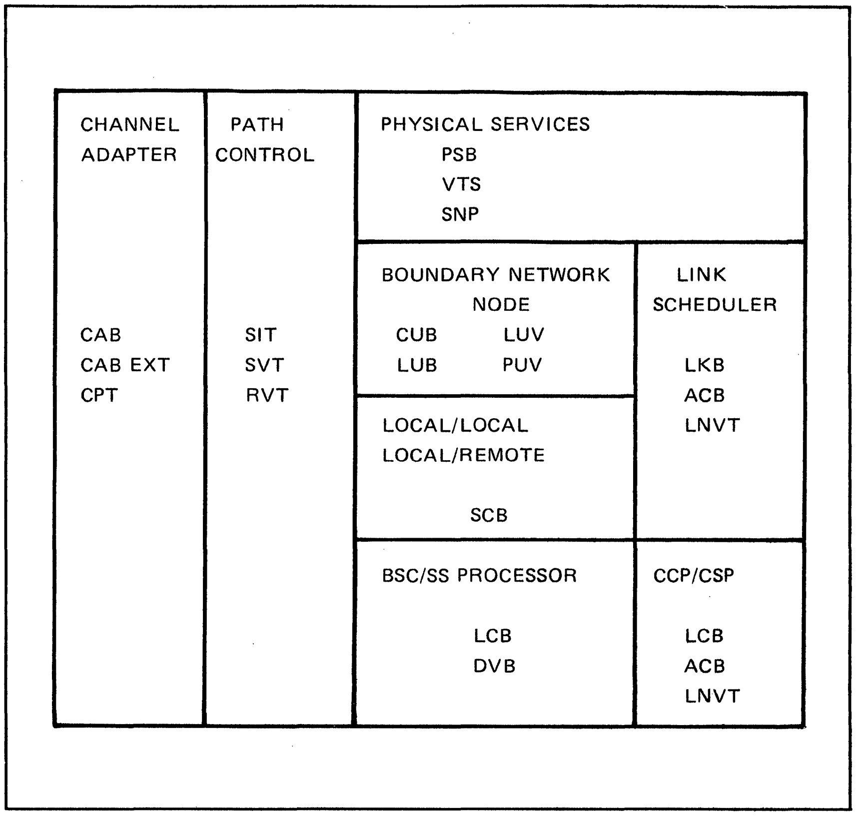


Figure 2.1. NCP Components

Network Control Program Supervisor

The NCP supervisor serves primarily as the interface between the background tasks running in level 5 and the routines running in levels 1, 3, and 4. When levels 1, 3, or 4 require data to be processed by background tasks, the tasks are scheduled via the supervisor. The supervisor queues the data and schedules the correct background processing task. Conversely, as background tasks require initiation of input or output, manipulation of queues, management of buffers, and similar tasks, the task requests are presented to the supervisor. The supervisor then processes those requests as required.

The supervisor executes in level 4. The primary control blocks used by the supervisor are: byte direct addressables (XDB), halfword direct addressables (XDH), word direct addressables (XDA), extended halfword direct addressables (HWE), extension of HWE (HWX), queue control blocks (QCB), path information units (PIU), and the SVC vector table. The supervisor code is executed and provides services for all of the routines identified in this section.

Channel Adapter IOS

The channel adapter module is used to monitor and control the hardware channel adapters within the 3705 controller during a data transfer to or from the host. There are four types of channel adapters; however, only three types are used in NCP mode for programming purposes within the controller. Therefore, there are three types of IOS operation: (1) type 1 adapter, (2) type 2 or type 3 adapter, and (3) type 4 adapter.

Type 1

A 3705 can have a type 1 CA for NCP mode or PEP mode, or the type 1 adapter can be used for emulation programming for (EP), with a second adapter (type 2 or type 3) for NCP mode only.

A type 1 channel adapter operates in data transfers of four bytes. Only one type 1 channel adapter may be installed in a 3705. If the type 1 is installed with a type 2 or type 3 channel adapter, the type 1 is used for emulation mode; the type 2 or type 3 is used for NCP mode. The type 4 and type 1 channel adapters may not be installed in the same 3705.

Type 2 or Type 3 Adapter

The 3705 operates in NCP mode with a type 2 channel adapter to a single processor. The type 3 channel adapter is used for a single processor with alternate path or as an interface to tightly coupled multiprocessors.

The type 2 and type 3 channel adapters operate in cycle steal mode for the capacity of channel words (CWs) as defined in the NCP HOST macro.

Two type 2 or type 3 channel adapters may operate concurrently in NCP mode on one 3705. If a type 1 or type 4 channel adapter is installed for emulation mode, only one type 2 or type 3 channel adapter is supported in NCP mode.

Type 4

A 3705 can have one to four type 4 channel adapters operating concurrently in NCP mode. Emulation mode is supported concurrently by two type 4 channel adapters. If a type 2 or type 3 channel adapter is installed in the 3705, only one type 4 channel adapter is allowed for emulation mode.

The type 4 channel adapter operates in cycle steal mode for the length of an NCP buffer or end of message, which ever is less.

Path Control

There are four routines referred to as path control; (1) intermediate network node (INN) 'path control', (2) boundary network node (BNN) 'path control out delayed', (3) boundary network node (BNN) 'path control in immediate', and (4) boundary network node (BNN) 'path control in delayed'.

Intermediate network node (INN) path control routes all PIUs from all NCP sources. 'INN path control' is also referred to as 'path control'.

The boundary network node (BNN) 'path control out delayed' converts the FID1 format to FID2 or FID3 format. After FID conversion, if the PIU length exceeds the PU MAXDATA value, the PIU is segmented. The PIU is then placed on the link outbound queue for transmission.

The boundary network node 'path control in immediate' is executed for all PIUs received on an SDLC link. If a PIU has been received from a type 4 PU 'path control in immediate' branches to 'INN path control' to locate the destination. If a PIU has been received from a type 1 or type 2 PU 'path control in immediate' queues the PIU for level 5 processing by BNN 'path control in delayed'.

BNN 'path control in delayed' is a level 5 dispatched task. All PIUs are initially associated with the PU. BNN 'path control in delayed' associates the PIU with the PU or one of the LUs. The PIU is converted from a FID2 or FID3 format to FID1 format.

INN Path Control

'INN path control' code is executed on all PIUs flowing in the NCP. A PIU received over the channel, local/local link, local/remote link, or from NCP physical services, SNA boundary node or BSC/SS processor is processed by path control.

INN path control directs the flow of path information units (PIUs) from all sources to its proper destination. INN path control uses the destination address field (DAF) from the PIU to access entries in the subarea index table (SIT), subarea vector table (SVT), and resource vector table (RVT). INN path control routine locates the appropriate path for the PIU and places the PIU on a queue control block (QCB) for processing by (1) a channel adapter block (CAB), (2) NCP physical services (PSB), (3) boundary network node (CUB or LUB), (3) local/local or local/remote link (SCB), or (5) BSC/SS processor. This module operates in program level 3. The entry is via a branch from the channel IOS or link scheduler IOS ('path control in immediate') or SVC (XPORT macro) from level 5 routines.

Boundary Network Node 'Path Control Out Delayed'

An outbound FID1 in boundary network node is processed by BNN 'path control out delayed' in level 5. BNN 'path control out delayed' converts the FID1 to a FID2 or FID3, segments the PIU if required, and places the PIU on a link outbound queue for transmission.

Boundary Network Node 'Path Control In Immediate'

When a PIU is received on an SDLC link, BNN 'path control in immediate' is invoked by a branch from the link scheduler. BNN 'Path control in immediate' checks for a PIU source of a type 4 PU; if the PIU is from a type 4 PU, the PIU is passed in level 3 to 'INN path control'. If the PIU is from a type 1 or type 2 PU, the PIU is queued on the PU CUB link inbound queue and BNN 'path control in immediate' exits from level 3.

Boundary Network Node 'Path Control In Delayed'

The PIU queued on the PU CUB link inbound queue invokes a level 5 task of BNN 'path control in delayed'. The PIU must be associated with the PU or one of the LUs. The FID2 or FID3 format is converted to FID1 before branching to an appropriate boundary network node connection point manager IN (CPM-IN) for additional processing.

Network Control Program Physical Services

NCP physical services represents the NCP as a function. PIUs addressed to NCP physical services are requests for NCP functions to be performed. NCP physical services provide functions such as activating or deactivating links, contacting physical units, and other control functions. These modules use the physical services control block (PSB).

The physical services routines operate in program level 5 via the task dispatcher. 'INN path control' schedules physical services by PIU requests. Responses are processed by 'INN path control' to locate the channel adapter block or station control block routing to an SSCP.

The NCP physical services has a 'connection point manager-in' queue (inbound error-handler queue), which is invoked by the link scheduler at the completion of a 'connect out' (dial), 'connect in' (answer), 'contact', or break in a link.

Boundary Network Node

The boundary network node (BNN) modules provide the interface to SDLC type 1 and type 2 devices. Local/local and local/remote support is not included in this code, as PIUs destined for a local or remote are enqueued directly on a station control block (SCB) by 'INN path control'.

BNN controls the session initiation and session status for the physical units and logical units attached to this 3705 controller. These modules operate in program level 5 via task dispatching. Type 1 and 2 physical units are defined by the common physical unit control block (CUB); logical units are defined by the logical unit control block (LUB). BNN modules are scheduled when they receive a PIU from 'INN path control' or 'path control in immediate'.

BNN processing includes conversion of FID1 to FID2 or FID3, FID2 or FID3 to FID1, segmenting, pacing between the host and BNN and pacing between BNN and the logical unit.

BNN enqueues outbound PIUs on a link outbound queue for the link scheduler. BNN passes inbound PIUs to 'path control' for routing.

Link Scheduler

The link scheduler executes in program level 3. The link scheduler is invoked for a specific link by the first 'activate link' command addressed to the link. The link scheduler has two basic functions: data transfer or command processing.

The link scheduler uses the service order table (SOT) to locate the physical units for that specific link. Each physical unit is checked for active status. If the physical unit is active, the link outbound queue is checked for outbound PIUs to transmit. After any allowed outbound PIU traffic has been sent, the physical unit is polled for inbound PIUs. When all physical units have been checked for data service at least once, the link scheduler switches to control functions. One control function ('connect out', 'connect in', 'contact', 'discontact') is attempted for one physical unit before the link scheduler returns to data transfer mode.

If there are no outbound PIUs for a link and if no active physical unit has inbound PIUs in response to polling, after the control cycle the scheduler suspends polling for a user-specified pause. Data queued to be transmitted is sent, but polling is suspended.

The link scheduler uses the link control block (LKB) to schedule link operations and maintain link status. The LKB is generated by a LINE macro of an SDLC group. The common physical unit block (CUB) or station control block (SCB) is used to schedule the station control and maintain station status for any SDLC physical unit.

SDLC Routines

The SDLC routines are used for the actual transmission of data on the link. The adapter control block (ACB) is used for link control. These routines operate in program level 2 via an interrupt from the hardware scanner.

SDLC routines are initiated by the link scheduler, providing addresses of processing routines in the character control block (CCB) and enabling the link for interrupts to begin processing.

BSC/SS Processor

The BSC/SS processor supports the BSC/SS devices in NCP mode that are attached to this communications controller. The processor uses the line control block (LCB) and the device control block (DVB) to schedule and control commands issued to these devices. Command processors are used to define the commands and the work scheduler is used to schedule the necessary tasks to complete the command. Command decoders and initialization routines initialize the lines and control their operation. Once initialized, the type 3 scanner operates in cycle steal mode for the length of an NCP buffer or end of data. The type 2 scanner requires character-service routines to handle the actual transmission of data across the line. Both types of routines use the adapter control block (ACB).

The command processors, work scheduler, and scheduler tasks operate in program level 5 via task dispatching. The command decoders and initialization routines operate in program level 3 via a PCI level 3. The character service routines operate in level 2 via a hardware interrupt from the scanner.

Unless BSC or SS devices are defined for NCP mode, BSC/SS processor support is not included in a network of SDLC terminals. The processor support routines are not included if BSC/SS devices are operated in emulation mode of a partitioned emulation program (PEP).

Overview Summary

There are four basic outbound destinations through the controller. The path that is taken depends upon the source and destination of the path information unit (PIU). The destination and sequences are as follows:

Physical services destination

1. From channel adapter IOS, INN path control, physical services processor
2. From SDLC routines, link scheduler, path control in immediate, INN path control, physical services processor

SDLC device or logical unit destination

1. From channel adapter IOS, INN path control, boundary network node, link scheduler, SDLC routines

2. From SDLC routines, link scheduler, path control in immediate, INN path control, boundary network node, link scheduler, SDLC routines

Type 4 PU destination

1. From channel adapter IOS, INN path control, link scheduler, SDLC routines
2. From SDLC routines, link scheduler, path control in immediate, INN path control, link scheduler, SDLC routines

BSC/SS processor destination

1. From channel adapter IOS, INN path control, BSC/SS processor, CCP
2. From SDLC routines, link scheduler, INN path control, BSC/SS processor, CCP

There are four basic inbound origins through the controller. The path that is taken depends upon the source and destination of the path information unit (PIU). The origin and sequences are as follows:

Physical services source

1. Physical services processor, INN path control, channel adapter IOS
2. Physical services processor, INN path control, link scheduler (SCB), SDLC routines

Type 1 or type 2 physical or logical unit source

1. SDLC routine, link scheduler, path control in immediate, path control in delayed, boundary network node, INN path control, channel adapter IOS
2. SDLC routine, link scheduler, path control in immediate, path control in delayed, boundary network node, INN path control, link scheduler, SDLC routines

Type 4 physical unit source

1. SDLC routine, link scheduler, path control in immediate, INN path control, channel adapter IOS
2. SDLC routine, link scheduler, path control in immediate, INN path control, link scheduler, SDLC routines

BSC/SS processor source

1. CCP, BSC/SS processor, INN path control, channel adapter IOS
2. CCP, BSC/SS processor, INN path control, link scheduler, SDLC routines

Chapter 3:

Network Control Program Supervisor

Purpose of the Supervisor

The NCP supervisor serves primarily as the interface between background tasks running in level 5 and routines running in levels 1, 2, 3, and 4. When levels 1, 2, 3, or 4 require data or a stimulus to be processed by the background tasks, the task is scheduled via the supervisor. The supervisor queues the data and schedules the correct background processing task. Conversely, as background tasks require initiation of input or output, manipulation of queues, management of buffers, etc., the task requests are presented to the supervisor. The supervisor then processes those requests as required.

Supervisor routines can be entered via a branch from levels 1, 2, 3, or 4 interrupt handler. Supervisor macros coded with the operand SUPV=YES generates a branch to a supervisor routine. The supervisor routine is then being executed as level 1, 2, or 3 code rather than level 4 code because it was entered directly, not because of an interrupt.

Levels 1 and 3 schedule the level 4 interrupt handler using the program controlled interrupt (PCI). Level 2 issues a PCI only to level 3. Levels 1 and 3 place a queue control block (QCB) on one of the four level 5 task queues and PCI to level 4 to dispatch a level 5 task.

Level 5 always uses a level 4 SVC interrupt to request supervisor services.

Entry to the level 4 interrupt handler at address X'180' is caused in one of two ways: a level 5 SVC macro or a level 4 PCI.

Additional information on the instructions and internal macros may be found in *IBM 3705 NCP Instructions and Supervisor Macros* SR20-4512.

The level 5 SVC is created by an EXIT instruction. The EXIT instruction and two-byte SVC code immediately following are generated by a level 5 macro which is coded with an operand of SUPV=NO. The level 4 interrupt handler uses the SVC code supplied by the level 5 macro expansion to index into the SVC vector table. This table contains pointers to the various supervisor macro routines. The SVC code is the first seven bits of the sixteen-bit field. The remaining nine bits are qualifiers of the SVC.

A level 4 PCI interrupt also causes the level 4 interrupt handler to get control. In this case, the level 4 interrupt passes control to one of three routines via a branch table.

Normally the first entry of the branch table points to the second entry and the second entry points to the third. The third entry always points to the dispatcher. A level 4 PCI interrupt normally causes the dispatcher to get control.

When the free buffer threshold is reached, the second entry is replaced with the address of the routine to generate a slowdown message. Each time the lease buffer routine (LEASE macro) is executed by a branch from level 3 or SVC from level 5, the count of remaining buffers is checked against the threshold value. If slowdown mode is required, the address of the slowdown message routine is placed in the branch table and slowdown bits are set in the direct addressable area.

If an unconditional buffer request is made and no buffers are available, levels 4 and 5 can be disabled. Level 5 is disabled by masking off level 5, and the address of the buffer allocation routine is placed in the first entry of the dispatcher branch table.

The entry code at X'180' is entered for SVC and PCI interrupts. An instruction of IN X'7F' provides a bit to define whether a PCI or SVC caused the interrupt. The result causes the supervisor to go either to the SVC interrupt handler or to the PCI branch table.

Task Management

A task in the network control program (NCP) is defined as a portion of code and a queue of data upon which the code operates. In the NCP, tasks are executed in level 5 only. If one portion of code operates upon two or more separate queues of data, the task dispatcher handles this portion of code as two or more separate tasks. The background level (level 5) of the NCP is made up of several routines that work together to schedule lines and process messages.

A task is defined at NCP generation when a queue control block (QCB) is assembled and linked to a unit of code. As queues become activated, their associated tasks are scheduled and initiated by the task dispatcher. Input queues (input to a task) are activated by the enqueuing of data to the queue. Enqueuing is provided by level 3 when a PIU is received over the communication lines or over the channel, or when the enqueuing is provided by one task passing control to another task. Pseudo-input queues (recording a stimulus for the task, but providing no data as input to the task) are activated by triggering the task upon the occurrence of some stimulus, such as a panel display request (panel interrupt key depressed).

There are several control blocks used by the dispatcher. Before we cover the method used by the supervisor, the topics that follow will acquaint you with some of the control blocks.

ACF/NCP/VS Network Control Program - Program Reference Summary (SY30-3043), can be used as a reference.

Direct Addressables (XDB, XDH, XDA)

There are three fixed areas of special pointers or special fixed data. These areas are:

- X'680' to X'6FF' Byte direct addressables (XDB)
- X'700' to X'77F' Halfword direct addressables (XDH)
- X'780' to X'7FF' Word direct addressables (XDA)

A special form of instruction with a base register of zero allows an implied base to refer to these fields, with the displacement providing the offset from the beginning of the area. The instructions are:

Insert Character

```
IC  5(0),16(0)
```

The 'insert character' instruction inserts the value at base location X'680' plus decimal 16 (X'10') into register 5 byte 0, for an effective address of X'690'. The true buffer size for this system, including the eight-byte prefix, is at X'690'.

Store Character

```
STC 5(0),16(0)
```

This instruction stores the value in register 5 byte 0 at location X'690' (X'680' plus X'10').

Load Halfword

```
LH  5,84(0)
```

The 'load halfword' instruction places the current free buffer count from X'700' plus decimal 84 (X'54') into register 5, creating an effective address of X'754'.

Store Halfword

```
STH 5,96(0)
```

The 'store halfword' instruction uses the value in register 5 to set the value of the system abend code at X'760' (X'700' plus decimal 96). The NCP sets a value at X'760' to indicate the reason the failure occurred.

Load

```
L  5,96(0)
```

The 'load' instruction moves the address of the last byte of storage from X'7E0' to register 5.

Store

```
ST  5,68(0)
```

The 'store' instruction records a pointer to the first free buffer at location X'7C4'.

The direct addressables provide key status indicators and pointers to the system control blocks. As the various NCP routines are covered, related direct addressables fields which provide status indicators as an aid in debugging are referenced. These are some of the initial fields which may be of special interest:

Byte direct addressables (XDB) X'680' to X'6FF'

- X'685' Control byte for dispatcher flags - bit 1 value of 1 indicates an active level 5 task
- X'687' BUILD macro buffer size
- X'689' Buffer pool and network status
- X'68A' General communications byte

- X'68B' Identifies program as NCP, EP, or PEP
- X'692' General communication byte
- X'693' SDLC subarea mask - 1 bits indicate MAXSUBA value
- X'694' SDLC element mask - 0 bits indicate MAXSUBA value

Halfword direct addressables (XDH) X'700' to X'77F'

- X'710' to X'72A' PEP emulation queue pointers
- X'736' to X'738' QCB for CCBs passed from level 2 to level 3
- X'73A' to X'742' Timer sub-control block
- X'754' Current free buffer count to slowdown; value of original buffers minus threshold (SLODOWN) percentage
- X'756' Free buffer threshold count plus one
- X'758' Number of defined communications lines
- X'760' System abend code
- X'770' Maximum byte count to host per host start I/O

Word direct addressables (XDA) X'780' to X'7FF'

- X'780' to X'7B8' Level 1/2 register save area
- X'7BC' Lagging address register (LAR)
- X'7C4' Pointer to first free buffer
- X'7D0' Remembrance of the last buffer in the buffer pool
- X'7D4' Remembrance of the first buffer in the buffer pool
- X'7D8' Pointer to extended halfword direct addressables (HWE)
- X'7DC' Pointer to HWE extension (HWX)
- X'7E0' Address of last byte of storage
- X'7E8' Pointer to the resource vector table (RVT) minus 2
- X'7F0' Pointer to the logical end of system free buffer pool

Extended Halfword Direct Addressables (HWE)

The address of extended halfword direct addressables is in a fullword at X'7D8'. The following are offsets:

- X'00' Initial free buffer count
- X'04' Address trace block pointer
- X'06' Check record pool pointer
- X'08' Line trace vector table pointer
- X'0A' Display/refresh/select table pointer
- X'0C' Panel control block pointer
- X'12' Line control select table pointer

- X'18' Address of last channel adapter block (CAB)
- X'2C' Non-device input queue pointer
- X'34' to X'43' Communication scanner control bytes
- X'44' Pointer to the physical services control block (PSB)
- X'48' Pointer to the subarea index table (SIT)
- X'4C' Pointer to the subarea vector table (SVT)
- X'50' Pointer to end of system immediate queue
- X'54' Pointer to beginning of system immediate queue
- X'58' Pointer to end of system productive queue
- X'5C' Pointer to beginning of system productive queue
- X'60' Pointer to end of system appendage queue
- X'64' Pointer to beginning of system appendage queue
- X'68' Pointer to end of system non-productive queue
- X'6C' Pointer to beginning of system non-productive queue
- X'70' Pointer to system active queue control block

HWE Extension (HWX)

The address of HWE extension is in a fullword at X'7DC'. The following are offsets:

- X'00' Dynamic Reconfiguration (DR) pointer to physical unit anchor block
- X'04' Dynamic Reconfiguration (DR) pointer to logical unit anchor block type 1
- X'08' Dynamic Reconfiguration (DR) pointer to logical unit anchor block type 2
- X'0C' Dynamic Reconfiguration (DR) pointer to RVT extension
- X'14' Performance Measurement Facility control block pointer
- X'18' 3705 II enhance features indicators

Queue Control Blocks (QCB)

The queue concept is basic to an understanding of the data flow within the NCP. A queue is a group of either data blocks (PIU or BCU) or queue control blocks (QCBs) connected first through last by address pointers. First in, first out (FIFO) is the basic mode of queue manipulation; however, last in, first out (LIFO) mode is also used.

A queue control block (QCB) has two queue pointers. One points to the first element in the queue. The first element points to the second, the second points to the third, etc. The second queue pointer points to the last element on the queue. If both addresses are zero, there are no elements in the queue.

The address field of a QCB is a fullword. The rightmost twenty bits are the address field. The leftmost bits are are used for other purposes.

There are three types of queues: input, pseudo-input, and work. Each type of queue provides different program support.

Input queues

An input queue contains elements to be processed by the task identified by the QCB. Some of the fields are:

- X'00' Major control block displacement, provides the displacement from the beginning of the control block which contains this QCB to the first byte of the QCB.
- X'00' to X'03' Address of first element queued
- X'04' to X'07' Address of last element queued
- X'08' 1010 1xxx indicates this is an input QCB
- X'08' to X'0B' Address of next QCB on this queue
- X'0C' Task and queue status
- X'0C' to X'0F' Task entry point

Figure 3.1 illustrates an Input QCB.

	INPUT QCB	
0	CTL BLK OFFSET	FIRST ELEMENT QUEUED
4		LAST ELEMENT QUEUED
8	PROTECT KEY	NEXT QCB QUEUED ADDRESS
C	TASK/Q STATUS	TASK ENTRY POINT
10	DISP. PRIORITY	SAVE AREA LIST ADDRESS
1A		PRIOR QCB QUEUED ADDRESS

Figure 3.1. Input QCB

Placing an element in a queue with the macro ENQUE ACTV=YES puts a task in the pending state. If no task is active, the pending task becomes the active task.

Pseudo-input Queue

A pseudo-input queue contains no elements. It has the same format as the input queue, but the task is triggered by a stimulus rather than by the enqueuing of an element. An example of a pseudo-input queue is the panel queue. When the interrupt key is pressed on the 3705 panel, a level 3 panel interrupt occurs. When level 3 determines that the interrupt was from the panel, level 3 branches to the level 4 supervisor routine which places the panel QCB on a dispatching queue.

The format of the pseudo-input queue is the same as the standard input queue. The only difference between an input queue and a pseudo-input queue is the means of dispatching the pseudo-input queue without data.

Work queue

A work queue does not have a task entry point. It is used as a queue to hold elements. The work queue is only sixteen bytes in length. The fields are as follows:

- X‘00’ to X‘03’ Address of first element queued
- X‘04’ to X‘07’ Address of last element queued
- X‘08’ 1010 0xxx indicates this is an work QCB
- X‘08’ to X‘0B’ Address of next QCB on this queue
- X‘0C’ Task and queue status
- X‘0D’ to X‘0F’ Reserved

Path Information Unit (PIU)

The element placed in a queue is either a queue control block (QCB), a block control unit (BCU) used in the BSC/SS processor, or a path information unit (PIU). The placing of a PIU on a QCB normally triggers scheduling. The flow of the network control program is initiated by receiving a PIU from the channel or line and passing the address from one queue to the next for processing.

The PIU is received in one or more NCP buffers. The PIU is made up of a transmission header (TH), request/response header (RH), and request/response unit (RU). The PIU is that portion received from the host or from the lines.

In addition to the area specified on the BUILD macro BFRS operand, each NCP buffer requires an eight-byte prefix for control purposes. The size of each buffer specified for the user is given in XDB at X‘687’. The true buffer size is in XDB at X‘690’. The buffer prefix field on each buffer is specified as follows:

- X‘00’ Buffer prefix chain field, four byte address of the next buffer in the chain, or zero if the last in a chain.

- X'04' and X'05' Reserved
- X'06' Buffer prefix data offset field. If the value is X'FF' the buffer is not allocated and is in the buffer pool. A value other than X'FF' provides the offset from the buffer prefix to the first byte of PIU text.
- X'07' Buffer prefix data count field. This field specifies the quantity of data from the location indicated by the offset that is valid in the buffer.

In the first buffer of a PIU is an event control block (ECB). The eighteen-byte ECB immediately follows the buffer prefix. The first buffer prefix offset of X'12' provides the offset past the ECB to the first byte of FID1 PIU. The PIU actually starts in the twenty seventh byte of the buffer, including prefix.

The ECB fields have various functions depending upon the type of PIU and current processing. Some of the ECB fields are as follows:

- X'08' ECB chain pointer
- X'0C' Block status flags. Specifies if the PIU is in a queue.
- X'0E' Set time interval, queued SDLC status, or PIU1 text count
- X'10' CAB address of 'Activate Physical' command, QCB for waiting task, address of LU-APPL QCB, or last buffer of PIU address
- X'12' Hold area for blocks N(s) or SDLC received C field
- X'14' LSA suspended status, number of host read CCWs, RVT destination status, or UIB status
- X'16' Reserved

Figure 3.2 illustrates the NCP buffer prefix and event control block fields.

If the FID type is FID0 or FID1, the next byte after the buffer prefix and event control block is the first byte of the PIU. The buffer prefix offset is X'12', X'1A' including buffer prefix.

The FID2 buffer prefix offset specifies an offset of X'16' and the four bytes (X'1A' through X'1D') are not used. The FID3 buffer prefix offset is eight bytes (X'1A' through X'21') and the eight bytes are not used.

Figure 3.3 illustrates the NCP buffer offset of FID0, FID1, FID2 and FID3 formats.

0	4	6	7
BUFFER CHAIN FIELD	RESERVED	DATA OFFSET *	BUFFER DATA COUNT

Buffer Prefix

8	9	C	D	E
ECB STATUS FLAGS	ECB CHAIN POINTER	BLOCK STATUS FLAGS	RESERVE	TIME INTERVAL/ QUEUED STATUS/ PIU TEXT COUNT

10	14	16
ACTPU ROUTE/ TASK QCB/ LU–APPL QCB/ LAST BUFFER OF PIU	LSA STATUS/ NO. HOST READ/ RVT–UIB STATUS	RESERVED

18
RESERVED

Buffer Event Control Block

* A value of X'FF' in the buffer data offset field indicates the buffer is in the buffer pool and not allocated.

Figure 3.2 NCP Buffer Prefix and Event Control Block Fields

FID0

1A		1C	1E	20	22	24	27	28	2A	2C	2E
FLAGS	UNUSED	DESTINATION ADDRESS FIELD (DAF)	ORIGIN ADDRESS FIELD (OAF)	SEQUENCE NUMBER	COUNT (RH+RU)	RH FLAGS	FID0 PAD	BTU COMMAND	BTU FLAGS	BTU RESPONSE	DATA

FID1

1A		1C	1E	20	22	24	27
FLAGS	UNUSED	DESTINATION ADDRESS FIELD (DAF)	ORIGIN ADDRESS FIELD (OAF)	SEQUENCE NUMBER	COUNT (RH+RU)	RH FLAGS	DATA

FID2

1A	1E	1F		21	22	24	27
FID2 ALIGNMENT BYTES	FLAGS	UNUSED	LOCAL DAF	LOCAL OAF	SEQUENCE NUMBER	RH FLAGS	DATA

FID3

1A	22	23	24	27
FID3 ALIGNMENT BYTES	FLAGS	LOCAL ID	RH FLAGS	DATA

Figure 3.3 FID Formats in NCP Buffers

FIDO

The FIDO PIU is used for all text transfers between a host application and a BSC/SS terminal. The FIDO is received from the host and sent to the BSC/SS converter. The FIDO is converted to a block control unit (BCU) for the BSC/SS processor. Text from a BSC/SS terminal is received in a BCU format buffer, sent to the BSC/SS converter, and converted to a FIDO before being sent to the host.

Including offsets from the beginning of the buffer, the format of the FIDO is as follows:

Transmission header (TH)

- X'1A' Transmission header. This field identifies the PIU as type 0 by xx00 xxxx in this byte.
- X'1B' Reserved
- X'1C' Destination network address
- X'1E' Origin network address
- X'20' Sequence number
- X'22' Text count of the RH plus RU (excludes TH)

Request/response header (RH)

- X'24' Request/response byte 0. this byte specifies that the PIU is a request or response, formatted or unformatted, sense or no sense included. PIU chaining by an application program is specified in this field, which specifies if this is the first, middle, last, or only PIU element.
- X'25' Request/response byte 1. This field specifies that an FME/DR1 is requested/sent, RRN/DR2 is requested/sent, and an exception response is requested/sent. The pace bit is not used in the BSC/SS processor.
- X'26' Request/response byte 2. Not used by BSC/SS Processor.
- X'27' Request/response byte 3. This field is a pad byte to align the RU on a halfword boundary.

Request/response unit (RU)

- X'28' RU0 byte 0. BTU command field. This field is covered in *ACF/NCP/VS Network Control Program - Program Reference Summary* (LY30-3043), Section 3: BTU Command and Modifiers.
- X'29' RU0 byte 1. BTU command modifier
- X'2A' and X'1F' RU0 bytes 2 and 3. BTU flags
- X'2C' RU0 byte 5. BTU system response.

This field is covered in *ACF/NCP/VS Network Control Program - Program Reference Summary* (LY30-3043), Section 8: BTU Responses.

- X'2D' RU0 byte 6. BTU extended response.
- X'2E' User data

FID1

The type 1 PIU is a field identification type 1 (FID1). This format is used for all control commands, and all text transfers between a host and boundary network node (BNN). If the PIU is being transferred over a local/local or local/remote link, the FID1 format is unchanged until it reaches the destination BNN.

Including offsets from the beginning of the buffer, the format of the FID1 is as follows:

Transmission header (TH)

- X'1A' Transmission header. This field identifies the PIU as type 1 by xx01 xxxx in this byte. This byte also specifies whether this PIU is the first middle, last, or only PIU segment. PIU segmenting occurs when a PIU from the host has a length greater than that defined by the MAXDATA operand coded on a PU macro.
- X'1B' Reserved
- X'1C' Destination network address
- X'1E' Origin network address
- X'20' Sequence number
- X'22' Text count of the RH plus RU (excludes TH)

Request/response header (RH)

- X'24' Request/response byte 0. This byte specifies whether the PIU is a request or response, formatted or unformatted, sense or no sense included. PIU chaining by an application program is defined in this field, which specifies whether this is the first, middle, last, or only PIU element.
- X'25' Request/response byte 1. This field specifies that an FME/DR1 is requested/sent, an RRN/DR2 is requested/sent, and an exception response is requested/sent. VPACING and PACING use the pace bit in this field.
- X'26' Request/response byte 2. This field specifies begin and end bracket, change direction, and code selection of EBCDIC or ASCII.

Request/response unit (RU) - Network commands from SSCP only.

A PIU with the formatted bit 'on' (RH byte 0, bit 4 -- hexadecimal value of X'x8' to X'xF') indicates the RU contains defined values. The RU length is variable. The values for formatted PIUs to or from an SSCP are provided in *ACF/NCP/VS Network Control Program - Program Reference Summary* (LY30-3043), Section 5: NCP Network Commands.

- X'27' User data begins at this address immediately following the TH and RH.
- X'27' Formatted SNA commands have a variable format based on the type of command. For additional information, refer to *ACF/NCP/VS Network Control Program Logic* (LY30-3041), Appendix A: Network Commands.

- X'2A' Network address for SSCP function management requests. A command to activate a link or contact a resource is addressed to NCP physical services in the DAF; the device to be affected by the command is addressed by this field.

FID2

The type 2 PIU is field identification type 2 (FID2). This format is used for all control commands and all text transfers between the boundary network node (BNN) routine and support of type 2 physical units (3274, 3276, 3770, 3600, 3650, 3660, 3790). The FID1 is received from the host and converted to a FID2 before being sent to the type 2 physical unit. A FID2 from a type 2 physical unit is converted to a FID1 by the BNN code before being sent to the host.

The FID2 is created from a FID1 by converting the two-byte OAF and DAF network address to a one-byte local address of the logical unit and session identifier and deleting the two-byte transmission header (TH) data count field. This conversion deletes four bytes from the FID1 requirements. Shifting the fields to the right places the following fields in the original FID1 buffer:

Transmission Header (TH)

- X'1A' Reserved four-byte area (residual left from the original FID1 on outbound PIUs)
- X'1E' Transmission header. xx10 xxxx in this byte identifies the PIU as type 2.
- X'1F' Reserved
- X'20' Destination (local address or session identifier)
- X'21' Origin (local address or session identifier)
- X'22' Sequence number

Request/response header (RH). Same as FID1

Request/response unit (RU). Same as FID1

FID3

The type 3 PIU is a field identification type 3 (FID3). This format is used for some control commands and all text transfers between boundary network node (BNN) code for support of type 1 physical units (3767, SDLC 3275 and 3277). The FID1 is received from the host and converted to a FID2 with the normal FID2 processing. The FID2 is converted to a FID3 before being sent to the type 1 physical unit. The FID3 commands directed to an SDLC 3275 or 3277 are processed by the NCP and are not sent on the link.

The FID3 is created from the FID2 by converting the six-byte transmission header (TH) of the FID2 to a two-byte TH of the FID3. The FID2 local address of one byte is converted to the low-order six bits of the last byte of the TH. The two leftmost bits specify the following:

- Bit 0 - 1=to/from application, 0=to/from SSCP
- Bit 1 - 1=to/from logical unit, 0=to/from physical unit

The first byte of the FID3 TH identifies the type of PIU. Deleting the four bytes of the TH from the FID2 makes four more alignment bytes available. Shifting the fields to the right provides the following fields in the original FID1 buffer:

Transmission header (TH)

- X'1A' Reserved eight-byte area (residual left from the original FID1/FID2 on outbound PIUs)
- X'22' Transmission header. This field identifies the PIU as a type 3 by xx11 xxxx in this byte.
- X'23' Application or SSCP indicator and local address

Request/response header (RH). Same as FID1

Request/response unit (RU). Same as FID1

Task States

At any given point in time, a task can be in any one of four logical states. The four states, under program control, are: active, pending, ready, or disconnected. Initially all tasks are in the 'ready' state. The state is specified in the QCB at an offset of X'0C' for all conditions.

When a QCB is generated the task status is 'ready'. A 'ready' task is available for execution, but there is no element in its queue or no stimulus to initiate it; therefore it is not in a dispatching queue.

When a QCB is placed in a queue by an ENQUE ACTV=YES macro or when a TRIGGER macro is executed, the task is changed from 'ready' to 'pending and disconnected', and is placed on one of the dispatching queues. The 'pending' status makes it available for execution. The 'disconnect' status identifies the QCB as having been triggered (in a dispatching queue) or not being eligible to be triggered; therefore, it will not be triggered again until the 'disconnect' status completes (a task should not be placed in the dispatching queue when it is already in the dispatching queue). Subsequent elements placed on a triggered QCB normally have an automatic trigger to the end of the QCB queue. When the level 4 supervisor looks for a task to make 'active', it takes the first pending QCB off the highest priority dispatching queue and schedules the task routine specified in the QCB field.

Only one task can be 'active'. The active task may issue SVC requests to level 4 for services, but remains the active level 5 task until it completes. If the active task is waiting for supervisor services (SVC), the second bit in byte X'0C' of the QCB is a '1' (task in wait state).

A task completes by issuing a SYSXIT macro. The SYSXIT macro ends the 'active' state. The 'disconnect' state is reset by a QPOST macro. If the task is to be available for immediate reexecution the normal macro sequence is QPOST followed by SYSXIT. If the first element on the QCB specifies the task is to be dispatched (offset X'08'), the QCB is triggered to the end of the dispatching queue and made 'pending'.

When a task is active, the byte direct addressable (XDB) at X'0685' has a value of x1xx xxxx. The address of the active task QCB is in the extended halfword direct addressables (HWE) at offset X'70'.

The following are the bit settings of the QCB in byte X'0C', which indicates the status:

```
0xx0 xxxx    Ready and not disconnected
1xx1 xxxx    Pending and disconnected
0xx1 xxxx    Active and disconnected
1xx0 xxxx    Disconnected
```

Figure 3.4 illustrates task state migration. All tasks which are in the 'ready' state are only in the ready state. If a task is 'pending' or 'active' the task is also 'disconnected'.

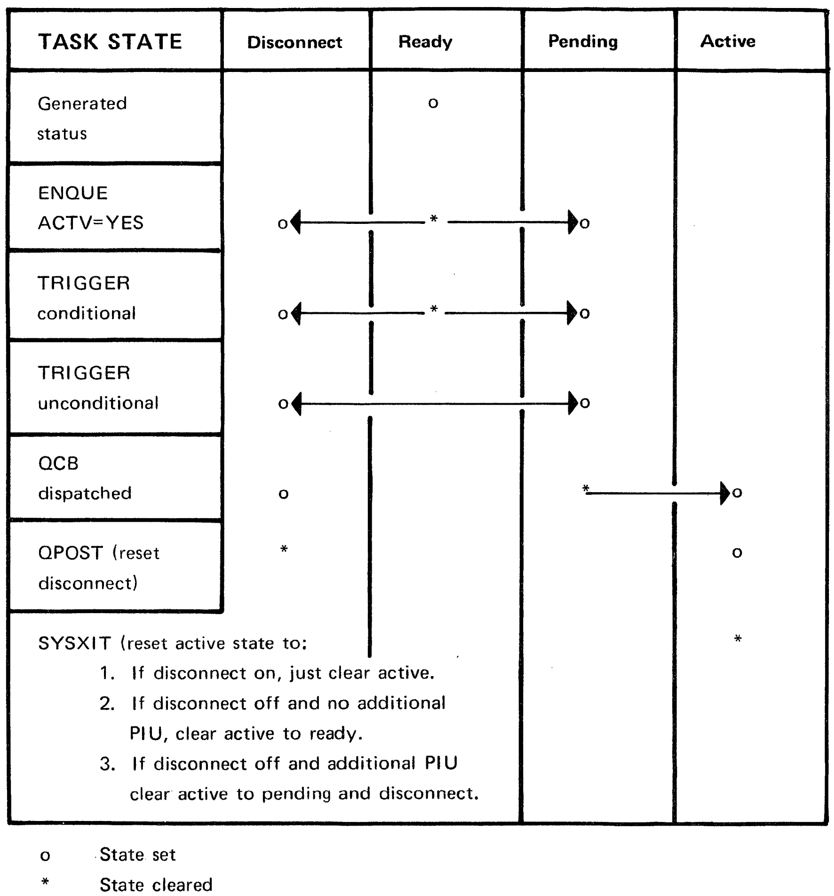


Figure 3.4. Task States

Task Dispatching Priorities

Tasks in the network control program have one of four task-scheduling priorities: appendage, immediate, productive, and nonproductive. All QCB tasks having the same priority are queued together.

Appendage tasks have the highest priority in the system. When the current active task relinquishes control, appendage tasks are dispatched from the appendage queue on a first-in, first-out (FIFO) basis. Appendage tasks are generally initiated by BSC/SS character service at the end of a line input or output operation. However, they can also be initiated by the supervisor or by level 5 tasks.

Immediate tasks have the second highest priority. Once processing for a line has started, all tasks necessary to initiate the input or output on the line are given the immediate priority.

Productive tasks have the third highest priority. A task is classified as productive if the end result of its execution is the initiation of output on either the channel or the communication line.

Nonproductive tasks have the lowest priority in the system. A task is classified as nonproductive if it is not capable of starting input or output operations.

There are definite reasons for having task scheduling priorities:

- Appendage tasks are used to handle an exceptional condition as soon as possible.
- Immediate priority improves performance. Lost subarea (LSA) tasks and BSC/SS line I/O tasks are executed as immediate priority. Once a routine associated with a BSC/SS line in the idle state receives control, the performance is better if all the routines necessary to initiate the transfer on this line are dispatched in succession before dispatching tasks associated with any other lines. The immediate priority accommodates such tasks.
- Productive tasks have a high potential for freeing buffers and a low potential for allocating buffers.
- Nonproductive tasks have a low potential for freeing buffers and a high potential for allocating buffers. Hence, productive tasks should be executed before nonproductive tasks.

The priority of a task can be changed dynamically by the CHAP macro.

The task dispatching queues are in the extended halfword direct addressables (HWE) at the offsets given below. The left offset is a fullword address of the first QCB queued and the right offset is a fullword address of the last QCB queued.

Task Queue	HWE Offset	
Appendage queue	X'64'	X'60'
Immediate queue	X'54'	X'50'
Productive queue	X'5C'	X'58'
Nonproductive queue	X'68'	X'6C'

The QCB identifies which task dispatching queue is to be used. At QCB plus an offset of X'10', the indicator to the TRIGGER macro is specified as follows:

```
10xx xxxx   Productive
010x xxxx   Immediate
001x xxxx   Appendage
000x xxxx   Nonproductive
```

Dispatching Tasks

When level 4 is entered by PCI or by SVC, the supervisor at CXABTST checks for PCI or SVC. A PCI uses the three branch-table entries for processing. If levels 4 and 5 are disabled for buffer allocation, the first entry is primed with CXALEAS, the buffer allocation routine. Normally, the first entry contains the address of the second entry. If the system is in slowdown, the address of the routine to generate the slowdown entry message is in the second entry (CXAEXSS). Normally, the second entry contains the address of the the third entry. The third entry contains the address of the task dispatcher (CXADISP). Figure 3.5 illustrates the supervisor processing.

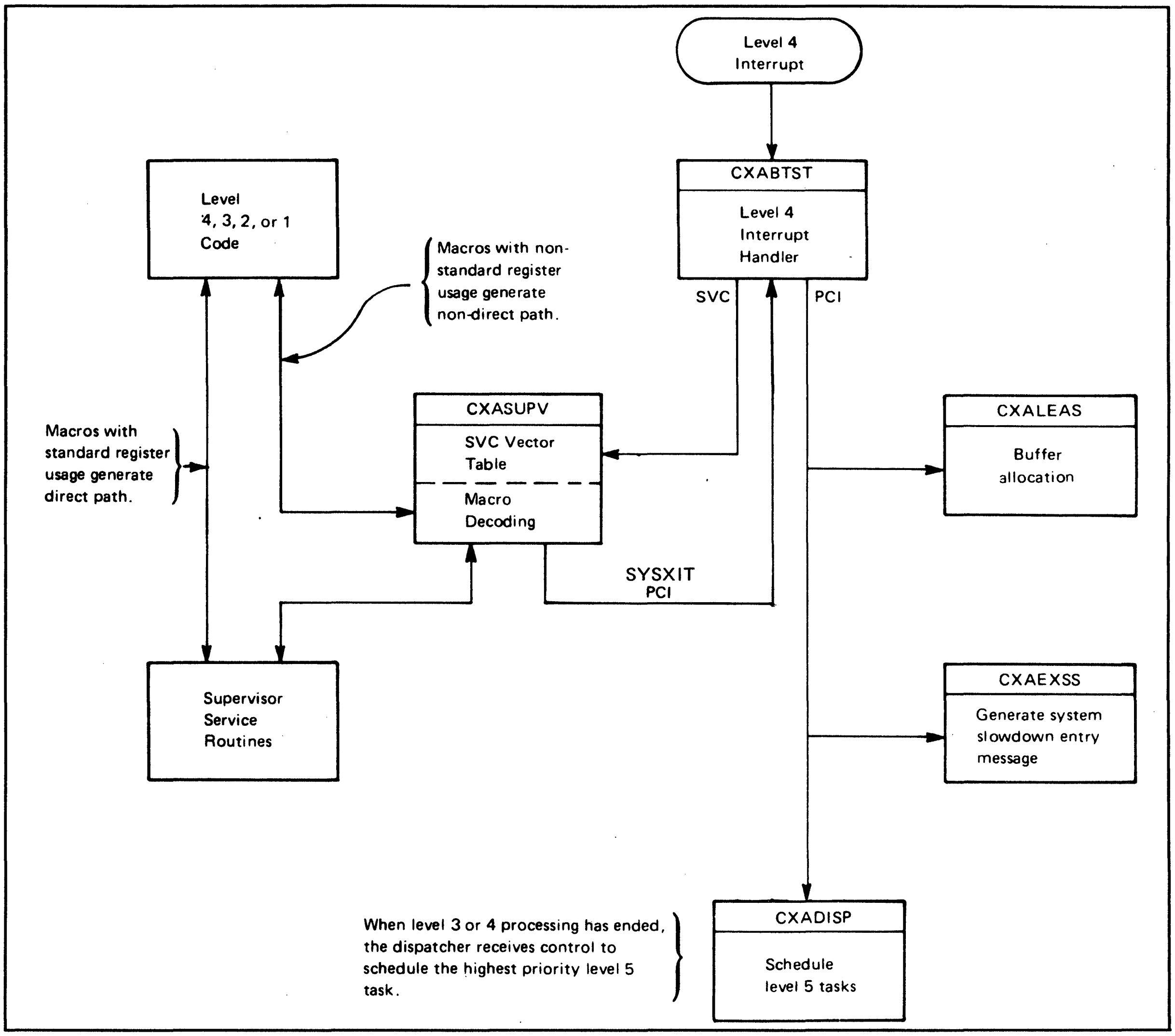


Figure 3.5. Dispatcher Entry Points

The CXADISP task dispatcher checks whether level 5 is enabled by testing the byte direct addressable (X'685') to see if dispatcher service is required. If service is not required the supervisor checks for an active task in byte X'685'. If this bit is on, an EXIT from level 4 returns control to the current active level 5 task. If there is no active task, the supervisor searches the dispatching queues.

The queues are scanned in a sequence of appendage, immediate, productive, and nonproductive. The first entry found is dequeued, the QCB address is placed in the level 5 register 2, the task entry point is placed in level 5 register 0, and the level 4 supervisor executes an EXIT to allow level 5 to begin execution. The active task's QCB address is saved at offset X'70' in the extended halfword direct addressables (HWE).

If no QCB is found, level 5 is disabled and an EXIT at level 4 places the controller in the wait state.

Figure 3.6 illustrates the dispatching sequence.

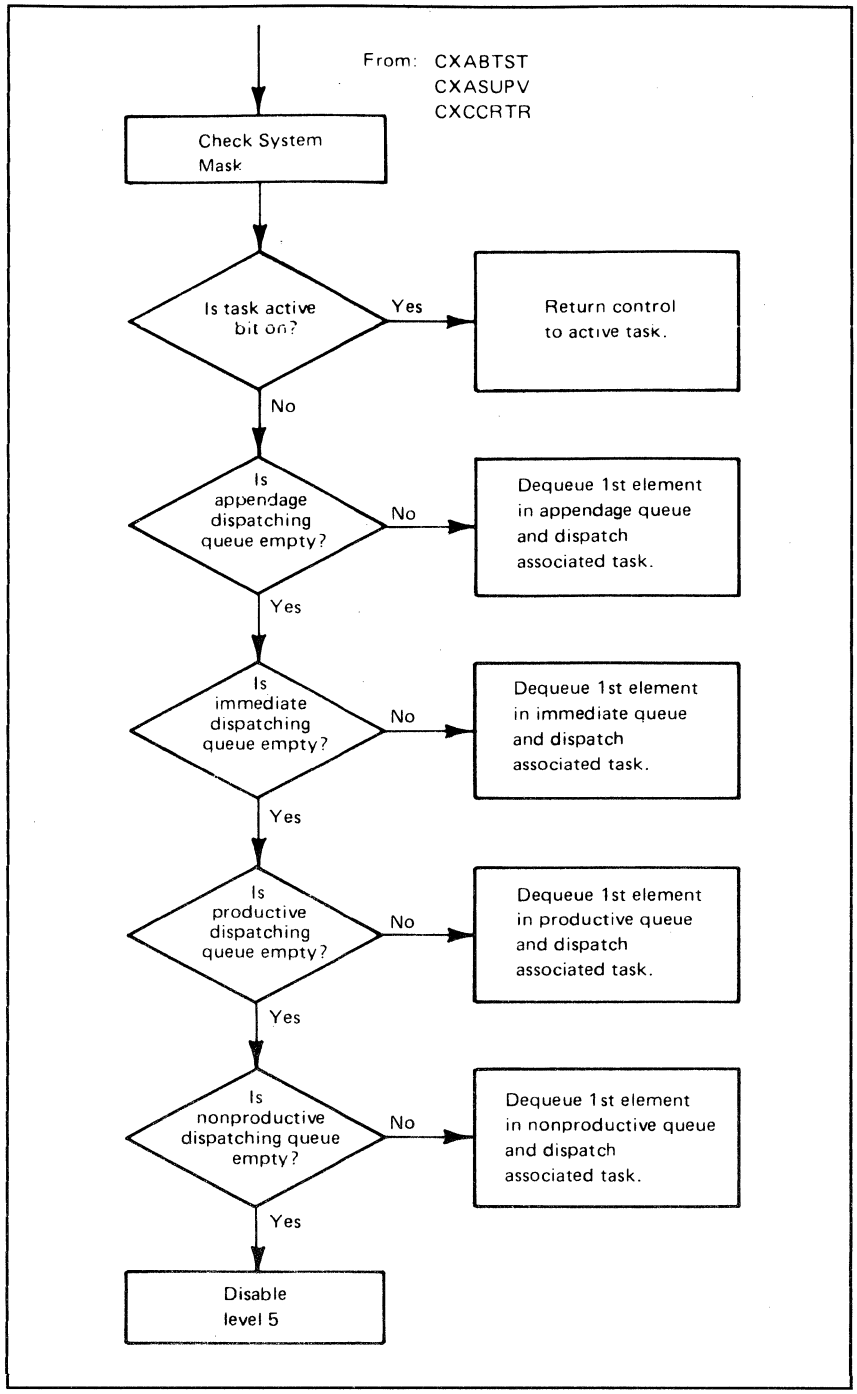


Figure 3.6 Dispatcher Execution Sequence

Supervisor SVC Services

When level 4 is entered by PCI or by SVC, the supervisor at CXABTST checks for PCI or SVC. An SVC goes to CXASUPV to decode the macro using the SVC decode table.

In addition to the task management routines, the supervisor provides queue management, buffer management, and supervisory services. All of these facilities are provided by macros. Macros at level 5 provide an SVC interrupt by an operand of SUPV=NO. Levels 1, 2, 3, and 4 branch directly to the appropriate routine by using a macro coded with an operand of SUPV=YES. The SVC is an EXIT instruction in level 5 with a 16-bit EXIT qualifier (seven-bit SVC identifier, and nine-bit SVC qualifier).

All of these services are covered in the *3705 Instructions and Macro Instructions Student Text* (SR20-4512).

Performance Measurement Facility

The performance measurement facility (PMF) which generates statistics on cycle and buffer utilization. PMF is only available on 3705 II and requires the cycle utilization counter (CUCR). The CUCR is a 15-bit binary counter that counts utilized machine cycles used for instruction execution, cycle steal operations, or maintenance.

The CUCR counter advances once for each 8 utilized cycles. CUCR data may be accessed using an Input X‘7A’ instruction and reset with an Output X‘7A’ instruction. For information about starting, stopping, and resetting the counter from the operator panel, refer to *Guide to Using the IBM 3705 Communications Controller Control Panel,* GA27-3087.

When the PMF is active, each 100 milleseconds PMF reads the CUCR and the current available buffer field count. PMF provides statistics for minimum, maximum, and average cycle usage and buffer availability for five time periods; 1.6 seconds, 25.6 seconds, 6 minutes 50 seconds, 1 hour 49 minutes, and 29 hours 8 minutes.

The PMF statistics are maintained in the following control blocks:

- Performance measurement facility (PFM) control block
- Dummy data buffer (DDB) - Performance measurement facility (PMF)

The address of the PMF control block is in the XDA at offset X‘7F8’ and the HWX at offset X‘14’. The PMF control block contains the address at offset X‘04’ of the DDB for the cycle utilization fields; at offset X‘08’ is the address of the DDB for buffer utilization.

The counters are binary values and displayed as hexadecimal values. Each time period statistics are accumulated fifteen times, and the sixteenth time overflow to the next time period. After the values overflow to the next time period counter, the values are reset.

Supervisor Summary

The supervisor provides service facilities to level 1, 2, and 3 routines by direct branch. The supervisor provides level 5 services facilities by SVC interrupts to level 4. Entry to level 4 by PCI normally causes the supervisor to search the dispatching queues for queued work to be dispatched in level 5.

If a partitioned emulation program (PEP) is defined by the BUILD macro operand of TYPGEN=PEP or TYPGEN=PEP-LR, a concurrent emulation and NCP program is generated. The NCP performance may be degraded by heavy emulation usage. All emulator code executes at levels 1, 2, and 3. Therefore, the emulation code has priority over the NCP dispatcher and level 5 dispatched routines.

When register 0 (instruction address register) addresses an EXIT instruction (X'B840'), the program level terminates. If the EXIT is in level 5, there is an additional 16-bit SVC qualifier.

Chapter 4:

Channel Adapter IOS

Channel Adapter Definition

To transfer data across the channel interface, we must give the NCP definitions of the channel adapter type and the host buffers.

The CA=operand and CHANTYP=operand in the BUILD macro defines the types and number of channel adapters installed in the 3705. This operand also selects the appropriate IOS module to be included in the generation. There are three modes of operation, (1) four-byte transfer (type 1 channel adapter), (2) cycle-steal for the length of a buffer (type 4 channel adapter), and (3) cycle-steal for multiple buffers (type 2 and type 3 channel adapter).

The control block for a channel adapter is the channel control block (CAB). Each channel adapter will have its own CAB. The initialization values of each CAB and CAB extension is not completed until an SSCP activates the NCP over a specific channel adapter. The initialization values are each host definition is maintained in a channel parameter table (CPT).

A channel parameter table (CPT) is generated for each subarea defined in the SUBAREA operand of a HOST macro. The HOST macro provides the correct values to be placed in a channel parameter table (CPT) for proper channel operation. The MAXBFRU and UNITSZ operands define to the NCP the input area that the host allocates on any channel read operation.

The maximum amount of data in one PIU that the NCP may transfer to the host in a single channel transfer is defined by the BUILD operand of TRANSFR. TRANSFR=count where count is the maximum number of NCP buffers to contain a single PIU. If BFRS=64, and TRANSFR=10, the maximum PIU size (TH, RH, RU) is 622 bytes; (64 x 10) - 18 byte first buffer ECB.

If TRANSFR is defined as a large value the HOST macro limits a single PIU to (MAXBFRU x UNITSZ) - BFRPAD. This calculation uses all allocated host buffers to contain one PIU.

The maximum number of PIUs which may be sent as a single channel transfer depends upon the size of the PIUs. In the previous paragraph, the example illustrated how one PIU may use all host buffers. If all PIUs sent to the host are small enough so that each PIU and the buffer pads fit in a single host buffer, as many PIUs may be sent in a single channel transfer as there are host buffers available (MAXBFRU quantity).

Normally the PIUs do not all fit in a single host buffer nor will all the host buffers be required for one PIU. A combination of PIU lengths occurs where some PIUs require one host buffer while others require multiple host buffers.

The INBFRS operand determines the number of buffers the NCP should LEASE for a host 'write' operation to the 3705. When the number of INBFRS is totally depleted, the INBFRS quantity is LEASED to continue the 'read' from the host. Once NCP buffers are allocated for a host 'write' they are allocated until used by this or a future host 'write'. At the end of a host 'write' the unused INBFRS are not returned to the available buffer pool.

Host Writes to the NCP

The host channel program must always start with a control command. A 'write' operation from the host must start with a 'write start zero' (WS0) or a 'write start one' (WS1). The first 'write' must be a WS0. After the first 'write', the 'write start' commands alternates between WS0 and WS1 with the successful completion of each write channel program. The 'write start' commands are X'31' for WS0 and X'51' for WS1.

When the channel adapter receives the control command, the CA generates a level 3 interrupt into IOS. When channel IOS receives the 'write start' (WS) control command, the command determines that the host wants to write to the 3705. The WS control command is compared to the expected WS command in the channel adapter control block (CAB at offset X'25'). If the two commands are equal, the expected WS command is flipped and the enqueue count and skip count are reset to zero. Data is transmitted until a complete PIU is received or until an unexpected control command signals an error condition on the channel interface. When a PIU is completely received, the PIU is passed to 'path control', and the enqueue count is incremented by 1. The address of the last PIU passed to 'path control' is recorded in the CAB extension at offset X'18'.

Channel error recovery for host writes consists of restarting the host write CCWs from the beginning. Restarting write CCWs results in NCP receiving an 'unexpected' WS command. If an unexpected WS control command is received, the enqueue count is added to the skip count. As each PIU is received a second time, rather than pass the PIU to 'path control' again, the skip count is decremented until the count is zero.

The enqueue count and skip count fields are reset to zero by IOS when IOS receives the next 'expected' WS control command.

The host can send multiple PIUs to the NCP with one host write. PIUs are separated logically by using a host channel control word (CCW) with command chaining between PIUs.

Host Reads from the NCP

The host 'read channel program' must start with a control command. The control command is either a 'read start zero' (RS0) or a 'read start one' (RS1). At the completion of the read channel program in the host, the 'read start' next expected (CAB at offset X'27') is reversed. The RS0 and RS1 commands alternate with the successful completion of each read channel program. The RS control commands are X'32' for RS0 and X'52' for RS1.

The host does not initiate the read channel program without first receiving an attention interrupt from the 3705 controller. The attention interrupt is the means by which IOS lets the host know that it has data to send across the channel.

PIUs to be sent to a host over a specific channel adapter are first queued on the CAB extension at offset X'44'. Additional processing transfers the PIU to the channel intermediate queue (CAB extension offset X'00'). When the host issues a RS command the PIUs to be sent are moved to the channel hold queue (CAB extension offset X'08'). The PIUs on the channel hold queue are held until the next RS command is received.

Before the data is put on the channel intermediate queue, the data length is checked to ensure that the PIU meets the host buffer requirements. IOS sets up the channel adapter to present the attention interrupt to the host. The attention interrupt causes the host to execute its read channel program starting with an RS control command.

On receiving the RS control command, IOS compares the received RS command with the 'expected' RS command in the channel control block (CAB) at a displacement of X'27'. If the RS command received is the expected command, IOS flips the expected RS command and purges previously sent PIUs from the hold queue. IOS removes the first PIU from the intermediate queue. While a PIU is being sent to the host the pointers to the PIU buffers are in the CAB extension. After the PIU has been sent the PIU is added to the hold queue.

Channel error recovery for host reads consists of restarting the host read CCWs from the beginning. Restarting read CCWs results in NCP receiving an 'unexpected' RS command. If an unexpected RS control command is received, the host is requesting a retransmission of the previous PIUs. The previously sent PIUs are available on the channel hold queue. The NCP retransmits all of the PIUs on the channel hold queue.

Before each PIU is sent to the host, pad characters may be required as a reserved area for host internal control. The count of pads is coded on the HOST macro BFRPAD operand. Following the pad, if the pad and PIU are less than or equal to the length of one host buffer, the IOS sends a complete PIU (UNITSZ value). IOS never lets the host CCW 'channel stop' (recognize a CCW zero count), but forces chaining to avoid a channel stop by the channel. If the end of a PIU forces chaining, the second PIU begins in the next host buffer, with leading pads sent before the PIU. If the original PIU had additional data beyond a single host buffer, the data continues into the subsequent host buffer.

When all PIUs in the hold queue have been sent, IOS presents ending status to the host. If more PIUs are available for the host, IOS adds an 'attention' to the status being sent back to the host. This 'attention' status indicates to the host that a new 'read' is needed for the 3705 controller. The host responds with a new 'read channel program'.

A second method by which the channel IOS indicates to the host that it has PIUs for the host is to send a status modifier (SM) at the end of the host 'write' portion of a write/read combination channel program. The SM tells the host to skip over a NOP that follows the 'write' CCWs and continue with the 'read' CCWs. This method eliminates the need for excess asynchronous interrupts on the channel. At the end of the read CCWs IOS presents final status of channel end, device end, and unit exception. This facility is specified on the HOST operand of STATMOD=YES.

The following channel programs illustrate the host channel program for a 'read', 'write', and the 'write/read' sequence.

Read Channel Program

```
CCW 32 or 52,*,X'60',1
CCW 02,BUF1,X'60',L'BUF1
  --
  --      Read Commands
  --
CCW 02,BUFn,X'60',L'BUFn
CCW 03,*,0,1      NO-OP
```

Write/Write Break Channel Program

```
CCW 31 or 51,*,C'60',1
CCW 01, BUF1,X'60',L'BUF1
  --
  --      Write and/or
  --      Write break commands
  --
CCW 09,BUFm,X'60',L'BUFm'
CCW 03,*,0,1      NO-OP
```

Write/Write Break and Read Combination Channel Program

```
CCW 31 or 51,*,X'60',1
CCW 01,BUF1,X'60',L'BUF1
  --
  --      Write and/or
  --      Write break commands
  --
CCW 09,BUFn,X'60',L'BUFn
CCW 03,*,0,1               (NOTE 1)
CCW 32 or 52,*,X'60'1
CCW 02,BUFn,X'60',L'BUF1
  --
  --      Read commands
  --
CCW 02,BUFn,X'60',L'BUFn
CCW 03,*,0,1      NO-OP
```

NOTE 1: This NO-OP is not essential for correct operation, although it may be desirable for compatibility when the status modifier option is selected. If the status modifier option is not selected, the 'write break CCW' may be command-chained to the 'read start CCW'. If status modifier is selected, the NO-OP should be included and should not be command-chained to the 'read start CCW'. If compatibility is desired, include the NO-OP in the channel program and turn the command chain flag on and off as needed.

Channel Adapter Types

Type 1 Channel Adapter

The type 1 channel adapter support requires an interrupt at level 3 for each four bytes transferred. As the number of INBFRS is depleted, the level 3 code branches into the level 4 supervisor routine to obtain a new supply of buffers equal to the INBFRS number. Then the 'read' (host 'write') operation continues to the completion of the host 'write' or another allocation of buffers. Buffers allocated to the channel and not used by the current NCP 'read' are held for a later 'read' operation.

If a type 1 channel adapter is used for NCP mode, the type 1 channel adapter must be the only channel adapter. If a type 1 channel adapter is installed for emulation mode only in a partitioned emulation program (PEP), one additional type 2 or type 3 channel adapter may be installed for NCP mode.

Type 4 Channel Adapter

The type 4 channel adapter support requires an interrupt at level 3 at the end of each NCP buffer or end of PIU. As the number of INBFRS is depleted, the level 3 code branches into the level 4 supervisor routine to obtain a new supply of buffers equal to the INBFRS number. Then the 'read' (host 'write') operation continues to the completion of the host 'write' or another allocation of buffers. Buffers allocated to the channel and not used by the current NCP 'read' are held for a later 'read' operation.

One to four type 4 channel adapters may operate concurrently in NCP mode (two maximum on a 3705 I). Two of the channel adapters may also be used concurrently for emulation mode in a partitioned emulation program (PEP). If a type 4 channel adapter is installed for emulation mode only in a partitioned emulation program (PEP), one additional type 2 or type 3 channel adapter may be installed for NCP mode.

Type 2 or Type 3 Channel Adapter

The type 2 or type 3 channel adapter uses cycle-steal for multiple NCP buffers. The facility requires IN or OUT control words (CW) which are similar to CCWs in the host.

Host Writes

When the first 'write start zero' is received, IOS leases buffers, builds 'IN' control words (CWs) and sets up the channel adapter to accept data from the channel. The 'IN' control words are executed one at a time, causing the channel adapter to cycle-steal the PIUs into the buffers. During the execution of the set of control words, no program intervention is required.

The next level 3 interrupt into IOS is from one of three conditions; a channel stop, zero count override, or an unexpected 'write start' command.

The channel stop condition occurs when the channel adapter receives 'command out' to a 'service in' request. The channel stop condition signals the end of a PIU and causes a level 3 interrupt into IOS. IOS increments the enqueue count, passes the PIU to 'path control', and sets up the channel adapter to continue receiving data.

A zero count override condition exists when all the control words (CWs) on the channel-in chain (CIC) have been executed and the host still has more data to transfer. At the completion of the last control word in the CIC, the channel adapter causes a level 3 interrupt into IOS. Since the data transfer has not completed for this PIU, IOS must rebuild the CWs in the CIC. IOS leases new buffers, chains them to the previous buffers and rebuilds the CWs. When the CWs are rebuilt, IOS sets up the channel adapter to continue transferring data, using the address of the first CW in the new CIC. This sequence occurs each time a zero count override is reached.

Receiving an unexpected control command is common to all adapter types and was covered earlier.

Host Reads

When the 'read start' is received, it is compared against the expected 'read start' control command. If the 'read start' is as expected, IOS flips the expected RS command and purges any PIUs in the hold queue (the previous PIUs to the host). IOS builds the 'OUT' control words (CWs) necessary to send each PIU to the host. After building the 'OUT' control words for a PIU, including any required buffer pad (HOST BFRPAD=value), the CWs on the channel-out chain (COC) are executed and the PIUs are sent to the host. When the COC chain has been executed, the PIUs which have been sent are held on the CAB hold queue until acknowledged by the next expected read start from the host.

When all of the CWs on the channel-out chain (COC) have been executed, the channel adapter generates a level 3 interrupt into IOS. IOS presents ending status to the channel adapter for the host. If more PIUs are available for the host on the intermediate queue, IOS adds 'attention' to the status for the host. This 'attention' status indicates that a new start I/O is needed from the host.

Control Blocks of Channel IOS

There are five control blocks used for channel IOS; (1) channel parameter table (CPT), (2) channel control block (CAB), (3) channel control block (extension) (CAB extension), (4) channel word (CW), and (5) channel word (CWX).

Channel Parameter Table (CPT)

A channel parameter table (CPT) is created for each subarea coded in a HOST macro SUBAREA operand. If the HOST macro is coded SUBAREA=(3,4), two CPTs are created for this one HOST macro.

Figure 4.1 illustrates a channel parameter table (CPT) and the HOST macro operands which provide the field values.

	CPT			
0	SUBAREA		MAXBFRU	
4	UNITSZ		DELAY	
8	TIMEOUT		INBFR	BFRPAD

Figure 4.1. Channel Parameter Table (CPT)

Channel Control Block (CAB)

A channel control block (CAB) is generated for each channel adapter defined by the BUILD macro CA operand. The pointer to the last CAB is in the HWE at offset X'18'. The pointer to each previously defined CAB is in the CAB at offset X'7C'.

Figure 4.2 illustrates some of the key fields of the CAB.

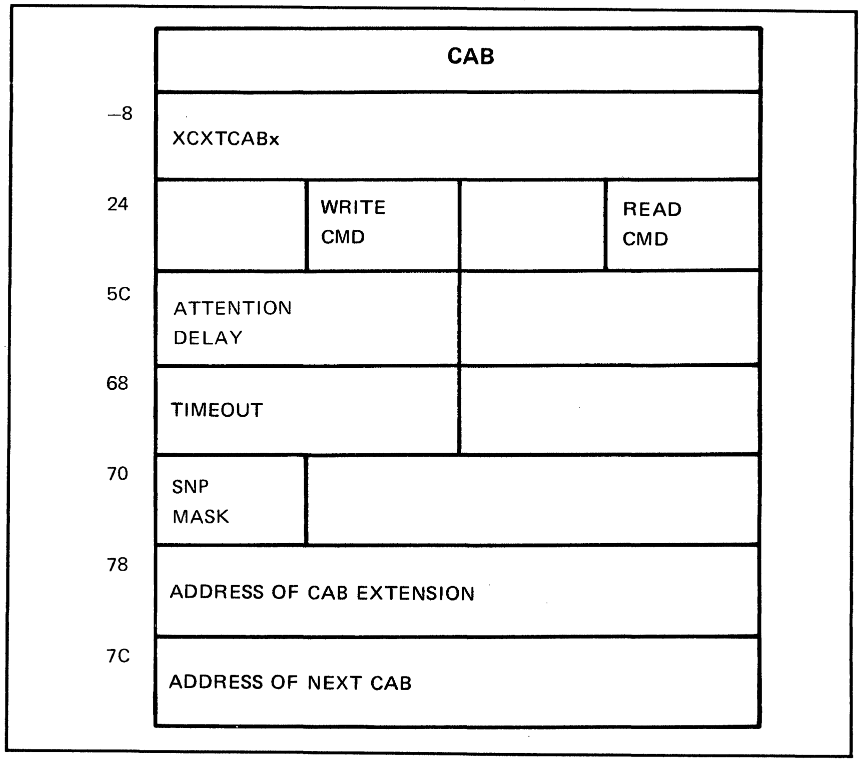


Figure 4.2. Channel Control Block (CAB)

The dump identifier at minus 8 has a value of XCXTCABx where x is 1, 2, 3, or 4 to identify the first, second, third, and fourth CAB.

Some of the fields from the channel parameter table (CPT) which initialize the CAB are:

```
X'54'  STATMOD
X'5C'  DELAY
X'68'  TIMEOUT
```

Other CPT operand locations are provided in the CAB extension.

Channel Control Block (CAB) Extension

A channel control block (CAB) extension is generated for each CAB. The pointer to the CAB extension is in the CAB at offset X'78'.

Figure 4.3 illustrates some of the key fields of the CAB extension.

Offset	CAB EXTENSION	
0	INTERMEDIATE QUEUE	
8	HOLD QUEUE	
20	PIU'S PASSED TO PATH CONTROL	PIU'S TO SKIP
24	PIU'S SKIPPED	INBFRS
60	UNITSZ	
64	MAXBFRU	
70	BFRPAD	
7C	ADDRESS OF CAB	

Figure 4.3. Channel Control Block (CAB) Extension

The CAB extension provides PIU pointers of the following:

- X'00' Intermediate queue, PIUs not sent to the host.
- X'08' Hold queue, PIUs sent to the host but not acknowledged.
- X'18' Address of last PIU passed to path control.
- X'44' Address of last PIU from path control; the PIU address is held until queued on the intermediate queue.

Additional PIU address pointers indicate current PIU and data address being sent or received.

Some of the fields from the channel parameter table (CPT) which initialize the CAB extension are:

```
X'26'  INBFRS
X'60'  UNITSZ
X'64'  MAXBFRU
```

Channel Words (CW and CWX)

Channel words are used in the NCP for type 2 and type 3 channel adapters. Channel words are coded in a four field format. The CW form is used for a 3705 of 256K or less storage. The CWX form is used for a 3705 of more than 256K storage. The following references to CW apply to CWX. The four fields in sequence specify:

1. Type (IN, OUT, or OUT STOP)

2. Chaining or no chaining.

3. Quantity of data to be read or written

4. Address of data area

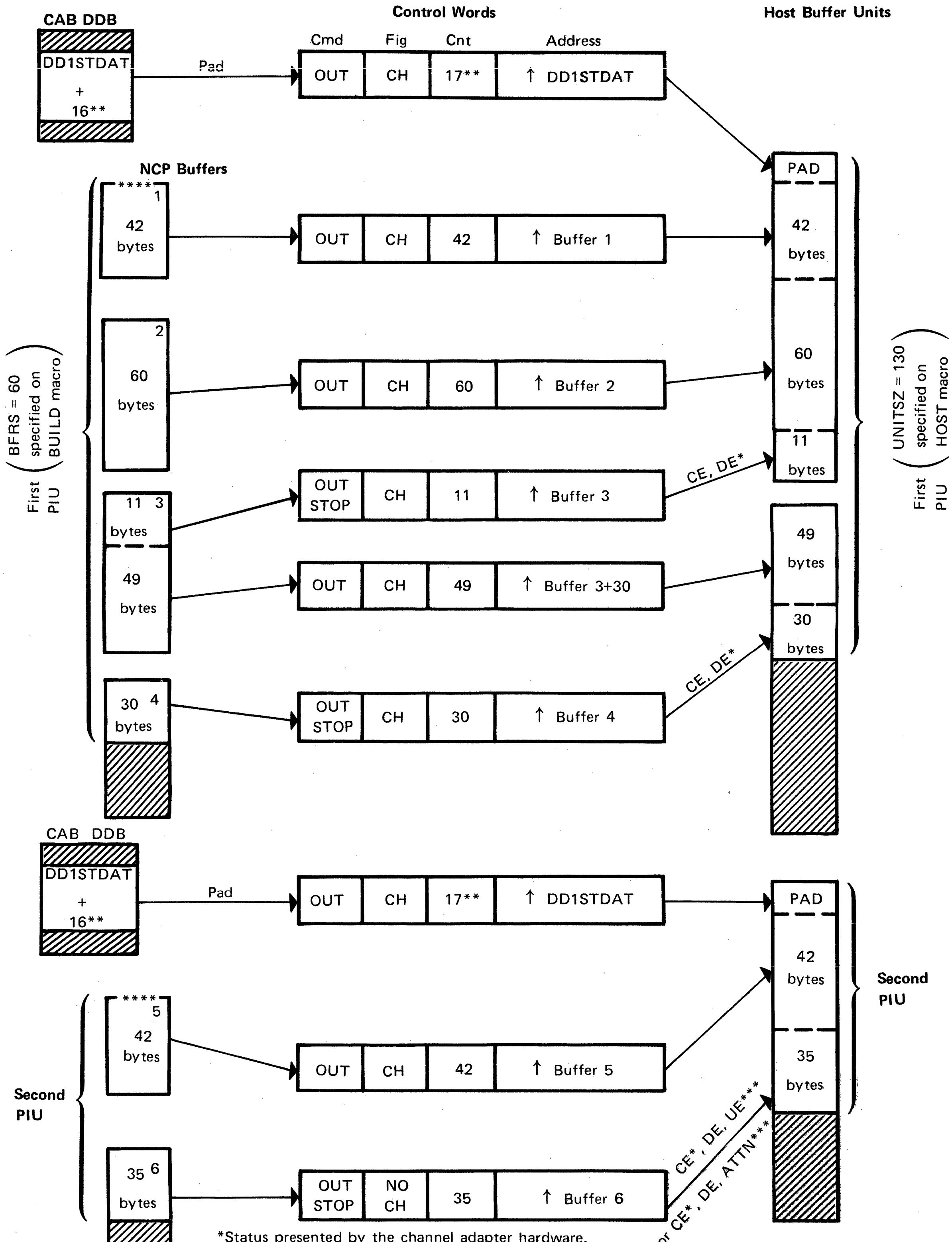
When IN CWs are built, all of the CWs are coded IN with the chaining bit 'on' in all CWs except the last. When the last CW completes without chaining, (1) a level 3 interrupt occurs to lease more buffers, (2) the last CW is chained to the new buffer chain, and (3) the CWs are rebuilt to point to the new buffers. The first CW points X'1A' offset into the buffer to reserve space for the event control block (ECB) and buffer prefix. All subsequent buffers are generated with an offset address of X'08' into the buffer to bypass the buffer prefix. The CW length field specifies the true buffer size less X'08'.

When a PIU is received, the host forces 'channel end', 'device end', *without* 'unit exception' by using a CCW with command chaining. This channel status stops channel transfer and generates a level 3 interrupt. The PIU is passed to 'path control'. The next available CW is modified for an offset address of X'1A' into the buffer, the count is modified to the remaining buffer length, and the channel is restarted. All of the delay is transparent to the host.

OUT channel command words are used with chaining until the last CW of a PIU is transmitted or until a host CCW is filled. The next data byte sent to the last-plus-1 position of a host CCW causes the channel to halt transfer on a 'zero count override' to access the next CCW. This channel halt is avoided by forcing the next CCW access without letting the zero count be recognized. If the end of a PIU is reached, the next PIU must start, with pads, in a new host buffer. Both of these conditions use the OUT STOP command to send channel end, device end, *without* unit exception. Chaining is used for both OUT and OUT STOP until the last OUT STOP CW.

The first CW of each PIU sends pad characters. The second CW addresses the PIU at an offset of X'1A', following the event control block (ECB).

Figure 4.4 illustrates the NCP-to-host transfer using OUT and OUT STOP CWs. The NCP buffer size is 60 bytes (without the buffer prefix) and the host buffer size is 130 bytes. The first PIU is 192 bytes, the second PIU 77 bytes. In each PIU the pad is sent to the host for BFRPAD length (17 bytes in the example). The first NCP buffer of each PIU has a 18-byte ECB which is not sent to the host; only 42 bytes are transmitted from a first NCP buffer.



*Status presented by the channel adapter hardware.
**17 pads (specified on BFRPAD operand of HOST macro).
***Additional status is generated by the NCP
****18 byte ECB, of the PIU, is not transferred.

Figure 4.4. Data Transfer from NCP to Host Buffer Units

Channel Adapter IOS Performance

Type 1 Versus Type 4 Versus Type 2 or 3 Channel Adapter

There are many factors which affect channel performance, one factor being the type of channel adapter.

The type 1 channel adapter must be installed on a byte multiplexer channel. Compared to the other channel adapters, the type 1 CA requires many additional communications controller cycles to execute the level 3 code after every four bytes of transfer. More commands are also processed in the channel, tying up the channel for greater periods of time than is the case with the type 2, 3 or 4 channel adapter. If the controller is not heavily loaded, machine cycles are available for servicing the type 1 adapter, rather than having the controller in the wait state.

The type 4 channel adapter may be on a byte multiplexer, block multiplexer or selector channel. The rate of data transfer is less for the type 4 channel adapter than for other devices, such as direct access devices. If the channel adapter is on a block multiplexer or selector channel it may seriously degrade other devices of greater transfer rate on the channel.

The type 4 channel adapter cycle-steals data for the length of an NCP buffer or end of data, which ever comes first. The level 3 code is not executed until the next NCP buffer or next PIU. The channel commands are executed once per buffer rather than once per four bytes as with the type 1 channel adapter.

The type 2 and type 3 channel adapter may be installed on a byte multiplexer, block multiplexer, or selector channel. The type 2 and 3 channel adapter cycle-steals data for multiple NCP buffers and multiple PIUs. The level 3 code and channel commands are executed once for the entire transfer to the host; once for the entire transfer from the host if the INBFRS quantity is not depleted.

Host and NCP Buffer Sizes

Host and NCP buffer sizes should not need be identical. For one thing, the NCP buffer has an 18-byte event control block (ECB) for control fields in the first buffer of a PIU, and this size does not match the host pad requirements.

The size of buffers should be related to the average size of the PIU in order to avoid unused space in large buffers for small PIUs and avoid excessive buffer chaining and unchaining of small buffers for large PIUs. Remember that CICS, IMS, TCAM, control commands, and probably many user applications have a response TH/RH of 13 bytes, and even control commands are short. The minimum NCP buffer size of 60 (BUILD BFRS operand) should be sufficient for responses. The maximum size of 240 for NCP buffers or the default of 60 may be excessive if data requests are short. The host size should be determined as the same size as NCP, plus the difference between the NCP control requirement (ECB 18 bytes) and the host buffer pad requirements as specified on the BFRPAD operand.

Using the default NCP buffer size of 60, a request PIU of 120 bytes requires three NCP buffers; the first NCP buffer contains the PIU event control block of 18 bytes, and contains 42 PIU bytes. In this example an NCP buffer size of 72 (fullword multiple) requires two NCP buffers for the request.

In addition to the size of PIUs as a factor in NCP buffer size, there is a critical factor of SDLC terminal buffer size specification. An operand of MAXDATA on the PU macro specifies the maximum PIU which can be sent to the device. There is an absolute requirement that the NCP buffer size must be at least TH less for segmenation than the smallest MAXDATA value. This NCP buffer size should never be a problem unless the MAXDATA is coded in error. Selecting NCP buffer sizes for segments is provided in a later topic of boundary network node.

Type 1 physical units have a 261-byte physical buffer (five-byte FID3 TH/RH plus 256 bytes of text), and type 2 physical units have a 265-byte physical buffer for receiving PIUs (nine-byte FID2 TH/RH plus 256 bytes of text). The largest NCP buffer size is 240.

If PIUs are larger than the MAXDATA operand, PIUs are segmented. A segment is a TH-plus-1 or a multiple of full NCP buffers. Segmenting affects the NCP buffer size. Segmenting is covered later under the topic boundary network node.

Note: VS1 VTAM requires the host buffer size to be an odd multiple of words. The HOST macro UNITSZ operand should not be divisible by 8 for VS1 VTAM.

Host Buffer Allocation

The NCP defines the number of host buffers on the HOST macro operand of MAXBFRU. The number multiplied by the host buffer size minus buffer pads ((MAXBFRU x UNITSZ) - BFRPAD) restricts the size of the largest PIU which can be sent to the host. The true maximum PIU is restricted to the number of NCP buffers defined by the BUILD macro operand of TRANSFR. There is no restriction in NCP on the size of a PIU from the host to NCP. If channel IOS determines that a PIU exceeds the length of the host capacity, IOS converts PIU requests to an error response and returns the PIU to the origin.

Another consideration is the number of PIUs sent to the host by a single host read. If DELAY is coded as nonzero on the HOST macro, a timer is set when the first PIU is enqueued to the intermediate queue. 'Attention' is not sent until the timer expires, allowing additional PIUs to be enqueued, or until the number of PIUs fills the number of host buffers, whichever condition occurs first. If the host completes a write to the NCP and STATMOD=YES is coded on the HOST macro, any PIUs in the intermediate queue are sent before the timer event.

If PIUs arrive on the average of one a second a delay of .1 seconds delays data, but does not result in multiple PIU tranfers. This may result in delaying PIUs when traffic is light resulting in even response times when traffic gets heavier.

If PIUs arrive on the average of one each .1 second a delay of .1 seconds results in multiple PIUs per transfer, either from STATMOD host writes or multiple PIUs per attention. If PIUs arrive at a rate where there is always more traffic than the host can accept the delay is ignored.

The delay technique has two benefits: (1) improvement in the host performance by reducing the number of 'attentions' and host buffer allocations; (2) improvement in NCP performance by reducing the number of channel initializations and termination processing of the intermediate queue to hold queue. When traffic is light, the PIU is delayed at the channel. When traffic is heavy, the delay is not used because the amount of data queued fills the host buffer allocation.

NCP Buffer Allocation

The INBFRS operand defines the number of buffers to be allocated for host-to-NCP transfers. When the last allocated buffer is filled, the NCP obtains more buffers as required. If a large number of NCP buffers is allocated to the channel and not used promptly, it deprives other users of free buffers and may result in slowdown. If few NCP buffers are allocated, the NCP must lease buffers more frequently, taking required controller cycles.

Delay

See Host Buffer Allocation.

Status Modifier

The STATMOD=YES operand of the HOST macro allows the NCP to send PIUs to the host at the completion of a host 'write'. When a host 'write' completes, rather than send the 'attention' separately or as a part of the write status and waiting for a host 'read', the PIUs can be sent as a continuation of the host 'write CCW' chain. If the NCP has traffic for the host, the status modifier causes the host 'write CCWs' to chain to 'read CCWs'.

Channel Timeout

If the HOST macro operand is coded TIMEOUT=NONE, the NCP sends 'attention' and waits indefinitely for the host to reply. If auto network shutdown support is included (BUILD ANS=YES), the operator can initiate auto network shutdown from the panel of the communications controller. Warning: Auto network shutdown from the panel initiates auto network shutdown for all owners.

If the host does not reply to the 'attention', TIMEOUT=value provides automatic entry to auto network shutdown for the owner of the channel adapter. For all resources owned by the SSCP on this channel adapter, all current pending line operations complete, resources are deactivated, and 'lost subarea' PIUs are sent to all active adjacent subareas and remaining owners.

Channel Adapter Summary

There are four types of channel adapters. Type 1 requires heavy program support. Type 1 or type 4 are required for emulation programming. Channel adapter types cannot be mixed in NCP mode except types 2 and 3. Type 2 is used for single processors; type 3 allows a dual interface to tightly coupled multiprocessors. Types 2, 3, and 4 are high-performance, cycle-steal channel adapters. Type 4 cycle-steals for the length of an NCP buffer or end of data. Types 2 and 3 cycle-steals for the length of multiple NCP buffers and PIUs. User definition of host and NCP buffer parameters and other channel-related operands on the HOST macro can have significant effect on performance.

Chapter 5:

Intermediate Network Node Path Control

Note:

Boundary Network Node (BNN) path control functions include 'path control out delayed', 'path control in immediate' and 'path control in delayed'. Each of these BNN path control functions are covered in the boundary network node section.

Introduction

INN path control directs the flow of path information units (PIUs) to the proper destination. INN path control uses the destination address field (DAF) from the PIU to access entries in the subarea index table (SIT), subarea vector table (SVT), and the resource vector table (RVT). The INN path control routine locates the appropriate path for the PIU and places the PIU on a destination queue. The valid destinations are (1) a channel adapter (CAB), (2) NCP physical services (PSB), (3) boundary network node (CUB or LUB), (4) local/local or local/remote link (SCB), or (5) the BSC/SS processor. The INN path control module operates in program level 3. The entry is via a branch from the channel IOS or BNN 'path control in immediate') or SVC (XPORT macro) from level 5 routines.

A PIU destined for a subarea over a channel adapter is queued on the channel control block (CAB) channel intermediate queue.

A PIU destined for a type 4 physical unit is queued on the station control block (SCB) link outbound queue (LOB). From the link outbound queue the PIU is transmitted to the local or remote by the link scheduler.

When the PIU destination is a type 1 or type 2 physical unit, the PIU is enqueued on the common unit physical block (CUB) or logical unit block (LUB), depending upon the PIU destination.

A PIU that is destined for NCP physical services is placed on the NCP physical services block (PSB) process queue.

If the PIU is destined for a BSC or SS device, the PIU is passed to the BSC/SS router, which converts the PIU to a BTU and enqueues the BTU on a device block (DVB).

Control Blocks of INN Path Control

INN path control routes PIUs to their proper destination. To accomplish this routing, INN path control uses several tables (created during NCP generation) in conjunction with the DAF portion of the PIU. These tables are the subarea index table (SIT), subarea vector table (SVT), and resource vector table (RVT).

Subarea Index Table (SIT)

The system pointer to the subarea index table (SIT) is in the extended half-word direct addressables (HWE) at offset X'48'.

The subarea index table (SIT) consists of one-byte entries that correspond to the network subarea addresses. NCP generation builds the SIT quantity of entries selected by the MAXSUBA operand of the the BUILD macro. If MAXSUBA is coded 15, there are MAXSUBA entries-plus-1, or 16 entries.

The first one byte entry contains X'00'. The remaining entries are filled according to the definitions of the network control program.

Each entry in the SIT represents one of the possible subarea values. The first SIT entry represents subarea 0, the second entry represents subarea 1, etc. The SIT one byte entries are initialized with X'00' for subareas which are not defined to the NCP. Defined subarea entries contain a nonzero value. Nonzero SIT entries contain an offset to an entry in the subarea vector table (SVT).

SIT entries are defined by (1) BUILD macro SUBAREA operand, (2) HOST macro SUBAREA operand, (3) PU macro TYPE=(4,LOCAL) or TYPE=(4,REMOTE) SUBAREA operands, and (4) PATH macro DESTSUB operand. The subareas identified in the PATH macro operand of ADJSUB does not define SIT entries; ADJSUB identifies the SIT offset to be used for all subareas coded in the DESTSUB operand.

If a channel attached host is defined as subarea 1, the subarea value of 1 provides the offset into the second entry in the SIT. The value of the second byte of the SIT table, multiplied by 8, provides the offset into the SVT. The SVT entry would provide the address of the channel control block (CAB).

If an NCP over a local/local link is defined as subarea 5, the sixth SIT table entry (offset value of 5) provides the one-byte offset multiplied by 8 to locate the station control block (SCB) entry in the SVT.

If this NCP is defined as subarea 3, the fourth SIT table entry (offset of 3) provides the one-byte offset into the SVT. The SVT entry would provide the address of the resource vector table minus four (RVT-4) which contains the network addresses of all resources defined to this NCP.

If the HOST macro is coded SUBAREA=1, the BUILD macro is coded SUBAREA=2, and two type 4 PU macros are coded SUBAREA=4 and SUBAREA=6, the second, third, fifth, and seventh SIT entries provide offsets to the SVT table.

Figure 5.1 illustrates an SIT of MAXSUBA=7 and subareas 1 through 5 defined to this NCP.

	SIT
0	00
1	02
2	03
3	01
4	04
5	05
6	00
7	00

Figure 5.1. Subarea Index Table (SIT)

Subarea Vector Table (SVT)

The system pointer to the subarea vector table (SVT) is in the extended halfword direct addressables (HWE) at offset X'4C'.

The subarea vector table is made up of eight-byte entries. The first two bytes of each entry consists of two type fields, which describes the type of subarea this entry represents. The third and fourth bytes are reserved. The last four bytes contain an address or zeroes. If the entry represents the resource vector table (RVT) the address field contains the address of the RVT-4. If the entry represents an adjacent or nonadjacent subarea over a local/local or local/remote link the field contains the address of the station control block (SCB) representing the adjacent subarea. An address entry of a channel control block (CAB) contains zeros until initialized by an 'Activate physical' command, and then contains the address of the CAB used for the 'Activate physical' command.

The first SVT entry contains a value of zero, and all SIT entries with undefined SUBAREAs index to this entry. Any request PIU with a subarea destination which results in reaching this entry is returned to the origin. Any response PIU is discarded.

The second entry is the address of the resource vector table (RVT). The BUILD macro SUBAREA operand value is used as an offset into the SIT to initialize that SIT entry with the value of X'01'. The SIT value of X'01' provides the offset to the second SVT entry, the resource vector table (RVT) address.

The third entry and subsequent address pointers vary depending upon the definition (or omission) of HOST macros and type 4 PU macros.

An SVT entry is created for each HOST macro. The address field of the SVT entry for a CAB contains zeros. The SIT is initialized for all subareas defined in the HOST SUBAREA=(n,...) to point to this SVT entry. If six HOST macros are defined, six SVT entries are created. When NCP receives an 'Activate physical' command addressed to NCP physical services, the NCP records the address of the CAB in the PIU ECB at offset X'10'. When the command is processed the CAB address from the ECB is placed in the SVT entry identified for that host subarea in the SIT. A 'Deactivate physical' command clears the SVT field.

An SVT entry is created for each type 4 PU macro defined with an operand of SUBAREA. A backup link for a type 4 physical unit is coded with PU operands of PUTYPE, IRETRY, and RETRIES. The type 4 PU macro entries (other than backup) are generated after any HOST macro entries. Each entry contains an address of the station control block (SCB) generated by the PU macro.

An SVT entry with the leftmost bytes value of X'FF' is created as a delimiter.

Figure 5.2 illustrates an SVT generated for an NCP which had two HOST macros and two type 4 PU macros.

	SVT		
0	00 00	00 00	00 00 00 00
8	80 A0	00 00	RVT – 4
	48 E0	00 00	CAB or 00 00
	48 E0	00 00	CAB or 00 00
	41 E0	00 00	SCB
	41 E0	00 00	SCB
	FF 00	00 00	00 00 00 00

Figure 5.2. Subarea Vector Table (SVT)

Resource Vector Table (RVT)

The system pointer to the resource vector table (RVT) is in the word direct addressables (XDA) at X'07E8' which points at the RVT-2.

The resource vector table (RVT) consists of a header and eight-byte entries. The header contains two fields. At RVT-4 is a two-byte field which contains the count of the total entries in the table. The total count multiplied by eight provides an offset into the RVT to the last entry. RVT-2 is a two-byte field which contains the count of BSC/SS devices. The BSC/SS count multiplied by eight provides an offset into the RVT to the last BSC/SS entry.

The first entry in the RVT has a type field of X'00' and the address of physical services control block (PSB). The PSB represents NCP physical services.

The second and subsequent entries may define NCPNAU macro generated control blocks for user programmed resources.

If BSC/SS devices are defined, the BSC/SS resource addresses would follow the previous entries. Following the last BSC/SS resource entry is an entry whose leftmost byte is X'FF'.

SDLC device addresses follow the BSC/SS delimiter entry, or if no BSC/SS devices are included, the first SDLC entry follows the PSB entry (or NCPNAU entries). The end of the RVT is identified by a X'FF' delimiter entry.

Each entry consists of two one-byte type fields, a two byte reserved field, and four-byte address of the control block represented by this entry.

Figure 5.3 illustrates a resource vector table (RVT) which includes NCPNAU, BSC/SS, SDLC, and user programmed resources.

	RVT			
		−4	TOTAL COUNT	BSC/SS COUNT
0	00 20	00 00	PSB	
	08 38	00 00	NLB (NCPNAU)	
	84 00	00 00	LCB (switched)	
	5C 00	00 00	DVB	
	80 00	00 00	LCB	
	5C 00	00 00	DVB	
	FF 00	00 00	00 00	00 00
	80 20	00 00	LKB	
	40 60	00 00	PU (type 4)	
	80 20	00 00	LKB	
	60 20	00 00	PU	
	08 20	00 00	LU	
	80 30	00 00	VLB	
	60 30	00 00	NPB	
	08 30	00 00	NLB	
	60 00	00 00	PUDR	
	08 00	00 00	LUDR	
	08 00	00 00	Invalid LU	
	FF 00	00 00	00 00	00 00

Figure 5.3. Resource Vector Table (RVT)

The NCP generation builds the RVT, with an entry for NCP physical services; user programming NCPNAU macros; BSC/SS definitions of LINE, CLUSTER, TERMINAL, and COMP macros; and SDLC definitions of LINE, PU, and LU macros; user programming (VIRTUAL=YES) of LINE, PU, and LU macros. If dynamic reconfiguration or switched SDLC links are defined, the last entries are addresses of physical units and logical units from the dynamic reconfiguration pools. These addresses are generated by the PUDRPOOL and LUDRPOOL macros. The LUPOOL macro may be required to support prior host releases.

INN Path Control Flow

INN path control receives control from (1) channel adapter IOS, (2) BNN path control in immediate from a type 4 PU, or (3) an SVC (XPORT) macro from NCP physical services, BNN 'connection point manager in' from a PU or LU, or BSC/SS processor.

The DAF of the FID0 or FID1 is used by path control to route the PIU properly. The first byte of the DAF contains the subarea address. The byte is shifted as required to delete any leftmost bits of element address, leaving the true subarea value. This subarea address is used to vector into the SIT to the entry for that subarea. The one-byte SIT entry contains an index value to be used with the SVT. This value, multiplied by 8, is used by path control to index into the SVT to the corresponding entry. The SVT entry contains flags describing the entry and a pointer to the control block representing that subarea.

The possible subarea entries in the SVT and their associated pointers are as follows:

- Invalid subarea (entry of zeros)
- Address of the resource vector table (RVT)
- Address of a channel control block (CAB)
- Address of a station control block (SCB)
- Address of a programmed resource control block (VLB, NPB, NLB)
- X'FF' delimiter entry

The action taken by INN path control differs for the various subareas.

A PIU with a destination of a channel control block (CAB) is enqueued on the CAB intermediate queue.

A PIU destined for a type 4 physical unit is enqueued on the station control block (SCB) link outbound queue.

A PIU destined for physical services is enqueued on the physical services control block (PSB).

PIUs for type 1 or type 2 physical units are queued on an appropriate CUB or LUB queue.

PIUs for user programmed resources are queued on an appropriate generated control block.

If the RVT entry is in the BSC/SS section of the RVT, the PIU is routed to the BSC/SS system router via a branch instruction.

INN Path Control Summary

All PIUs from all sources are processed by INN path control to locate the destination queue.

If the destination subarea is not defined the subarea index table (SIT) the entry contains a zero offset into the subarea vector table (SVT). The SVT zero offset entry for invalid subareas results in the request PIU being marked in error and being returned to the origin. Response PIUs are discarded.

If the destination subarea is for this NCP the subarea index table (SIT) entry provides an offset into the subarea vector table (SVT). This SVT entry contains the address of the resource vector table (RVT). After the RVT is located the element portion of the destination address field (DAF) is used as an index to locate the address of the destination resource.

If the destination subarea is for a channel connected destination the SIT entry provides an offset to an SVT entry with an address of a channel control block (CAB). The PIU is queued on the CAB intermediate queue.

If the destination subarea is for a link connected destination the SIT entry provides an offset to an SVT entry with an address of a station control block (SCB). The PIU is queued on the SCB link outbound queue.

Chapter 6:

Network Control Program Physical Services

Purpose of NCP Physical Services

The NCP physical services component is a collection of routines necessary for the control and/or modification of the communications network. NCP physical services are divided into two functional areas: (1) system control (SC) and (2) function management (FM). The required services are selected via request codes in the PIU.

Session Hierarchy

The requirement for physical services is based upon the session control of the network and the need to change network status. Before data can be transferred through the communication network, a physical and logical connection must be established between the origin and destination of the data request. The logical connection is referred to as a session.

The commands which request a session contain the rules to be used during the session. The session rules are defined in sets under the categories of ' function management (FM) profiles' and 'transmission subsystem (TS) profiles'. The profiles define session rules, session commands required, and session commands not supported. A positive response to the session command indicates agreement to use the profile rules.

There are various types of sessions in ACF/NCP. The session between cross domain resource managers (CDRMs) is transparent to the NCP. The BSC/SS session FID0 commands are used only by the BSC/SS processor. There are four types of SNA sessions involving the NCP which are controlled by network FID1 commands.

Figure 6.1 illustrates the four FID1 NCP session types.

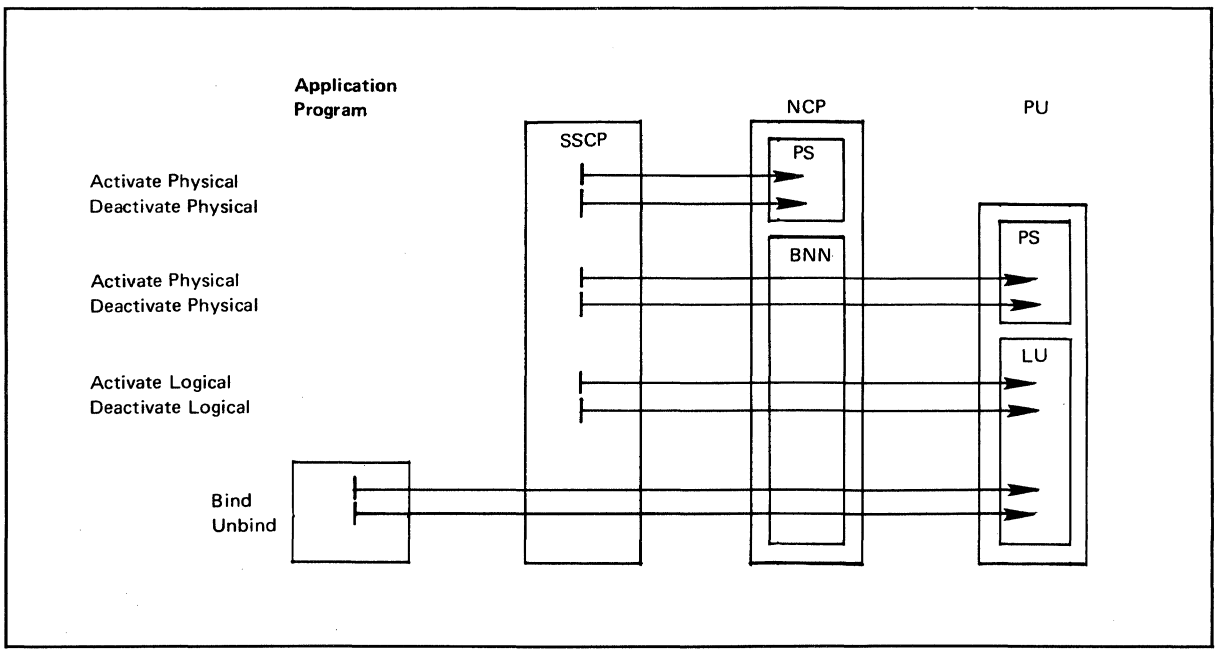


Figure 6.1. Session Hierarchy

SSCP and NCP Physical Services

This session is initiated with an 'activate physical' command to NCP physical services from SSCP and is ended with a 'deactivate physical' command. The function management (FM) profile is in RU byte 2 bits 0 to 3 of the 'activate physical' command. The transmission subsystem (TS) profile is in RU byte 2 bits 4 to 7. The TS profile requires a 'start data traffic' command to enable data flow in the SSCP/NCP session. Data sent to physical services consists of requests to change the network status.

Before the other sessions can be initiated, the links must be 'activated' and physical units 'contacted'. The 'activate link' and 'contact' commands from an SSCP to NCP physical services identify the command and the network address of the target resource in the PIU RU.

An 'activate link' function management (FM) request is required to activate a link. The 'activate link' request to NCP physical services causes the link scheduler to be initiated for this link. Bit 1 of LKBSTAT (X'1A') in the link control block (LKB) is set to 1 to indicate that an 'activate link' is in progress. The LKBSTAT bit 0 is set to 1 to indicate an active link. For nonswitched links only the modem is enabled. Switched links require a 'connect in' (answer) or 'connect out' (dial) command, and other switched commands which are covered later under the topic Boundary Network Node Switched Support.

An 'activate link' command is accepted only if it is from an owner of the NCP. The origin address field (OAF) of the command is checked against the network addresses in the SSCP-NCP session control blocks (SNPs). A positive response to an 'activate link' command provides ownership of a link to the SSCP issuing the command.

Ownership of an SDLC link is recorded in the link control block (LKB) at offset X'20' with the SNP owner mask. Ownership of a BSC or SS line is recorded in the line control block (LCB) at offset X'41'. Nonswitched SDLC links have shared ownership. Switched SDLC links and BSC and SS lines are owned serially.

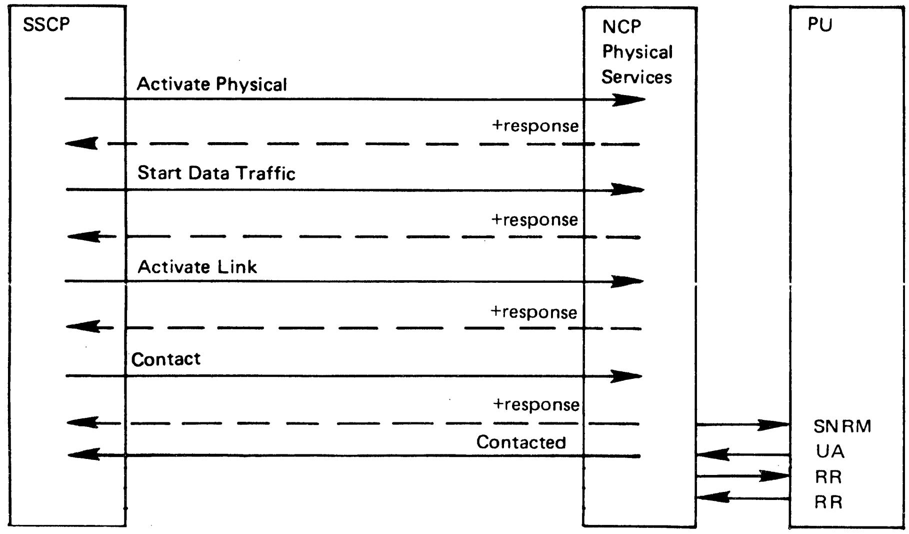


Figure 6.2. SSCP to NCP Physical Services Command Sequence

A 'contact' command is required to obtain ownership of an SDLC physical unit. Figure 6.2 illustrates the request sequence of a contact command.

A 'contact' command is accepted only from an owner of the link. A positive response to a contact command provides ownership of the physical unit to the SSCP issuing the command. Ownership is recorded in the common physical unit block (CUB) at offset X'1C'.

The contact request is acknowledged by physical services with a response to SSCP. The contact request also schedules a 'set normal response mode' (SNRM) SDLC command to the physical unit by setting the SNRM bit in the CUB plus X'2F'.

An SNRM SDLC command to a physical unit results in a (1) timeout, (2) 'disconnect mode' (DM), (3) 'unnumbered information frame' (UA), or (4) 'request initialization mode' (RIM). A timeout response results in the SNRM being retried on a user-specified basis. The DM response is sent by the secondary station on a local/local link after an 'activate link' but prior to a 'contact' command. If an UA or RIM response is returned by the physical unit, a 'contacted' PIU is generated by NCP physical services and sent to SSCP.

An RIM response indicates the PU requires loading. After the host sends the load program in response to the RIM, the 'contact' is rescheduled to obtain an UA response.

An UA response indicates the PU is active and available for sessions. The common physical unit block (CUB) CUBSSCF (X'2E') bit 2 (not operational bit) is turned off to indicate that the device is available for sessions to be established. Figure 6.3 illustrates the SSCP-PU and SSCP-LU activation sequence.

SSCP and PU Physical Services

The SSCP/PU session is established with an 'activate physical' command addressed to the common physical unit block (CUB) defined by a PU macro. The session is ended by a 'deactivate physical' command. The function management (FM) profile is in RU byte 2 bits 0 to 3 of the 'activate physical' command. The transmission subsystem (TS) profile is in RU byte 2 bits 4 to 7.

The SSCP/PU session must exist before any sessions can be established with logical units. The 'activate link' to the link and 'contact' command to the device must complete successfully before this session can be established. Type 2 and type 4 physical units receive and respond to the 'activate physical' command. The NCP processes this command for type 1 physical units.

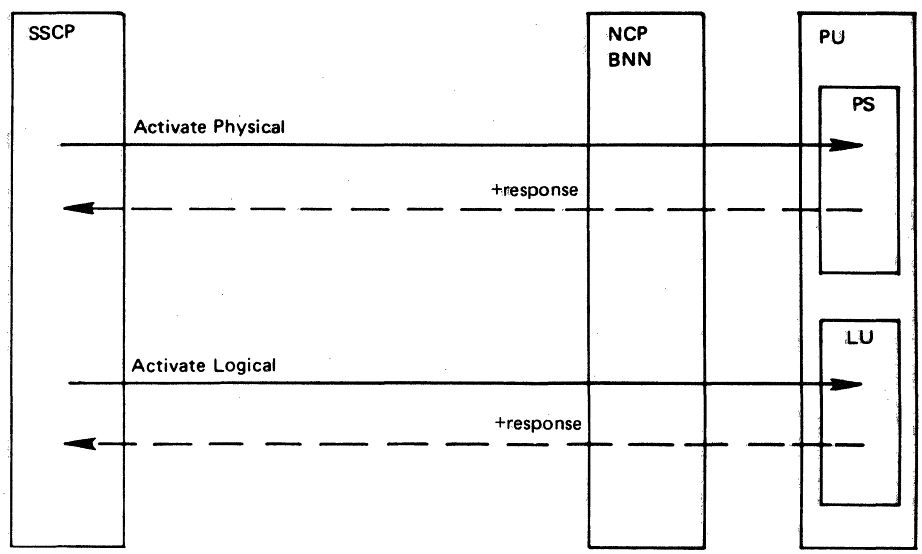


Figure 6.3. Activate Physical and Activate Logical Commands

SSCP and LU

The SSCP/LU session is initiated with an 'activate logical' command addressed to the logical unit block (LUB) defined by a LU macro. The session is ended by a deactivate logical command. The function management (FM) profile is in RU byte 2 bits 0 to 3 of the 'activate logical' command. The transmission subsystem (TS) profile is in RU byte 2 bits 4 to 7. This session must exist before a APPL/LU (host application/logical unit) session can be established. This command is processed by type 1, type 2, and type 4 physical units, except for the type 1 SDLC 3270. The NCP performs the processing and issues all responses for all commands addressed to the SDLC type 1 PU 3270.

Host Application and LU

The APPL/LU (host application/logical unit) session is initiated with a 'bind' command addressed to the LUB. The 'bind' command selects the 'function management (FM) profile' (RU byte 2) and 'tranmission services (TS) profile' (RU byte 3). If the TS profile is type 3, 4, or 5 a 'start data traffic' command is required before data flow can occur. The session is ended by an 'unbind'. Figure 6.4 illustrates the APPL/LU activation sequence for a TS profile type 3, 4, or 5.

The NCP performs the processing and issues responses for all session control commands addressed to the type 1 SDLC 3270 on the APPL/LU session.

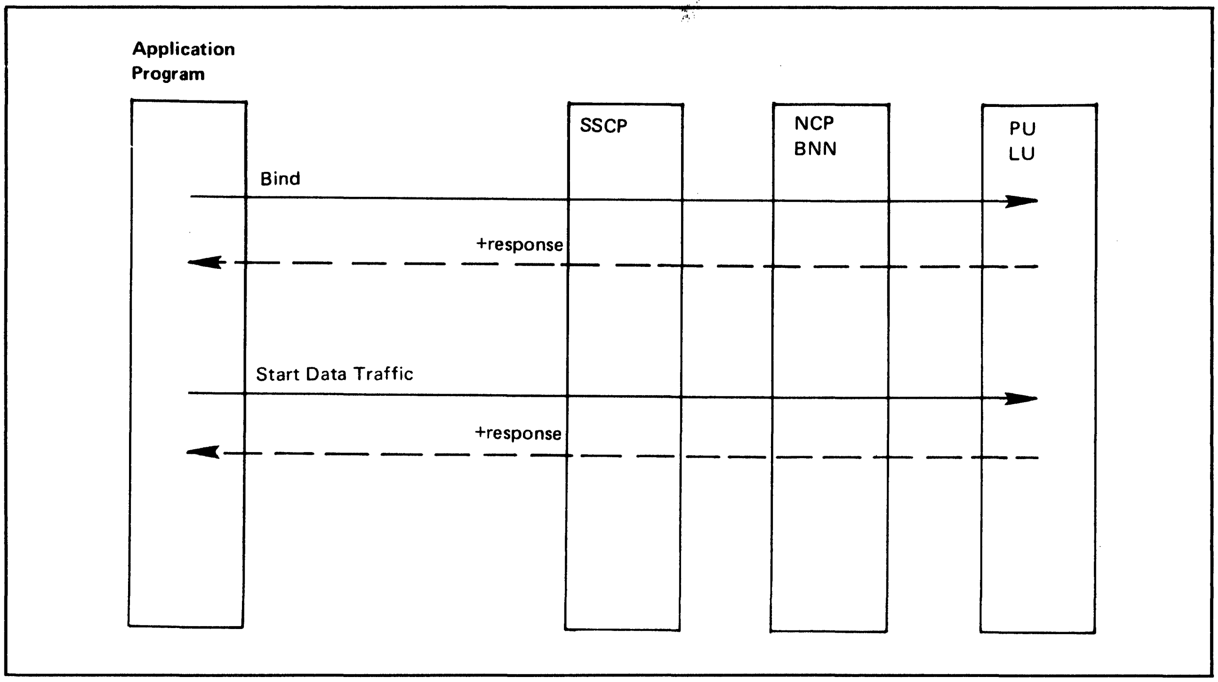


Figure 6.4. Bind and Start Data Traffic Commands

Control Blocks of NCP Physical Services

The NCP physical services component is represented by the physical services block (PSB). The NCP may have sessions with eight SSCPs as selected by the BUILD macro operand of MAXSSCP. When an 'activate physical' command from an SSCP to NCP physical services is received the SSCP information is recorded in a vector table of SSCP-NCP sessions (VTS) and an SSCP-NCP session control block (SNP).

An 'activate physical' command to NCP requests ownership of the NCP. Ownership is recorded as an SSCP-NCP session (SNP) bit. Each SSCP which activates the NCP is assigned one of eight bits. Ownership of NCP resources is recorded with the assigned bit 'on' in an SNP owner mask field.

Physical Services Block (PSB)

The system pointer to the physical services block (PSB) is in the extended halfword direct addressables (HWE) at offset X'44'.

The physical services block (PSB) contains the process queue control block for NCP physical services. The PSB also contains all of the control information common to all owners. Owner dependent information is provided in the vector table of SNPs (VTS) and SSCP-NCP session control block (SNP).

Figure 6.5 illustrates the fields of the PSB.

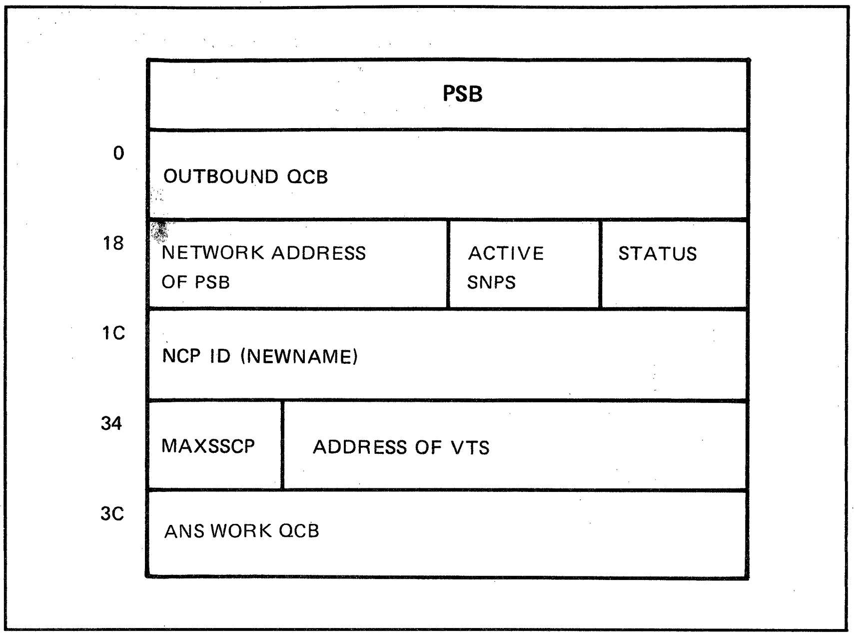


Figure 6.5. Physical Services Block (PSB)

The SNP owner mask (X‘1A’) has a bit ‘on’ for each current SSCP owner. The BUILD macro NEWNAME operand is located at X‘1C’. The total count of vector table of SNPs (VTS) entries (MAXSSCP) is recorded at X‘34’. The address of the vector table of SNPs (VTS) is in a fullword at X‘34’.

Vector Table of SNPs (VTS)

The system pointer to the vector table of SNPs (VTS) is in the physical services block (PSB) at offset X‘2C’.

The vector table of SNPs (VTS) consists of an eight-byte entry for each SSCP. The number of entries is selected by the MAXSSCP operand of the BUILD macro, with a minimum of 1 and maximum of 8.

Figure 6.6 illustrates a vector table of SNPs (VTS).

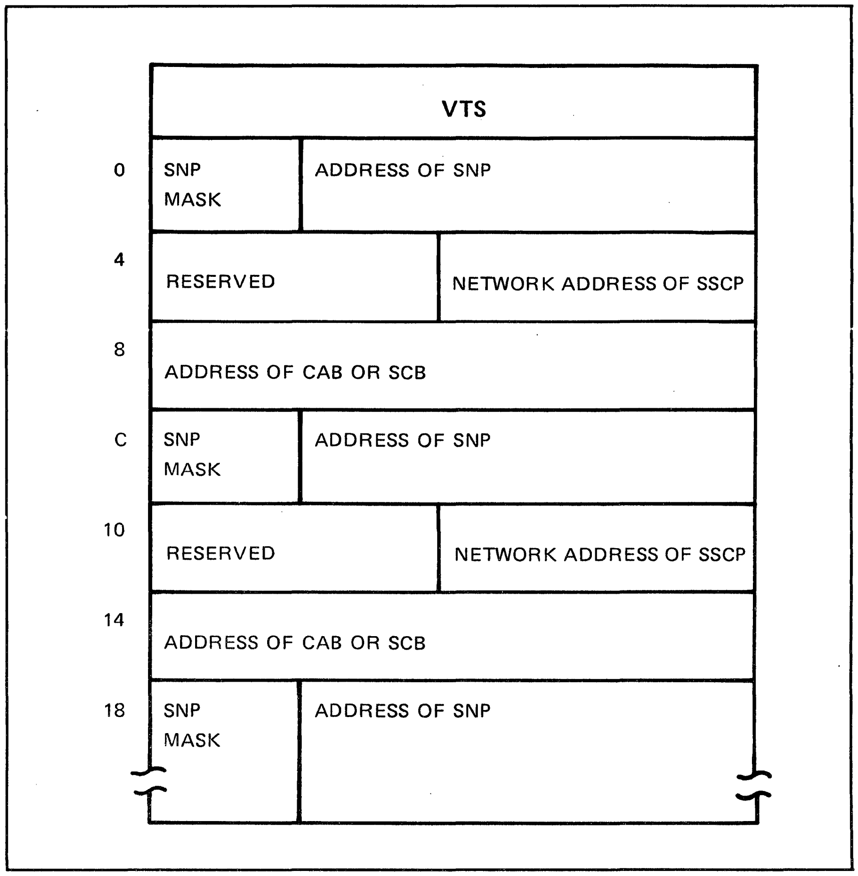


Figure 6.6. Vector Table of SNPs (VTS)

Each SSCP entry contains the SNP owner mask in byte 0. SNP owner masks are assigned by the NCP during processing of the 'activate physical' command. A mask value of X'00' indicates an available entry (not currently assigned).

The fullword at X'00' contains the address of the SSCP-NCP session control block (SNP). The SNP contains SSCP dependent information.

Each SSCP is reached via a channel adapter or link. If the SSCP activated the NCP via a channel adapter the channel adapter block (CAB) address is in the last fullword of a vector table of SNPs (VTS) entry. If the SSCP activated the NCP via a link the station control block (SCB) address is in the last fullword of a VTS entry.

SSCP-NCP Session Control Block (SNP)

The system pointer to an SSCP-NCP session control block (SNP) is in the vector table of SNPs (VTS).

Figure 6.7 illustrates the fields of a SNP.

Offset	SNP			
0	NETWORK ADDRESS OF SSCP		STATUS	
4	SNP MASK	RESTART STATUS	ANS CODE	
8	INBOUND SEQUENCE NUMBER		OUTBOUND SEQUENCE NUMBER	
C	VTS ENTRY ADDRESS			
10	ANS QCB			

Figure 6.7. SSCP-NCP Session Control Block (SNP)

The SSCP network address is stored in the first halfword. The remaining fields provide status bytes, the SNP mask, inbound and outbound sequence numbers for this SSCP-NCP session, an address pointer to the VTS, and ANS QCB.

Physical Services Control Block Initialization

The NCP physical services control blocks (PSB, VTS, and SNP) are initialized by an 'activate physical' command from an SSCP to NCP. The 'activate physical' may flow over any of the (1) four channel adapters or (2) type 4 PU links. Subareas reached via a link are assigned in the subarea vector table (SVT) at generation; the entry is reset by switched network backup command from the host. Subareas reached via a channel are dynamically assigned to a specific channel; before an 'activate physical' is passed from channel IOS, the address of the CAB is added to the ECB at offset X'10'.

A session is established by:

- Allocating a SNP for the session
- Setting the SNP mask in the SCB or CAB
- (channel source only) Placing the CAB address into the appropriate SVT entry
- (channel source only) Moving the appropriate CPT fields into the CAB
- Sending an initialization complete to the SSCP
- Sending the response to the 'activate physical'

Physical Services Components

The NCP physical services component interfaces with the 'system services control point' (SSCP) to provide control functions for the NCP. NCP physical services is made up of four sections: connection point manager out (CPM-OUT), connection point manager in (CPM-IN), system control (SC) router, and function management (FM) router. The system control (SC) router is common to NCP physical services and NCP boundary network node physical services.

Physical Services Connection Point Manager Out (CPM-OUT)

Physical Services CPM-OUT receives a PIU addressed to NCP physical services. The PIU is validated and, according to the contents of the request/response header (RH) byte 0, CPM-OUT calls either the system control (SC) router or the function management (FM) router.

Physical Services Connection Point Manager In (CPM-IN)

Physical services CPM-IN validates a PIU and XPORTs it to INN path control to be sent to a host SSCP. As an example, when link commands (connect out (dial), connect in (answer), contact,) complete, NCP CPM-IN is triggered to change status fields and build a PIU to inform the host.

System Control Router

The system control (SC) router receives control for a system control (SC) category PIU (from either NCP physical services CPM-OUT or boundary network node physical services CPM-OUT). The PIU request unit (RU) request code is resolved and through a table lookup routine, the appropriate processor for that request code is given control. The values of bits in the RH and RU determine whether system control (SC) or function management (FM) gets control.

The system control (SC) router is shared by NCP physical services and boundary network node (BNN). Not all SC functions are used by NCP physical services. The following identifies the commands and modules for the given RH/RU system control (SC) router values:

RH byte 0 x11xxxxx

```
RU Byte 0

0D   Activate logical CXDBSIL
0E   Deactivate logical CXDBSTL
11   Activate physical (BNN) CXDBSIP
11   Activate physical (NCP) CXDBAPH
12   Deactivate physical (BNN) CXDBSTP
12   Deactivate physical (NCP) CXDBDPH
31   Bind CXDBSIA
32   Unbind CXDBSTA
A0   Start data traffic CXDBSDF
A1   Clear
A2   Set and test sequence numbers
A3   Request recovery
```

There are data commands addressed to the system router which have an RH byte 0 value of x01x xxxx. The commands are:

RH byte 0 x01xxxxx

```
RU byte 0  Command

05   Lost subarea (NC) adjacent
07   Auto network shutdown complete
50   Initialization complete
51   Switch line to NCP mode (BSC/SS)
52   Switch line to EP mode (BSC/SS)
```

Function Management (FM)

The function management (FM) router validates FM requests, selects a table of processors according to the RVT type field and, by using a table lookup routine, selects the appropriate processor according to the PIU RU request code. If the PIU RH byte 0 has a value of x00x xxxx, the function management (FM) router is given control.

The PIU RU byte 1 value determines which of four FM categories is used. The PIU RU byte 2 contains the request code.

For a list of PIU commands in RU sequence, refer to *ACF/NCP/VS Network Control Program - Program Reference Summary* (LY30-3043), Section 5: NCP Network Commands. For a list of PIU commands in the sequence for activation, refer to Appendix A: PIU Command Sequence.

Network Control Program Physical Services Flow

Physical services CPM-OUT receives control via an enqueue macro with the ACTV=YES operand. This macro is issued by INN path control. This queueing occurs when INN path control receives a PIU with a DAF destined for NCP physical services. The PIU is enqueued on the physical services outbound queue in the physical services block (PSB). The task entry pointer for the PSB QCB points to the NCP physical services CPM-OUT.

If the contents of the RU byte 0 is a X'11' request code ('activate physical') is received and a SNP is available, a session is established by:

- Allocating a SNP for the session
- Setting the session mask in the SCB or CAB
- (channel source only) Placing the CAB address into the appropriate SVT entry
- (channel source only) Moving the appropriate CPT fields into the CAB
- Sending an initialization complete to the SSCP
- Sending the response to the 'activate physical'

If the contents of the RU byte 0 is not a X'11' request code ('activate physical'), CPM-OUT searches the vector table of SNPs (VTS) for an entry of the origin SSCP network address. The VTS entry contains the SNP address which contains session information for this session. Session status and sequence numbers are verified.

Physical services CPM-OUT uses bits 1 and 2 of the PIU RH byte 0 to determine the type of request. Both bits 'off' signifies a function management (FM) request. If the PIU is a system control request, the system control (SC) router is called.

Function management performs more verification on a request by checking the sequence number of the PIU against the SNP at offset X'0A'. CPM-OUT assumes that the first FM data request following the 'activate physical' from the SSCP to physical services must have a sequence number of X'0001' in its transmission header. Each subsequent function management request is expected to have a sequence number one greater than the previous request.

The system control (SC) router and the function management (FM) router both use a table lookup routine in conjunction with the PIU request code to select a processor. There are significant differences between the two routers.

The system control (SC) router first uses the DAF from the TH and the UIB1TYPE byte of the PIU to set an indicator showing the destination type for this PIU. The indicators are as follows:

```
X'80' Request is for NCP physical services
X'00' Request is for BNN physical services
X'40' Request is for a BNN logical unit
```

The indicator is used as the second byte of a two-byte table search argument. The request code from the RU1RCO byte of the PIU is used as the first byte of the second argument.

The search argument is compared to the first two bytes of each entry of the system control router table (SCRT). When a match is found, the routine pointed to in that entry is given control. X'FFFC' indicates the end of the SCRT.

The function management (FM) router activates links, contacts physical units, and performs similar services.

Function management requests are divided into four subcategories. The type of subcategory is determined by the contents or the RU1BT1 byte of the PIU as follows:

```
X'00' BSC/SS service request
X'02' Physical configuration services request
X'03' Physical maintenance request
X'06' Session services request
```

Once the function management (FM) router determines which subcategory the request is for, the RVTTYPE bytes within the RVT are used to select the proper table within that subcategory. An example of this table selection is the physical configuration subcategory which contains three tables:

```
Link configuration table
NCP configuration table
Station configuration table
```

Finally, the function management (FM) router uses the request code in the RU1RC2 byte of the PIU as a search argument for the selected table. When a match is found, the routine pointed to in that entry is given control. The function management (FM) router tables are delimited by a X'80'.

Physical services CPM-IN receives control via a branch from other routines. Physical services CPM-IN does not have a QCB and therefore cannot be dispatched. As an example, when a permanent link error occurs or a contact command completes, the link control block (LKB) QCB is dispatched. When the LKB processing is complete, physical services CPM-IN is executed via a branch using the LKB QCB. Physical services CPM-IN sends the 'inoperative' on an error or appropriate command required (contacted, discontact, etc.) for the completed contact command.

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 as explained below. (Any lines currently operating in emulation mode are unaffected by shutdown of lines in network control mode.)

The orderly procedure is called automatic network shutdown (ANS). The ANS facility is required and may not be omitted if multi-domains are defined (multiple channels or local to local link is defined). The ANS facility is included in the network control program for single domain definitions unless you specifically exclude it by coding ANS=NO in the BUILD macro. Separate from automatic network shutdown, individual lines and stations may be deactivated and reactivated by commands from the access method to the network control program.

The network control program performs automatic network shutdown for network resources on behalf of the SSCP that currently owns the resources when the network control program loses its ability for any reason to communicate with that SSCP -- a condition referred to as *lost subarea.* The network control program detects the loss of a subarea via:

- a timeout over the channel or link
- an unexpected link command (SNRM, DISC, etc)
- an unexpected 'activate physical'
- an adjacent network control program may notify the present network control program by a lost subarea (LSA) command

Automatic network shutdown of the network or a part thereof occurs under the following conditions:

Local network control program:

- An adjacent access method fails to respond to the network control program 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 parameter of the HOST macro that represents the access method.
- An adjacent network control program fails to respond to the network control program within a specified interval. For an NCP defined as the primary link (polling) the interval is specified in the REPLYTO and RETRIES operands of the LINE macro. For an NCP defined as the secondary link (polled) this interval is not user defined.
- An adjacent network control program notifies the present NCP that it has lost contact with a subarea in the network.
- A shutdown request is entered at the control panel of the communications controller (automatic shutdown occurs for all owners).

Remote network control program:

- The remote network control program detects a lapse in communication activity over the local/remote link it is presently using to communicate with the local network control program. The lapse may occur either through outright failure of the link or by exhaustion of error recovery procedures performed by the local NCP. The lapse interval is determined by the value you specify in the ACTIVTO operand of the GROUP macro representing the SDLC link. This interval must be sufficiently long for the local NCP to complete its error recovery procedures for the link (see REPLYTO and RETRIES operands on the LINE macro).
- The local network control program, upon entering ANS mode for the owner of the remote, signals the remote NCP to shutdown the lines controlled by the remote program.
- A shutdown request is entered at the control panel of the remote controller.

A failure requiring ANS is always associated with a channel control block (CAB) or station control block (SCB). All subareas associated with a specific CAB or SCB are located using the INN path control tables of subarea index table (SIT) and subarea vector table (SVT). The 'lost subarea' commands are sent identifying the lost subareas to all adjacent subareas and owners of the NCP.

NCP physical services uses the resource vector table (RVT) to locate resources owned by the lost owner. The action physical services takes differs for each kind of line and station undergoing shutdown, as follows:

For SDLC links, physical services:

- 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.

For SDLC stations for which ANS=STOP is specified in the PU macro, the network control program:

- Sends a 'Discontact' command and stops polling the station.
- Breaks the switched connection, if any, to the station.
- Terminates intensive mode and link test, level 2
- Dissociates the station from the owning SSCP with which communication has been lost.
- Cancels any sessions in which the station is currently active.

For SDLC stations for which ANS=CONT is specified in the PU macro, the network control program:

- Dissociates the station from the owning SSCP with which communication has been lost.
- Cancels any sessions in which the station is currently active.
- Cancels the session with the SSCP with which communication has been lost.
- Permits to continue any existing sessions with logical units not affected by loss of the owning SSCP (cross-domain sessions).

The action the network control program takes for BSC and SS devices during automatic network shutdown differs for each kind of line and station undergoing shutdown, as follows:

For BSC and SS 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 SS 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.
- Cancels any sessions in which the station is currently active.
- Resets the station from monitor mode, if that mode is currently in effect.

Physical services not only resets SNP masks in the control blocks of the lost owner, but also purges all pending messages within the NCP for that subarea. This results in duplicate responses to prior messages. As an example, an inbound request PIU reaches a host over a channel and a positive response is sent to the logical unit. The original request, however, is still on the channel hold queue until the next channel transfer. A lost subarea for that channel results in the original request on the hold queue converted to a *negative* response, link failure sense data, and a second response returned to the origin.

When the network control program recognizes a 'lost subarea' condition:

1. a 'network control lost subarea' command is sent to all adjacent hosts and network control programs.
2. a 'network services lost subarea' command is sent to all SSCP owners of the NCP.

The LSA identifies all subareas lost from the link failure. Each host or NCP receiving the 'adjacent' LSA message propagates the LSA to its adjacent hosts and NCPs unless the lost subareas are available over alternate paths. The 'SSCP owner' LSA is sent directly to the owner.

An example of the need for both types of LSA is in figure 6.8.

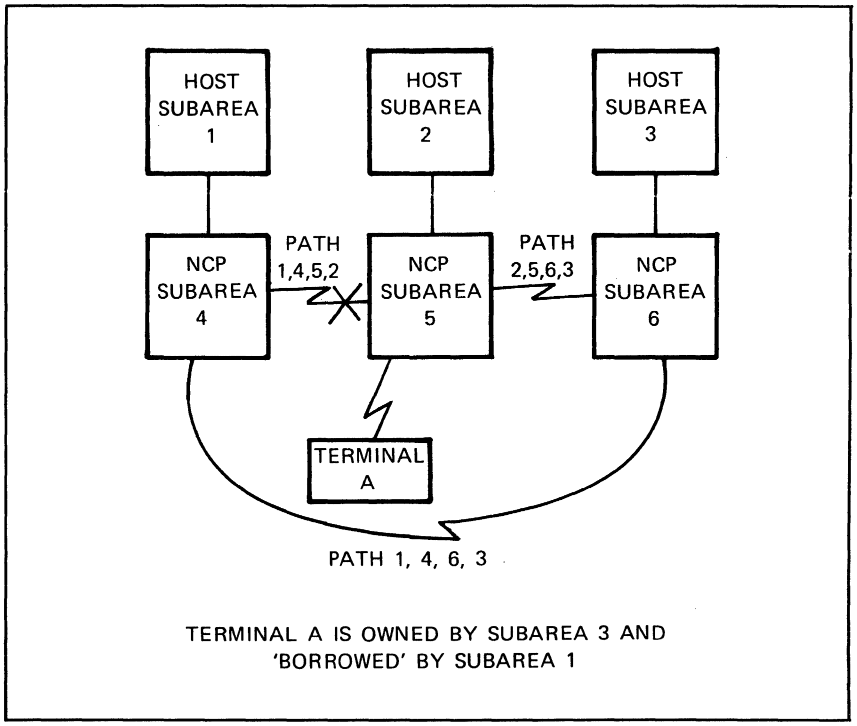


Figure 6.8. Lost Subarea Example

Figure 6.8 illustrates a path from the owning host, subarea 3, to terminal A from subarea 3 to 6 and 6 to 5. The CDRM session between subarea 1 and subarea 3 flows across the link between subareas 4 and 6 for subarea 1 to 'borrow' terminal A. The data flow between subarea 1 and terminal A occurs on the link between subarea 4 and subarea 5.

A link failure between subareas 4 and 5 results in subarea 5 sending an 'adjacent' LSA message identifying subareas 1 and 4 to subareas 2 and 6. Subarea 2 uses the failed path and identifies its sessions. Subarea 6 uses an alternate path (path 1,4,6,3) to reach subareas 1 and 4, and therefore does not propagate the 'adjacent' LSA to subarea 3. Subarea 3, which owns terminal A, would not be aware of the lost path (and 'lost' terminal) between subarea 1 and terminal A.

Subarea 4 recognizes the path failure of path 1,4,5,2, and sends LSAs to owners and adjacent subareas also. Subarea 6 receives an 'adjacent' LSA identifying subareas 2 and 5. Subarea 6 would not propagate the LSA as subarea 6 has a path to subareas 2 and 5 over path 2,5,6,3.

The link failure between subareas 4 and 5 results in subarea 5 sending an 'owner' LSA message identifying subareas 1 and 4 to its owners (subarea 2 and 3). Subarea 2 received the 'adjacent' and 'owner' LSA; subarea 3 received the 'owner' LSA only. Subarea 3 can recover terminal A.

In some cases the 'owner' LSA duplicates the 'adjacent' LSA, however Figure 6.8 illustrates the need for both types.

Network Control Program Physical Services Summary

NCP physical services is represented by a connection point manager in (CPM-IN), connection point manager out (CPM-OUT), system control (SC) services, and function management (FM) services. The control blocks of physical services are (1) the physical services control block (PSB), vector table of SNPs (VTS), and SSCP-NCP session control block (SNP).

One to eight SSCPs may establish a session with an ACF/NCP. The 'activate physical' command requests a session, obtains a SNP mask and SNP control block for the SSCP.

Physical services provides services for system control (SC) requests and function management (FM) requests. The session initiation with the NCP, activation of lines, initial contact of devices, etc., all are performed by physical services. Host control requests are sent to physical services in the PIU RU with the command type, command, and resource address of the element to be affected by the command.

Chapter 7:

Boundary Network Node (BNN)

Introduction

The NCP boundary network node (BNN) is the interface for type 1 and type 2 physical units between INN path control and the link scheduler. The BNN processes PIUs containing control requests and data associated with sessions between:

- SSCP and the physical units (SSCP/PU)
- SSCP and the logical unit (SSCP/LU)
- Host application and the logical unit (APPL/LU)

The major elements of BNN outbound are:

- BNN 'connection point manager out' (CPM-OUT)
- BNN 'path control out delayed'

The major elements of BNN inbound are:

- BNN 'path control in immediate'
- BNN 'path control in delayed'.
- BNN 'connection point manager in' (BNN CPM-IN),

BNN outbound consists of processing PIUs travelling to a physical unit (PU) or logical unit (LU) on an SDLC link. BNN inbound consists of processing PIUs received from a physical unit (PU) or logical unit (LU).

There are three distinct paths through the BNN for PIUs travelling in either direction. These paths relate to the session which can be established with PUs or LUs. The possible sessions are SSCP/PU, SSCP/LU, and APPL/LU.

Boundary Network Node Outbound Processing

Outbound PIUs in an SSCP/PU session are queued on the BNN SSCP/PU 'connection point manager out' (CPM-OUT) queue. System control requests, such as 'activate physical' and 'deactivate physical', are passed to the system control (SC) router for processing.

Session status is recorded in the common physical unit block (CUB). SSCP/PU CPM-OUT branches to BNN 'path control out delayed' for conversion of the FID1 to FID2 or FID3 format.

SSCP/LU CPM-OUT

Outbound PIUs in an SSCP/LU session are queued on the BNN SSCP/LU 'connection point manager out' (CPM-OUT) queue. System control requests, such as 'activate logical', and 'deactivate logical' are passed to the system control (SC) router for processing. The 'bind' command is queued by path control to this queue, checked by the SSCP/LU CPM-OUT, and queued to the APPL/LU CPM-OUT for additional processing.

Session status is recorded in the logical unit block (LUB). SSCP/LU CPM-OUT branches to BNN 'path control out delayed' for conversion of the FID1 to FID2 or FID3 format.

APPL/LU CPM-OUT

Outbound PIUs in an APPL/LU session are queued on the BNN APPL/LU 'connection point manager out' (CPM-OUT) queue.

The APPL/LU session data flow is scheduled by pacing definitions. Pacing is covered in the topic Boundary Network Node (BNN) Pacing.

System control requests, such as 'bind' and 'unbind' are passed to the system control (SC) router for processing.

Session status is recorded in the logical unit block (LUB). APPL/LU CPM-OUT branches to BNN 'path control out delayed' for segmenting and conversion of the FID1 to FID2 or FID3 format.

BNN Path Control Out Delayed

BNN path control out delayed is entered from the three BNN CPM-OUT routines. BNN path control out delayed converts the FID1 PIUs to FID2 or FID3, segments the PIU (if required) and places the PIU on the physical unit 'link outbound' queue for transmission on the link.

Boundary Network Node Inbound Processing

All inbound PIUs from SDLC devices are processed by BNN path control in immediate. If the PIU has been received from a type 1 or type 2 PU the PIU is queued on the 'link inbound' queue of the physical unit. If the PIU has been received from a type 4 PU, BNN path control in immediate branches to INN path control.

BNN Path Control In Delayed

BNN path control in delayed is initiated by a PIU queued on the physical unit 'link inbound' queue by BNN path control in immediate. BNN path control in delayed converts the PIU from FID2 or FID3 to a FID1 and, depending upon the session, branches to the BNN SSCP/PU CPM-IN, SSCP/LU CPM-IN, or APPL/LU CPM-IN.

BNN SSCP/PU CPM-IN

BNN SSCP/PU CPM-IN receives inbound PIUs from BNN path control in delayed. Responses are checked for session status to update NCP status fields. Requests and responses are passed to INN path control.

BNN SSCP/LU CPM-IN

Inbound PIUs in an SSCP/LU session are passed from BNN path control in delayed. Responses are checked for session status to update NCP status fields. Requests and responses are passed to INN path control.

BNN APPL/LU CPM-IN

Inbound PIUs in an APPL/LU session are passed from BNN path control in delayed.

The APPL/LU session data flow is scheduled by pacing definitions. Pacing is covered in the topic Boundary Network Node (BNN) Pacing.

Inbound PIUs are passed to INN path control.

BNN Control Blocks

The control blocks used by the boundary network node (BNN) code are:

- Common physical unit block (CUB)
- Logical unit block (LUB)
- Logical unit vector table (LUV)

References are made in this section to NCP physical services control blocks (covered in the previous section on physical services).

The boundary network node (BNN) code processes FID1, FID2, and FID3 PIUs. The formats of the PIU were covered in the Network Control Program Supervisor section.

Common Physical Unit Block (CUB)

The system pointer to a common physical unit block (CUB) is in the resource vector table (RVT).

Figure 7.1 illustrates some of the primary fields of the CUB.

Offset	CUB	
0	INBOUND QCB (LINK INBOUND QUEUE)	
18	LINK OUTBOUND QUEUE HEAD	
1C	SNP MASK	LINK OUTBOUND QUEUE TAIL
20		LINK OUTSTANDING QUEUE HEAD
24	STATION TYPE	LINK OUNSTANDING QUEUE TAIL
28	SDLC ADDR	LKB ADDRESS
2C	CUB NETWORK ADDRESS	POLL/CONTACT STATUS
64	LUB COUNT	LUV ADDRESS
68	OUTBOUND QCB (CPM—OUT)	

Figure 7.1. Common Physical Unit Block (CUB)

The common physical unit block (CUB) is generated by a PU macro with an operand of PUTYPE=1 or PUTYPE=2. The CUB represents the physical device for SDLC control.

Outbound PIUs for a type 1 or type 2 PU are queued on the SSCP/PU CPM-OUT queue at CUB offset X‘68’. SSCP/PU CPM-OUT processes the command. Commands for a type 1 CUB are processed within the NCP. The request PIU is converted to a response and given to INN path control. Commands for a type 2 CUB are processed within the NCP to record pending session status. The PIU is converted from a FID1 to a FID2, and placed on the CUB link outbound queue (LOBQ). The LOB queue is at CUB offset X‘18’. The level 3 link scheduler searches the LOB queue for PIUs to send on the link.

Inbound PIUs from a type 1 or type 2 PU are queued on the CUB link inbound queue at CUB offset X‘00’. PIUs on the link inbound queue initiate BNN path control in delayed. BNN path control in delayed determines the

PIU session and branches to the SSCP/PU CPM-IN, SSCP/LU CPM-IN, or APPL/LU CPM-IN.

Figure 7.1 illustrates the common physical unit block (CUB). The CUB has a four-byte address at offset X'64' which contains a pointer to the logical unit vector table (LUV) and a count of LUV entries. The count of LUV entries defaults to the quantity of LU macros following a PU macro, or is coded in the MAXLU operand on a PU macro. On a nonswitched PU, if MAXLU is greater than the number of defined LUs, dynamic reconfiguration is used to add additional LUs without a new NCP generation. On a switched PU, MAXLU defines the maximum number of LUs for a switched connection.

Logical Unit Vector Table (LUV)

Each common physical unit block (CUB) has a logical unit vector table (LUV). The address pointer to the logical unit vector table (LUV) is located in each CUB at offset X'64'.

Figure 7.2 illustrates a logical unit vector table (LUV). The quantity of LUV entries defaults to the number of LU macros following a PU macro. The PU macro operand of MAXLU defines the quantity of LUV entries generated.

	LUV	
0	LOCAL ADDRESS	LUB or NLB ADDRESS
4	FLAGS	
8	LOCAL ADDRESS	LUB or NLB ADDRESS
C	FLAGS	
10	LOCAL ADDRESS	LUB or NLB ADDRESS
14	FLAGS	

Figure 7.2. Logical Unit Vector (LUV) Table

The LUV is initialized at NCP generation for the generated LUs on a non-switched line. Additional entries, reserved by the MAXLU operand, are initialized in the same manner as entries for switched lines.

NCP generation reserves a quantity of type 1 and/or type 2 logical unit blocks (LUBs) as defined in the LUDRPOOL macro. The host requests the NCP to add a specified number of LU's to a specified PU and return the network addresses of the added logical units. The request is the command 'request network address assignment' (RNAA). The request RU header is X'410210'. The PU address is in RU bytes 3 and 4. The RU identifies the logical unit dynamic reconfiguration pool and quantity of logical unit blocks (LUBs) to be allocated.

Each logical unit block (LUB) must be initialized by a 'set control vector' command (RU X'010211xxxx04'). The command 'free network addresses' is used to release logical units blocks (LUBs) to the pool. A generated LUB cannot be released unless the LU macro was coded LUDR=YES.

For additional information, see the following topic on dynamic reconfiguration.

Note: ACF/NCP release 2 supports back level hosts with SDLC switched support and use of the LUPOOL defined logical units. NCP physical services receives an 'assign network addresses' command to allocate logical unit blocks (LUBs) from the LUPOOL logical unit pool. The LUB local address and storage address of the LUB is stored in a LUV entry. When the switched connection ends the LUBs are returned to the LUPOOL logical unit pool and the LUV entries cleared by a 'free network addresses' command.

Logical Unit Block (LUB)

The system pointer to a logical unit block (LUB) is in the resource vector table (RVT). Logical unit blocks (LUBs) are generated by an LU or LU-DRPOOL macros. NCP provides support functions (such as sequence number support) for type 1 LUBs; therefore, additional fields are required for type LUBs.

The format of a type 2 LUB is illustrated in Figure 7.3. The format of a type 1 LUB is identical to the type 2 LUB with the addition of a PU type 1 extension at offset X'50'. The format of a type 1 LUB is illustrated in Figure 7.4.

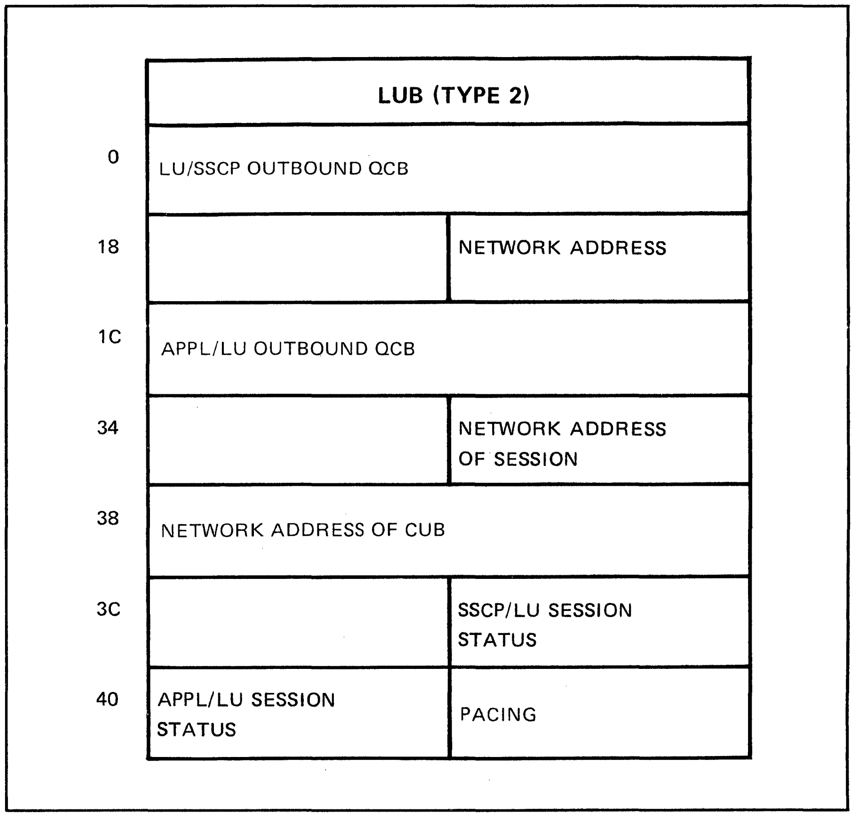


Figure 7.3. Type 2 Logical Unit Block (LUB)

The logical unit block (LUB) contains the queues for the SSCP/LU CPM-OUT (LUB offset X'00') and APPL/LU CPM-OUT (LUB offset X'1C').

The logical unit block (LUB) is generated by a LU or LUDRPOOL macros.

Logical units for a type 2 PU have a length of X'50' bytes. Logical units for a type 1 PU have an extension at LUB offset X'50' to allow NCP to maintain SSCP/LU normal and expedited identification fields and APPL/LU inbound and outbound sequence numbers.

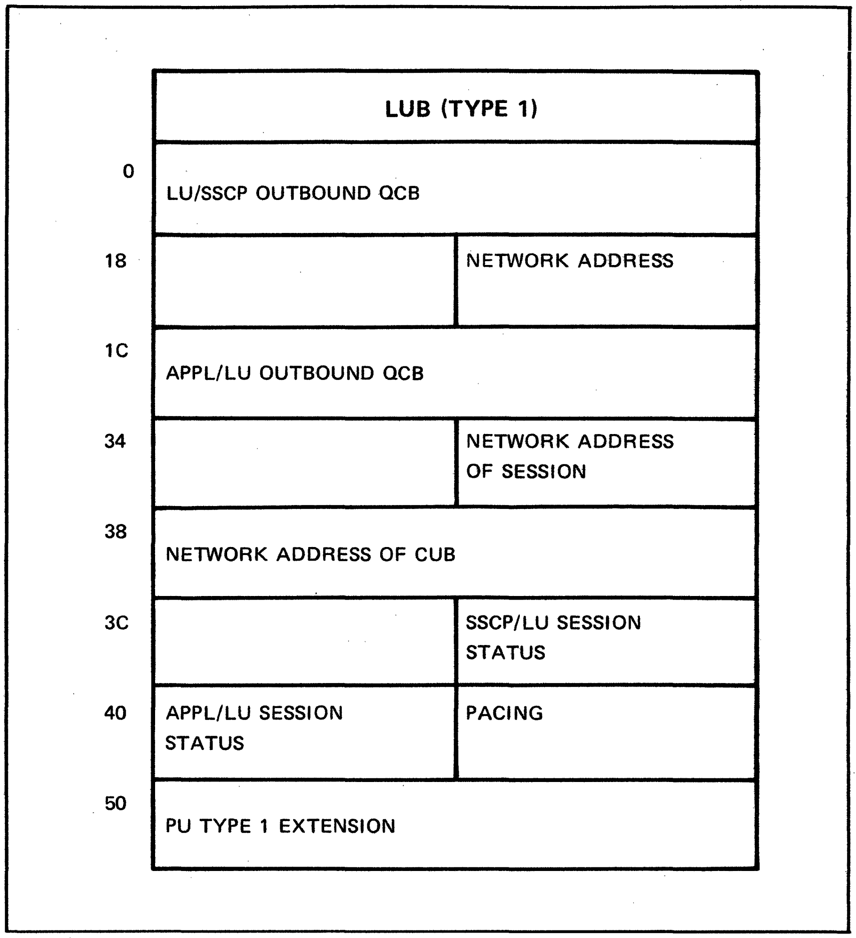


Figure 7.4. Type 1 Logical Unit Block (LUB)

Dynamic Reconfiguration

Dynamic reconfiguration provides the ability for access methods to add and delete nonswitched type 1 and type 2 PUs and switched and nonswitched LUs without going through an NCP generation.

Dynamic reconfiguration allows the definition of pools of physical units (CUBs) and logical units (LUBs) for allocation at execution time. The PUDRPOOL macro defines the number of CUB control blocks to be generated for PU dynamic reconfiguration. CUBs may be added only to SDLC nonswitched lines. A LINE macro must be coded with the MAXPU operand value to reserve an entry in the line's PU vector table. The SERVICE macro must be coded with the MAXLIST operand value to reserve an entry in the service order table for the added CUB.

The LUDRPOOL macro defines the number of type 1 logical unit blocks and number of type 2 logical unit blocks to be generated for LU dynamic reconfiguration and SDLC switched support. LUBs may be added to nonswitched SDLC lines for dynamic reconfiguration, and switched lines. A PU macro must be coded with the MAXLU operand value to reserve an entry in the PU's LU vector table.

Generated physical units may be released from a generated position and the CUB and LUV reallocated if the PU is coded PUDR=YES. Generated logical units may be released from a generated position and the LUB reallocated if the LU is coded LUDR=YES. The original network address is lost when a control block is released. For a control block to be released and reallocated a new network address must be reserved by the BUILD macro operand of RESOEXT=count. The count operand defines the quantity of addresses (entries in the RVT) to be generated. When a control block has been assigned from a dynamic reconfiguration pool, the network address is not lost when the control block is returned to the pool.

SDLC type 1 3270 command support is provided by NCP. If the generated NCP does not include SDLC type 1 3270 definitions, and if SDLC type 1 3270 support is to be added using dynamic reconfiguration, the programming support must be included at NCP generation. The programming support is included by coding the BUILD macro operand of DR3270=YES. This operand is *not* required for SDLC type 2 3270 support.

The host requests the NCP to add a specified number of PU's to an SDLC nonswitched line and return the network addresses of the added physical units. The request is the command 'request network address assignment' (RNAA). The request RU is X'410210'. The LINE address is in RU bytes 3 and 4. The RU identifies the physical unit reconfiguration pool (RU byte 5 X'44) and quantity of CUBs to be allocated.

The host may send a free network address (FNA) command with RU bytes 3 and 4 X'0000' and bytes 7 and 8 with the address of the CUB; NCP builds the link address in the response to the host without releasing the CUB.

If a PU has previously been added by dynamic reconfiguration, and a host asks for a dynamic reconfiguration add, the network address of the assigned CUB control block is sent to the host.

Each CUB must be initialized by a 'set control vector' command (RU X'010211xxxx03'). The command 'free network addresses' is used to release CUBs back to the pool. A generated CUB cannot be released unless the PU macro was coded PUDR=YES.

The host requests the NCP to add a specified number of LU's to a specified PU and return the network addresses of the added logical units. The request is the command 'request network address assignment' (RNAA). The request RU is X'410210'. The PU address is in RU bytes 3 and 4. The RU identifies the logical unit dynamic reconfiguration pool (RU byte 5 X'41') and quantity of LUBs to be allocated.

If an LU has previously been added by dynamic reconfiguration, and a host asks for a dynamic reconfiguration add, the network address of the assigned LUB control block is sent to the host.

Each logical unit block (LUB) must be initialized by a 'set control vector' command (RU X'010211xxxx04'). The command 'free network addresses' is used to release LUBs back to the pool. A generated LUB cannot be released unless the LU macro was coded LUDR=YES.

If generated CUB or LUB control blocks are released the generated network address is lost. The released control blocks are added to the appropriate dynamic reconfiguration pool only if a network address is available to be assigned from the extension.

BNN Outbound Flow

Outbound PIUs destined to BNN are received from INN path control. If a PIU is for a PU from SSCP, the PIU is enqueued on the SSCP/PU queue within the common physical unit block (CUB) at offset X'68'. If the PIU is for an LU from SSCP, the PIU is enqueued on the SSCP/LU queue of the logical unit control block (LUB) at offset X'00'. If the PIU is for an LU from an application program, the PIU is enqueued on the APPL/LU queue of the logical unit control block (LUB) at offset X'1C'.

PIUs for the three types of sessions are enqueued on an input QCB which has a task entry point of a 'connection point manager out' (CPM-OUT). Each session has a separate CPM-OUT because the processing is different. The ENQUE macro issued in INN path control includes the ACTV=YES operand which causes the associated task to be triggered if the task is in the ready state. When the task is dispatched, the appropriate CPM-OUT has control.

If the PIU is a system control command the system control (SC) router is called. This is the same system control (SC) router which is used by NCP physical services.

The APPL/LU CPM-OUT data flow is scheduled by pacing definitions and the dispatching state of the APPL/LU task. Pacing is covered later in this section in the topic Boundary Network Node (BNN) Pacing.

BNN path control out delayed is initiated by a branch from SSCP/PU CPM-OUT, SSCP/LU CPM-OUT, and APPL/LU CPM-OUT. BNN path control out delayed converts the FID1 PIU to FID2 or FID3 PIU. For APPL/LU sessions, the PIU is segmented if the length exceeds the physical unit MAX-DATA operand value. Finally, BNN path control out delayed issues an XIO LINK which causes the PIU to be enqueued on the common physical unit block (CUB) link outbound queue.

Figure 7.5 illustrates the flow of an outbound PIU through BNN. Depending upon the session a PIU is enqueued to the SSCP/PU CPM-OUT queue, SSCP/LU CPM-OUT queue, or APPL/LU CPM-OUT queue.

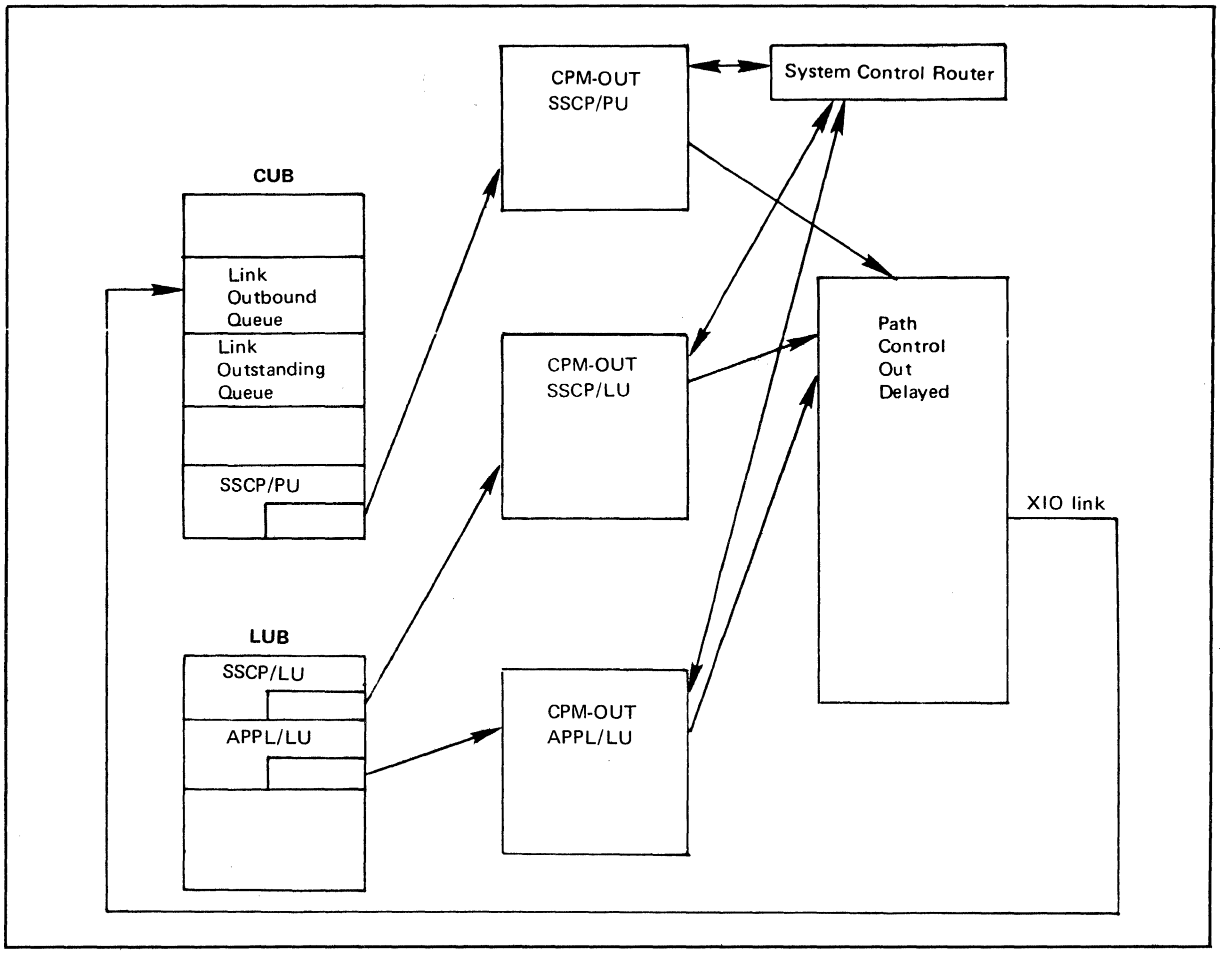


Figure 7.5 Boundary Network Node Outbound Path Flow

SSCP/PU CPM-OUT

Outbound PIUs in an SSCP/PU session are queued on the BNN SSCP/PU 'connection point manager out' (CPM-OUT) queue. The PIU is validated as a FID1 format. Only a FID1 format is valid input to the SSCP/PU CPM-OUT.

The PIU origin address field (OAF) is compared to the network address of the SSCP, which is stored in the SSCP-NCP session control block (SNP). The PIU must be from the SSCP owner which issued the 'contact' command.

The CUB cannot accept any SSCP/PU commands unless the PU is operational. This operational status occurs by means of a function management (FM) 'contact' command from SSCP to NCP physical services.

The contact command claims ownership for the origin and schedules a 'set normal response mode' (SNRM) SDLC command to the device. An 'unsequenced acknowledgement' (UA) reply indicates that the command was received and the physical unit is ready for session initiation. Bit 3 of CUBSSCF at offset X'2E' of the common physical unit block (CUB) is set to zero to indicate that the CUB is operational.

If the PIU is a control request, with an x11x xxxx in byte 0 (RH1B0) of the RH, the system control (SC) router is called. This is the same system control (SC) router which is used by NCP physical services. If the control command in byte 0 of the RU is X'11' ('activate physical'), CPM-OUT checks for a session established at bit 0 of X'5C' (CUBPSTAT) in the CUB. If a session is already established, the request is rejected and returned to origin.

If the physical unit is a type 2, bit 1 of byte X'5C' CUBPSTAT is turned on to indicate that a session initiation request is being processed. CPM-OUT branches to 'path control out delayed' to convert the FID1 to a FID2 and enqueue the PIU for transmission to the CUB link outbound queue at CUB plus X'18' (CUBLOBH).

If the physical unit is a type 1, the 'activate physical' command is processed by the NCP and not transmitted to the physical device. The 'session established' bit in the CUB at offset X'5C' (bit zero) is set on. The request is converted into a response and sent to the origin.

If the device is a type 1 SDLC 3271 or 3275 (BNNSUP=YES), all commands are processed by the NCP, and all replies on behalf of the 3271 or 3275 are created by NCP and sent to the host.

If the PIU is not a control request (RH byte 0 value of x00x xxxx), the CUB is checked at offset X'5C' for bit 0 value of 1 to confirm that a session has been established. If a session has been established, CPM-OUT branches to BNN 'path control out delayed' to convert the FID1 to FID2 (or FID3) and enqueue the PIU for transmission to the CUB link outbound queue at CUB plus X'18' (CUBLOBH).

SSCP/LU CPM-OUT

Outbound PIUs in an SSCP/LU session are queued on the BNN SSCP/LU 'connection point manager out' (CPM-OUT) queue. The PIU is validated as a FID1 format. Only a FID1 format is valid input for the SSCP/LU CPM-OUT.

Path control queues bind commands on this queue. SSCP/LU CPM-OUT queues the bind on the APPL/LU CPM-OUT queue for additional processing.

The PIU origin address field (OAF) is compared against the network address of the owner, which is stored in the SSCP-NCP session control block (SNP). Only the owner of the physical unit can create this SSCP/LU session.

The SSCP/LU session cannot exist unless the SSCP/PU session is established. The CUB is checked for a 1-bit in bit 0 of X'5C' (CUBPSTAT), indicating an active SSCP/PU.

If the PIU is a control request with an x11 xxxx in byte 0 of the RH, the system control (SC) router is called. This is the same system control (SC) router which is used by NCP physical services.

If the control command in byte 0 of the RU is X'0D' ('activate logical'), the LUB is checked for an existing session at LUB plus X'3C' (LUBCPSET) indicated by a 1 in bit 0. If no session exists, bit 3 in X'3C' (LUBCPSET) is set to 1 to indicate that an 'activate logical' command is being processed.

CPM-OUT branches to BNN 'path control out delayed' to convert the FID1 to a FID2 or FID3 and to enqueue the PIU for transmission to the CUB link outbound queue at CUB plus X'18' (CUBLOBH).

If the PIU is not a control request (RH byte 0 value of x00x xxxx), the LUB is checked at offset X'3C' (LUBCPSET) for a bit 0 value of 1 to confirm that a session has been established. If a session has been established, CPM-OUT branches to BNN 'path control out delayed' to convert the FID1 to a FID2 or FID3 and enqueue the PIU for transmission to the CUB link outbound queue at CUB plus X'18' (CUBLOBH).

APPL/LU CPM-OUT

Outbound PIUs in an APPL/LU session are queued on the BNN APPL/LU 'connection point manager out' (CPM-OUT) queue. Enqueuing a PIU dispatches the CPM-OUT task if the task is in the ready state.

The APPL/LU CPM-OUT is scheduled by pacing values and the task state. The PIU processing may be deferred by pacing requirements or by the task being nondispatchable. Pacing is covered later in this section in the topic Boundary Network Node (BNN) Pacing.

The PIU is validated as a FID1 format. Only a FID1 format is valid input for the APPL/LU CPM-OUT.

The APPL/LU CPM-OUT processor checks to verify that the SSCP/LU session exists. The LUB is checked for a 1-bit in bit 0 of X'3C' (LUBCPSET).

If the PIU is a control request with an x11x xxxx in byte 0 of the RH, the system control SC) router is called. This is the same system control (SC) router which is used by the NCP physical services. If the control command in byte 0 of the RU is X'31' ('bind') the LUB is checked for an active APPL/LU session (bit 0 value of 1) in LUB plus X'40' (LUBAPSET).

If no 'bind' command has established a session, bit 3 of byte X'40' of the LUB is set to 1 to indicate that a 'bind' is being processed. CPM-OUT branches to 'path control out delayed' to convert the FID1 to a FID2 or FID3 and to enqueue the PIU for transmission to the CUB link outbound queue at CUB plus X'18' (CUBLOBH).

If the PIU is not a control request (RH byte 0 value of x00x xxxx), the LUB is checked at offset X'40' (LUBAPSET) for a bit 0 value of 1 to confirm that a session has been established. If a session has been established, CPM-OUT branches to BNN 'path control out delayed' to convert the FID1 to a FID2 or FID3, segment data PIUs as required, and enqueue the PIU for transmission to the CUB link outbound queue at CUB plus X'18' (CUBLOBH).

If the PIU flows within the APPL/LU session, the address of the APPL/LU CPM-OUT is placed in the PIU ECB at offset X'10'. (In a segmented PIU, all non-last segments have a value of zero at offset X'10', and the address is provided in the ECB of the last segment.) When the PIU has been processed and placed on the CUB link outbound queue the APPL/LU CPM-OUT task exits without a QPOST leaving the task in a nondispatchable state.

Additional PIUs may be added to the APPL/LU CPM-OUT queue and a pacing response may be received for this queue without dispatching this task. When the link scheduler dequeues the PIU from the link outbound queue, if the value in the buffer prefix at offset X'10' is nonzero, the link scheduler provides an unconditional trigger to the task address at X'10'. The unconditional trigger queues the APPL/LU CPM-OUT task for dispatching.

Allowing only one PIU on the link outbound queue for one APPL/LU session ensures an even distribution of PIUs for all sessions.

If only one APPL/LU session exists with this physical unit the second PIU should be queued on the link outbound queue before the first PIU has been sent on the link. This allows multiple PIUs to a PACING, MAXOUT, or PASSLIM limit with a single APPL/LU session.

BNN Path Control Out Delayed

BNN path control out delayed is initiated by a branch from SSCP/PU CPM-OUT, SSCP/LU CPM-OUT, and APPL/LU CPM-OUT. BNN path control out delayed converts the FID1 PIU to FID2 or FID3 PIU. For APPL/LU sessions, the PIU is segmented if the length exceeds the physical unit MAXDATA operand value. Finally, BNN path control out delayed issues an XIO LINK which causes the PIU to be enqueued on the common physical unit block (CUB) link outbound queue.

FID1 to FID2 Conversion

When the PIU is received by BNN path control out delayed, the PIU is checked to ensure it is a valid FID1 PIU. Conversion does not change the request/response header (RH) or request/response unit (RU). The only change is to the transmission header (TH). Figure 3.3 illustrates the FID conversion within NCP buffers.

The TH1DCF count field at offset X'22' and one byte from the OAF and DAF fields are deleted. The TH1SNF sequence number field at offset X'20' is moved to X'22'.

Both the destination address field (DAF) and the origin address field (OAF) are two-byte fields in a FID1 PIU. The FID2 format provides only a single byte for each of these fields. The PIU has reached the destination point of the network address by being queued to the specific control block which defines the physical destination point. The full network address is no longer required. For outbound PIUs the destination address need identify only the device local address; the origin address need identify only that the PIU flows in the SSCP session or application session.

If the FID1 origin address field (OAF) is from an SSCP, the FID2 OAF field is set to a value of X'00'. If the PIU is from an application program, the field is set to X'01'. The FID2 OAF is located at TH2OAF at offset X'21' where the original FID1 sequence number was located.

If the DAF is a type 1 physical unit the command is processed by NCP by the SSCP/PU CPM-OUT. BNN path control out delayed is not used.

If the DAF is a type 2 physical unit the local address of the physical unit is X'00'. If the DAF is a logical unit the local address for the FID2 DAF is obtained from the LUB at offset X'47'. The FID2 DAF is located in the NCP buffer at TH2DAF at offset X'20'.

The TH1B0 from X'1A' is moved to X'1E', with bit 3 set to 0 (FID1 indicator) and bit 2 set to 1 (FID2 indicator).

A four-byte gap has been created from X'1A' through X'1D'. The buffer offset is incremented by 4 and the buffer data count field is decremented by 4 to adjust for the change. The PIU FID1-to-FID2 conversion is complete.

FID2 to FID3 Conversion

Type 1 PU SSCP/PU sessions are processed by the NCP in FID1 format. No conversion occurs for PIUs within an SSCP/PU session for type 1 PUs.

If the bit settings in the common physical unit block (CUB) at offset X'24' bit 5 is a 1 (type 1 PU), the FID2 must be converted to a FID3.

Conversion of the FID2 to a FID3 format affects only the transmission header (TH) fields. Four more bytes in the original buffer are converted and become reserved fields and only two bytes of TH are used. The first byte of FID3 TH contains the FID3 identifier at buffer offset of X'22'. The offset of X'23' bits 0 and 1 contains two bits of information defining the session as follows:

- Bit 0 - 1=to/from application, 0=to/from SSCP
- Bit 1 - 1=to/from logical unit,0=to/from physical unit

The remaining six bits contain the device local address of the destination of this PIU. The local address is obtained from the FID2 OAF field with the two leftmost bits deleted.

PIU Segmenting

A data PIU from an application is segmented only if the FID2 or FID3 PIU length (TH, RH, RU) exceeds the MAXDATA operand on a PU macro. If the PIU length is greater than MAXDATA, the segmentation routine (CXDBSEG) divides the PIU into segments which are equal to or less than the length of MAXDATA.

A first segment contains the TH, RH, and a portion of the RU, in multiples of full NCP buffers, the total length of which is less than or equal to MAXDATA size. A nonfirst segment consists of the TH (copied from the first segment into a separate buffer) plus one or more full NCP buffers of less than or equal to MAXDATA size. The first buffer of each nonfirst segment contains the eighteen-byte event control block and TH; the buffer chain field contains the address of a data buffer. If PIUs of more than MAXDATA length are used, the NCP buffer size should be selected to provide an efficient segment length. If segmenting is not normal, the segment length should not be a consideration in selecting an NCP buffer size.

To calculate an appropriate buffer size for segmentation, subtract the TH size from MAXDATA. If MAXDATA=265 and TH is 6 bytes, then nonfirst segment RUs may be 256 bytes in length. If MAXDATA=261 and TH is 2 bytes, then nonfirst segment RUs may be 256 bytes in length. Each nonfirst segment contains a copy of the ECB and TH in a separate buffer. Because segmentation does not work with partial NCP buffers, a buffer size of 60

(BFRS=60) results in the copied TH plus three NCP buffers (180 bytes of RU) sent per segment. A buffer size of 64 (BFRS=64) results in the copied TH plus four NCP buffers (256 bytes of RU) sent per segment.

Each segment is placed on the link outbound queue. The task processes all segments before exiting, and therefore all segments are chained together; the chain of segments queued on the link outbound queue are without intervening PIUs.

Chaining of PIU segments is the same as chaining multiple PIUs. The address of the following segment is in the event control block at offset X'08' in the buffer which contains the TH. The last segment chain field contains zeros. Buffers within a segment are chained with the buffer chain field (buffer offset X'00').

The first and middle segments contain a value of zero in the buffer at offset X'10' to avoid an unconditional trigger to the APPL/LU CPM-OUT by the link scheduler as each segment is sent. The last segment contains the address of the APPL/LU CPM-OUT to be triggered.

Figure 7.6 illustrates a PIU which requires segmenting. The FID1 PIU contains 581 bytes (568 bytes of RU). The physical unit definition is coded MAXDATA=265. The NCP buffers are defined as 64 bytes. Segment size is in full NCP buffers. The segments sizes are:

- First segment, TH=6, RH=3, and RU=225, from the first four NCP buffers. The RU is made up of 33 bytes of the first buffer and 192 bytes of the second, third and fourth buffers.
- Middle segment, TH=6 and RU=256. The TH is copied from the first buffer into a leased buffer. The RU is from buffers five, six, seven, and eight.
- Last segment, TH=6 and RU=87. The TH is copied from the first buffer into a leased buffer. The RU is from buffer nine and ten.

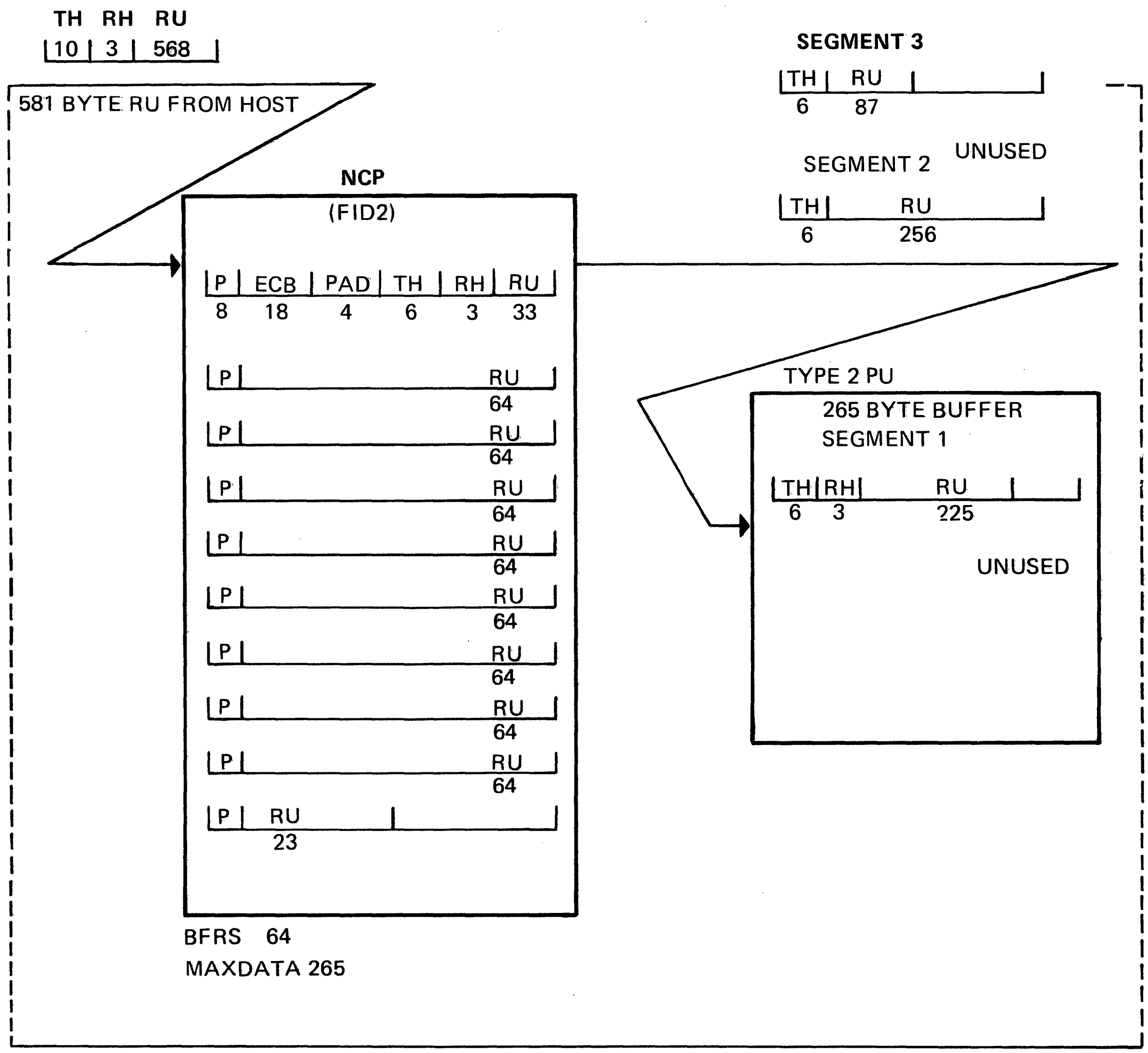


Figure 7.6. Segmentation Example

A PIU from BNN path control out delayed is placed on the link outbound queue by an XIO LINK macro. If the PIU is segmented, all segments are queued together on the link outbound queue. Pacing occurs on complete PIUs, *not PIU segments*. Keep in mind that segments of a PIU may overrun the PU physical line buffers.

Segmenting may not be supported by a specific terminal type. In addition, you should not confuse *segmenting* by BNN path control out delayed (TH indicated) and *chaining* by an application program (RH indicated).

Summary BNN Outbound Flow

A PIU from an SSCP to a physical unit is enqueued on the CUB BNN CPM-OUT queue at X'68'. This queuing triggers the SSCP/PU connection point manager out (CPM-OUT). If the PIU is a system control command, the PIU is passed to the system control (SC) router for processing; control is returned to SSCP/PU CPM-OUT. Type 1 PU commands are processed by NCP, and responses are queued to SSCP/PU CPM-IN. A type 2 PU PIU is passed to BNN path control out delayed for conversion to a FID2. The PIU is passed to the link scheduler by placing the PIU on the CUB link outbound queue at CUBLOBH (CUB plus X'18').

SSCP/LU

A PIU from SSCP to a logical unit is enqueued on the LUB SSCP/LU CPM-OUT queue at LUL1ECB (offset X'00'). This queueing triggers the SSCP/LU connection point manager out (CPM-OUT). If the PIU is a system control command, control is passed to the system control (SC) router; control is returned to SSCP/LU CPM-OUT. PIUs are passed to BNN path control out delayed for conversion to FID2 or FID3 and queueing on the CUB link outbound queue at CUBLOBH (CUB plus X'18').

APPL/LU

A PIU from an application to a logical unit is enqueued on the APPL/LU CPM-OUT queue at LUA1ECB (offset X'1C'). Unless the task is disconnected the queueing triggers the APPL/LU connection manager out (CPM-OUT). When a PIU is placed on the link outbound queue of the CUB the task exits in the disconnect state (no QPOST). When the link scheduler dequeues the PIU from the link outbound queue the APPL/LU CPM-OUT is unconditionally triggered to process the following PIU.

The APPL/LU CPM-OUT is scheduled by pacing values. The PIU processing may be deferred by pacing requirements. Pacing is covered later in this section in the topic Boundary Network Node (BNN) Pacing.

If the PIU is a system control command, the PIU is passed to the system control (SC) router for processing; control is returned to SSCP/LU CPM-OUT. PIUs are passed to BNN path control out delayed for conversion to FID2 or FID3, segmenting, and queueing on the CUB link outbound queue at CUBLOBH (CUB plus X'18').

BNN Path Control Out Delayed

BNN path control out delayed is entered from the three BNN CPM-OUT routines. BNN path control out delayed converts the FID1 PIUs to FID2 or FID3, segments the PIU (if required) and places the PIU on the physical unit 'link outbound' queue for transmission on the link.

BNN Inbound Flow

PIUs travelling inbound in BNN are received from the link scheduler. The link scheduler branches to BNN path control in immediate in interrupt level 3. BNN Path control in immediate validates the PIU and checks for the type of PU.

Note: The link schedule is common for type 1, 2, and 4 PUs and branches to path control in immediate for all SDLC stations. If the PIU is from a type 4

PU, BNN path control in immediate branches to INN path control. PIUs received over a local/local or local/remote link are immediately routed by INN path control in level 3.

If the PIU is from a type 1 or type 2 PUs and LUs, BNN path control in immediate enqueues the FID2 or FID3 to the common physical unit block (CUB) link inbound queue at CUB1ECB (offset X'00'). The CUB link inbound queue task triggered is BNN path control in delayed in level 5.

BNN path control in delayed converts the FID2 or FID3 to a FID1 and branches to one of three BNN connection point manager in (CPM-IN) routines.

PIUs destined for the SSCP from the physical unit are processed by the SSCP/PU CPM-IN. When the PIU is a response to an 'activate physical' or 'deactivate physical' the session status in the CUB is updated.

PIUs destined for the SSCP from the logical unit are processed by the SSCP/LU CPM-IN. When the PIU is a response to an 'activate logical', 'deactivate logical', or 'clear', the session status in the LUB is updated.

PIUs destined for the application program from the logical unit are processed by the APPL/LU CPM-IN. When the PIU is a response to an 'bind' or 'unbind' the session status in the LUB is updated. An FM data response PIU is checked for a 'pacing response' which initiates pacing processing and triggers BNN CPM-OUT. Pacing is covered later in this section in the topic Boundary Network Node (BNN) Pacing.

Figure 7.7 illustrates the BNN inbound flow for a type 1 or type 2 physical unit.

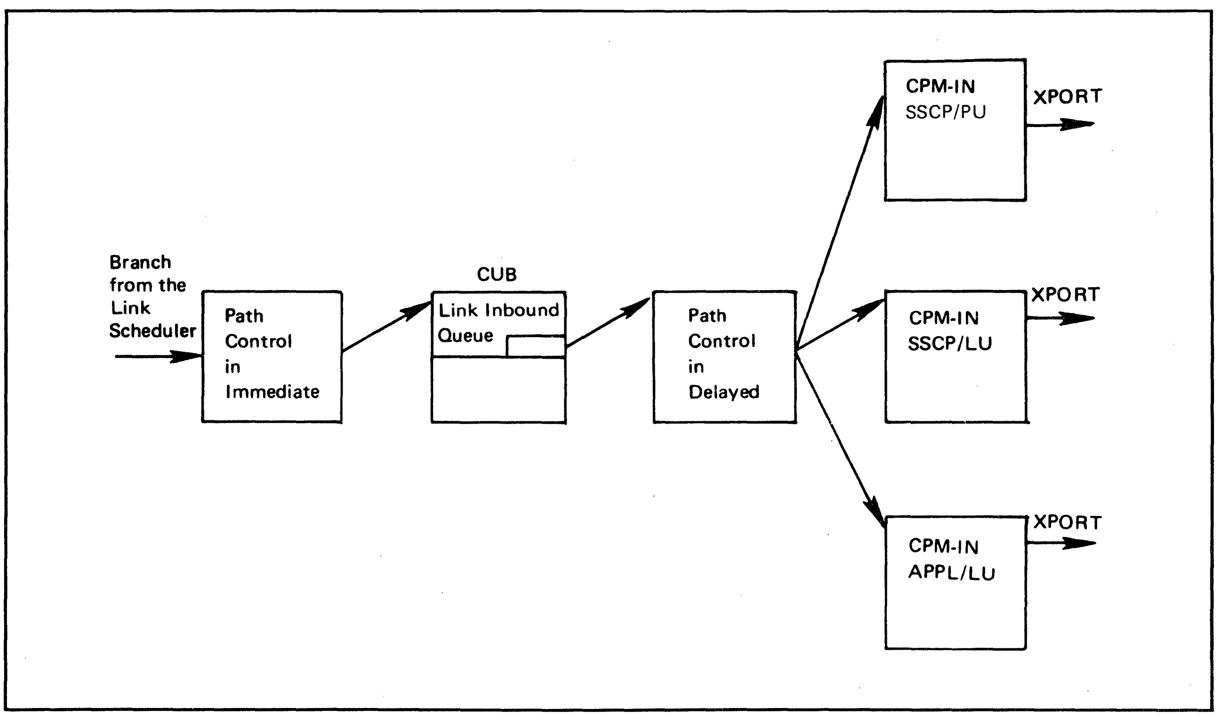


Figure 7.7. Boundary Network Node Inbound Path Flow

Path Control In Immediate

When a PIU has been received on a link the link scheduler branches to the level 3 BNN 'path control in immediate'. The PIU is validated for appropriate FID format and length.

A PIU from a type 4 physical unit must be a FID0 or FID1 and passed to INN path control.

A PIU from type 1 or type 2 PUs or LUs must be a FID2 or FID3 and queued on the common physical unit block (CUB) link inbound queue at CUB1ECB (offset X'00'). The ENQUE ACTV=YES triggers BNN 'path control in delayed' in level 5. BNN 'path control in delayed' is triggered to convert a FID2 or FID3 PIU to FID1 and schedule the correct BNN CPM-IN.

BNN Path Control In Delayed

Conversion from FID2 or FID3 to FID1 format occurs within the first NCP buffer of the PIU. When the response to the poll is received, the link scheduler leases a buffer and sets up the appropriate offset for the type of device polled. A type 4 physical unit sends and receives a FID0 or FID1 which requires an offset of X'1A'. A type 2 physical unit sends and receives a FID2 which requires an offset of X'1E'. A type 1 physical unit sends and receives FID3, which requires an offset of X'22'.

FID3 to FID2 Conversion

A FID3 is converted to a FID2. The conversion from a FID3 to a FID2 obtains most of the basic information to rebuild the FID2 from the FID3, however the sequence number field is in the logical unit block (LUB).

The buffer prefix offset and count are changed to reflect the expanion from the FID3 TH to FID2 TH. The FID3 TH3B0 is moved from X'22' to X'1E' with bits 3 and 4 changed from '11' (FID3 indicator) to '10' (FID2 indicator). TH2B1 is a reserved field.

The FID3 offset of X'23' contains the following information defining the session as follows:

- Bit 0 -- 1=to/from application, 0=to/from SSCP
- Bit 1 -- 1=to/from logical unit,0=to/from physical unit
- Bits 2 through 7 -- local address

The FID2 DAF at offset X'20' is set to X'00' if the FID3 DAF at offset X'23' bit 0 is 0, and set to X'01' if bit 0 is 1.

The FID3 local address from the rightmost six bits of offset X'23' are expanded with leftmost zeros to fill the FID2 OAF at offset X'21'.

The sequence number field must be obtained from the logical unit block (LUB) at offset X'50' through X'5D'. The CUB is known, as the device was selected from the service order table for polling. At CUB offset X'64' is the address of the logical unit vector table. The local address from the FID2 OAF at offset X'21' and the local address from the LUV entry at offset X'00' are compared. The matching local address in the LUV entry contains the fullword address of the LUB. The sequence number field is obtained from the LUB type 1 PU extension at offset X'50'.

If no matching entry is found (an undefined logical unit) the PIU is returned to the origin with a sense code.

FID2 to FID1 Conversion

The buffer prefix offset and count fields are changed to reflect the expansion of the FID2 TH to FID1 TH. The FID2 TH2B0 at offset X'1E' is moved to FID1 TH1B0 at offset X'1A'. The FID1 TH1B1 at offset X'1B' is a reserved field.

Some fields from the logical unit block (LUB) are required to to convert to the FID1 format. At CUB offset X'64' is the address of the logical unit vector table. The local address from the FID2 OAF at offset X'21' and the local address from the LUV entry at offset X'00' are compared. The matching local address in the LUV entry contains the fullword address of the LUB.

The destination address field of the FID2 identifies the FID2 as an SSCP (X'00') or application (nonzero). If the FID2 destination is the SSCP, the SSCP network address is in the CUB at offset X'62'. The SSCP network address is copies to the PIU at offset X'1C'. If the FID2 destination is an application the network address of the session partner is in the LUB at offset X'36'. This session partner network address is copied to the PIU at offset X'1C'.

The origin network address is in the LUB at offset X'1A'. This address is copies to the PIU at offset X'1E'.

The sequence field is move from the FID2 offset X'22' to the FID1 offset X'20'. As the PIU was received the received text count is maintained in the ECB at offset X'0E'. The total text count, minus the TH, is stored in the FID1 at offset X'22'.

There are three connection point manager in (CPM-IN) routines. BNN 'path control in delayed' determines which of the three CPM-IN routines to call, depending upon the session type (SSCP/PU, SSCP/LU, or APPL/LU).

SSCP/PU CPM-IN

SSCP/PU CPM-IN verifies that a session is established or pending by checking the CUB at offset X'5C'. A request PIU has a value of 0xxx xxxx in RH1B0 at X'24'. A request PIU with an established session is passed to INN path control.

A response PIU has a value of 1xxx xxxx in RH1B0 at X'24'. If the RH1B0 contains 111x xxxx, the response is to an 'activate physical' or 'deactivate physical'. A response requires that status be changed in the CUB at offset X'5C'.

A positive response to 'activate physical' turns on 'session established' and turns off the 'processing session initiation' bit in the CUB CUBPSTAT byte. A 'deactivate physical' response turns off the 'session established' and 'processing session termination request' bits of the same byte. A negative response requires the bit indicating that a command is in process be set to 0. The response is passed to INN path control.

SSCP/LU CPM-IN

SSCP/LU CPM-IN verifies that a session is established or pending by checking the LUB at offset X'3C'. A request PIU has a value of 0xxx xxxx in RH1B0 at X'24'. A request PIU with an established session is passed to INN path control.

A response PIU has a value of 1xxx xxxx in RH1B0 at X'24'. If the RH1B0 contains 111x xxxx, the response is to an 'activate logical', 'deactivate logical', or 'clear'. An 'activate logical' or 'deactivate logical' response requires that status be changed in the LUB at offset X'3C'.

A positive response to 'activate logical' turns on 'session established' and turns off the 'processing activate logical' bit in the LUB LUBCPSET byte. A 'deactivate logical' response turns off the 'session established' and 'processing deactivate logical' bits of the same byte. A negative response requires the bit indicating that a command is in process be set to 0. The response is passed to INN path control.

APPL/LU CPM-IN

APPL/LU CPM-IN verifies that a session is established or pending by checking the LUB at offset X'40'. A request PIU has a value of 0xxx xxxx in RH1B0 at X'18'. A request PIU with an established session is passed to INN path control.

A response PIU has a value of 1xxx xxxx in RH1B0 at X'24'. If the RH1B0 contains 111x xxxx, the response is to a 'bind', 'unbind', or other system control command. A 'bind' or 'unbind' response requires that status be changed in the LUB at offset X'40'.

A positive response to 'bind' turns on 'session established' and turns off the 'processing bind' bit in the LUB LUBAPSET byte. A positive 'unbind' response turns off the 'session established' and 'processing unbind' bits of the same byte. A negative response requires the bit indicating that a command is in process be set to 0. The response is passed to INN path control.

If the RH1B0 contains x00x 0xxx, the response is function management (FM) unformatted user data. The response PIU is checked for a pacing response in RH1B1 at X'25'. If the pace bit is 1, pacing processing occurs, and BNN CPM-OUT is triggered. If The response is passed to INN path control. Additional information on pacing is provided in the topic Boundary Network Node (BNN) Pacing.

Summary BNN Inbound Flow

Figure 7.7 illustrates BNN inbound flow of a PIU. There are three paths for inbound processing. All three paths are the same until BNN 'path control in delayed' enqueues the PIU to one of three connection point managers in (CPM-IN). The CPM-IN passes the PIU to INN path control.

BNN Path Control In Immediate

A PIU is passed from the link scheduler to BNN 'path control in immediate' at level 3. 'Path control in immediate' checks to see if the PIU is from a type 1, type 2 or type 4 PU. A PIU from a type 4 physical unit is checked for FID0 or FID1 and passed to INN path control. A PIU from a type 1 or type

2 physical unit is enqueued to the link inbound queue of the CUB which was polled, triggering BNN 'path control in delayed'.

BNN Path Control In Delayed

BNN 'path control in delayed' is a dispatched level 5 task triggered by the PIU enqueued from BNN 'path control in immediate'. The PIU is converted from FID2 or FID3 to FID1 and, depending upon session type, passed to SSCP/PU CPM-IN, SSCP/LU CPM-IN, or APPL/LU CPM-IN.

SSCP/PU CPM-IN

The SSCP/PU CPM-IN processes control responses to reflect correctly the session status of the SSCP/PU session, and passes the PIU to INN path control.

SSCP/LU CPM-IN

The SSCP/LU CPM-IN processes control responses to reflect correctly the status of the SSCP/LU session, and passes the PIU in INN path control.

APPL/LU CPM-IN

The APPL/LU CPM-IN processes control responses to reflect correctly the status of the APPL/LU session. Data PIU responses are checked for pacing responses from the device. Pacing processing occurs as required. The PIU is passed to INN path control.

Boundary Network Node Switched Support

The command sequence required for nonswitched boundary network node support was provided earlier in this section. The required nonswitched command sequence is:

- 'Activate physical' from SSCP to NCP physical services
- 'Activate link' from SSCP to NCP physical services identifying the link
- 'Contact' from SSCP to NCP physical services identifying the PU
- 'Contacted' from NCP physical services to SSCP identifying the PU
- 'Activate physical' from SSCP to PU
- 'Activate logical' from SSCP to LU
- 'Bind' from host application to LU

The command sequence required for switched support includes all of previous commands and additional commands. The added commands are destined for NCP physical services with the address of the resource to be affected identified in the RU at offset X'2A'.

The NCP generation of SDLC switched support includes defining a group of lines for dial out, dial in, or dial in/out operations. The macro instructions that define switched SDLC operations are GROUP, LINE, PU, and LUDRPOOL.

Note: ACF/VTAM release 1 and ACF/TCAM release 1 require the LUPOOL support.

The PU macro specifies the number of LUBs required during a connection by the operand of MAXLU. When a connection is made, the LUBs are obtained as required from the LUDRPOOL.

The switched SDLC CUB contains the address of the logical unit vector table (LUV) at offset X'64'. The LUV entries generate with an indicator of entry not in use at offset X'04'. The number of entries in the LUV is defined by the MAXLU operand of the PU macro.

The NCP provides three modes of operation for switched SDLC links:

1. Manual dial. The NCP enables the link and allows the operator to dial.
2. Autodial. The NCP enables the link and performs the dial operation using the dial digits provided with the command.
3. Answer. The NCP enables the link and allows the switched stations to call in. The link remains in answer mode until the SSCP terminates it. If the SSCP issues a dial command to the link, the answer mode is temporarily suspended until the dialed connection is broken.

When a switched connection is completed NCP sends an SDLC XID command with a X'FF' general poll address. The response provides the true SDLC address of the terminal. The response is sent to the host for identification.

The SSCP identifies the CUB from the XID information. The SSCP sends the defined values for the connected CUB in the 'set control vector' command (RU byte 5 X'03').

Logical unit control blocks (LUBs) are dynamically assigned to logical units when a switched connection is made. For this reason, a number of LUBs for switched lines must be allocated during NCP generation. Using the 'request network address assignment' (RNAA) command, the SSCP requests LUBs from the LUDRPOOL.

The NCP allocates LUBs from the pool, initializes the LUV table, and provides the network addresses of the assigned LUBs in the RNAA response. Each LUB must be initialized by a 'set control vector' command (RU byte 5 X'04'). When the SSCP breaks the connection, the SSCP issues a 'free network addresses' command to return the LUBs to the pool.

Additional commands from SSCP to NCP physical services provide the control of switched link support. The function management data indicator (x00x xxxx) in RH byte 0 and X'02' in RU byte 1 indicate a request to physical configuration services. The commands for the control of switched links include the following:

- X'0E' Connect out (dial). Causes the NCP to initiate an outbound call on the indicated switched SDLC link. For auto dial, the NCP performs the dial operation with the dial digits provided in the command. For manual dial, the NCP enables the link and the operator performs the dial operation.
- X'0F' Abandon connection. Causes the NCP to terminate a switched connection.
- X'11' Set control vector-LU:

RU, byte 5 = x'03'. Changes dynamic fields in the common physical unit block (CUB) which are associated with the specified physical unit.

RU, byte 5 = X'04'. Changes dynamic fields in the logical unit control block (LUB) and completes initialization of the logical unit vector (LUV) table.

- X'16' Connect in (answer). Causes the NCP to put the specified link in answer mode. Connect in enables the link to accept incoming calls.
- X'17' Abandon connect in. Causes the NCP to discontinue answer mode on the specified link.
- X'18' Abandon connect out (dial). Causes the NCP to halt the dialing operation over the specified link.
- X'410210' Request network address assignment. Requests the NCP to add a specified number of LUs to a specified PU and return the network addresses of the added elements.
- X'1A' Free network addresses. Causes the NCP to free the network addresses that were assigned to a physical unit.
- X'84' Request contact (off hook). Informs the SSCP that a physical connection has been established between the NCP and a physical unit. The PIU contains the station ID.

Appendix A provides the command sequence required to create the connections and sessions.

Figure 7.8 illustrates the command sequence and SDLC sequences of a switched connection.

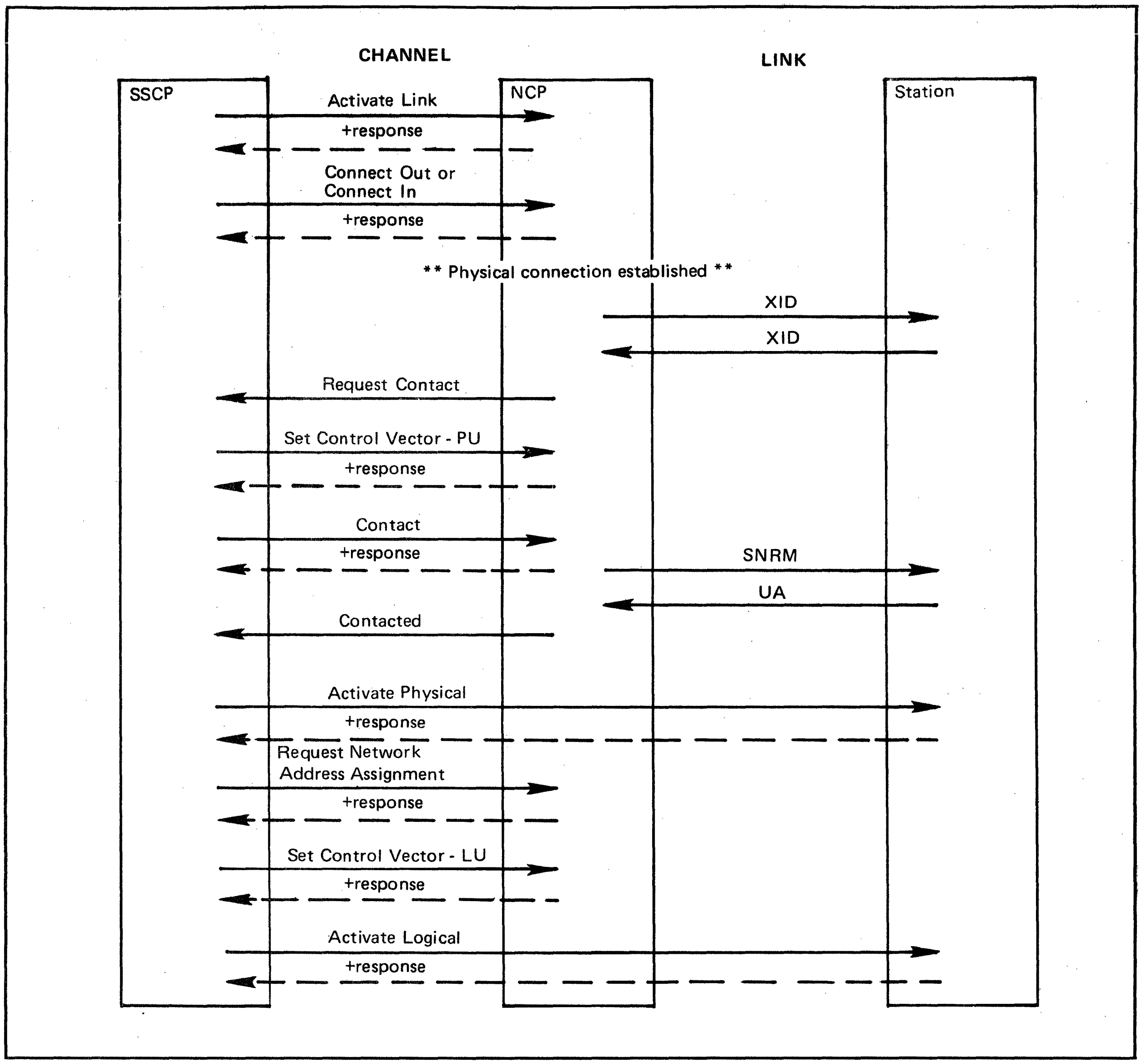


Figure 7.8. SNA and SDLC Switched Command Sequence

The switched SDLC link connection is broken by the following sequence of commands to terminate the connection:

1. The SSCP issues a 'deactivate logical' command for each of the logical units. This command terminates the SSCP/LU session.

2. The SSCP issues a 'free network addresses' command to release the assigned LUBs and return them to the LUDR pool.

3. A 'deactivate physical' command terminates the session between the SSCP and the physical unit. If the physical unit is a type 1 device, the NCP does not transmit the command to the device, but responds to the 'deactivate physical' command. Type 2 devices receive the command and reply.

4. The 'discontact' command causes the NCP to send a 'disconnect' (DISC) SDLC command to the station. The station replies with an UA and disconnects from it's modem.

5. The 'abandon connection' command causes the NCP to disable the link and return it to 'on hook' status. If the link was previously in connect in (answer) mode, the NCP reenables the link.

Figure 7.9 illustrates the network commands and SDLC sequences for breaking a switched SDLC connection.

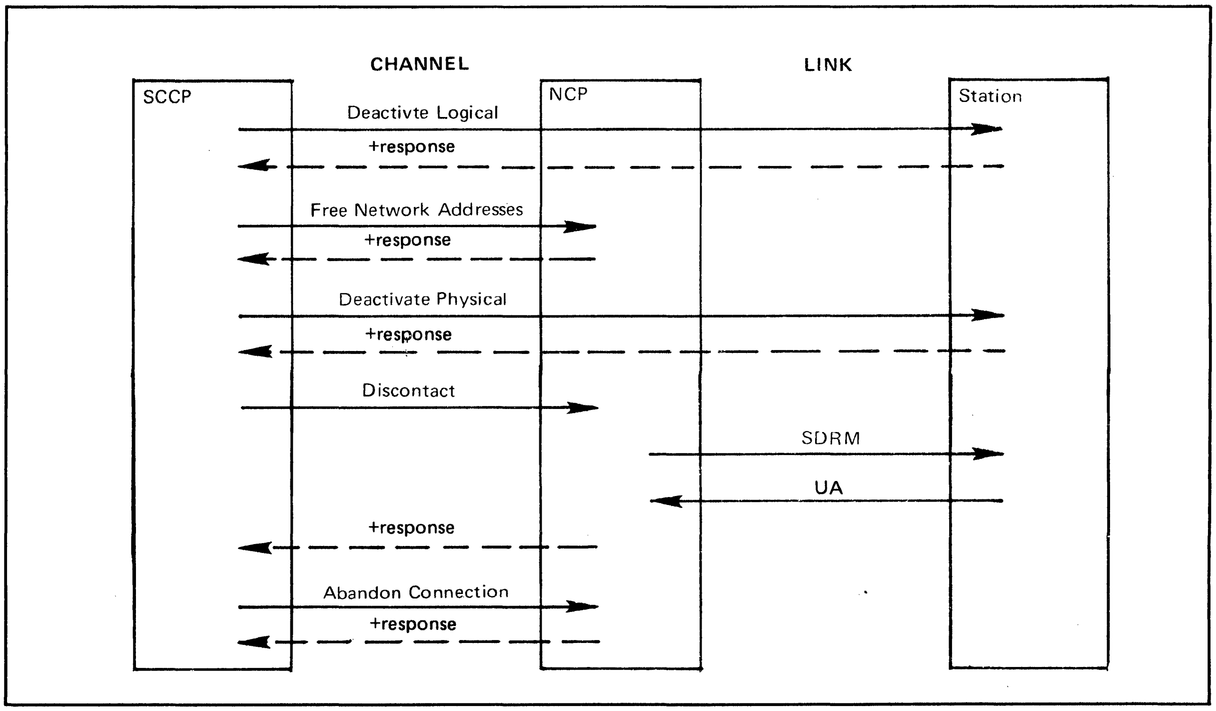


Figure 7.9. SNA and SDLC Commands to Terminate Switched Connections

Boundary Network Node (BNN) Pacing

Introduction

The boundary network node supports inbound and outbound pacing. The boundary network node supports *outbound* pacing (1) between the host application and BNN APPL/LU CPM-OUT, and (2) between the BNN APPL/LU CPM-OUT and the logical unit. The boundary network node supports *inbound* pacing between the logical unit and the host application by giving priority to pacing responses in APPL/LU CPM-OUT.

Pacing or the lack of pacing can have a significant effect on the performance of the network. Pacing occurs in an APPL/LU session only.

Pacing definitions may be provided on a session basis in the bind command. A nonzero value in the bind pacing fields provides the pacing value for duration of the session. A zero value in the bind defaults to the operand coded values for outbound pacing. Inbound pacing is defined only in the bind.

The bind parameters for pacing are located at:

X'2F' (RU byte 8) - logical unit/host

X'30' (Ru byte 9) - boundary node/logical unit

X'33' (RU byte C) - host/boundary node

Pacing values defined in operands are PACING for boundary node to logical unit, VPACING for VTAM host to boundary node, and OPACING for TCAM host to boundary node. These values are used if the bind parameter definition is zero.

A pacing request is indicated by RH byte 1, bit 7, with a value of '1' (offset X'25) in a request PIU. A pacing response is indicated by RH byte 1, bit 7, with a value of '1' (offset X'25) in a response PIU. The same bit is used for each type of pacing. It is necessary to identify the direction of flow (to or from the host) and current location of the PIU (host/boundary node or boundary node/logical unit) to determine which pacing flow is involved.

Pacing uses a request PIU for pacing requests. Pacing responses may 'piggy back' on a regular response, however pacing responses are normally an 'isolated pacing response'. An 'isolated pacing response' has the pace bit on, definite response 1 off, definite response 2 off, and execption response off. The isolated pacing response provides additional traffic in the network.

If the session is interactive or definite response, pacing is not needed. Pacing provides controls for batches of data which exceed the buffer requirements at the destination of the pacing route. If pacing is not needed you should code the pacing operands for zero pacing and code the bind for zero pacing. If a logical unit requires pacing for some sessions, code the pacing operands with zero and provide pacing controls on a session basis.

If the application does not control PIU traffic (interactive or definite response) and if pacing is not defined, the PIUs are processed as they are received. Uncontrolled traffic from the host to NCP would soon deplete NCP buffers and create a buffer slowdown condition. Uncontrolled traffic from NCP to an LU would overrun LU buffers. Uncontrolled inbound traffic from an LU to an application may overrun NCP as well as host buffer allocations.

One logical unit without pacing can tie up buffers in a host, NCP or physical units while the remaining logical units are waiting for PIUs for a lack of buffers.

The PACING operand allows two suboperands of N and M to be coded. A pacing operand of (N,M) specifies that N PIUs are to be sent to the logical unit before waiting for a pacing response. The M value defines which of the N PIUs carries the request for a pacing response. VPACING and OPACING allow the N value to be defined and provide a fixed value of M=1 for the second operand.

The PIU RH1B1 field at offset X'25' bit 7 is the pace bit. The pace bit is used for pacing requests and pacing responses, inbound and outbound, between APPL/BNN, BNN/LU, and LU/APPL.

The pace bit 'on' in a request is a 'request for pacing response'. The pace bit 'on' in a response is a 'response to a pacing request'. The pace bit applies to pacing between the components within the flow of the PIU (APPL/BNN, BNN/LU, and LU/APPL).

- Outbound request PIU in APPL/BNN is an APPL/BNN pacing request.
- Inbound response PIU in APPL/BNN is an APPL/BNN pacing response.
- Outbound request PIU in BNN/LU is a BNN/LU pacing request.
- Inbound response PIU in BNN/LU is a BNN/LU pacing response.
- Inbound request PIU in LU/APPL is an LU/APPL pacing request.
- Outbound response PIU in LU/APPL is an LU/APPL pacing response.

A pacing request is always indicated in a data PIU. A pacing response may be indicated in a response to a data PIU or in an 'isolated pacing response' (IPR). An IPR is a response PIU with the pace bit 'on' and all response bits 'off' (DR1, DR2, and exception bits).

Inbound pacing responses are always IPRs. Outbound pacing responses in host/boundary node flow are IPRs when (1) the PIU containing the pacing request does not request a definite response, or (2) no additional PIUs are queued for this session. Outbound pacing responses in boundary node/logical unit are terminal dependent.

Figure 7.10 illustrates the three areas of pacing. Note that VTAM VPACING and TCAM OPACING define pacing between a primary logical unit and (1) a secondary logical unit in a host or (2) the NCP boundary network node. NCP PACING defines pacing between NCP boundary network node and an SDLC logical unit. The BIND parameter defines inbound pacing, and may be used to override PACING and VPACING/OPACING. The arrows indicate the direction of a pacing *request.* The pacing *response* flow is in the reverse direction.

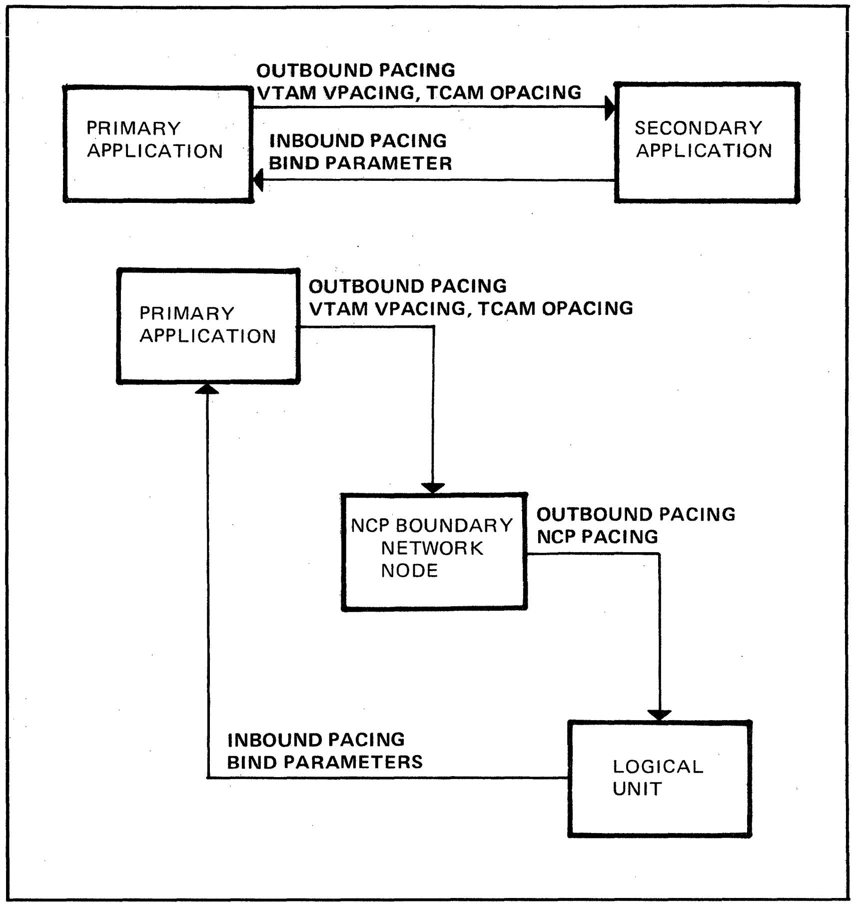


Figure 7.10 Pacing Flow

The following topics provide an introduction to the APPL/BNN, BNN/LU, and LU/APPL pacing flows.

Host/BNN

Pacing between a host application and boundary network node (host/BNN) is defined to the host. Host/BNN pacing is defined to TCAM in the OPACING operand and to VTAM in the VPACING operand.

Request PIUs (data) are sent for 'n' pacing quantity. The 'm' PIU has the pace bit 'on'.

The request PIUs are processed by APPL/LU CPM-OUT. When the request PIU with the pace bit 'on' is processed a pacing response is scheduled or sent. If the PIU does not request a definite response, an isolated pacing response (IPR) is created and sent immediately. If the PIU requests a definite response (DR1 or DR2) and there is an additional PIU queued for this LU/LU session, a 'pace required by host' is set in the logical unit block (LUB); the next response PIU to arrive within the session carries the pacing response. In either case, the pace bit is set to 0. When the last PIU on the queue is processed, if the 'pace required by host' bit is on, an IPR is sent to the host, and the 'pace required by host' bit is set off.

If CPM-OUT is triggered and no PIUs are on the CPM-OUT queue, LUB offset X'48' is checked for 'pace required by host'. If pace is not required by the host CPM-OUT exits. If pace is required by the host, a buffer is leased to build an isolated pacing response (IPR) and the 'pace required by host' is turned off. The IPR is passed to INN path control.

When CPM-OUT is dispatched the logical unit block (LUB) is checked at offset X'48' for 'awaiting pace from LU'. The request PIUs in the APPL/BNN flow are not processed if CPM-OUT is 'awaiting pace from LU'.

A IPR response PIU, response to inbound pacing, is processed and placed on the link outbound queue even when 'awaiting pace from LU'. This avoids an interlock where the logical unit is waiting for an inbound pacing response and NCP is waiting for BNN/LU pacing response.

BNN/LU

Pacing between boundary network node and the logical unit (BNN/LU) is defined to the NCP. BNN/LU pacing is defined on the LU macro operand of PACING. The bind command provides alternate pacing values for the duration of the APPL/LU session; a nonzero value in the RU at offset X'09' provides the 'n' value with 'm' defaulting to X'01'.

Request PIUs (data) are sent for 'n' pacing quantity. The 'm' PIU has the pace bit 'on'.

APPL/LU CPM-OUT maintains a 'n' and 'm' limits, a 'pacing counter' of PIUs sent, 'awaiting pace from LU', and 'received pace response from LU'. Request PIUs (data) are sent for 'n' pacing quantity. The 'm' PIU has the pace bit 'on'.

When 'n' PIUs are sent in the BNN/LU flow CPM-OUT checks the 'received pace response from LU' bit.

If the 'received pace response from LU' bit is off, CPM-OUT turns 'on' the 'awaiting pace from LU' and suspends processing of request PIUs until a BNN/LU pace response is received. The suspended processing holds request PIUs in the APPL/BNN flow and controls the APPL/BNN pacing.

If the 'received pace response from LU' bit in on, the response to the 'm' PIU was received before the 'n' PIU was sent. The pacing response allows the 'pacing counter' and 'awaiting pace from LU' bits to be set to '0' and a new group of 'n' PIUs to be transferred. The pacing response allows the 'pacing counter' to be reset.

Inbound responses from the LU are received by CPM-IN in the BNN/LU flow. The inbound response with the pace bit 'on' sets the 'received pace response from LU' bit 'on' in the LUB and triggers CPM-OUT for processing more outbound requests.

If the response is an isolated pacing response (IPR) the buffer is returned to the NCP buffer pool. If the response PIU continues to the host, the pace bit in the PIU is set 'off'. BNN/LU pacing flow is complete.

CPM-IN begins APPL/BNN pacing response checks. If the 'pace required by host' bit is 'on' a pacing response must be sent to the host. The pace bit in the response PIU is set 'on' and the 'pace required by host' bit is set 'off'. The response PIU is passed to INN path control.

Note: In most cases only IPRs flow to the host in the host/BNN flow. To 'piggy back' pacing responses (1) the pacing bit must be on in a request PIU with a definite response bit on, and (2) the BNN CMP-OUT queue must have continuous queued PIUs until an inbound definite response is processed.

LU/APPL

Pacing between a logical unit and a host application (LU/APPL) is defined in the host. The 'n' value is sent in a 'bind' command to the logical unit of a secondary logical unit. The 'n' value is received and processed by BNN APPL/LU CPM-OUT for a logical unit of a type 1 physical unit. The 'm' value is assumed to be '1'.

In a LU/APPL flow pacing requests are carried from the LU in request PIUs; pacing responses are carried from the APPL in response PIUs.

A LU/APPL pacing request PIU is received by BNN CPM-IN and passed unchanged to INN path control. LU/APPL pacing requests are not processed in NCP.

A LU/APPL pacing response PIU is always an isolated pacing response (IPR). An IPR provides a pacing response and is not a response to a PIU request. A LU/APPL pacing response PIU is queued on BNN CPM-OUT as the first PIU on the queue.

BNN CPM-OUT may be 'awaiting pace from LU'; however, waiting does not apply to IPR pacing responses in the LU/APPL flow. Both BNN/LU and LU/APPL flows can be waiting for a pacing response. The LU/APPL response PIU is sent to avoid an interlock.

The NCP only participates in LU/APPL inbound pacing by allowing outbound isolated pacing responses (IPR) to be sent to a LU when all other traffic is held for a BNN/LU pacing response.

BNN Pacing Control Fields

PIU Pacing Indicator

Pacing control uses RH byte 1 bit 7 in a request PIU as a pacing request indicator. The RH byte 1 bit 7 in a response PIU is used as a pacing response.

Pacing responses are included in a data response PIU or as an isolated pacing response (IPR). Use of an IPR is implementation dependent.

Pacing Fields in the Logical Unit Block (LUB)

The control fields in the logical unit block (LUB) are:

- X'42' LUBM -- BNN/LU pacing 'm' value for this session
- X'43' LUBN -- BNN/LU pacing 'n' value for this session
- X'44' LUBMG -- LU/APPL PACING 'm' value (generated)
- X'45' LUBNG -- LU/APPL PACING 'n' value (generated)

- X'46' LUBPC -- BNN/LU pacing counter
- X'48' LUBPS -- Pacing flags for APPL/BNN, BNN/LU, and LU/APPL flows

```
1xxx xxxx  Pace required by host
x1xx xxxx  Send pace to host early
xx1x xxxx  Pace sent to host early
xxxx 1xxx  Awaiting pace from LU
xxxx x1xx  Received pace response from LU
xxxx xxx1  I pacing response on LU/APPL queue
```

Pacing Flow

Outbound Flow

The following flow follows a group of PIUs from a host application to a logical unit. The flow checks for request and response PIUs with pacing indicators.

The host application sends 'n' PIUs to NCP with the pace bit (RH byte 1 bit 7) 'on' in the 'm' PIU. No additional PIU requests or responses flow from the host until a pacing response is received in the host *except* isolated pacing responses (IPRs) in the LU/APPL flow.

If BNN/LU pacing is not defined the LUBN field has a value of X'00'; CPM-OUT never waits for 'awaiting pace from LU'. If pacing is not defined, CPM-OUT processes and sends PIUs as rapidly as CPM-OUT is dispatched. CPM-OUT is not dispatched again until the PIU is scheduled on the link by the data link control (DLC) manager.

If LUBN has a nonzero value, the following pacing processing occurs:

1. CPM-OUT checks the CPM-OUT queue for a PIU to process. If there are no PIUs to be processed (CPM-OUT triggered by CPM-IN by a BNN/LU pacing response), CPM-OUT exits.
2. CPM-OUT locates the first PIU on the CPM-OUT queue and checks for a response to inbound pacing, an 'isolated pacing response' (IPR); pacing (RH byte 1 bit 7 of '1'), response (TH byte 0 bit 0 of '1'). An IPR (LU/APPL pacing response) is queued to the CUB link outbound queue for transmission.
3. If the PIU is not an IPR, the LUB is checked at LUBPS for an 'awaiting pace from LU'. If a wait is indicated, the PIU remains in the APPL/LU queue on the LUB and CPM-OUT EXITs. CPM-OUT is triggered again by (1) CPM-IN when a BNN/LU pacing response is received from the LU, or (2) the next PIU arriving on the CPM-OUT queue (which may be an IPR for step 1).
4. If step 2 did not suspend processing, the PIU is checked for a pacing request; pacing (RH byte 1 bit 7 of '1'), request (RH byte 0 bit 0 of '0'. A request PIU with the pace bit 'on' indicates an APPL/BNN request for pacing response.

If the PIU requests a definite DR1 or DR2 response bit 0 in LUBPS is set to '1' (pace required by host). If a definite response to the PIU is not required CPM-OUT leases a buffer and builds an isolated pacing response (IPR) to be sent to the origin address field. The IPR 'pacing response' is passed to INN path control. The pace bit in the outbound PIU RH byte 1 is set to '0'. The RH pace bit is cleared before BNN/LU pacing processing begins.

5. The PIU is removed from the CPM-OUT queue for processing. The LUBPC (LUB offset X'46') pacing counter is incremented by 1 and compared to LUBM. If LUBPC is equal to LUBM, the current PIU carries the BNN/LU pacing request. The pace bit (RH1B1 bit 7) is set to '1'. The shifted address of the CPM-OUT QCB is stored in the first buffer of the PIU at offset X'10'. The PIU is queued to the CUB link outbound queue for transmission.

6. CPM-OUT checks the CPM-OUT queue for additional PIUs. If the queue is empty, and if the 'pace required by host' bit is 'on', an IPR is sent, and 'pace required by host' bit is turned 'off'.

7. The pacing counter, LUBPC, is compared to LUBN. If fields are not equal, CPM-OUT exits. If the LUBPC and LUBN fields are equal, the LUBPS 'received pace response from LU' bit is checked for a value of '1'. If the pacing response has been received the LUBPC pacing counter is reset to zero and a new group of 'n' PIUs may be sent.

If the LUBPC and LUBN fields are equal, and the LUBPS 'received pace response from LU' bit is '0', a pace response has not been received. The LUBPS 'awaiting pace from LU' bit is set on. No additional PIUs are sent in the BNN/LU flow until a pacing response from the LU is received except isolated pacing responses (IPRs).

8. CPM-OUT exits without a QPOST leaving the CPM-OUT task in the 'disconnect' state. When the data link control (DLC) manager schedules the PIU on the link, the shifted address of the CPM-OUT QCB is obtained from the buffer at offset X'10'. DLC unconditionally triggers CPM-OUT.

Inbound Flow

The following flow follows a group of PIUs from a logical unit to a host application. The flow checks for request and response PIUs with pacing indicators.

The logical unit sends 'n' PIUs to the host application with the pace bit (RH byte 1 bit 7) 'on' in the first PIU. The 'n' value is coded to the host and provided to the logical unit in RU byte eight of the 'Bind' command.

If LU/APPL pacing is not defined the Bind command contains X'00' and PIUs are sent to the host application as they are available.

If LU/APPL pacing has a nonzero value, the following pacing processing occurs:

1. The logical unit sends 'n' request PIUs. The first of the 'n' PIUs has the pace bit 'on' (RH byte 1 bit 7). The logical unit waits for a pacing response.

2. The 'n' PIUs are received and processed by BNN CPM-IN. Inbound request PIUs (LU/APPL) are not checked for pacing. Inbound pacing

requests are passed unchanged from BNN CPM-IN to INN path control.

Inbound response PIUs (BNN/LU) are checked for pacing. A BNN/LU pacing response sets the logical unit block (LUB) offset X'48' bit 5 'on', 'received pace response from LU'. BNN APPL/LU CPM-OUT is triggered.

3. When a BNN/LU pacing response is received CPM-IN checks 'pace required by host'. If 'pace required by host' is '1', a pace response is sent to the host, and the 'pace required by host' is set to '0'.

4. When SSCP passes a request PIU to the host application the pace bit in the RU is checked. If the pace bit is 'on' SSCP builds a LU/APPL isolated pacing response (IPR) to send to the logical unit.

Pacing Summary

Pacing control fields are in the PIU RU byte 1 and logical unit control block (LUB) bytes X'42' through X'48'.

A request PIU with a pace bit 'on' is a pacing request. A response PIU or isolated pacing response (IPR) with a pace bit 'on' is a pacing response.

An outbound request PIU with a pace bit 'on' flowing between the host and BNN CPM-OUT is in the APPL/BNN flow. An inbound response PIU with a pace bit 'on' flowing between the BNN CPM-IN and the host is in the APPL/BNN flow.

An outbound request PIU with a pace bit 'on' flowing between the BNN CPM-OUT and the logical unit is in the BNN/LU flow. An inbound response PIU with a pace bit 'on' flowing between the logical unit and BNN CPM-IN is in the BNN/LU flow.

An inbound request PIU with a pace bit 'on' flowing between the logical unit and the host application is in the LU/APPL flow. An outbound response PIU with a pace bit 'on' flowing between the host application program and the logical unit is in the LU/APPL flow.

The NCP supports pacing in the boundary network node in APPL/LU CPM-OUT and APPL/LU CPM-IN. When outbound pacing limits are reached CPM-OUT suspends processing of outbound request PIUs until CPM-IN records 'received pace response from LU' and CPM-IN triggers CPM-OUT. CPM-IN processes outbound pacing responses (LU/APPL) as they arrive.

Boundary Network Node (BNN) Summary

All PIUs in a session involving an SDLC link are processed by BNN routines. These routines handle session control requests and responses, and convert PIUs to the required format. Data flow requests and responses in an application/logical unit session are processed by pacing routines. On output to a physical unit, the PIUs are segmented as required by the PU macro MAXDATA operand. The NCP performs sequence-number processing on PIUs to and from type 1 physical units.

The user definition of pacing is vital to system performance. Interactive and definite response traffic should define pacing operands and bind pacing parameters as zero to avoid unnecessary pacing processing and IPR responses. If pacing values change by session the bind parameters override the generated values.

If uncontrolled PIU traffic may flow, host/BNN pacing is the basis between the host and the NCP to avoid buffer depletion. BNN/LU pacing schedules PIUs on a logical unit basis between the NCP and the physical unit to avoid depleting physical unit buffers and having one logical unit lock out other logical units. LU/APPL pacing schedules PIUs between a secondary logical unit and host application.

Segmenting breaks up PIUs when the length of a PIU exceeds MAXDATA. A first segment contains the TH, RH, and RU to full NCP buffers of equal to or less than MAXDATA size. A nonfirst segment is TH (copied from the first segment into a separate buffer) plus one full NCP buffer or a multiple of buffers of less than or equal size of MAXDATA. If PIUs of more than MAXDATA length will be received, the NCP buffer size should be selected to provide an efficient segment size.

Figure 7.11 illustrates the flow of PIUs through BNN for CUBs and LUBs. The numbered text that follows identifies the components and processing:

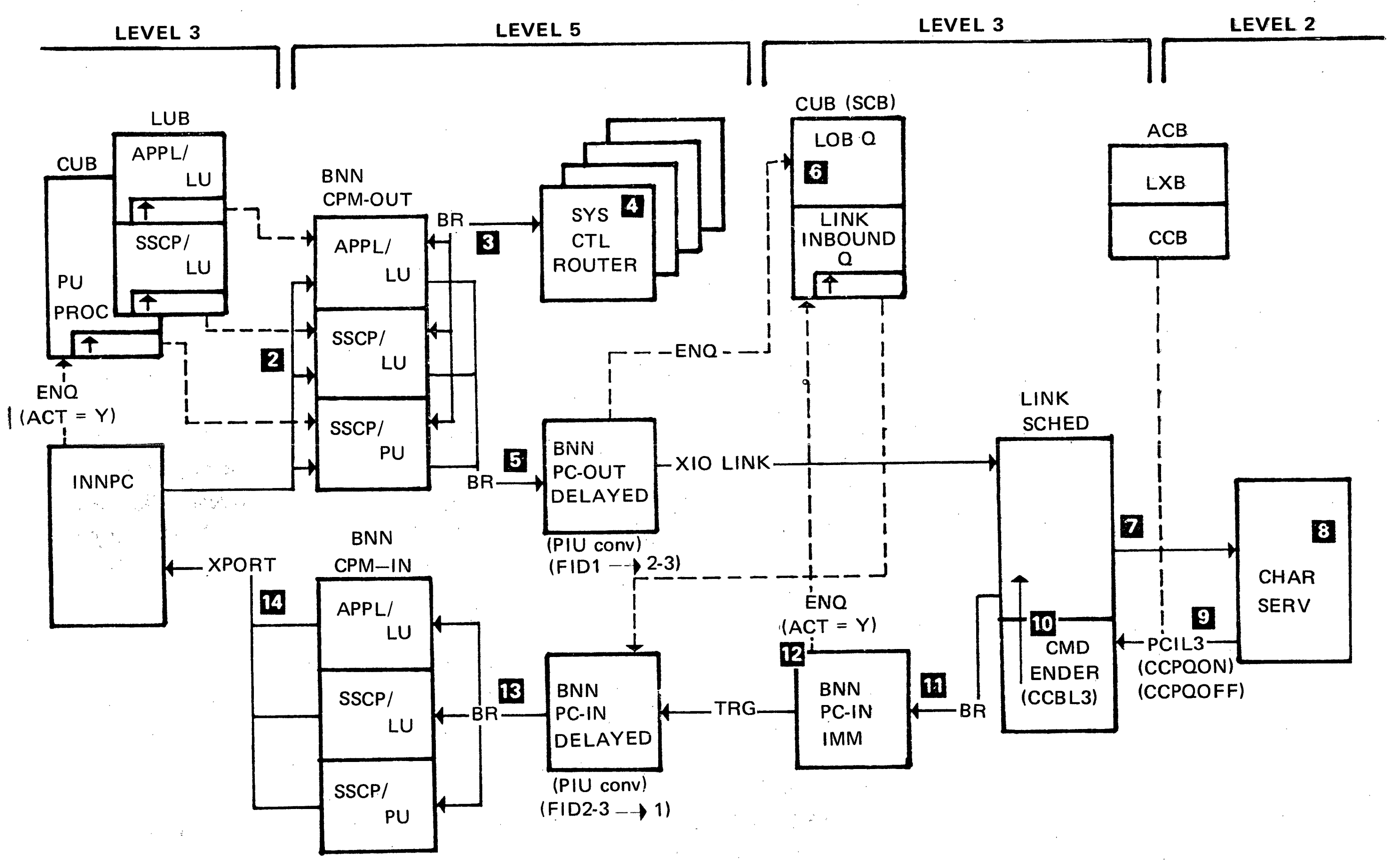


Figure 7.11. BNN CUB and LUB PIU Processing

1. Channel IOS branches to 'path control out'. Using the DAF to access the SIT, SVT, and RVT, path control enqueues the PIU to a CUB or LUB.
2. The enqueuing triggers the BNN CPM-OUT.
3. If the PIU is a session control request, the system control router gets control via a branch.
4. The system control router selects the proper subroutine and returns.
5. BNN CPM-OUT processes the PIU and calls 'BNN path control out'.
6. The PIU (FID2) is placed on the link outbound queue (LOB) and XIO is issued to the link.
7. The link scheduler locates the PIU on the LOB, then sets up the CCBL2 and ICW.
8. CSP handles the 'transmit' or 'receive'.
9. When level 2 ends, level 2 sets up CCPQON/OFF and issues a PCI to level 3 to return.
10. The 'command ender' routine uses CCBL3 to continue level 3 link scheduler processing.
11. The link scheduler branches to 'BNN path control in immediate'.
12. 'Path control in immediate' enqueues the PIU to the link inbound queue on the CUB.
13. 'Path control in delayed' selects the proper CPM-IN, using the FID1 or FID2 origin to locate the CUB or LUB queue.
14. The CPM-IN processes the PIU and branches to INN path control to be routed to the destination.

Chapter 8:

Data Link Control

Introduction

The routines which control SDLC links execute in levels 2, 3 and 5. After the NCP is loaded no link routines are active. The activate link command is sent to NCP physical services with the network address of the link to be activate in the PIU at offset X'2A'. The level 5 task of the link control block (LKB) is triggered to begin level 3 link scheduler operation for that link.

Level 5 schedules level 3 for the link to be activated via PCI and providing level 3 with the address of the ACB. No device on the link is available for work, therefore the adapter control block (ACB) is placed on a timer queue for 2.2 seconds. Each time the 2.2 second expires, the devices defined for that link are searched for an 'operational' physical unit (CUB offset X'2E'). When any device on the link is operational the link scheduler initiates the interface control word (ICW).

Level 2 is never initiated by PCI. Level 2 interrupts occur when an interface control word (ICW) requires service. Level 3 link scheduler searches for a physical unit for a transmit or receive operation. The link scheduler initializes the ICW primary control field (PCF) and secondary control field (SCF) for a transmit or receive. The adapter control block (ACB) is initialized with the address of a level 2 routine to be executed on the next level 2 interrupt. The ACB is also initialized with the address of a level 3 routine to be executed upon completion of the transmit or receive. Finally, the ICW is enabled for an interrupt.

If the ICW is set up for a transmission, the SDLC flag is sent and a level 2 interrupt is scheduled to obtain the physical unit SDLC address and place it in the parallel data field (PDF). If the ICW is set up for a receive the ICW monitors for an SDLC flag and no interrupt occurs until a flag is received.

The interface control word (ICW) is initialized for sending or receiving and enabled for level 2 interrupts. The ACB is placed on a timer queue as defined on the GROUP macro REPLYTO operand. If a level 2 interrupt does not occur before the REPLYTO expires the timer routine places the ACB on the XDHCCPQ queue at X'736' and PCIs to level 3. If a level 2 interrupt occurs, ICW service is provided by level 2 routines until an end of SDLC frame or error. On an end of frame or error level 2 places the ACB on the XDHCCPQ queue at X'736' and PCIs to level 3.

A level 3 interrupt for link service dequeues the ACB from X'736'. The ACB is checked for errors. If an error has occurred, the link control block (LKB) task is triggered, and level 3 exits without rescheduling the link scheduler. If level 3 does not find an error the link scheduler searches the physical units for the next PIU to send or physical unit to poll.

Note: For additional information see

ACF/NCP and Related Host Traces (SR20-4510)

ACF/NCP/VS Network Control Program Logic (LY30-3041), Appendix D, Command Sequence Charts, Command Sequence for SDLC Links.

Note the labels of CCBL2 and CCBL3 routines for SDLC transmit and receive.

The link scheduler has two modes of operation: normal service and control service or contact poll service. Normal service is transmitting PIUs or polling for PIUs. When the link scheduler (1) completes a pass of the service order table without PIUs to send and negative responses to polling or (2) SERVLIM passes through the service order table, normal service is suspended. Control or contact poll service searches for a pending 'contact' (SDLC SNRM), 'discontact' (SDLC DISC), etc., and attempts one command. If a timeout occurs, the link scheduler resumes normal polling. If a response to contact polling occurs the link control block (LKB) task is triggered, and level 3 exits without rescheduling the link scheduler. A PIU is sent to the host indicating the contact poll command has completed. The link scheduler is automatically restarted from the LKB task.

The link scheduler is initiated for a link with an activate link command. The link scheduler is suspended for level 2 link service or for a timer queue. The link scheduler is reactivated by a level 2 PCI to level 3 for service or by a timer interrupt.

Link Performance Flow Control

Link performance begins with scheduling PIUs on a link outbound queue of a CUB or SCB. The state of the APPL/LU CPM-OUT task and pacing parameters control the quantity of PIUs (not PIU segments) for each logical unit. Once the PIUs are on a link outbound queue link performance depends upon PU, LINE, GROUP and SERVICE macro operands.

The GROUP macro REPLYTO operand applies to polling and addressing for control commands (such as contact) as well as text. The LINE macro operands of REDIAL and RETRIES may retry transmissions for excessive time periods degrading other units on the link.

The link scheduler services non-switched physical units on a link in the sequence the physical units are defined in a service order table. Each non-switched primary SDLC link must have a SERVICE macro which identifies all of the physical units on that link.

The sequence of the service order table controls the link scheduler servicing sequence. If each physical unit is defined once in the table, each physical unit is searched for service once per link scheduler pass of the service order table.

If one or more physical units are to receive a priority at the link scheduler level, those physical units can be repeated within the service order table. If four physical units are on a link with labels of PU1, PU2, PU3 and PU4; and if PU1 is to receive priority, one method of coding the SERVICE macro is:

```
SERVICE ORDER=(PU1,PU2,PU1,PU3,PU1,PU4)
```

Physical unit PU1 is searched for service by the link scheduler after each of the other physical units are serviced.

Dynamic reconfiguration adds a physical unit and places the address in the service order table at the end of current SOT entries. When physical units are deleted all SOT entries for that device are removed from the SOT.

Polling cycle

The 'polling cycle' extends from the moment service begins with the first entry in the service order table to the moment service returns to the first entry; the service order table is treated as a wrap-table.

A minimum time for a polling cycle is coded in the PAUSE operand of the LINE macro. Before beginning a polling cycle, a pause time value is set up. If the time value elapsed during the polling cycle, a new polling cycle is started immediately. If the time value has not elapsed during the polling cycle, the line enters the 'poll-wait' state until the time value expires. Allowing a pause to elapse when activity on the link is relatively low can reduce the amount of processing time consumed by unproductive polling. The PAUSE value must be selected based upon the number of devices in the service order table.

The delay is only for polling. If a PIU is received to be transmitted to the PU, the scheduler is immediately triggered to send the PIU.

Link Normal Service

The operation of an SDLC link at the primary (polling) end consists of two operations; (1) normal service, and (2) control service. Normal service consists of one or more polling cycles (passes of the service order table); the number of polling cycles is (1) selected in the LINE macro SERVLIM operand or (2) a polling cycle without data transfer ends normal service. Control service consists of searching the service order table for a 'contact poll' command. The combined sequence of 'normal service' and 'control service' is called the 'service cycle'.

The normal service consists of searching the service order table (SERVICE macro) for physical units enabled for sending and receiving data. The service order table is used to locate physical units for data to transmit or poll to receive.

On a full-duplex link the scheduling of the receive leg (transmission of polling charcters) has priority over transmssion of data. On a half duplex link data transmission has priority over receiving. Once the link scheduler has selected a physical unit for service, PIUs are sent until one of the following conditions occurs:

- No PIUs remain on the link outbound queue of this physical unit. The link scheduler searches for the next physical unit in the service order table.
- The physical unit MAXOUT quantity of frames have been sent and polling for the SDLC response is required. The physical unit is polled for an SDLC response before continuing with PIU transmission.
- The physical unit PASSLIM quantity of frames have been sent. The link scheduler searches for the next physical unit in the service order table.

Note: The IBM 3271 type 1 physical unit require all segments of a PIU to be sent before any data transfer to or from the physical unit occurs for other logical units on the physical unit. Segmented full screen processing with an

NCP buffer size of 134 may require PASSLIM=14 to transmit all segments to a single screen.

Normal processing continues from the first entry to the last entry in the service order table. Before returning to the first entry, the polling cycle delay (PAUSE) must expire. If the number of polling cycles has reached the SERVLIM value, normal service is suspended, and the control service is initiated.

A full duplex link operation requires the receive link to be scheduled by sending a poll over the transmit link. Therefore, when the transmit link becomes available (completes a transmission) the receive link is checked. If the receive link is free (the last polling operation or pause is complete) poll characters are transmitted, scheduling the receive link. If the receive link is scheduled (receiving, waiting for a polling response, or in a polling pause) PIUs are sent on the transmit link.

A full duplex terminal may receive and transmit at the same time. A half duplex terminal may not receive and transmit at the same time. On a full duplex link if the polling pointer and transmit pointer both point to a half duplex terminal, full duplex operation is suspended until completion of the current operation.

Link Control Service

The control service consists of searching for a physical unit with a pending contact poll (contact, discontact, or other contact poll command). The search for a physical unit with a contact poll command begins with the first entry in the service order table following the previous physical unit found with a contact poll command (CCB offset X'64'). The search continues until (1) a physical unit is located with a pending command, or (2) the service order table is wrapped without locating a pending command.

If a pending command is found, the contact command is attempted. A successful completion or timeout ends the control service. As an example, a contact command initiates an SDLC set normal response mode (SNRM). If the physical unit addressed by the SNRM is turned off, a timeout of GROUP REPLYTO=value occurs. If this is the only pending control command the SNRM and timeout occur every control service. At the end of control service the polling cycle is restarted with normal service.

Link NCP Buffer Control

The NCP buffer limit for one data PIU is coded in the NCP LINE macro operand of TRANSFR. This operand specifies the maximum number of NCP buffers that the NCP is to use to receive one SDLC frame on a link. If the end of the PIU has not been reached by the time the buffer limit is reached, the NCP discards all the data received and sends a negative acknowledgment to the sending station.

Link Control Components

Data link control (DLC) provides the scheduling and control for link operations. Data link control is made up of three main parts:

1. Communications interrupt control program (CICP)
2. Link scheduler

3. Synchronous data link control (SDLC)

The communications interrupt control program (CICP) interfaces with the background tasks and drives the link scheduler. The link scheduler schedules, initiates, and ends all SDLC link operations. The synchronous data link control (SDLC) communicates with the link scheduler to indicate an end of a transfer operation or end of buffer; for a type 2 scanner SDLC routines transfers the data between the data buffers and the hardware scanner.

The first 'activate link' command initiates the link scheduler for a specific link. The activate link command triggers the LKB task called the 'run initiator'. During 'run initiator' execution the LKB task address is changed to the 'run terminator'. The completion of a control command or permanent link error results in the execution of the LKB 'run terminator' task. The 'run terminator' leases a buffer to advise the host of the ending condition, and changes the LKB task to the 'run initiator' address. For control commands (except deactivate link) the 'run initiator' task is automatically reinitiated.

After a link is activated the link scheduler searches the devices in the service order table (SOT) for work to schedule. If work is found the link scheduler initiates an SDLC link operation. The link scheduler suspends activity for this link until the SDLC link operation ends. If no work is found the link scheduler suspends activity on this link for 2.2 seconds.

A half-duplex (HDX) link has a send priority. A full-duplex (FDX) link priority schedules the 'receive' link by transmitting polling characters before using the 'transmit' link for sending data. This topic covers data link control in relation to a full-duplex (FDX) link.

The flow of control from initiation of the link scheduler to termination of the link scheduler involves the components covered in the previous sections. NCP physical services process the commands to 'activate link' and 'contact' the physical devices on the line. The 'activate link' initiates the link scheduler. As each 'contact' completes, the 'run terminator' gets control, but the link scheduler is reinitiated for a new 'run' command. After the contact commands the session initiation commands to physical units must be sent on the link to establish the session with the physical units. The session commands and responses are processed as data by the data link control support.

If a contact command is pending for a physical unit the link scheduler initiates a 'contact poll' (from a 'contact' command). A response to a contact poll triggers the run terminator. The run terminator reissues the XIO RUN, which remains in effect until a response to a contact poll to another device, discontact command, permanent error, or deactivate link command.

Once the link is operative, the service order table is used to locate a link outbound queue of a CUB or SCB. If no element is queued and the device session is established, the 'receive' leg is scheduled with a poll. With the 'receive' leg now committed, the send ACB can search for a service order table entry with PIU to send to a station other than to the polled station.

The first link outbound queue (in service order table sequence) sends one to seven PIUs depending on several factors. If there is only one PIU, only one is sent before going to the next SOT entry. In addition, there are two operands on the CUB or SCB which qualify the number of PIUs sent on a link. MAX-

OUT specifies that one to seven frames may be sent before an SDLC response is required. PASSLIM specifies the maximum number of frames sent before going to the next entry in the SOT. The type 4 SCB PASSLIM may be set to 254 frames maximum on a full duplex link. On a full duplex link, after each frame the 'receive' link is checked for a busy condition. If the 'receive' leg is released, a poll is sent between frames to a device other than the one currently being transmitted to.

The LINE macro SERVLIM operand (default of 4) specifies the number of passes through the service order table for normal service (polling and addressing) before control service (command processing) is scheduled. Control service is a search for a command (contact, discontact, deactivate link, etc.). If a command is found, one command is attempted before returning to normal data traffic scheduling.

Any pass through the service order table without a PIU to send or with no incoming traffic from polling causes control service to be scheduled immediately.

The PAUSE operand defines a time value for one pass through the service order table. If the time value has expired before the end of the service order table is reached, normal processing continues. If the time value has not expired, the link scheduler suspends service on the link until (1) the time expires or (2) a PIU is enqueued to a CUB link outbound queue to be transmitted on this link. If the link scheduler is triggered for sending a PIU, the PIU is sent, but no polling occurs until the time has expired.

When a PIU is dequeued from the link outbound queue the buffer at offset X'10' is checked for a nonzero value. The nonzero value is the address of an APPL/LU CPM-OUT to be unconditionally triggered. When an APPL/LU CPM-OUT places a PIU on the CUB link outbound queue the task exits without a QPOST to reset the task to a dispatchable state. This method restricts the CUB link outbound queue to one PIU per APPL/LU session at a time. An even distribution of PIU traffic for all active sessions is provided.

Each line is initially disabled for interrupts to level 2. When the line is enabled from level 3, the CCB is primed on each interrupt with the address of the next character service routine at CCB offset X'24' (CCBL2). When the sequence at level 2 is complete, a PCI to level 3 gives control to the routine specified at CCB offset X'4C' (CCBL3). The level 3 processing passes input to 'path control in immediate' and schedules the next poll. Output PIUs are retained on the 'link outstanding' queue until an SDLC response confirms a good transmission; then the buffers are released.

Data Link Control (DLC) Control Blocks

There are several control blocks generated from a LINE macro definition at NCP generation. In addition to the LINE macro, the GROUP macro- and SERVICE macro-generated control blocks are used by data link control.

Line Group Table (LGT)

The line group table (LGT) is generated by the GROUP macro. SDLC groups generates a X'18' byte LGT. At LGT byte 0 an X'8C' value identifies an SDLC primary station. An X'8E' indicates an SDLC secondary station.

Figure 8.1 illustrates the LGT.

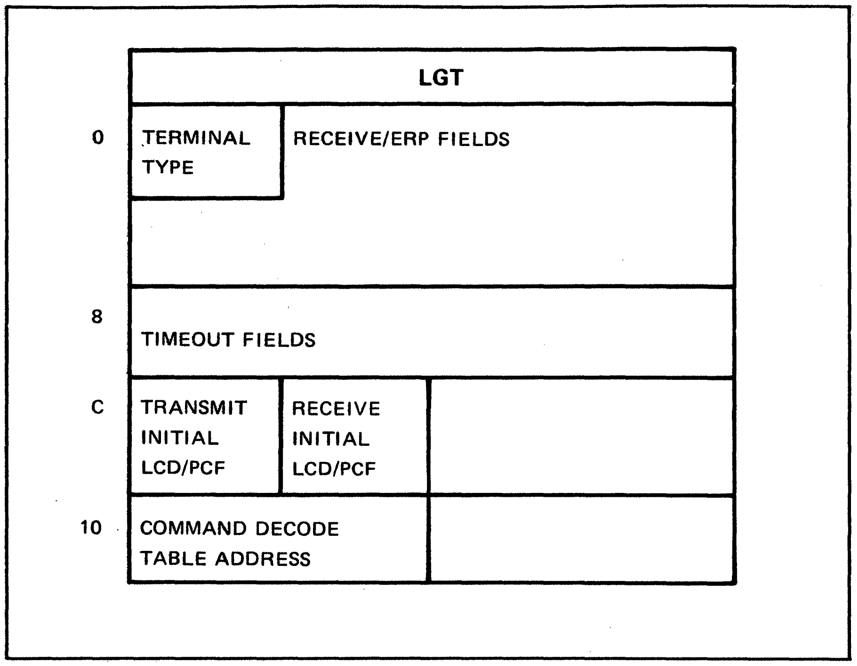


Figure 8.1. Line Group Table (LGT)

Link Control Block (LKB)

The link control block (LKB) contains fields for scheduling link operations and for maintaining link status information. The LKB is generated for each link from the LINE macro. The resource vector table (RVT) contains a pointer to the LKB.

Figure 8.2 illustrates some of the primary fields of the LKB.

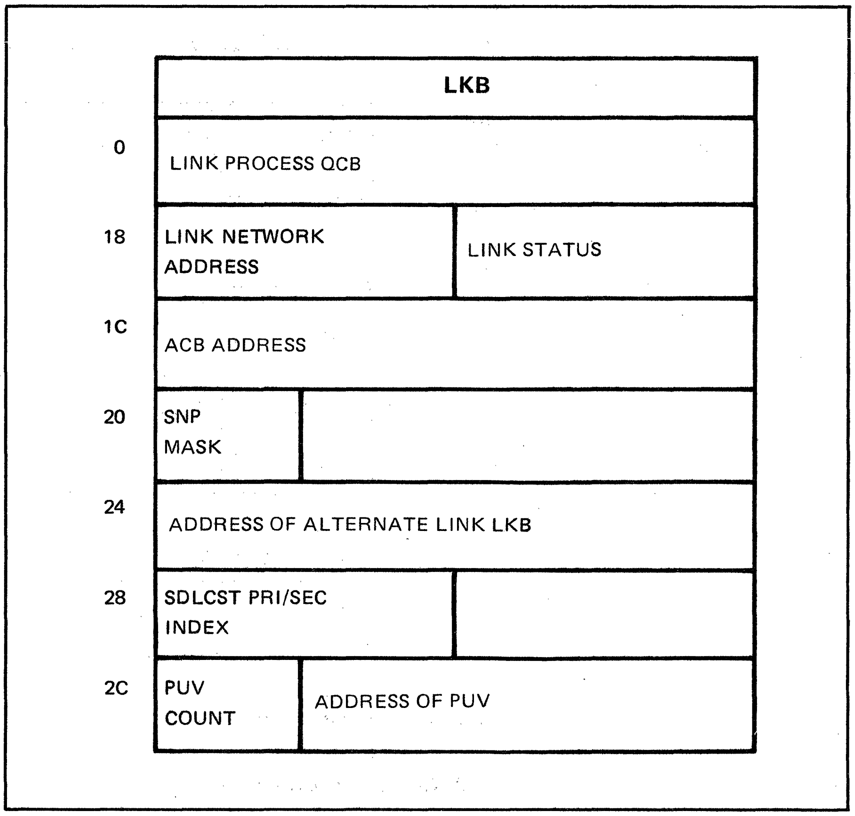


Figure 8.2. Link Control Block (LKB)

Special notice should be given to the following fields:

- X'00' Shifted address to first element queued.
- X'02' Shifted address to last element queued.
- X'0C' Run initiator task address at generation time. When the link is enabled by XIO LINK, the address is replaced by the address of the run terminator.
- X'18' Network address of the link. This network address is used at PIU plus X'2A' for 'activate link' and 'deactivate link' commands processed by physical services.
- X'1A' Status of link. Bit 0 indicates an 'active link', bit 1 an 'activate link in progress', bit 2 indicates indicates 'deactivate link in progress'.
- X'1B' Link type. Specifies this link is leased, switched, one or more type 1 PUs, one or more type 2 PUs, one or more type 4 PUs, and whether the link is primary or secondary.
- X'1C' Address of adapter control block (ACB).
- X'20' SNP mask of SSCP owners.
- X'28' Index into SDLCST for primary SDLC link definition.
- X'29' Index into SDLCST for secondary SDLC link definition.

Physical Unit Vector Table (PUV)

Each link control block (LKB) has a physical unit vector table (PUV). The address pointer to the physical unit vector table (PUV) is located in each LKB at offset X'2C'.

Figure 8.3 illustrates a physical unit vector table (PUV). The quantity of PUV entries defaults to the number of PU macros following a LINE macro. The LINE macro operand of MAXPU defines the quantity of PUV entries generated.

	PUV	
0	SNP MASK	CUB or SCB ADDRESS
4	FLAGS	
8	SNP MASK	CUB or SCB ADDRESS
C	FLAGS	
10	SNP MASK	CUB or SCB ADDRESS
14	FLAGS	

Figure 8.3. Physical Unit Vector (PUV) Table

The PUV is initialized at NCP generation for the generated PUs on a link.

NCP generation reserves a quantity of common physical unit blocks (CUBs) as defined in the PUDRPOOL macro. The host requests the NCP to add a specified number of PU's to a specified LKB and return the network addresses of the added logical units. The request is the command 'request network address assignment' (RNAA). The request RU is X'410210'. The PU address is in RU bytes 3 and 4. The RU identifies the physical unit dynamic reconfiguration pool and quantity of common physical unit blocks (CUBs) to be allocated.

Each common physical unit block (CUB) must be initialized by a 'set control vector' command (RU X'010211xxxx03'). The command 'free network addresses' is used to release (CUBs) to the pool. A generated CUB cannot be released unless the PU macro was coded PUDR=YES.

For additional information, see the topic on dynamic reconfiguration.

Service Order Table (SOT)

The service order table (SOT) is generated by a SERVICE macro to identify the sequence of service to devices on a line. A pointer in the link adapter control block (ACB) at X'60' points to the current entry in the table for service. All SDLC links, except the secondary or SDLC-switched, have a service order table.

Figure 8.4 illustrates an SOT for SDLC devices.

	SOT			
0	00	00	MAXIMUM ENTRIES	ENTRIES IN USE
4	01	CUB or SCB ADDRESS		
8	02	CUB or SCB ADDRESS		
C	03	CUB or SCB ADDRESS		
	XX	00	00	00

Figure 8.4. SDLC Service Order Table (SOT)

The first halfword contains X'0000'. Offset X'02' contains the maximum number of entries in the SOT. Offset X'03' contains number of entries in use.

The default number of fullword entries in the service order table (SOT) is the number of labels coded in the ORDER operand. Additional SOT entries are reserved for adding devices using dynamic reconfiguration by coding MAXLIST=number.

The leftmost byte of each fullword entry contains a negative offset to the first entry. The remaining bits of each entry except the last contain an address of a CUB or SCB. The last entry address field contains a value of zero.

Adapter Control Block (ACB)

The adapter control block (ACB) is generated by a LINE macro. The ACB consists of an auto call unit (ACU) prefix, link XIO control block (LXB), and character control block (CCB). The ACB contains line control information and the status of input or output operations for SDLC links. The link control block (LKB) has an address pointer at offset X'1C' to the only ACB for a half-duplex link or to the 'receive' ACB for a full-duplex link. In the 'receive'

ACB at offset X'66' is the address of the 'transmit' ACB for a full-duplex line.

The ACB can be located from the line vector table (LNVT) using the line address. The ACB contains the link XIO block (LXB) from X'00' to X'23' and the character control block (CCB) from X'24' to X'6B'. The fields are covered under the LXB and CCB which follow.

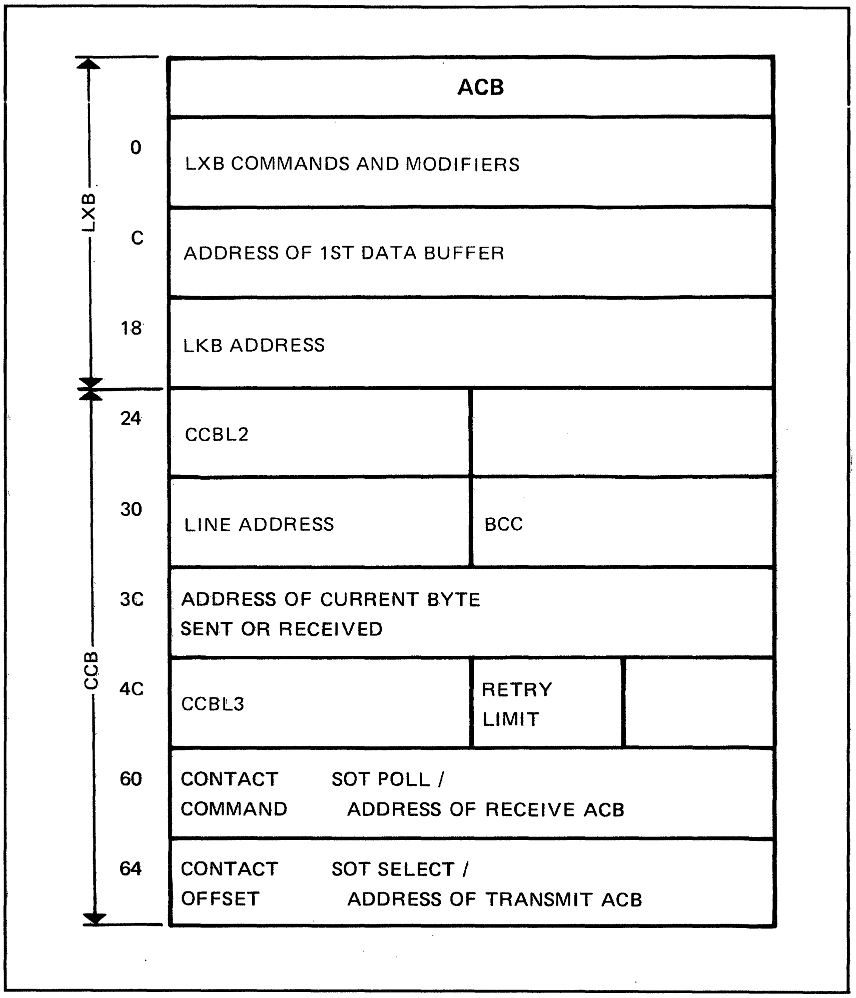


Figure 8.5. Adapter Control Block (ACB)

Link XIO Block (LXB)

The link XIO block (LXB) is generated as the first X'24' bytes of the adapter control block (ACB) and contains the status of link operations. Some of the primary fields are as follows:

- X'00' Immediate control command flags
- X'01' I/O commands. The only valid commands are X'00' (no I/O occurred), X'8D' (enable), X'32' (run initial), X'30' (run SDLC link), X'8F' (dial), and X'83' (disable).
- X'04' Command ending status and completion code status

- X'0A' Received block size (number of data characters stored)
- X'0C' Address of first buffer of data received
- X'18' Pointer to link control block
- X'1C' Address of final buffer of data received

Character Control Block (CCB)

The character control block (CCB) is generated as bytes X'24' through X'6B' of an adapter control block (ACB) and contains line control operations. Some of the primary fields as offsets from the ACB are as follows:

- X'24' Address of current level 2 character service routine
- X'26' Pointer to character service state address table
- X'30' Line vector table (LNVT) entry address
- X'34' Pointer to the line group table (LGT)
- X'3C' Address of current data byte being sent or received
- X'40' Address of current buffer
- X'4C' Address of next level 3 routine to be executed
- X'54' Expected ending status of the level 2 operation
- X'60' Pointer to current service order table (SOT) for half-duplex and duplex receive leg.
- X'60' 'Contact poll' command executed (see X'64')
- X'62' Duplex link pointer to 'receive' leg ('transmit' leg ACB only)
- X'64' Pointer to current service order table (SOT) for 'transmit'.
- X'64' Offset into SOT of current 'contact poll' device (see X'60')
- X'66' Duplex link pointer to 'transmit' leg ('receive' leg ACB only)

Line Vector Table (LNVT)

The line vector table (LNVT) is generated from the CSB macro and initialized by the LINE macro.

A two-byte entry is generated in the line vector table (LNVT) for each possible line address (maximum=96) for each defined scanner (CSB macro). The first scanner position generates 96 halfword entries from X'800' to X'8BF'. Each subsequent CSB macro reserves an additional 96 halfwords. An undefined line address has an entry with a value of X'00DB'. A nonvalue of X'00DB' indicates that the halfword contains the address of the adapter control block (ACB) for this line. The first X'20' entries from X'800' to X'83F' are always invalid because the first scanner has only 64 lines starting at line address X'20'.

If a line address is known the LNVT entry can be calculated by multiplying the line address by 2 and adding X'800'. The LNVT allows the level 2 routines to find the ACB (LXB and CCB) for a line when only the line address is known.

User line control definitions (LEVEL2=name, LEVEL3=name) do not generate entries in the LNVT. The user adapter control block (UACB) addresses are provided in the user line vector table (ULVT).

Figure 8.6 illustrates the LNVT.

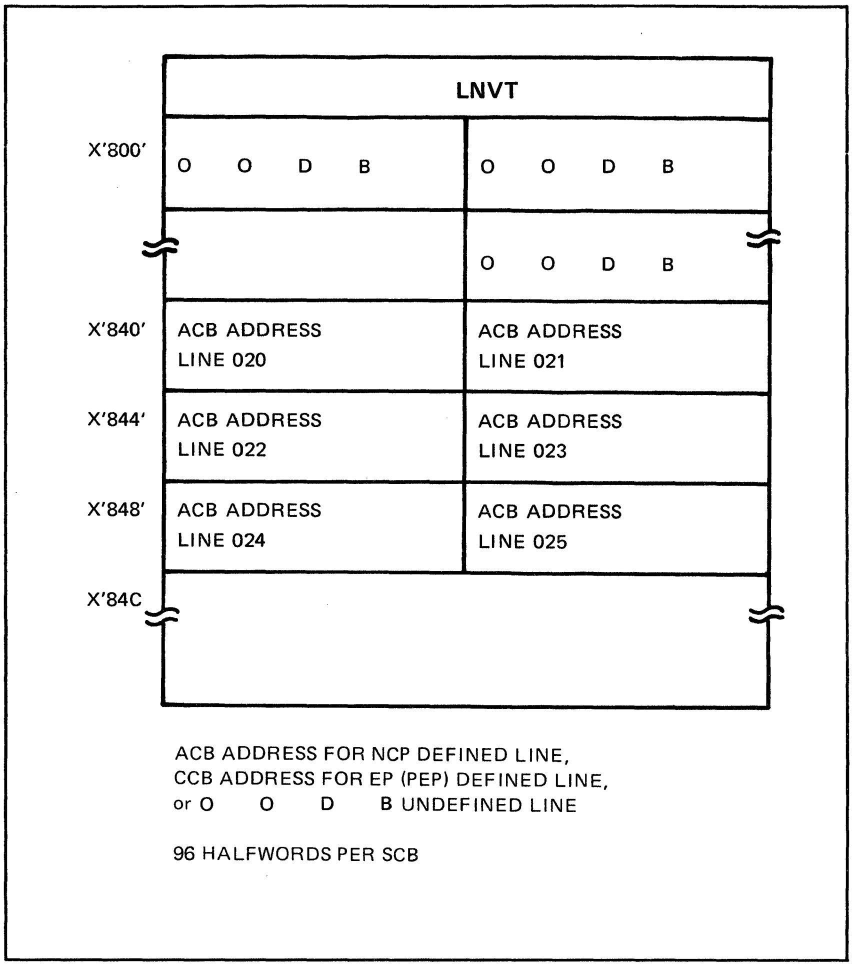


Figure 8.6. Line Vector Table (LNVT)

Selection Table Entry (STE)

The selection table entry (STE) is generated from the SDLCST macro. SDLCST macros are coded for a secondary local/local link (POLLED=NO) that is attached to an NCP with the remote program load (RPL) feature. The purpose of an STE is allow a failed 3705 with an RPL feature to be loaded as a remote; the original secondary must become primary to control polling.

The IPL initial command received at the secondary (SDLCST=(label1,label2), POLLED=NO) to switch the line definition to primary (POLLED=YES). When the 3705 with the RPL is reloaded as a local the local/remote link is deactivated (or fails). The link must be returned from primary to secondary. An activate link command switches a link defined with an SDLCST operand to secondary.

The first selection table entry is located on a linkage edit map at label CXTSTE. The link control block (LKB) at X'28' contains an offset from the first STE entry to the primary mode entry; at X'29' contains an offset to the STE for the secondary mode entry.

Figure 8.2 illustrates the STE. The fields are generated from the SDLCST operands which correspond to LINE and PU operands.

Link Scheduler Data Flow

Link Scheduler Initiation

At the termination of the 'activate link' process, the 'run initiator' receives control and issues the XIO LINK macro with the run command stored in the LXBCMAND field of the LXB. The XIO macro causes an SVC level 4 interrupt into the supervisor. The supervisor uses the SVC code to vector into the branch table for a pointer to the CICP at entry CXECMDCO. The CICP passes control to level 3 via a PCI level 3 interrupt. Level 3 is used to eliminate any interference while setting up to start the command.

The CICP running in level 3 checks to see if the link is busy by checking the receive-CCB control field (CCBCTL) phase bits for 00. Finding the link not busy, CICP zeros out the status fields in the LXB. No check is made to see whether command initialization should be delayed. If no delay is required, the receive-CCB is checked for any outstanding status.

Next, the transmit-CCB is checked for any outstanding status. With no outstanding status on either leg, the link's PCF field is set to zero to prevent any level 2 interrupts on this line from changing any fields that will be set up now. CICP vectors into the command decode vector table, using the LGTCMD pointer from the line group table (LGT) and the command from the LXB. The pointer at the vector is loaded into the instruction address register (IAR), causing a direct branch to the link scheduler at entry point CXELNKSI. The entry point is the 'run' command initialization entry into the link scheduler. Here the phase bits (CCBCTL) are set to indicate command active, then a branch is taken to the scheduler to schedule run command activity.

When scheduling run command activity, the link scheduler decides whether to schedule a poll or a data transmission. The first test determines whether the 'transmit' leg of the link is busy. If not, the 'receive' leg is checked to see whether it is busy. With both legs of the link idle, the scheduler branches to the poll subroutine to schedule a poll operation. The poll operation is started by scanning the service order table (SOT) for a station control block (SCB) or common physical unit block (CUB) to poll. The scheduler first checks the service-seeking control flags (SCBSSCP) and the service-seeking output control flags (SCBOCF) to be sure that this entry has not already been polled or that a second level error recovery program is not in progress. If the station has not been polled and there is no second level error recovery in progress, the scheduler proceeds to poll the station.

Before the actual transmission of the poll frame, a test is made for the station type. Is the station a type 1, type 2 or type 4 physical unit? Type is checked so that the 'receive' buffer used to store the response can be set up properly for the type of FID. A branch is then taken to the 'receive initialization' subroutine (RCVINIT) to set up the 'receive' leg to handle the response from polling. This routine prepares the 'receive' leg to monitor for flags, then returns to the scheduler. The scheduler sets up the 'transmit' leg to send an 'RR' poll command (CCBCFLD). The CCB, if not already transmitting continuous flags, is set up to transmit a flag character. Returning from the transmit initialization subroutine, the scheduler sets the data length field in the CCB (CCBCHAR) to zero for the poll, and exits from the program level.

The SDLC character service program sends a flag byte on the link. The program also prepares the CCBL2 pointer to send the address field from the CCB (CCBAFLD) at the next level 2 interrupt. When the complete poll frame has been sent, the scanner is set up to transmit continuous flags until the link scheduler finds the next service order table (SOT) entry that can be polled. The ACB is then queued to the ACB queue (XDH X'736' and X'738') and a PCI level 3 interrupt is issued. The PCI level 3 interrupt causes the link scheduler to get control again, via the CCBL3 pointer, to poll the next entry in the service order table (SOT).

XIO LINK to the Link Scheduler

The XIO LINK macro is used to put PIUs on the link outbound queue (LOB) to be transmitted down the link. The XIO LINK macro stores the pointer to the PIU in the LOB and checks for an active 'run' command. With a 'run' command already active, XIO LINK does not have to trigger the link scheduler. During its normal scan, the link scheduler finds the PIU on the LOB and transmits it on the link.

Entry to the link scheduler is at CSELNKX for normal scan. Before sending the PIU, the link scheduler checks to see if the 'receive' leg is busy as a result of the last poll. If the receive leg is busy, the link scheduler tries to select a CUB or SCB with data on its LOB. The current service order table (SOT) select pointer from the 'transmit' leg (LXBSEL) is used to get the station's SOT pointer. A test of the station's output control flags (SCBOCF) is made to see if the station is ready and data is waiting. If the station is not ready, the link scheduler advances to the next SOT select entry. With the station ready the link scheduler branches to the SENDPIU subroutine.

The SENDPIU subroutine first checks the basic link unit (BLU) outstanding count (SCBOCL) defined by the MAXOUT operand and the pass count (CCBPASCT) defined by the PASSLIM operand. If either MAXOUT or PASSLIM have been reached, SENDPIU returns to the calling routine (SELECT). SELECT increments the select pointer and continues to the next SOT entry except for a CUB or SCB MAXOUT condition. A local/remote link SOT pointer is not incremented until the PASSLIM operand value is reached. If the counts have not been exceeded, the PIU is sent.

SENDPIU increments the BLU outstanding count, takes the PIU off the LOB queue and puts it on the link outstanding queue (LOS). The first buffer of the PIU at offset X'0A' is checked for a nonzero value. A nonzero value is the address of an APPL/LU CPM-OUT. The APPL/LU CPM-OUT is unconditionally triggered to provide the next PIU in this APPL/LU session to the link outbound queue. Next an 'I' format BLU command is built and passed to the XMTINIT subroutine along with the ending process pointer

(CSELNKX). A branch is taken to the XMTINIT subroutine to initiate the transmission. The XMTINIT subroutine stores the ending processor address in the CCBL3 field and the BLU command in the CCBCFLD.

Level 2 interrupts are now disabled and a test is made to see if flags are already being transmitted. If so, XMTINIT loads the CCBL2 field with the address to the SDLC send address routine (CSBDLXZ). Level 2 interrupts are enabled again and a 'transmit' time-out is started. XMTINIT now returns to SENDPIU to complete the setup of the CCB for transmission. In the CCB, SENDPIU stores the character count (CCBCHAR), the pointer to the current buffer (CCBSTART), and the pointer to the first data character (CCBDATA). SENDPIU returns to the calling routine (SELECT). SELECT stores the SOT pointer in the LXB (LXBSEL) and EXITs from the program level.

The following flow is for a type 2 scanner. The type 3 scanner cycle-steals data into or from storage for the length of an NCP buffer without a level 2 interrupt.

The SDLC character-service routines take over to transmit the PIU on the link. The link scheduler subroutine (XMTINIT) previously setup the CCBL2 pointer to point to the 'transmit address' routine (CSBDLXA). When the scanner hardware finishes sending a flag character, a level 2 interrupt is generated to the 'transmit address' routine. This routine initializes the BCC field (CCBBCC), than passes the address field (CCBAFLD) and the next CCBL2 pointer (CXBDlXC) to the BCC accumulation routine. The BCC accumulation routine sends the address to the scanner, accumulates the BCC character, stores the CCBL2 pointer passed to it in CCBL2, and then EXITs from the program level.

The next level 2 interrupt is to the 'transmit control field' routine (CXBDLXC). This routine sends the control character from CCBCFLD and tests the character count (CCBCHAR) for zero. If the count is not zero, CCBL2 is set up to transmit data (CXBDLXI). If it is zero, the CCBL2 is set up to transmit the first BCC character (CXBDLXB1). On the next level 2 interrupt, with the CCBL2 pointer set to CXBDLXI, the character-service routine sends the data out on the link. This routine loops until all the data has been sent on the link. With no more data to send, this routine sets up the CCBL2 pointer to transmit the BCC character that has been accumulated in the CCB.

The next level 2 interrupt transmits the rightmost byte of the BCC field (CCBBCC) in the CCB. The CCBL2 is set up to transmit the leftmost byte of the BCC field on the next level 2 interrupt. After the second BCC character is transmitted, CCBL2 is set up (CXBDLXFF) to transmit a flag to end the frame.

On the next level 2 interrupt, a check made for half-duplex to see whether a turnaround is needed on the line. If no turnaround is needed, 'frame transmitted' status is stored in the CCB (CCBCMPCD) and the line is set to transmit continuous flags (LCD/PCF=9D). A branch is taken to QACBL3 to queue the ACB to the ACB queue for level 3 processing and a PCI level 3 is set. Returning from QACBL3, the CCBL2 pointer is set to ignore interrupts from the line (CSBDLIDL) before EXITing from the program level. At this point, one PIU has been sent on the link.

The PCI level 3 interrupt gives control back to the link scheduler at entry CXELNKX, via the CCBL3 pointer. The CCB is passed to level 3 on the queue at XDH X'736' and X'738'. The 'transmit leg busy' flag is reset and a check is made to see if the last frame transmitted was an 'I'-format frame. If not, the frame must have been in 'S' or 'NS' format, so execution returns is to the scheduler at entry CXELNKSX to continue link activity.

Link Poll to Path Control Inbound

A poll frame has been sent down the link. The correct offset for the type of FID has been stored in the CCB (CCBOFSET), based on the secondary station type, and the 'receive' leg CCBL2 pointer has been set to monitor for flags. The CCBL3 pointer has been set up with the normal read-end processor address (CXELNKR).

When the first flag character is received, the 'monitor for flags' routine sets CCBL2 to receive the address (CXBDLRA). The hardware in the scanner handles any other flags received without causing any level 2 interrupts. The next level 2 interrupt comes with the first nonflags character received. This character should be the address field of the frame. The 'receive address' routine (CXBDLRA) checks the character received to see if it is the address expected. If it is not, then the link is reset to monitor for flags again. If the character is the expected address, the address is stored in the CCB (CCBAFLD). The BCC (CCBBCC) is initialized next and the CCBL2 pointer is set to receive the control field (CXBDLRC). Before EXITing from the program level, the BCC is accumulated for the address field which was received.

The next level 2 interrupt gives control to the control field routine (CXBDLRC). If the character received was a flag, a format error has occured. The status is set to indicate format exception and the ACB is queued back for level 3 processing. If the ACB cannot be queued back for processing, 'block overrun' status is set and the remainder of the frame is flushed. If the character received is the control field, tests are made for the frame format. If the frame is an 'I' format, a buffer is leased and initialized. The CCBL2 pointer is set to receive data (CXBDLRI) before EXITing from the program level.

The 'receive data' routine accepts characters until a flag is received. (The type 3 scanner cycle-steals data for the length of an NCP buffer.) The characters are stored in the buffer leased by the control field routine (CXBDLRC). If more buffers are needed, buffers are leased one at a time. When the flag character is received, the pointer to the data buffers is stored in the LXB (LXBDATAP). Next, three checks are made, testing for: (1) correct BCC for this frame, (2) the expected address, and (3) final frame. If this is not the final frame, the ACB is queued back to level 3 to process this frame and the CCB is set up to receive the next frame, starting with the address field (CXBDLRA). If this was the final frame (P/F bit on), the transmitting terminal sets 'poll/final' status and queues the ACB back to level 3 to process the frame.

The return to the link scheduler is via CXELNKR from the CCBL3 pointer. If the frame received is an 'I' format, a branch occurs to PROCIFMT to process the frame. PROCIFMT computes the frame length and sets the data offset and counts in each buffer. The last buffer count is adjusted for the BCC characters that were stored. The total data count is stored in the NCP buffer prefix. The N(C) count is updated by 1. A branch is made to BNN path control in immediate for routing the PIU.

Termination and Restart of an XIO Run Command

The 'run' command is ended by triggering the link process queue in the LKB. The task pointer in the link process queue is for the 'run terminator' task. There are only six valid reasons for ending the run command:

1. Reset immediate ('deactivate link in progress', or 'contact' command)
2. Permanent link error (hardware or XMIT error)
3. Station counters overflow
4. Buffer pool end
5. Valid response or ERPs exhausted on 'contact poll'
6. Unrecoverable station error during poll

When the SDLC character-service routines uses PCI to return to level 3 for processing, the link scheduler checks the link status to see if 'run termination' is required. If 'run termination' is required, (CXELNKSS) the 'stop run' command is set in both of the CCB's (CCBCTL) for a full-duplex link. When both legs of the link become idle, the link scheduler ENDRUN subroutine triggers the run terminator task.

Based on the error status received, the ENDRUN subroutine flushes the LOB and LOS queues. For hardware or transmit error status, the ENDRUN subroutine flushes the LOB and LOS for all stations on the link. All PIUs on the station's LOB and LOS are set with 'path error' status, and put on the link inbound queue for that station. The link inbound queue gets triggered along with the run terminator task. For this type of error, the 'run' command is not reissued.

For other exceptions, only the current stations LOB and LOS queues are flushed. Again, all the PIUs on the current station's LOB and LOS are set with 'path error' and are put on the station's link inbound queue. The link inbound queue is triggered along with the run terminator task, but in this case the run terminator reissues the RUN command when the task finishes its processing.

The run terminator determines the reason for termination and the appropriate routine is called to handle the status. In an example of a permanent link error, the link is set inactive (LKBSTAT), the 'active links count' in the PSB is decremented by 1, an 'inoperative' request is built and sent to the SSCP, and an MDR record is returned to the host. All the stations on the link are checked for FM data requests and if any are found they are returned to the SSCP with an error indication. All stations are left with 'inoperative' and 'poll skip' flags on.

Data Link Control Summary

The link scheduler is initiated for this link by an 'activate link' command addressed to NCP physical services. NCP physical services identifies the link to be activated in the PIU RU1NA field, and an 'enable' is processed for nonswitched links.

The link scheduler has a three-part cycle:

1. SERVLIM data passes are made through the service order table as long as one PIU is sent or received per pass. The first pass without a PIU transfer invokes part 2.
2. The physical units are searched for a 'contact' command to be processed. The search begins with the first physical unit following the last unit serviced (CCB offset X'64') for a contact command and ends when a command has been serviced or all physical units have been scanned.
3. If in the last data pass (see point 1), the time specified in the PAUSE operand had not expired, the link scheduler waits until (a) a PIU is enqueued for a PU on this link and then transmits only; or (b) the time expires to begin polling.

The flow of an 'activate link' for an SDLC link is illustrated in Figure 8.7. The numbered items that follow identify the flow of the 'activate link' command and processing that takes place.

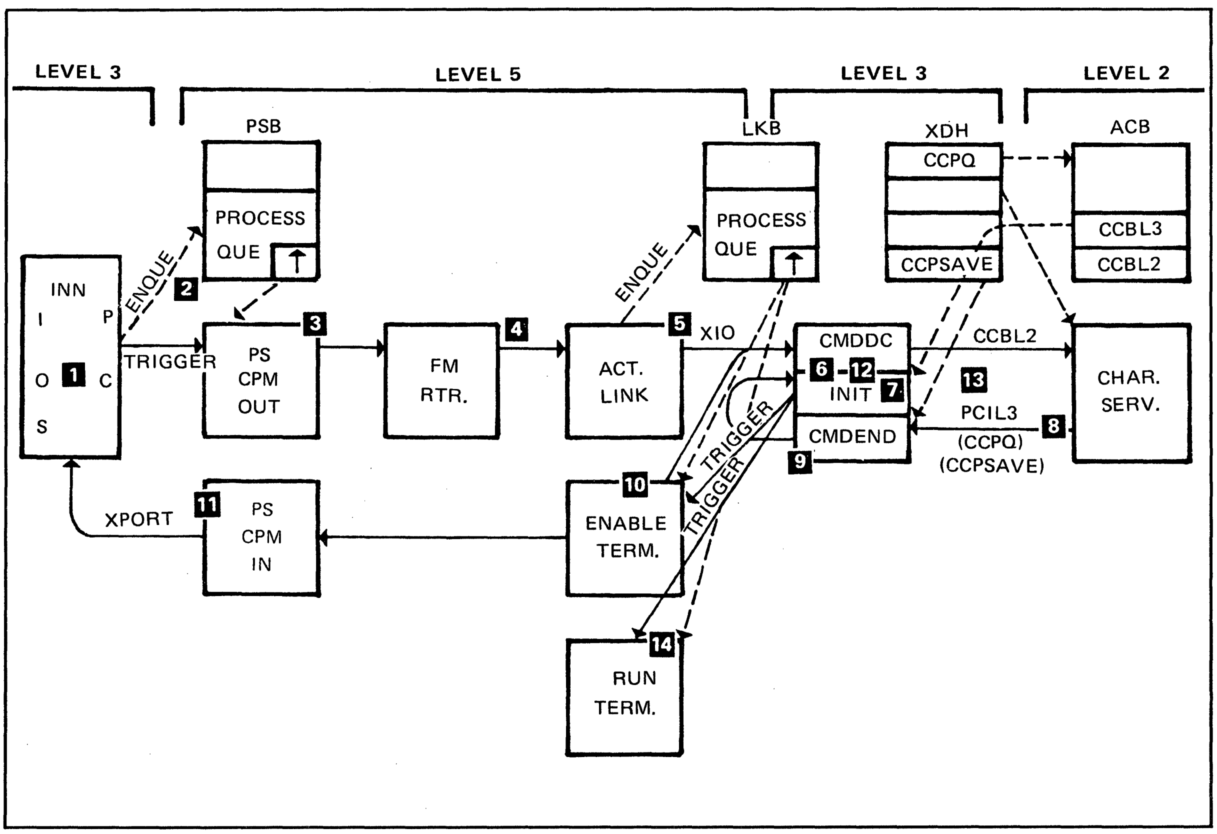


Figure 8.7. Activate Link Command Flow

1. At channel stop, IOS passes the PIU to path control via a branch.
2. Using the PIU DAF to access the SIT, SVT and RVT, path control enqueues the PIU to NCP physical services.
3. The PSB task is dispatched (PSB CPB-OUT), which calls the function management router.
4. Using RU1BT1 and RU1RC2 (bytes 1 and 2 of the RU), the 'function management router' selects and calls the 'activate link' processor.
5. 'Activate link' enqueues the PIU to the LKB, sets the LKB task pointer to 'enable terminator', sets 'activate link in progress', and issues 'enable XIO'.
6. Command decoder selects the proper initialization routine for 'enable'.
7. CCBL3 is set for the proper return from level 2, the ICW is set to 'data terminal ready', and CCBL2 is set to the proper level 2 routine to wait for 'data set ready'.
8. At 'data set ready', level 2 enqueues the ACB on the ACB queue (CCPQON) and issues a PCI to level 3.
9. CCPSAVE contains the address of the command ender, which gives control to the CCBL3 pointer.
10. The LKB is triggered, which schedules the 'enable terminator' task. The enable terminator task changes the LKB task pointer to 'run terminator', sets 'link active', and issues 'run XIO'.
11. NCP physical services CPM-IN sends a response to the channel queue for routing to the host.
12. The command decoder resolves the link scheduler as the initialization routine for 'run XIO'.
13. The link scheduler begins polling and selection for this link until the termination of the run command.
14. Should the 'run' command terminate, the 'run terminator' is dispatched because of the LKB task pointer.

Chapter 9:

Local/Local and Local/Remote Definitions

Introduction

An NCP link attached to another NCP is defined by PU macros with an operand of PUTYPE=(4,LOCAL) or PUTYPE=(4,REMOTE). The PUTYPE=(4,LOCAL) defines a local/local link connection to an NCP on a channel-attached host with or without a remote program loader (RPL). The PUTYPE=(4,REMOTE) defines a local/remote link connection to an NCP which has a remote program loader (RPL) and no channel adapter.

The link to a type 4 physical unit requires the same activation as a link to type 1 and type 2 physical units. Each end of a local/local link receives an activate link command. Only the local end of a local/remote link receives an activate link command.

Once the link is active, a 'set control vector - NCP subarea' command is sent to NCP physical services. The 'set control vector - NCP subarea' contains:

- X'2A' Network address of the link.
- X'2C' A value of X'02' qualifies the set control vector command as a 'NCP subarea' type.
- X'2D' The subarea address (left justified) of the adjacent subarea.

The subarea address (offset X'2A') is checked against the SIT subarea value to locate the SVT entry. The network address of the link (offset X'2A') is used to locate the RVT entry of the link. If the non-backup link is being activated the address of the SCB which immediately follows this link is in the SVT entry. If the backup link is being activated the following processing occurs:

- Checks to ensure the primary link is not active.
- Checks that the primary link and backup link are not the same.
- Checks that the primary/secondary bit is set in both the current and backup SCB.
- Checks that the backup SCB is not in the 'contacted' state.
- Copies from the non-backup SCB to the backup SCB the SDLC address, network address, maximum outstanding limit (MAXOUT), pass limit (PASSLIM), and station type (PUTYPE=(4,LOCAL) or PUTYPE=(4,REMOTE)).

Once the link is active and the 'set control vector - NCP subarea' provides a positive response, a 'contact' command is sent for the SCB. Each type 4 PU of a local/local link receives a contact command; only the local PU of a local/remote receives a contact command.

Once the path is available, INN path control passes an outbound PIU to the PU generated station control block (SCB) by queuing the PIU to the SCB link outbound queue. Inbound PIUs are passed from the link scheduler to BNN 'path control in immediate' at level 3. When BNN 'path control in immediate' determines that the PIUs are from a type 4 PU, the PIU is passed to INN path control for routing.

Local/local and Local/remote Control Blocks

Station Control Block (SCB) PU Type 4

The station control block (SCB) is illustrated in Figure 9.1. The common physical unit block (CUB) is illustrated in Figure 7.2 in the boundary network node section.

If you compare the type 4 PU station control block (SCB) with the common physical unit control block (CUB), the fields are identical for the length of the SCB. The type 1 and type 2 CUB has an extension added for a QCB for outbound PIU processing.

PIUs destined to a type 4 PU are enqueued by INN path control to the SCB link outbound queue at SCBLOBH (X'18). PIUs from a type 4 PU are passed from the link scheduler to BNN path control in immediate and to INN path control.

	SCB	
0	INBOUND QCB (LINK INBOUND QUEUE)	
18		LINK OUTBOUND QUEUE HEAD
1C	SNP MASK	LINK OUTBOUND QUEUE TAIL
20		LINK OUTSTANDING QUEUE HEAD
24	STATION TYPE	LINK OUTSTANDING QUEUE TAIL
28	SDLC ADDR	LKB ADDRESS
2C	CUB NETWORK ADDRESS	POLL/CONTACT STATUS

Figure 9.1. Station Control Block (SCB)

Selection Table Entry (STE)

The selection table entry (STE) is generated from the SDLCST macro. SDLCST macros are coded for a secondary local/local link (POLLED=NO) that is attached to an NCP with the remote program load (RPL) feature. The purpose of an STE is allow a failed 3705 with an RPL feature to be loaded as a remote; the original secondary must become primary to control polling.

The IPL initial command received at the secondary (SDLCST=(label1,label2), POLLED=NO) to switch the line definition to primary (POLLED=YES). When the 3705 with the RPL is reloaded as a local the local/remote link is deactivated (or fails). The link must be returned from primary to secondary. An activate link command switches a link defined with an SDLCST operand to secondary.

The first selection table entry is located on a linkage edit map at label CXTSTE. The link control block (LKB) at X'28' contains an offset from the first STE entry to the primary mode entry; at X'29' contains an offset to the STE for the secondary mode entry.

Figure 9.2 illustrates the STE. The fields are generated from the SDLCST operands which correspond to LINE and PU operands.

STE

0 COUNT OF STE ENTRIES	1 MAXOUT	2 LGT ADDRESS	
4 ERP SECOND LIMIT	5 RUN MODIFIER	6 LINE TYPE	7 ERP RETRY LIMIT
8 MAXDATA	9 SERVLIM	A SDLC ADDR	B PASSLIM
C ERP DELAY	D RESERVED		

Figure 9.2. Selection Table Entry (STE)

Local/local Flow

The local/local link definitions are considered equals. Within the NCPs one end of the link is defined as 'primary' to perform polling and the other end is defined as 'secondary' to respond to polling.

The activation from each host is equal. The commands of activate link and contact must be issued from both hosts to enable data flow. Figure 9.3 illustrates the SNA and SDLC command flow for initiating local/local links.

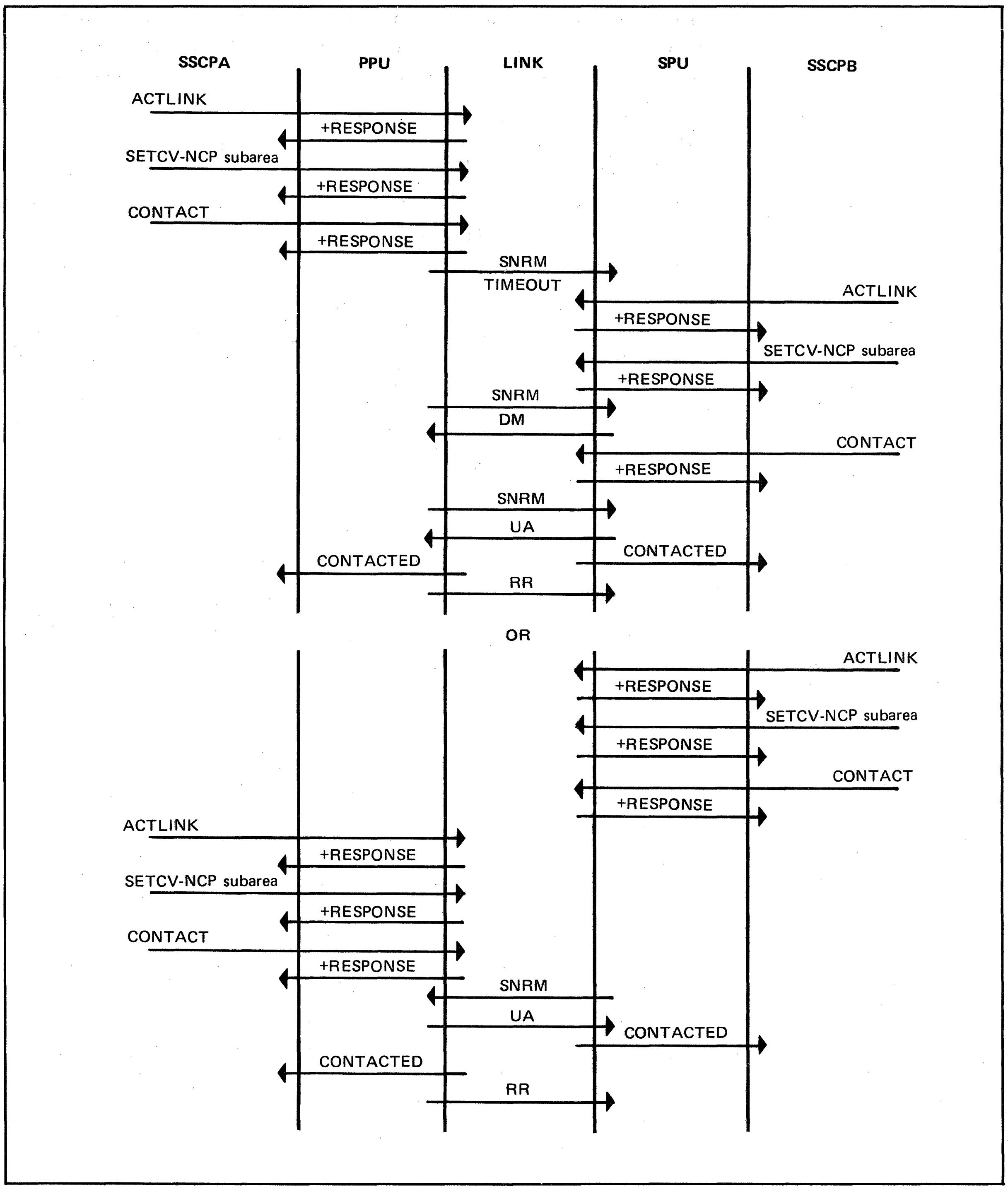


Figure 9.3. Initiate Local/local Link

The first example illustrates activation of the primary (polling) definition. The activate link enables the modem. Contact initiates the SDLC SNRM command.

The response to a SNRM from a secondary (polled) NCP where the activate link has not been processed is a timeout. The first SNRM followed by the timeout occurs until an activate link is processed by the secondary.

Once the secondary has processed an activate link, but before a contact command, the response to a SNRM is an SDLC 'disconnect mode' (DM) response. After activate link, but before a contact command, the DM is sent from the secondary until the contact is processed.

After the secondary has processed the contact command the response to the SNRM is an SDLC unnumbered acknowledgment (UA). The SNRM from the primary results in an UA to the primary and a 'contacted' command being sent to the secondary host owner. The UA received by the primary initiates a 'contacted' command to the primary host owner.

The second example in Figure 9.3 illustrates an activation first from the secondary. Note that after both activate link and contact commands at the secondary end no SDLC traffic flows. After the activate link and contact at the primary end the first SNRM flows, the UA response is returned, and both primary and secondary send 'contacted' commands to their host.

After the local/local link initialization data PIUs flow in both directions until a permanent error or local/local termination.

Figure 9.4 illustrates the command sequence for terminating a local/local link connection.

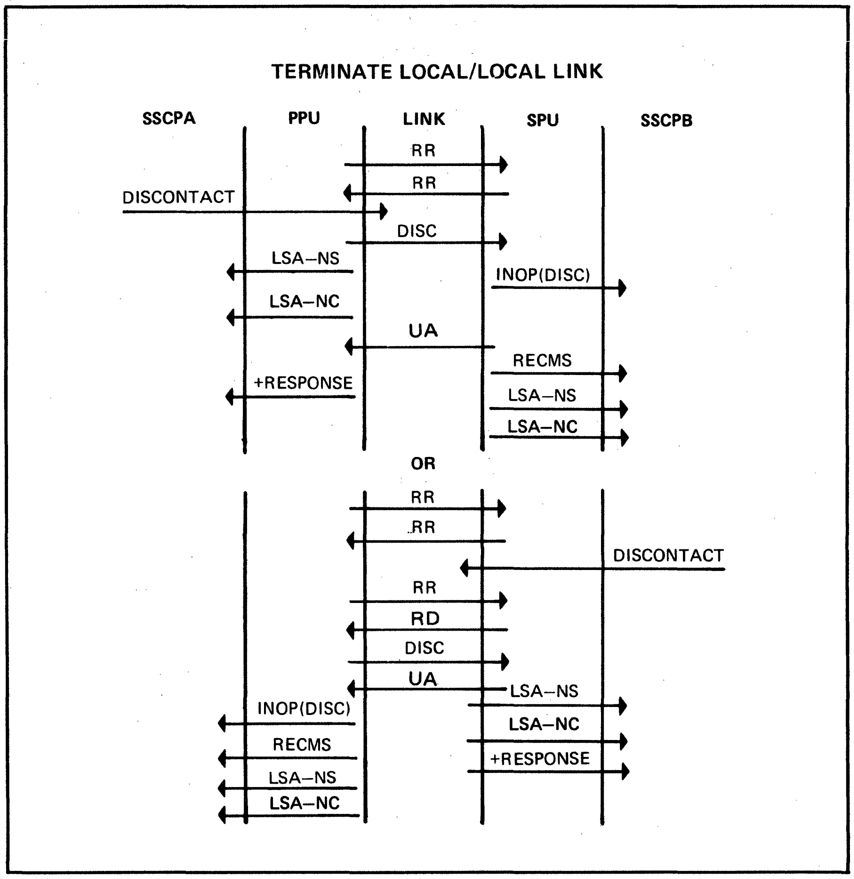


Figure 9.4. Terminate Local/local Link

The first example illustrates a discontact issued by the primary. The primary sends an SDLC discontact (DISC) is sent to the secondary. The primary sends lost subarea (LSA) commands to all adjacent type 4 and type 5 PUs (NS format) and to all NCP owners (NC format). When the secondary receives the SDLC DISC the secondary replies to the primary with an UA; the primary sends a response PIU to the discontact command. The secondary sends the link owners an 'inoperative, discontact' (INOP(DISC)) command and record maintenance statistics (RECMS). The secondary sends lost subarea (LSA) commands to all adjacent type 4 and type 5 PUs (NS format) and to all NCP owners (NC format).

The second example illustrates a discontact issued by the secondary. The secondary issues the discontact command. The primary must poll with an SDLC RR command before the secondary can reply with a 'request disconnect' (RD). The primary responds to a RD with a disconnect (DISC). The secondary response to a DISC is the unnumbered acknowledgment (UA).

Because the secondary initiated the disconnect, the primary generates the 'inoperative, discontact' (INOP(DISC)) command and record maintenance statistics (RECMS). The primary sends lost subarea (LSA) commands to all adjacent type 4 and type 5 PUs (NS format) and to all NCP owners (NC format).

The DISC from the primary initiates an SDLC UA response to the host, lost subarea (LSA) commands to all adjacent type 4 and type 5 PUs (NS format) and to all NCP owners (NC format), and a response PIU to the discontact command.

Initializing the Remote

The definition of a remote in a local is by a type 4 PU macro. The PUTYPE=(4,REMOTE) identifies an NCP without channel adapters (or treated as having no channel adapters). The PUTYPE=(4,LOCAL) identifies an NCP with one or more channel adapters and may or may not have a remote program loader (RPL). The PUTYPE=(4,LOCAL) definition must be primary (polling) to load and control a remote.

If the original definition in the local is secondary (polled), and an SDLCST macro provides primary values, the link is switched from secondary to primary by a command from the host. After the link is in primary mode the activate link, set control vector - NCP PU', and contact commands are required.

The contact command initiates an SDLC set normal response mode (SNRM) command to the remote. The response of unnumbered acknowledgment (UA) indicates the remote is loaded and available for activation. The response of request initialization mode (RIM) indicates the remote is available for loading.

Figure 9.5 illustrates the SNA and SDLC command sequence which occurs to a remote NCP which requires loading.

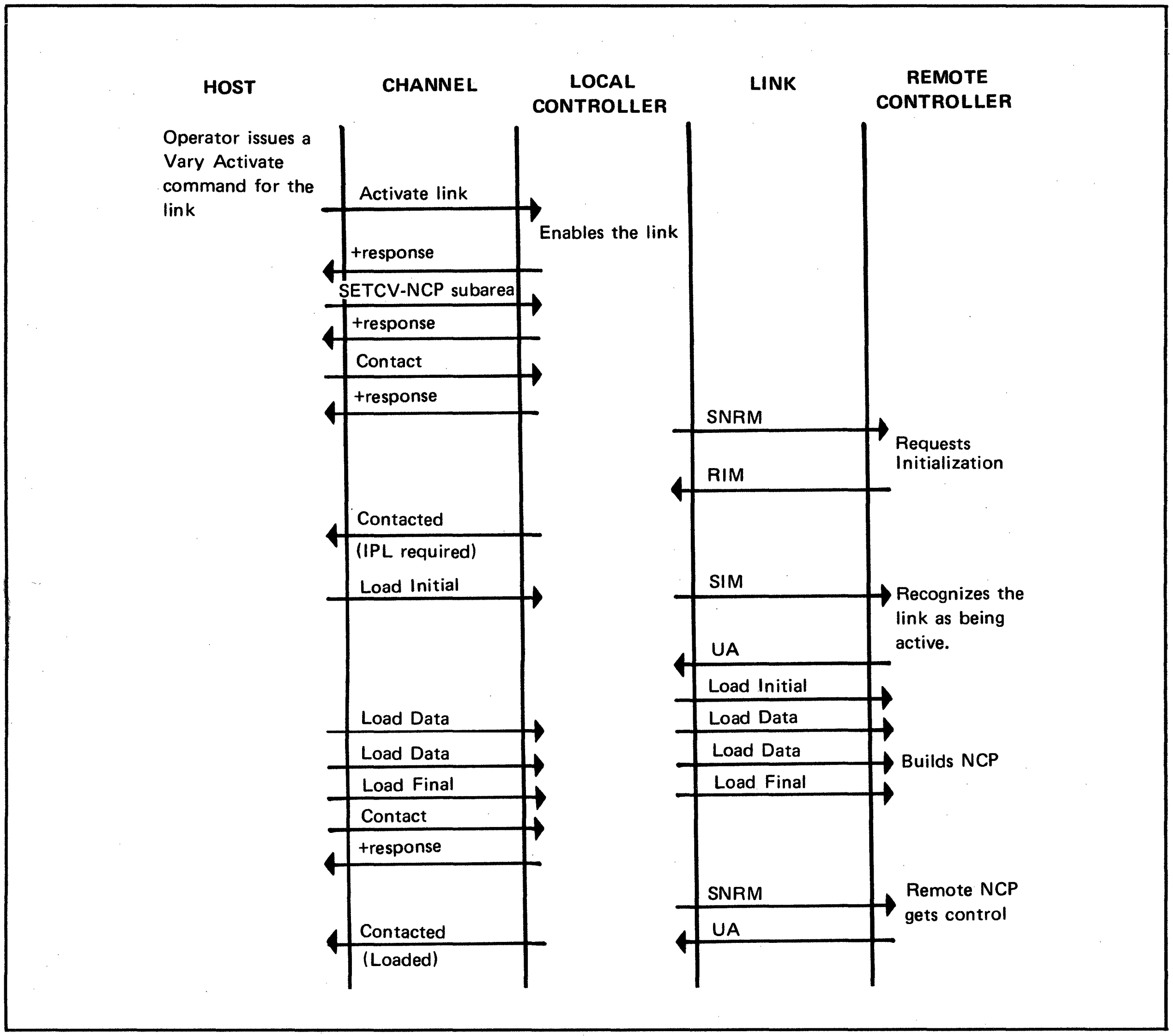


Figure 9.5. Command Sequence to Activate a Remote

In response to an RIM (contacted, IPL required), the SSCP obtains the remote load module and sends the load initial, load data, load final, and a second contact command.

The load initial to the local NCP physical services schedules a 'set initialization mode' (SIM) command and receives a unnumbered acknowledge (UA). The 'load initial', 'load data', and 'load final' are transmitted to the remote.

The second contact command initiates a new 'set normal response mode' (SNRM) to the remote and a response to the SSCP, to acknowledge that the 'contact' command was received. Now that the remote is operational, it can reply to the SNRM with an UA. The UA response results in a contacted command being sent to the SSCP.

Now that the remote is loaded, the same SSCP and application sessions are established in the remote as are established in the local. An SSCP/PSB 'activate physical', 'start data traffic', 'set state vector', 'activate link', and other session command sequences must be established between the SSCP and remote physical and logical units.

PIUs destined to the remote are processed by INN path control. INN path control at level 3 validates the FID0 or FID1, verifies that the local/remote link is active, and enqueues the PIU to the SCB link outbound queue (SCB plus X'10). The link scheduler locates and transmits the PIU to the remote.

The PIUs received on a link are passed at level 3 from the link scheduler to BNN path control in immediate. Path control in immediate checks the station type at the control block address plus X'24'. If the device is a type 4 PU SCB the PIU is validated as FID0 or FID1 and passed to INN path control for routing. If the remote link had a failure, the SCB connection point manager in (CPM-IN) is triggered for error recovery.

Remote NCP

The remote NCP has basically the same facilities as the local NCP. The remote controller does not have a channel adapter and therefore does not have the channel adapter IOS. The link is serviced by the link scheduler and outbound PIUs are passed to INN path control. INN path control enqueues the PIUs to physical services, the link outbound queue of the SCB, a BNN CPM-OUT, or the BSC/SS processor. The same session control sequence is required among SSCP, applications and remote elements as was required in the local.

Loading a Remote NCP

The remote 3705 controller includes a diskette which contains programs used to test the remote hardware and to load and dump the remote NCP. The diskette is prewritten with the configuration data set (CDS) file. This file must be configurated before the remote controller is used. The CDS defines the link to be monitored for communication and the pointers to the diskette data sets.

Loading and dumping of a remote NCP is performed by the load/dump program that resides on the diskette. This program is loaded into the high 8K of storage when one of the following occurs:

- Power is turned on
- The load pushbutton on the remote console is pressed
- The remote NCP terminates abnormally
- An error occurred during a load or dump
- Host issues a load or dump network command

Before loading the load/dump program into high storage, the NCP checks to see if the high 8K of storage should be saved and written on the disk. Also, checks are made to see if any diagnostics or initial tests are to be executed. Figure 9.6 illustrates the format of the remote disk files.

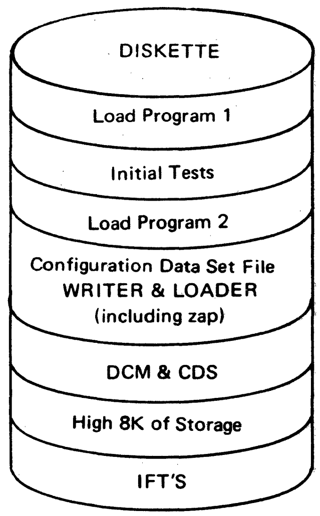


Figure 9.6 Remote Disk Format

When the load/dump program is loaded into storage, control is passed to program level 1. External register X'6B' contains IPL flags; general register 6 of program level 3 contains a line address for the load/dump program to monitor. This line must have been defined in the remote configuration data set (CDS) file. A byte in the CDS file entry determines if this line is to be used for loading and dumping of the remote NCP. This check prevents unauthorized loading and dumping of a remote controller.

After the load/dump program is initialized, the program executes in levels 2 and 3 performing link scheduler and SDLC functions. Level 1 is reentered when control is passed to the remote NCP after it is loaded.

If a 'load' is to be performed, after the link is activated and the remote contacted, the host sends PIUs containing the remote version of the NCP to the local NCP. Physical services in the local determine that the PIU is a function management request and call the function manager. The FM router uses the RU of the PIU to select the remote PIU decoder routine from the appropriate FM table.

The remote PIU decoder (CSDKRPD) determines that the request is a 'load initial'. It sets up the station control block for the remote and sends a 'set initialization mode' (SIM) SDLC command to the remote. The load/dump program in the remote controller responds with the 'unnumbered acknowledgement' (UA). The UA ends the run command in the local (CSDKRNT) and passes control to the SIM terminator (CSDKRST). The SIM terminator checks that an UA was received and issues an XIO LINK to send the load initial PIU to the remote controller. The 'load data' and 'load final' commands that follow are all processed through the local NCP's physical services to the remote PIU decoder (CSDKRPD), which issues XIO LINK commands and sends them to the remote controller. Figure 9.5 illustrates the sequence of commands for loading a remote NCP.

After the load final PIU is sent, a contact is sent by the host SSCP to the local NCP. The local NCP issues a 'contact poll' to the remote controller (send SNRM). On receiving the SNRM, the remote load/dump program passes control to the remote NCP which has been loaded. The remote NCP responds with an UA to the local NCP. The local NCP sends a contacted response PIU back to the SSCP indicating that the remote is loaded.

A remote NCP may have one owner. All links and resources in a remote are owned by the owner of the NCP.

Dumping a Remote NCP

If a printout of remote storage is to be made, the SSCP sends a dump request to the local NCP physical services, which forwards the 'dump initial', 'dump data', and 'dump final' network commands to the remote controller. Figure 9.7 illustrates the command sequences for a dump.

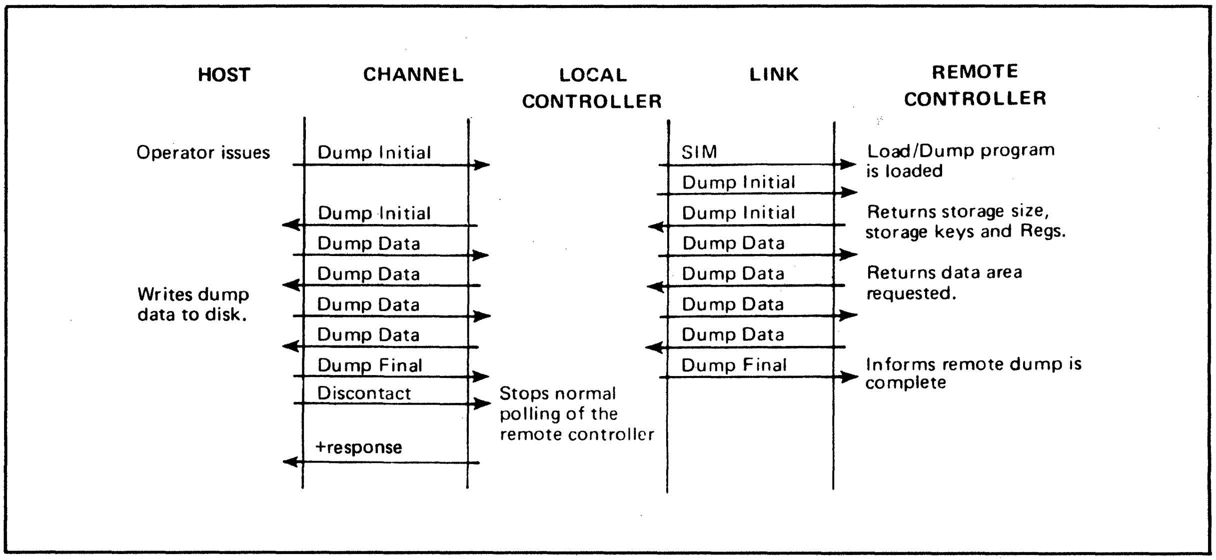


Figure 9.7. Command Sequence to Dump a Remote

The processing of the dump commands is similar to the load process using the remote PIU decoder (CSDKRPD) and the remote SIM terminator (CSDKRST). The dump data requests are sent to the remote load/dump program which returns the requested data area. The local NCP returns the dump data PIUs to the SSCP for writing to a host disk dump file. After the 'dump final' command is sent to the remote, a 'discontact' command is sent to the local NCP to stop normal polling of the remote controller.

Link Failure to a Remote

If a load is to be performed due to a permanent link failure, the SSCP activates the alternate link. Once the alternate link has been activated, the load and dump process is the same as described above. Figure 9.8 illustrates the command sequence for a recovery from a link failure and loading of a remote NCP.

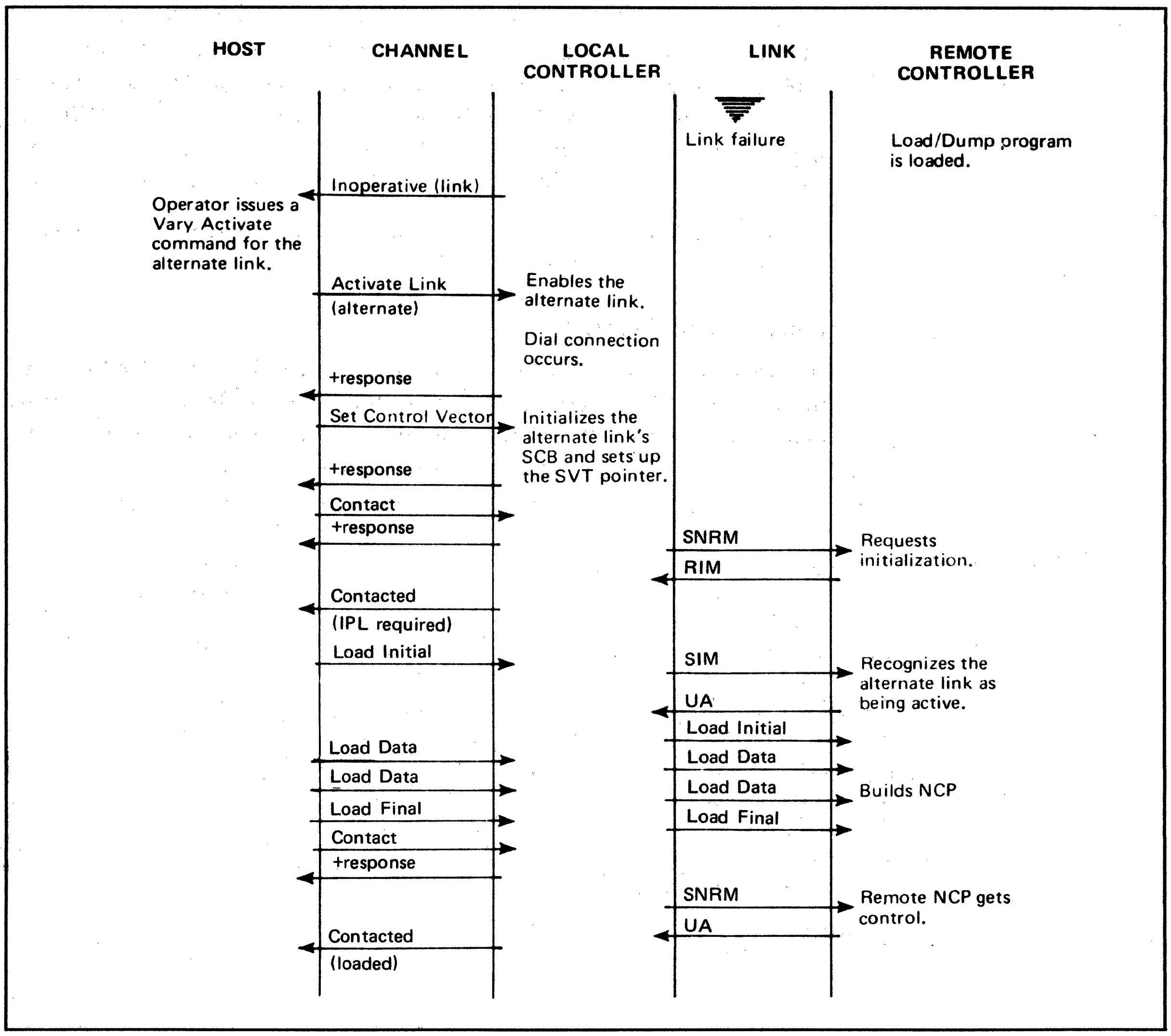


Figure 9.8. Command Sequence for Alternate Link

Remote Power Off

The SSCP may power off a remote controller by issuing a 'remote power off' command to the local NCP physical services. Function management remote PIU decoder (CSDKRPD) sets up the SCB for the remote to send a SIM to the remote NCP. Upon receiving the SIM, the remote NCP link scheduler causes an abend condition to load the load/dump program. Figure 9.9 illustrates the command sequence.

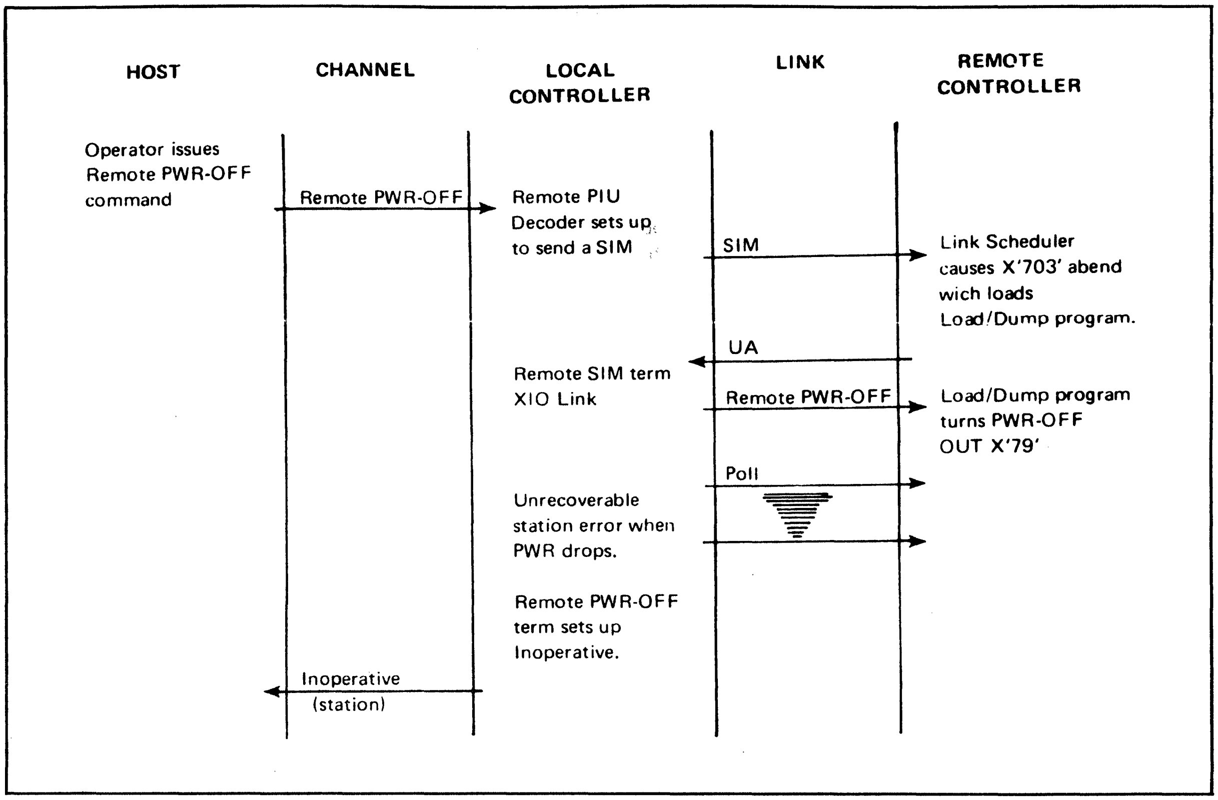


Figure 9.9. Command Sequence for Remote Power Off

The load/dump program responds to the local NCP with a 'unnumbered acknowledgment' (UA), causing the remote SIM terminator in the local to get control. The remote SIM terminator issues an XIO LINK to send the 'remote power off' command to the remote controller. The load/dump program checks for a 'remote power off' command and, finding it, issues an OUT X'79' instruction to power off the controller.

Local/local and Local/remote Summary

The type 4 PU station control block (SCB) represents an NCP on a link.

The local/local definition is a definition of equals; however, one end is primary (polling) and the other secondary (polled). A local may have eight concurrent owners, with ownership via a channel or link. A secondary may be switched to primary by a host command to allow the original primary to be loaded as a remote.

The local/remote definition connects a channel attached primary to a link attached secondary with a remote program loader (RPL). The load, dump, or power off a remote controller are all directed through physical services in the local NCP. A remote may have one owner. Once the remote is active, the sessions must be established for the remote physical services, CUBs and LUBs as for the local. Session commands and data PIUs directed to the remote are queued by INN path control in the local to the link outbound queue of the remote. PIUs received in the remote are routed by INN path control in the remote.

All inbound PIUs in the remote are queued on the SCB link outbound queue for the local. PIUs received in the local from the remote a passed to BNN path control in immediate and then to INN path control.

Chapter 10:

BSC/SS Processor

BSC/SS Introduction

The BSC/SS Processor was originally provided in NCP release 1 and 2 before SNA support. The code has been modified to provide support for BSC and SS devices in an SNA network.

The primary addition in ACF/NCP provides recording of ownership of a BSC or SS line in the line control block (LCB) at offset X‘41’. All resource on a line are owned by the owner of the line. A BSC/SS device can be 'loaned' for a cross-domain session, however all BSC/SS devices sessions are ended when the path to the owner fails.

Note: BSC 3270 sessions end when the path to the owner fails.

Definition of the BSC/SS Processor

The BSC/SS processor is that part of the NCP that processes requests for BSC/SS resources. Instead of the PIU, the basic unit of work is the basic transmission unit (BTU). Therefore, the BSC/SS processor must convert a FID0 PIU received from the host to a BTU and convert a BTU destined for the host to a FID0 PIU.

Processing within the BSC/SS processor is totally different from SDLC support for SDLC resources in boundary network node support. The routing of information to a BSC/SS resource includes the system router, command processor, work scheduler, I/O line task (including I/O line subtasks), and character-service routines. Also, in addition to using a BTU instead of a PIU, many of the queues and control blocks are different.

This topic presents the data format, control blocks, components, and data flow used in the BSC/SS processor for communicating with BSC/SS resources and BSC/SS supporting routines.

BSC/SS Major Control Blocks

Block Control Unit (BCU)

When the BSC/SS router receives a FID0 PIU, the PIU/BTU converter builds the block control unit (BCU) from the first NCP buffer of the FID0.

The BCU consists of:

- X‘00’ Buffer prefix
- X‘08’ Event control block
- X‘14’ Work area
- X‘20’ Basic transmission unit

The BTU contains 14 bytes of control information from the FID0 and may contain text. The BCU may be contained in one buffer or in many buffers, depending on the size of the buffers and the amount of text in the BTU.

Figure 10.1 illustrates the offsets from the first NCP buffer for a conversion to or from FID0 and BTU.

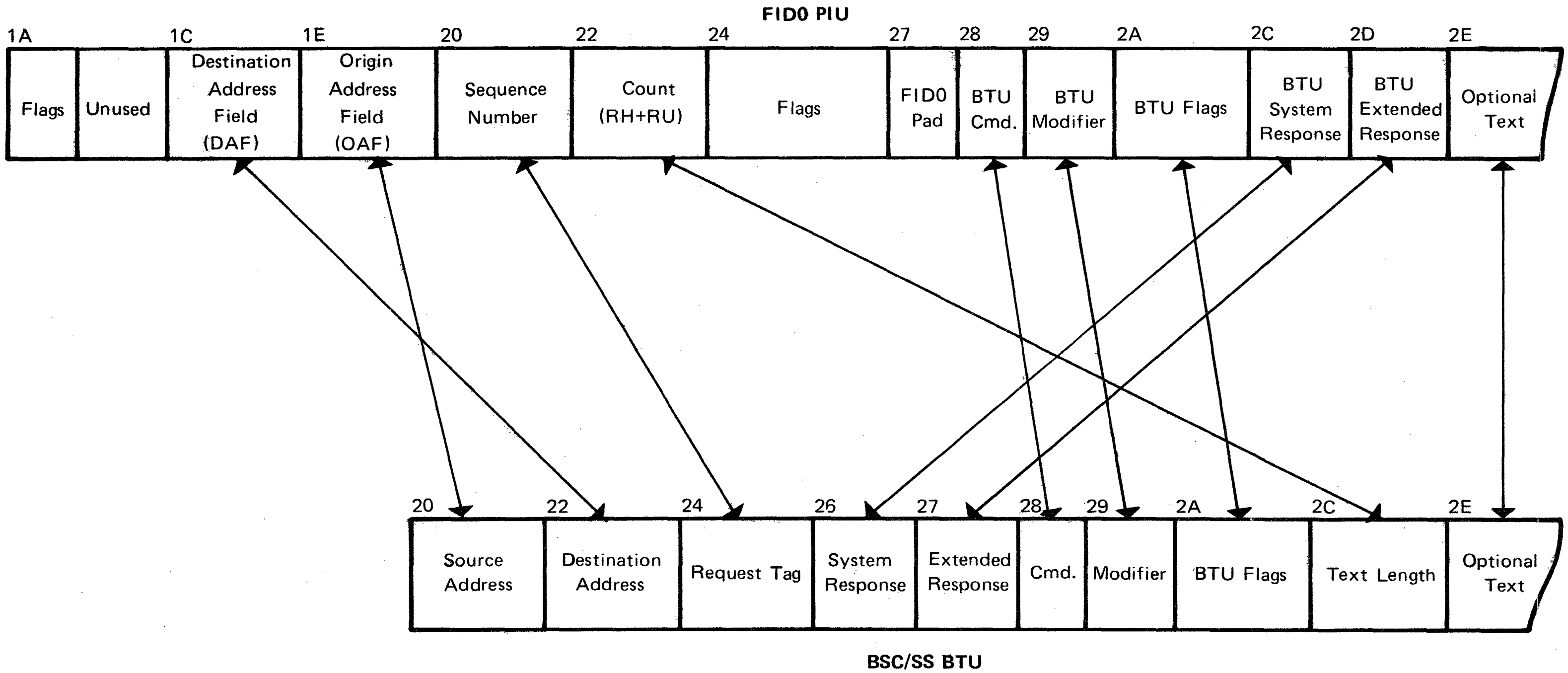


Figure 10.1. FID0/BTU Conversion

BTU commands and modifiers are covered later in this topic.

Resource Vector Table (RVT)

The BSC/SS portion of the RVT contains an entry for each LINE, CLUSTER, TERMINAL, and COMP macro in the BSC/SS portion of the NCP generation. Each LINE macro causes an entry to be built describing the type of entry and containing a pointer to the LCB representing that line. Each TERMINAL, COMP, or BSC/SS CLUSTER macro causes an entry to be built describing the type of entry and containing a pointer to the DVB representing that entry. The entries are built as the macros are encountered in the generation.

The format of the resource vector table (RVT) was described in the section on INN path control. Figure 5.3 illustrates an RVT with BSC/SS and SDLC resources.

Line Group Table (LGT)

The line group table (LGT) is generated by the GROUP macro. BSC groups generates a X'36' LGT. SS groups generates a X'33' LGT. At LGT byte 0 identifies the BSC type or SS terminal type.

Figure 8.1 illustrates the LGT without the BSC and SS extensions. BSC and SS extensions begin a offset X'17' and contain control characters.

Line Control Block (LCB)

At NCP generation time, a line control block (LCB) is built for each BSC/SS line connected to the controller. The LCB contains information required for scheduling line operations. The LCB also has fields for maintaining line significant status information and three queue control blocks: (1) line I/O queue, (2) line work queue, and (3) the suspended sessions queue when the LCB represents a multipoint line. Depending upon the line type, the LCB may have nonswitched point-to-point, multipoint, or switched extension.

The key fields of the line control block (LCB) are illustrated in Figure 10.2.

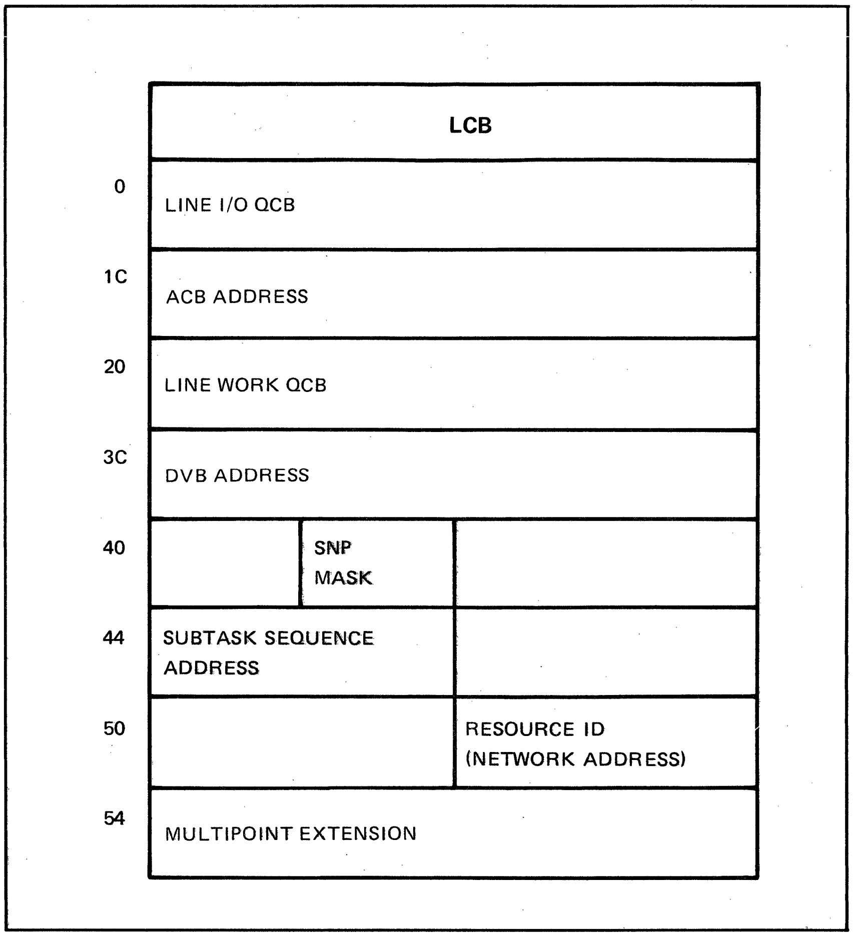


Figure 10.2. Line Control Block (LCB)

Line Type Command Table (LTCT)

The LTCT contains the system command table, the offset table, and a collection of subtask sequence tables. The system command table is a table of all valid BTU command/modifier combinations. The line work scheduler finds the position in the system command table corresponding to the command and modifiers specified in the BTU. The corresponding position of the offset table gives the offset to the appropriate entry in the subtask sequence table.

BTU commands and modifiers are covered later.

Adapter Control Block (ACB)

At NCP generation, an ACB is built for each line defined in the NCP. A BSC/SS ACB contains an input/output block (IOB) and a character control block (CCB). All ACBs are located in the first 64K of storage.

The ACB consists of:

- -X‘08’ Auto call prefix (ACU)
- X‘00’ Input/output block (IOB)

- X'24' Character control block (CCB)

Figure 10.3 illustrates some of the key fields of an ACB.

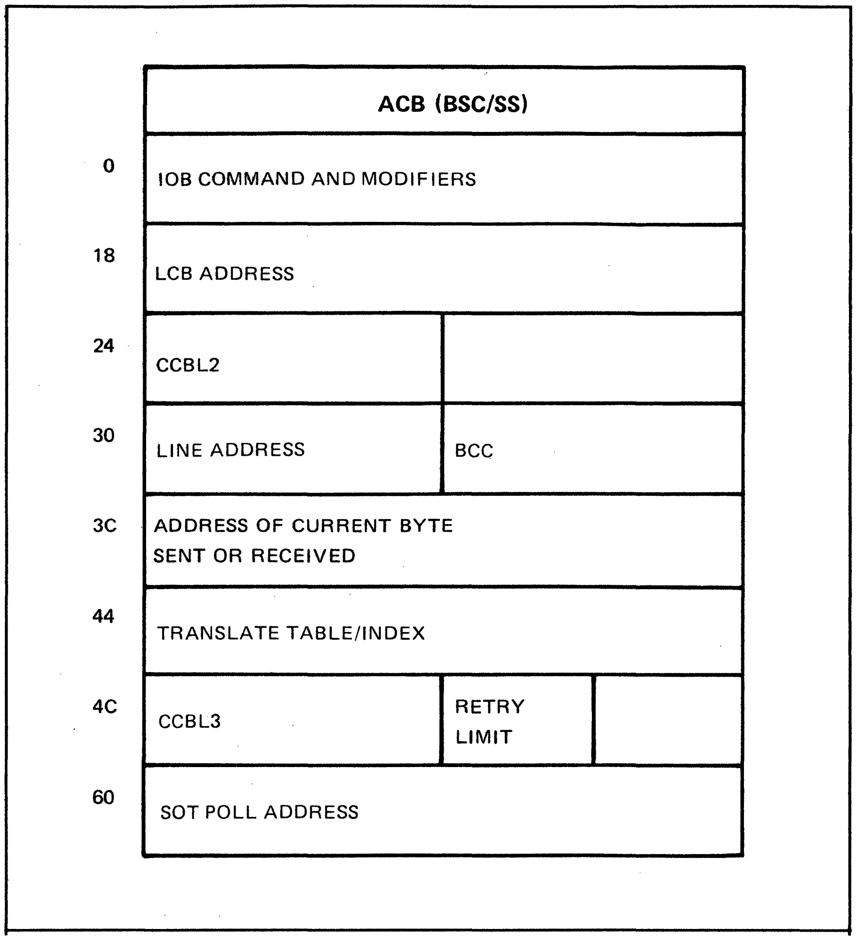


Figure 10.3. BSC/SS Adapter Control Block (ACB)

Input/output Block (IOB)

The input/output block (IOB) is contained within the adapter control block (ACB) at offset X'00'. The IOB contains the command and modifier to indicate the I/O operation to be performed. The IOB also contains status fields to indicate the outcome of the operation, and pointers to the beginning point and ending point of data sent or received, if any data is present.

Some of the key fields of the input/output block are:

- X'00' Flags, I/O command and modifiers
- X'0C' Pointer to first buffer in the block
- X'18' Pointer to the line control block (LCB)
- X'1C' Pointer to last buffer in the block
- X'21' Partitioned emulation (PEP) flags

Character Control Block (CCB)

The character control block is contained in the adapter control block (ACB) at offset X'24'. The CCB contains current information on the physical operation of the line and the data being transferred to or from the line. Some of the contents of the CCB are a pointer to the translate/decode tables, a CCBL2 pointer, a CCBL3 pointer, and counters that maintain the position of data being accessed within buffers.

Some of the key fields of the CCB are as follows:

- X'24' Address of current level 2 character-service routine (CCBL2)
- X'30' Line vector table (LNVT) entry address
- X'44' Address of receive translate table
- X'46' Leftmost byte of transmit translate table (rightmost byte is character to be translated)
- X'4C' Address of next level 3 routine (CCBL3)

Line Vector Table (LNVT)

The line vector table (LNVT) is generated from the CSB macro and initialized by the LINE macro.

The line vector table (LNVT) generates a two-byte entry for each possible line address (96) for each defined scanner (CSB macro). A single scanner generates 96 halfword entries from X'800' to X'8BF'. Each subsequent CSB macro reserves an additional 96 halfwords. An undefined line address has the rightmost bit set to 1 in a halfword entry. A bit of 0 indicates that the halfword contains the address of the adapter control block (ACB) for this line. Because the first scanner has 64 lines starting at line address X'20', the first X'20' entries from X'800' to X'83F' are set as invalid.

If a line address is known, the LNVT entry is calculated by multiplying the line address by 2 and adding X'800'. The LNVT allows the level 2 routines to find the ACB (and CCB) for a line when only the line address is known.

User line control definitions use the user line vector table (ULVT) for LNVT functions. User line control address are indicated as undefined in the LNVT.

Figure 8.4 illustrates the LNVT.

Service Order Table (SOT)

The BSC/SS service order table (SOT) is generated by a SERVICE macro to identify the sequence of service given to devices on a multipoint line. A pointer in the line control block (LCB) at X'4C' points to the current entry in the table for service. All BSC/SS multipoint lines have a service order table.

Figure 10.4 illustrates the BSC/SS service order table (SOT).

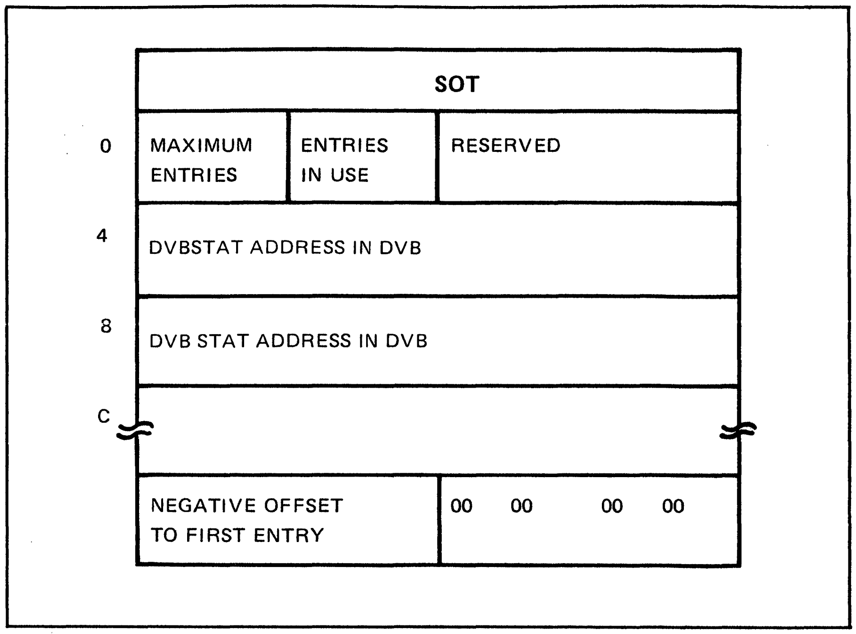


Figure 10.4. BSC/SS Service Order Table (SOT)

Device Base Control Block (DVB)

The device base control block (DVB) contains an input QCB for the device input queue and a work QCB for the device work queue, as well as all parameters needed to operate a device. One DVB is built at NCP generation time for each CLUSTER, TERMINAL, and COMP macro coded (except for CLUSTER macros coded without the GPOLL operand). The DVB may have one or more external extensions, depending on the type of device and the features of the device represented.

Figure 10.5 illustrates the key fields of the DVB.

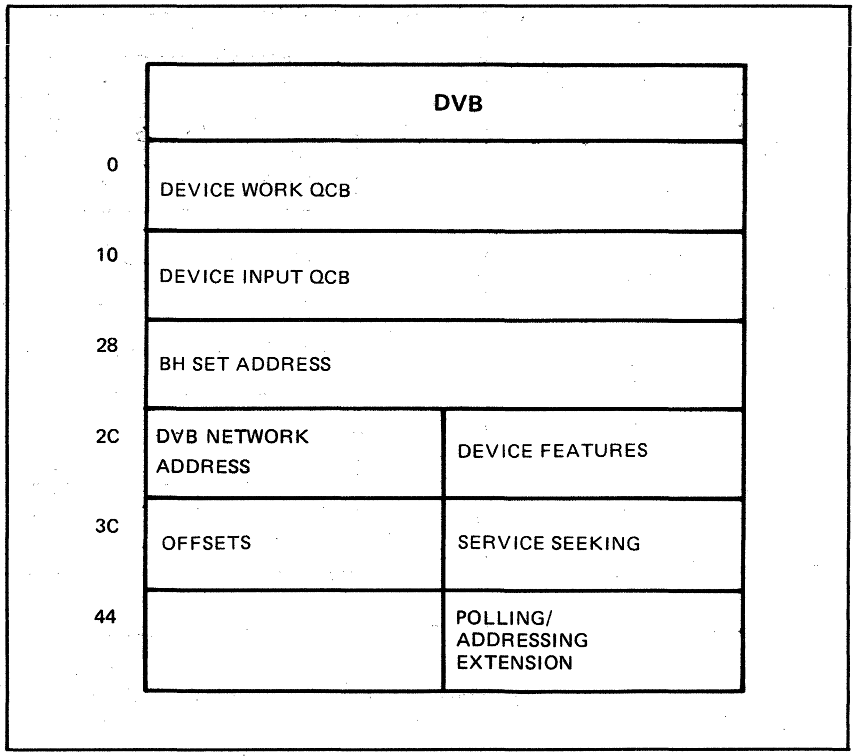


Figure 10.5. Device Base Control Block (DVB)

There are variable extensions to the DVB, depending upon the options selected when the generation definition is coded. Following are the control block extensions to the DVB (offsets to the extensions, if included, are in the DVB from X'38' to X'3D'. The format and values of the extensions can be found in the *ACF/NCP/VS Network Control Program - Program Reference Summary* (LY30-3043) under 'DVB'.):

- BHR Block handler routine extension
- BUE Switched backup extension
- CGP Cluster general poll extension
- CIE Callin extension
- COE Callout extension
- DAE Device addressing extension

BSC/SS Processor Components

Processing within the BSC/SS processor is totally different than in the SDLC support for SDLC resources. The routing of information to a BSC/SS resource includes the system router, command processor, work scheduler, input/output line task (including input/output subtasks), and character-service routines.

This section describes the BSC/SS processor components, providing information about the functions each component performs, the control blocks each component uses, and the manner in which each component passes control to the next. Figure 10.6 illustrates the components and processing flow.

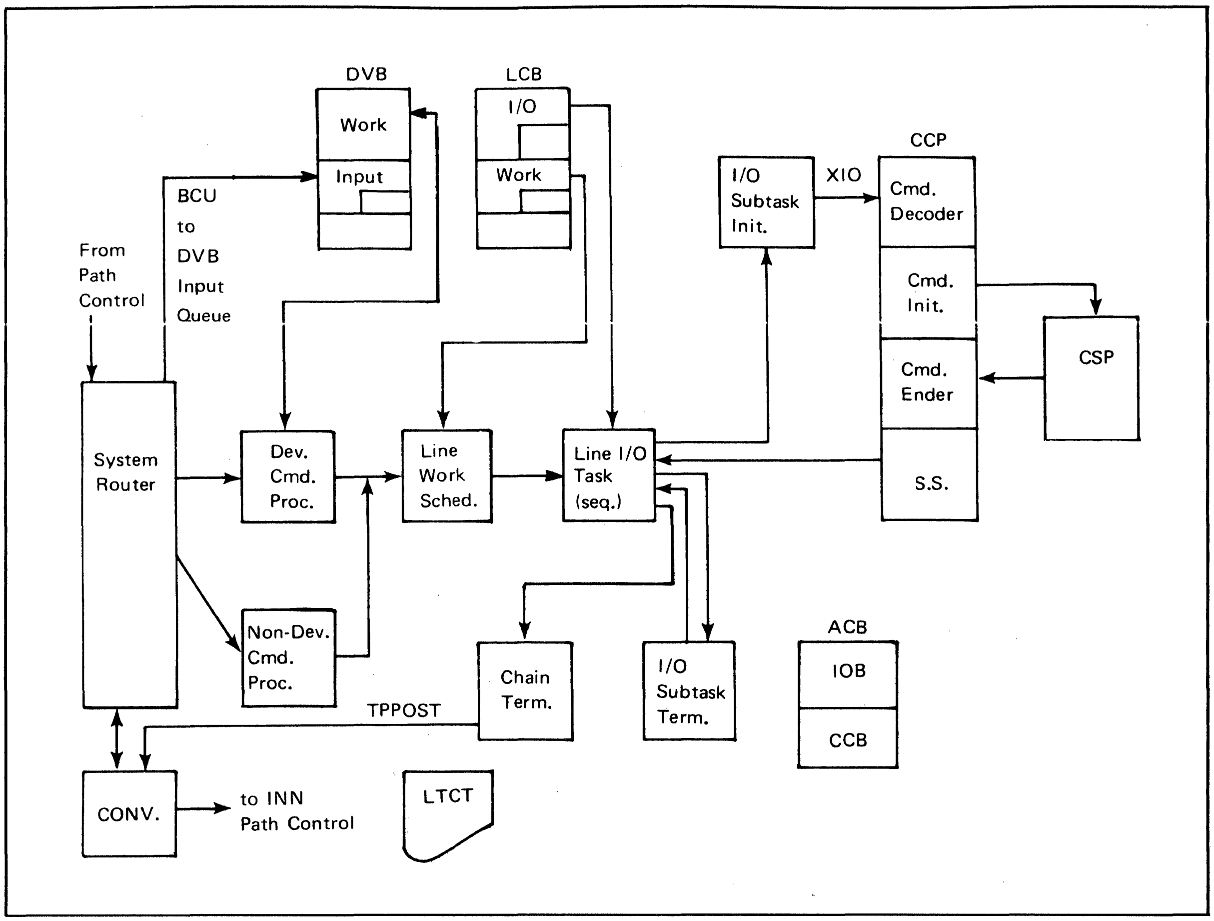


Figure 10.6. BSC/SS Processor Components

System Router

The system router receives control and data PIUs from INN path control via a branch. The system router branches to the PIU/BTU converter and, after conversion, control is returned to the system router. The system router resolves the BCU resource ID using the BSC/SS portion of the RVT. From the RVT, the system router obtains resource type information and the address of the control block representing this resource. The control block may be a device base control block (DVB) or a line control block (LCB). A DVB represents a terminal, component, or a BSC/SS cluster (these are considered to be device-type resources). An LCB represents a line (nondevice).

If the resource is a device, the system router enqueues the BCU on the input queue of the DVB representing that resource. If the command and modifier for the device indicate a critical control command (bit 1=1), the BTU is enqueued on the devices input queue ahead of data and noncritical control commands. The ENQUE macro used contains the ACTV=YES operand which results in a trigger.

Nondevice Command Processor

NCP releases 1 and 2 provided NCP physical services functions on a queue which is called 'nondevice input queue'. All supported functions have been transferred to NCP physical services, however, the nondevice input queue is still defined. If a FID0 was directed to a link, not a device, the system router would enqueue the BCU on the non device input queue. There is only one nondevice input queue, the address of which is in the extended halfword direct addressable (HWE) at X'2C'.

Device Command Processor

If the system router enqueued a BCU on the input queue for a DVB, the device command processor is triggered. The device command processor validates the BTU command and modifiers, dequeues the BCU from the DVB input queue, enqueues it to the DVB work queue, and triggers the line work scheduler.

Line Work Scheduler

The line work scheduler uses the BTU command and modifiers as a search argument against the line-type command table (LTCT). In the LTCT, the chain of subtasks necessary to process the BTU command and modifiers is found. The line work scheduler also dequeues the BCU from the DVB work queue, enqueues it to the line I/O queue, and triggers the line I/O task.

Line I/O Task

The line I/O task is made up of the sequencer and line I/O subtasks.

The sequencer simply gives control sequentially, as required, to the line I/O subtasks contained in the selected chain.

The line I/O subtask chains are made up of pairs of subtask initiators and subtask terminators (different pairs according to the BTU command and modifiers), plus a chain terminator (read or write version).

Each I/O subtask initiator stores an IOB command in the IOB (contained in the ACB for a BSC/SS line) and issues an XIO macro to pass control to the communications control program (CCP), which runs in level 3.

After the CCP and level 2 processing is completed for a given IOB command, each I/O subtask terminator gets control when the line I/O sequencer is triggered by the CCP. The I/O subtask terminator checks to see if the command completed successfully; if so, the terminator passes control back to the sequencer, which gives control to the next I/O subtask initiator or the chain terminator for this chain.

The chain terminator updates the response field in the BCU and issues a TPPOST macro which branches to the BTU/PIU converter. The converter then passes the PIU to the channel adapter I/O supervisor.

Communications Control Program

The BSC/SS CCP is made up of (1) the command decoder, (2) the command initializer, (3) the command ender, and (4) the BSC/SS service-seeking module.

The command decoder receives control from the XIO macro issued in an I/O subtask initiator. The decoder selects the proper initialization routine by using the command that was placed in the IOB. The command decoder then passes control to the command initializer.

The command initializer initializes the CCB (contained in the BSC/SS ACB) and the communication scanner in whatever way is necessary to accomplish the level 2 processing for the IOB command. When the command initializer is finished, level 2 interrupts begin to occur on this line for the IOB command.

When level 2 has finished processing the IOB command, the command ender receives control via a level 3 PCI initiated by level 2. The command ender checks whether a good completion occured; if so, it triggers the line I/O task.

Character Service Program (CSP)

The CSP processes level 2 interrupts from the communications scanner. Processing initially begins according to how the command initializer sets up the CCB level 2 pointer. From then on, CSP updates the CCBL2 pointer as required. For a type 2 scanner (CSB macro-coded with an operand of TYPE=TYPE2), the CSP moves a character at a time into the scanner's ICW for a given line (for 'write' operations) or removes a character at a time from the ICW for a given line (for a 'read' operation). For a type 3 scanner (CSB macro coded with an operand of TYPE=TYPE3), the data characters are transferred by cycle steal to the end of an NCP buffer or end of block. When CSP processing is complete for an IOB command, a level 3 PCI is issued to pass control back to the CCP.

BTU Commands for BSC/SS Resources

The basic transmission unit (BTU) is the unit of transfer within the BSC/SS processor. Data that passes between the host and the BSC/SS processor must be converted between the PIU and the BTU formats. In the buffer, the BTU is contained within the block control unit (BCU).

For data transfer to or from the BSC/SS resources, the BSC/SS processor uses three units of transfer: block, message, and transmission.

A block is the smallest unit recognized by the network control program. For SS devices, the data between two end-of-block (EOB) characters; for BSC devices, the data between a start-of-text (STX) or start-of-header (SOH) character and an end-of-transmission block (ETB).

A message for SS devices is the same as a transmission, that is, the data between a start-of-data (circle D) and end-of-block (EOB), end-of-transmission (EOT); for BSC devices a message is the data between a start-of-text (STX) or start-of-header (SOH) character and ended by an end-of-text (ETX) character.

A transmission for SS devices is the same as a message. For BSC devices, a transmission is ended by an end-of-transmission (EOT).

For large amounts of data coming from a terminal, the BSC/SS processor can run in subblocking mode. The BSC/SS processor passes data to the host as a subblock before receiving an EOB, ETX, or EOT. Subblocking is in full multiples of NCP buffers, as specified on the TRANSFER operand of the LINE macro, and CUTOFF specifies the number of subblocks before the data is flushed. The BSC/SS processor automatically sends data to the host on an EOB or ETX.

There are five BTU commands that are used with all BSC/SS devices. The commands are: 'invite', 'contact', 'read', 'write', and 'disconnect'. Even though the physical operation may be different for each device, the same commands are used. Each of these commands is discussed in detail later in this manual.

The 'invite' and 'contact' commands establish a BSC/SS session. The 'invite' command implies a 'read'. The 'read' and 'write' commands transfer data between the host and a device. The 'disconnect' command ends a session. By the use of command modifiers a number of commands can be combined into one request from the host.

There are two other commands the host can send to the BSC/SS processor: 'control' and 'test'. The 'control' command is used to alter or examine the status of a line or device. The 'test' command is used to test a BSC/SS device or line. The online line test (OLLT) tests BSC/SS lines; the online terminal test (OLTT) tests BSC/SS terminals. For these tests, the text portion of the BTU contains interpretive commands.

The BTU response information is contained within the system response byte of the BTU and the extended response byte of the BTU. The system response byte identifies a response as an error response or normal response. The system response byte also contains the phase (0, 1, 2, or 3) to which this response applies and the system response code. The extended response byte contains the initial status of the line and the final status of the line.

The BTU commands given below may be found in *ACF/NCP/VS Network Control Program - Program Reference Summary* (LY30-3043), Section 3: BTU Commands and Modifiers.

Contact Command

The 'contact' has a BTU command of X'06'. The modifiers are:

- X'00' Contact normal
- X'01' Return resource ID of line used to establish the dial connection

The 'contact' does not imply any data transfer, but only assures that a connection is available for data transfer. When a response to a 'contact' is received by the host, the host may then issue either a 'read' or 'write' command. Control commands, such as 'reset device queues' or 'reset immediate' can request termination of a 'contact' command.

Invite Command

'Invite' has a BTU command of X'05', with several modifiers available to qualify the 'invite', as follows:

- X'00' Invite normal. Unit of data for this command (block, message or transmission) is specified on the NCP macro defining the device. Default is block.
- X'01' Invite block. Unit of data for this command is a block. Ended by EOB.

- X'02' Invite message. Unit of data for this command is a message. Ended by ETX (BSC) or EOT (SS). Message and transmission are the same for SS.
- X'03' Invite transmission. Unit of data for this command is a transmission. Ended by EOT (BSC).
- X'04' Invite transmission with disconnect. Executed as an 'invite transmission' command followed by a 'disconnect' command.
- X'05' Invite with auto restart. Executed as unbounded series of 'invite with disconnect' commands. This command must be terminated with a 'reset' command.
- X'06' Invite perpetual. Valid only for clusters. Executed as an unbounded series of 'invite transmission' commands with no intervening 'disconnect' commands.

If an 'invite' is pending (no response from the terminal), and data is available to send to the device, the 'invite' can be terminated by control commands of 'reset invite', 'reset conditional', or 'reset at end of command'.

A 'write' to a BSC 3270 occurs without a reset of the 'invite perpetual'.

Read Command

'Read' has a BTU command of X'01' with several modifiers, as follows:

- X'00' Read normal. Unit of data for this command (block, message, transmission) is specified on the NCP macro which defines the device. Default is block.
- X'01' Read block. Unit of data for this command is the block. Ends with an EOB.
- X'02' Read message. Unit of data for this command is the message. Ends with an ETX (BSC) or EOT (SS). The message and transmission are the same for SS.
- X'03' Read transmission. Unit of data for this command is a transmission. Ends with an EOT (BSC).
- X'04' Read transmission with disconnect. Executed as a 'read transmission' command followed by a 'disconnect' command.
- X'05' Read with invite. Executed as a 'read transmission with disconnect' followed by an 'invite normal' command.

The read command can be terminated by a 'reset device queues', 'reset immediate', 'reset conditional', or 'reset at end of command'.

Write Command

'Write has a BTU command of X'02' with several modifiers, as follows:

- X'00' Write normal. Unit of data is one block. Ended by an EOB.
- X'01' Write with end-of-message. Unit of data is one block followed by the appropriate control sequence for an end of message.

- X'02' Write with end of transmission. Unit of data is one block followed by the control sequence for end of transmission.
- X'03' Write with disconnect. Executed as a 'write transmission' command followed by a 'disconnect command'.
- X'06' Write with read. Executed as a 'write with end of transmission' followed by a 'read' command.
- X'07' Write with invite. Executed as a 'write with end of transmission' followed by a 'disconnect' command and then an 'invite' command.
- X'08' Write with contact. Executed as a 'contact' command followed by a 'write normal' command. Ended with an EOB.
- X'09' Write with contact. Executed as a 'contact' command followed by a 'write with end of message'. Ended with ETX (BSC) or EOT (SS).
- X'0A' Write with contact. Executed as a 'contact' command followed by a 'write with end of transmission'. Ended with EOT.
- X'0B' Write with contact and disconnect. Executed as a 'contact' command followed by a 'write with end of transmission' followed by a 'disconnect' command.
- X'0E' Write with contact and read. Executed as a 'contact' command followed by a 'write with end of transmission' followed by a 'read normal' command.

The 'write' command can be terminated by 'reset device queues', 'reset immediate', 'reset conditional', or 'reset at end off command'.

Disconnect Command

'Disconnect' has a BTU command of X'07' with several modifiers, as follows:

- X'00' Disconnect normal. No modifier
- X'01' Disconnect with invite. Executed as a 'disconnect normal' followed by an 'invite normal' command.
- X'02' Disconnect with end of call. For switched lines, this modifier results in the physical connection between the terminal and the communications controller being broken. For nonswitched lines, this modifier is the same as 'disconnect normal'.
- X'03' Disconnect with end of call and invite. Executed as a 'disconnect with end of call' followed by an 'invite' command.

The 'disconnect' command is reset by 'reset immediate'.

Control and Test Commands

The control commands have a BTU command of X'08' with many modifiers. The test commands have a BTU command of X'03' with many modifiers. A listing of commands and modifiers is given in *ACF/NCP/VS Network Control Program - Program Reference Summary* (LY30-3043), Section 3: BTU Commands and Modifiers.

BSC and SS Sessions

A general discussion of BSC and SS sessions is available in *ACF/NCP Programming* (SR20-4620) in the topic BSC and SS Sessions.

The ability of the NCP BSC/SS processor to conduct multiple sessions on the same multipoint line depends upon the fact that data transfer does not occur continuously for the duration of a session. For example, for inquiry/response applications, 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. This elapsed time is the result of operator 'think' time. The interval during which the terminal is not using the line can profitably be used to service other terminals on the same line.

The number of concurrent sessions to be conducted on a line depends upon several factors. Among these are (1) the relative amount of time a terminal in use does not need the line, and (2) the permissible delay between the time the operator is ready to use the terminal and the time the line is available to that terminal. The number of concurrent sessions on a line is specified by the user in the SESSION operand of the LINE macro. This value is called a session Limit.

The sequence by which the BSC/SS processor attempts to establish sessions on a multipoint line is determined by the service order table associated with the line. This table is defined by the SERVICE macro, directly following the LINE macro.

Logical Connections

A session is active when the BSC/SS processor is communicating with, or is ready to communicate with, the associated device. If the NCP is not communicating with, or is not ready to communicate with, the associated device, the session is either suspended (but within an active session) or inactive.

In most applications it is necessary to limit the amount of time a session is permitted to be active in order to prevent a device, once in session, from monopolizing the line. The period during which a session is active is called a logical connection. The length of a logical connection is the maximum number of transmissions that may be transferred in either direction between the BSC/SS processor and the device during the logical connection. The limit is specified in the XMITLIM operand of the CLUSTER, TERMINAL, or COMP macro representing the device. The user can indicate that the XMITLIM specifies the number of blocks, rather than transmissions, by coding ENDTRNS=EOB on the macro.

Once a session has been established, the BSC/SS processor repolls the device for each subsequent transmission solicited from the device. 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. The number of polling operations allowed during this period is specified in the POLIMIT operand of the LINE macro. The value specified in the POLIMIT operand is called the negative response limit. The best performance results occur with a value of 1.

Once the negative response limit is reached, the BSC/SS processor can proceed in one of three ways:

1. NOWAIT. The BSC/SS processor breaks the logical connection and cancels the read request that caused the polling. The host is informed.

2. WAIT. The BSC/SS processor maintains the logical connection, holding the line; informs the host the negative poll limit has been reached, and waits for a new command from the host. This operand is required for the IBM 3735.

3. QUEUE. The logical session is suspended, the command is queued for the next logical connection for this device, and the host is notified that the negative poll limit was reached.

Most types of I/O errors that occur during an active session cause suspension of that session. The host is notified of the error.

Session-Servicing and Service-Seeking

The activity of attempting to establish a new session on a BSC or SS line is called service-seeking. Service-seeking occurs by searching all DVBs for an 'invite' or 'contact pending' bit in DVBSTAT, in the sequence of the service order table (SERVICE macro). If either the 'invite' or 'contact pending' bits have a value of 1, the device is polled or addressed (if it is a polled device) or otherwise enabled for communication. These bits are 0 once a session is established, but the 'connection exists' bit indicates that 'read' or 'write' commands may be processed during a logical connection. The 'disconnect received' bit in DVBSTAT, set when a 'disconnect' command is received, indicates that the session is to be terminated.

The activity of servicing existing sessions is called session-servicing. Session-servicing occurs by sequentially servicing all of the DVBs queued on the suspended session queue in the LCB for that line. Servicing a session consists of establishing a logical connection, then sending or receiving data (or both) until the logical connection ends.

Session-servicing and service-seeking alternate in a sequence of operations called a service cycle. A service cycle consists of service-seeking and session-servicing if at least one session exists. If no sessions exist, only service-seeking is performed. If the existing sessions equal the session limit, only session-servicing is performed.

The maximum number of devices with which the program attempts to establish a session during each service-seeking operation is called the service-seeking limit. To specify the service-seeking limit, the SERVLIM operand of the LINE macro should be coded with the maximum number of devices with which the program is to attempt service-seeking during one service-seeking operation. Service-seeking attempts are in service order table (SOT) sequence for this count, even if the DVBs searched are currently in session; each DVB is scanned for an 'invite' or 'contact pending'. If response time is poor for existing sessions, you may improve performance by coding SERVLIM with a low value; the default is one-half the entries in the service order table. The default will normally provide best total performance.

You may also specify whether service-seeking or session-servicing is to have priority. This option is specified by coding SERVPRI=OLD if session-servicing is to have priority, or SERVPRI=NEW if service-seeking is to have priority. If response time is poor for existing sessions, you may improve performance significantly by coding SERVPRI=OLD.

Nonproductive polling and the associated processing overhead can be minimized by specifying a service-seeking pause. The pause is in effect only when there are no established sessions and only service-seeking occurs for this line. The pause is specified in the PAUSE operand of the LINE macro. When the first session is established, the pause becomes inoperative until all active sessions have terminated for this line.

Session information can be changed by commands from the host. The control command (X'08') with the following modifiers can be used for dynamic tuning of the network:

- X'84' Change line service-seeking pause
- X'85' Change line negative poll response limit
- X'86' Change session limit
- X'8C' Change device transmission limit

The fields which are changed by these commands are in the line control block (LCB) in the multipoint extension. These values can also be changed from the 3705 control panel.

In specifying session limits, special consideration must be given to devices which use general polling. The BSC 3270 uses the general poll ('invite perpetual') to obtain input. A response to a general poll may include data from all 3277s on the cluster controller, exceeding the session limit. If the session limit is reached by a general poll of one cluster, other clusters on the same line are not polled. BSC 3270s should have a session limit equal to the total entries in the service order table; one per cluster controller plus one per 3277.

Write operations to BSC 3270s are queued to the DVB which represents the terminal. Read operations occur when the DVB representing the cluster controller is processed for service-seeking.

BSC/SS Flow

The scheduling of the BSC/SS request for execution begins with the device command processor, operating from the device input queue, and is composed of a main routine and several subroutines. The device command processor decodes the command, makes various error checks, depending upon the command, and, if no errors are found, accepts the command. If errors are found, the proper response code is moved to the BTU and is returned to the host.

Accepting the command, the device command processor enqueues the BCU on the device work queue (DVB X'00'). Since the work queue has no executable code associated with it, no task is triggered as a result of the enqueuing. If the line work scheduler is idle, the device command processor must trigger the line work scheduler to ensure that the input/output operation is initiated.

The device command processor also processes control commands directed to a device. If the command is a control command, the device command processor enqueues the BCU to the device work queue, provided the control command is noncritical and the device is in session. If the command is critical, or if the device is not in session, the device command processor passes control to the control router.

The nondevice command processor, operating from the nondevice input queue, processes all control commands that are not directed to a device. The processor dequeues the BCU from the nondevice input queue and uses the resource vector table (RVT) to determine the address of the line control block (LCB) representing that line.

If the control command is supported in the system, the nondevice command processor calls the control router. The control router scans the supported control command tables looking for a match. When a match is found, the control router branches to the routine. When the control command routine has finished its processing, the control router triggers the line work scheduler.

The same line work scheduler gets control regardless of the line type. A different subroutine of this task exists for each type of line: point-to-point (which also supports switched callin), switched callout, and multipoint. The line work scheduler assigns a subtask sequence chain to the request by decoding the command, using the line type command table (LTCT). The LTCT contains an offset table and a collection of subtask sequence tables. The offset table corresponds to the system modifier table (a table of all valid command/modifier combinations). The line work scheduler finds the position in the system command table corresponding to the command and modifiers specified in the BTU. The corresponding position in the offset table of the LTCT gives the offset to the appropriate entry in the subtask sequence table. Each entry in the subtask sequence table is a series of pointers to the I/O subtasks necessary to process a particular command. Each pointer is the fullword address of an I/O subtask.

Once the initial pointer to the required subtask sequence has been established by the line work scheduler, the offset into the subtask sequence is stored in the LCB (LCBSSP) for the line. The line work scheduler then enqueues the request BCU on the line I/O queue.

The next task to get control is the line I/O task which was triggered by the line work scheduler. The line I/O task consists of a line I/O sequencer and a series of initiator/terminator subtasks. The subtasks associated with a line I/O task vary, depending upon the command being processed at any given time, and the type of line, switched or nonswitched.

When the line I/O QCB is activated, the line I/O sequencer receives control. The line I/O sequencer updates the subtask sequence pointer passed to it by the line work scheduler and gives control to the next subtask in the sequence. The initiator/terminator subtasks are structured to perform a series of I/O operations. A complete sequence of subtasks executes all the I/O operations necessary to perform a requested function. Initiator subtasks structure the line's input/output block (IOB) by inserting the required I/O command and modifier codes and initializing other appropriate fields, then issuing an XIO macro to start the I/O operation.

When the I/O operation completes, the terminator subtask checks the I/O completion status, initiates any error recovery procedures, and prepares the line I/O task for the next operation. If a terminator does not initiate any action that requires supervisor dispatching (such as issuing an XIO or TRIGGER macro), control returns to the supervisor via the SYSXIT macro to allow other level 5 tasks to compete for level 5 system time.

BSC/SS XIO Processing

BSC/SS XIO processing is handled by the communications control program (CCP). The CCP for BSC/SS processing consists of the command decoder (CXECMDC), command initializers (CSECMDI), the command ender (CXECEND) and the BSC/SS service-seeking module (CXESVSK). The CCP routines initiate and terminate data transmissions on the line; the character service program accomplishes the actual data transmission.

After an XIO is issued to a communications line by the level 5 I/O subtask, the communications control program (CCP) gets control from the supervisor. One of three decode routines is entered, depending upon which type of XIO was executed. The three types of XIO commands are:

- Normal IOB (CSEMDC0)
- Set mode (CSECMDCI)
- Immediate control (CXECMDC2)

Normal IOB Command (CSECMDC0)

With information about the command in the line IOB, the nucleus routine for decoding normal XIO commands performs a number of initialization steps common to all such commands.

Certain IOB fields (status fields, input block size, and immediate control field) are set to zero, and the connection between the adapter control block (ACB) and line vector table (LNVT) is validated. If the ACB/LNVT connection is invalid, the XIO SVC is abended.

The character control block (CCB) is checked to see if that block is already busy executing a command or subblocked operation. The phase bits in CCBCTL are nonzero if the line is busy. If the CCB is found to be busy, the XIO SVC is abended unless the new command is appropriate for completing an outstanding subblocked operation.

The line adapter (ACB) is placed in the NO-OP state unless the ACB is completing a subblocked operation. This state permits level 2 to be enabled while command initialization steps are performed to the CCB, without interference from level 2 interrupt processing on this particular line.

At the start of command initialization, CCB fields are checked for any outstanding status conditions. Such conditions prevent the new command from being executed at this time. The new command is ended, using the outstanding status as its ending status and using the phase set to indicate the clearing of outstanding status.

The command control byte (CCBNCFL) is checked to see whether command initialization can proceed immediately or must be suspended until (1) something is received from the terminal or (2) a timeout completes.

As far as the decoder is concerned, there are three classes of commands: normal, common control, and subblock mode.

The first class consists of the normal data transfer commands, such as 'read initial' and 'write with end of transmission', since their execution is dependent on the particular type of line control. The initialization routine to be branched to is located through the command decode table pointer of the line group table (LGT) for the line. The initialization routines for these commands reside in the CSECMDI CSECT.

The common control commands include 'enable', 'dial', and 'disable'. Their functions are common to different types of line control, so the initializing routines are the same for all common control commands, which reside in the CSECMDC CSECT.

The subblock mode commands are normally accepted only if the line is busy. The decoder branches to the subblock command initializer, which resides in the CXECMDI CSECT.

The function of the command initializer routines is to examine the command and set up the adapter control block (ACB) with the proper values to handle the I/O on the lines. This module also sets initial timeouts and sets up the interface control word (ICW). Upon completion of level 2 operations, if the expected ending status is satisfied, control returns to the address contained in CCBL3.

Set Mode Command (CSECMDC1)

The entry point into the communications control program (CCP) for an XIO 'set mode' is CXECMDC1. This code validates the 'set mode' command, vectors into the set mode command decode vector table, using parameters passed from level 5, then branches to the routine for execution. When the set mode function has completed, control returns to the level 5 routine via the supervisor.

Immediate Control Command (CXECMDC2)

The entry point into the communications control program (CCP) for an XIO immediate is CXECMDC2. Several different types of resets are provided. Some resets are conditional and some are unconditional. The purpose of the resets is to terminate an IOB command operation or an ongoing subblocking operation. If the reset is successful, storing of received data is halted; if the line is subblocking, the receive buffers are released and the data is lost. If the line is transmitting when the reset is executed, the transmission is terminated in an orderly way.

The reset routine contains a branch tree to determine exactly what type of operation is to be reset: receive text, receive control data, transmit text, or transmit control data. For all 'reset immediate' routines, the linkage through the command ender is established via the CCBL3 pointer to the common reset end routine in CXECEND (CXECENDY), and by zeroing expected status. Then if the reset operation is completed without any hardware errors, the IOB command is ended with IOB status set to 'special' and 'reset', and phase set 'on' in the error flags byte. The phase of 'reset' is always 'control'.

There are two other types of 'immediate control' XIO commands. The first causes the break signal (SS only) to be sent if the line is currently executing a 'read' type IOB command. The second type places the line in monitor mode, provided the line is not executing an IOB command. If not busy or if handling a subblock between commands, the line is set to monitor mode, in which the line triggers the LCB's input/output task if an ending status condition occurs.

While in monitor mode, the line is busy to all IOB XIO commands issued to it. The result of an XIO being issued is to abend the task that executed the XIO, if monitor mode has not ended due to ending status or reset.

Character Service Program (CSP) Flow

The function of the character service program (CSP) is to maintain the line discipline while transmitting or receiving data. There are two types of CSP: one for BSC line control, one for SS line control.

Each CSP is made up of routines to handle the line discipline in addition to the transmission and reception of data. Each CSP routine sets up the CCBL2 pointer for the next function needed to complete the I/O operation.

When the I/O operation has completed, the last level 2 CSP routine that gets control queues the ACB to the CCP ACB queue (CCPQOFF) for processing to the level 3 command ender via a PCI level 3 interrupt.

When the CSP routines have finished processing, the last routine issues a PCI level 3 after queueing the ACB to the ACB queue. The PCI level 3 interrupt passes control to the command ender routine in the CCP in level 3. The command ender removes the ACB from the ACB queue, and, if the queue is empty, resets the PCI level 3.

The command ender compares the ending status of the current command with the expected status stored in the CCBESTAT field. If the two agree, and no error bits were flagged, the routine exits to the routine pointed to by CCBL3. The CCBESTAT and CCBL3 fields were both set during command initialization. If CCBESTAT=0, the command ender automatically accepts the results.

If the ending status does not agree with the expected status, or if any error bits were flagged, the error recovery program (ERP) setup routine schedules the appropriate ERPs.

The ERP setup routine uses the phase bits (CCBCTL) and CCBEND1 to vector to the correct ERP branch table within the command ender. The ERP setup routine branches to the ERP routine, where a check is made for retry limits. If the limit has not been reached, the ERP routine branches to the initialization routine to retry the operation. If the limit is reached, the command is ended and the ERP routine triggers the line I/O task. The line I/O task gives control to the I/O terminator subtask to check whether the second level ERP limit has been reached. If the second level ERP limit has not been reached, the I/O terminator subtask reschedules the I/O initiator subtask. Error recovery continues either until the limit is reached or the I/O operation has completed successfully. In either case the I/O terminator subtask TPPOSTs the appropriate response back to the host.

Figure 10.7 illustrates the flow of the BSC/SS Processor. The command sequence is identified in the numbered points following the figure.

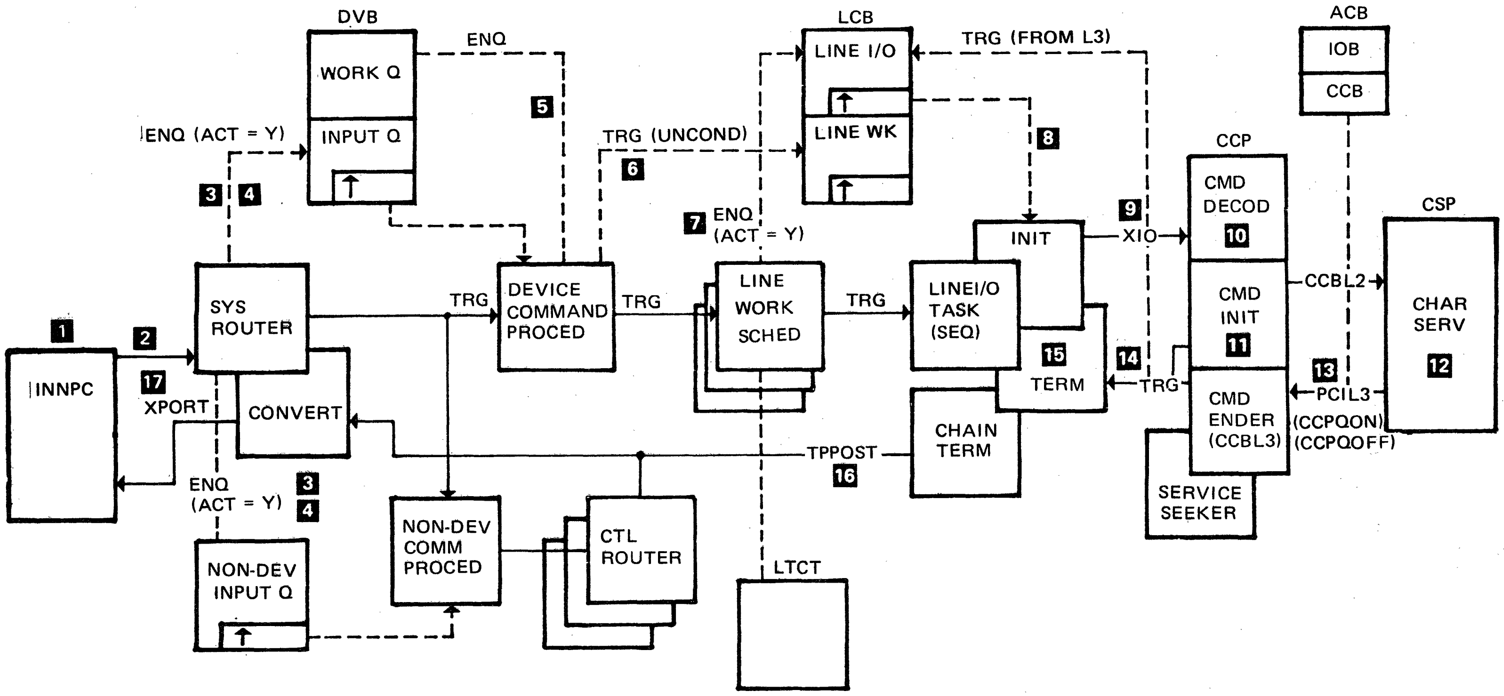


Figure 10.7. BSC/SS Processor Flow

1. INN path control receives the FID0 from (1) channel IOS if the FID0 arrives from a channel, or (2) BNN path control in immediate if the FID0 arrives from a station control block (SCB).
2. INN path control uses the DAF to access the SIT, SVT, and RVT in order to identify a BSC/SS resource. INN path control branches to the system router, which converts the FID0 PIU to a BTU.
3. The system router enqueues the BTU to either (1) a DVB queue, or (2) if the destination is a line, to the nondevice input queue.
4. The enqueuing triggers the DVB or nondevice processor.
5. The device command processor moves the BTU to the DVB work queue.
6. The line work queue is unconditionally triggered.
7. The line work scheduler moves the BTU to the line I/O queue and selects the proper subtask sequence.
8. The line I/O task sequences the initiators and terminators via branches.
9. The initiators issue the XIO.
10. Command decode selects the command initiator.
11. The command initiator sets up the line ICW, CCB, and CCBL2.
12. CSP handles the transmission or receive.
13. The end of the command at level 2 initiates a PCI to level 3 to the command ender.
14. The line I/O queue is triggered.
15. The terminator subtask (or error routine, if necessary) returns control to the line I/O task sequence (item 8).
16. The chain terminator TPPOSTs the BTU to the convert routine to change the BTU to a FID0 PIU.
17. The convert routine branches to INN path control for routing.

Refer to the following publication for additional information:

ACF/NCP/VS Network Control Program Logic (LY30-3041). See:

- Appendix B: BSC/SS Control Command Cross Reference Table
- Appendix C: Sequences of I/O Subtasks for BSC/SS Processing in Level 5
- Appendix D: Command Sequence Charts (identifies the CCBL2 and CCBL3 Routines)
- Appendix F: Online tests

Block-handler Routines

The BSC/SS processor provides three points at which user-written routines or IBM-supplied routines may be executed for the manipulation of data. These data manipulation routines are called block-handler routines (BHR).

Block-handler routines are data-oriented. The routines are given access to blocks that contain data at the following times:

1. Before the output is sent to a device: blocks accompany 'write' commands only (execution points 1 and 2).
2. After input is received from a device: blocks accompanying commands of 'read', 'invite', 'write conversational', 'write with read modifier' (in read phase), 'write with contact and read modifiers' (in read phase) (execution points 2 and 3).
3. When the block is in error (execution points 2 and 3).

Block-handler routines (BHR) are grouped into units called block-handlers (BH). A block-handler is designated at NCP generation to be executed at one of the three points listed below. Up to three block-handlers (a possibility of one for each execution point) are grouped to form a block-handler set (BHS). Each device can be assigned a BHS at NCP generation. The BHS can be flagged as initially executable or it can be activated later by a control command. A control command can also be used to assign a BHS dynamically to a device, or to change the BHS association specified at NCP generation.

The BCU being edited by the BHRs resides on a different queue at each of the three points of execution. At point 1, the BCU is on the device input queue (DVB); at point 2, the BCU is on the line input/output queue; at point 3, the BCU is on the point 3 BHR queue extension to the DVB.

The three execution points are as follows:

Point 1

The point 1 entry is used for BCUs to be written to the BSC or SS device. Point 1 BHRs are executed after the BTU is received from the host and before the line has been scheduled for the I/O operation. The device command processor is the interface with the BHR mechanism for point 1. No BCUs are in error at this point.

Point 2

Point 2 BHR is invoked during execution of certain initiator and terminator subtasks by the BHEXIT macro. Since BHRs are data-oriented, the only initiator subtask that invokes BHRs at point 2 is the write initiator. All terminator subtasks that represent termination of a read command invoke BHRs at this point. The subtasks include the common read terminator, the display service-seeking terminator, and the chain terminator (read entry point only). The following subtasks invoke BHRs at point 2 for BCUs that are in error: the 'write terminator', the 'read terminator' routine, the 'common read terminator', the 'display service-seeking terminator', the 'contact terminator', and the 'error retry' routine.

Point 3

The TPPOST routine is the interface for point 3. At this point all processing on the BCU has been completed and the BCU is ready to be sent to the host. The TPPOST routine puts the BCU on the point 3 BHR queue of the DVB. When the BHRs have completed processing, an XIO macro instruction is used to send the BCU to the host.

Block-handler Control Blocks

Figure 10.3 shows the relationships of the control blocks associated with block-handler routines. The paragraphs that follow explain the function of each block. The BHR extension to the DVB exists for those devices specified to have block-handler routines associated with them. The extension reserves space for a pointer to the block-handler set. If a block-handler set is defined at generation time, the address of a block-handler set is assigned at the same time. The pointer is changed if the block-handler set for this device is changed via an optional control command. The BHR extension also contains the QCB for a BHR queue, which is used at point 3 if the device macro is coded with an operand of PT3EXEC=YES.

The block-handler set (BHS) contains pointers to the block-handler driver tables (BHD) that are to be executed at each of the three points (or the BHS entry contains zero if no block-handler is defined for a point).

The block-handler driver table (BHD) defines the block-handler routines that are to be executed for each block-handler. Each entry contains a pointer to the BHR, control information related to the BHR, and a one-byte parameter or an address to a parameter list.

A block-handler set table (BST) has an entry for each block-handler set (BHS) defined in the NCP. Each entry contains control flags, plus the address of the block-handler set (BHS). This table is used in modifying block-handler sets associated with particular devices.

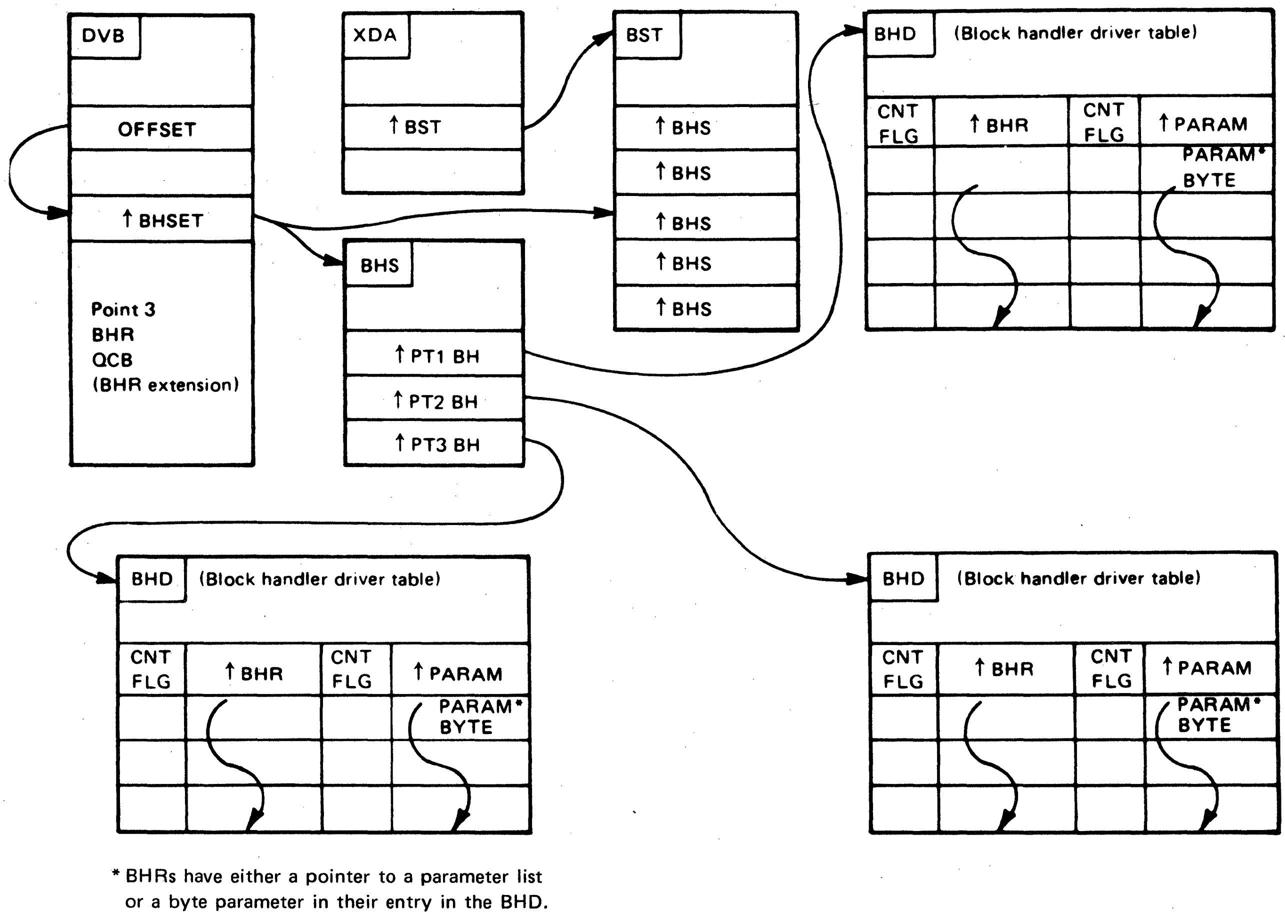


Figure 10.8. BSC/SS BLock-handler Control Block Relationships

The following information provides more detail on the control blocks used with block-handlers and block-handler routines.

BHR Extension to the DVB

Terminals that have BHRs associated with them have a DVB with a BHR extension. This extension contains a pointer to the block-handler set (BHS) for this terminal, and also contains the point 3 BHR QCB (if PT3EXEC=YES was coded on the CLUSTER, TERMINAL, or COMP macro).

The DVB address relating to BHRs is:

- X'38' Offset to BHR extension

The two fields in the block-handler extension to the DVB (BHR) are:

- X'00' Pointer to the block-handler set (BHS)

- X'04' Point 3 QCB

Block-handler Set (BHS)

The block-handler set (BHS) contains pointers to the one, two, or three block-handlers that are to be executed for this set. If a block-handler is not defined, the address pointer contains zeros. If a block-handler is defined, the pointers are addresses of a block handler driver table.

The following are the fields of a block-handler set:

- X'00' Pointer to point 1 BHD
- X'04' Pointer to point 2 BHD
- X'08' Pointer to point 3 BHD

Block-handler Driver Table (BHD)

The block-handler driver table (BHD) defines the block-handler routines (time and date, edit, and user block-handler) that are to be executed, at a point, for a block-handler. The BHD is created by the STARTBH, DATETIME, EDIT, UBHR, ENDBH macro grouping. Each entry in the BHD contains a pointer to the BHR, control information, and parameter information.

The BHD contains one entry for each coded macro of DATETIME, EDIT, or UBHR, with the following fields:

- X'00' Pointer to the block-handler routine
- X'04' Pointer to parameter list for edit

The parameter list for the EDIT BHD contains the following fields:

- X'00' Backspace character
- X'01' Flags
- X'02' Record descriptor masking configuration

Block-handler Set Table (BST)

The block-handler set table (BST) contains an entry for each block-handler set defined in the NCP generation. The address of the BST is in XDA at X'7F4'. This table is used for dynamic block-handler set association.

The block-handler set table (BST) contains one entry for each block-handler set (BHS) defined. Each entry contains an address of a block-handler set (BHS).

User Block-handler Routines

User block-handler routines are identified in a block-handler by the UBHR macro. The user routine must be preassembled in the library identified by the USERLIB (and QUALIFY) operand of the BUILD macro.

The user routine is written with 3705 communications controller instructions, assembler instructions, and internal macros. *IBM 3705 NCP Instructions and Supervisor Macros* (SR20-4512) provides user coding information.

At entry to a user routine, register 2 contains the address of the QCB which contains the block to be processed. The NCP abends if a valid BCU is not available when the user code returns control to the NCP.

Multiple Terminal Access (MTA)

The multiple terminal access (MTA) feature of the network control program allows the communications controller to communicate with several common types of SS terminals over a single switched network port. When a terminal calls in over a line identified at NCP generation as an MTA line, the NCP identifies the type of terminal and the transmission code being used, and initializes the line's adapter control block (ACB) accordingly. The NCP then communicates with the terminal normally until the session ends.

The types of terminals supported by MTA are:

- IBM 2741
- IBM 3767 (at 300 baud in 2741 compatability mode
- Western Union TWX
- IBM 2740 transmit control (with or without checking)
- IBM 1050
- IBM 2740 basic (with or without checking)

The NCP terminal identification procedure always tests for terminal type (of terminals defined in the NCP system), in the order listed above.

Multiple Terminal Access (MTA) Control Blocks

In order to identify the type of terminal and to establish the appropriate operating parameters (speed and transmission code) once the terminal is identified, the NCP uses several tables. This section describes the function and relationships of these tables.

MTA GROUP, LINE, and TERMINAL

The actual line interface definition requires a GROUP, LINE, and TERMINAL macro definition. The TERMINAL macro has an operand of TERM=MTA. These macros create the line group table (LGT), line control block (LCB), adapter control block (ACB), and device base control block (DVB) which are used for the initial connection. The incoming MTA call is received using these control block definitions.

MTA List

The MTA list is a table of one-byte entries, each entry representing one of the five terminal types that can call in on an MTA line. The list consists of a group of entries for each combination of terminal types on MTA lines in the telecommunication subsystem. The entries in a group are always in the order in which the NCP tests for terminal type. The following values represent the given terminal type in the MTA list:

- X‘00’ 2741
- X‘01’ TWX
- X‘02’ 2740 transmit control

- X'03' 1050
- X'04' 2740 basic
- X'05' 3767 or 3767 and a 2741 on the same MTA line

A group of entries is delimited by a byte containing the value X'FF'.

The MTA identification routine uses the MTA list to determine which types of terminals to test for. The initial offset into the list is in IOBSTOFS field of the line's IOB.

The multiple terminal access (MTA) identification routine uses the MTA list as an index into the code for testing the terminal type. The routine sets a timeout at the beginning of each test. If the terminal does not respond before the timeout expires, the MTA routine is reentered and the next MTA list entry is used to test for the next terminal type. If the delimiter entry X'FF' is reached, the routine disconnects the line and ends the command.

The sequence for terminal checking occurs in the following manner:

The MTA identification routine checks for both 2741 (3767 in 2741 mode) and TWX at the same time (if either terminal type is specified in the MTA list). The routine sends an EOT to the device and sets a timeout. If the terminal responds with an EOA, the routine assumes the terminal is a 2741/3767. If leading graphics are present and the response is the WRU character, the routine assumes a TWX terminal.

To test for 2740 transmit control, the MTA routine sends a slash-space (/b) character sequence. If the terminal responds with an EOA, the routine assumes the terminal is a 2740 with transmit control. If the response is a NAK, a 1050 terminal is implied.

To test for a 1050, the MTA routine must poll the terminal. The routine fetches and sends the polling characters for each device on the 1050 polling list until it receives a response. If the response ends in EOT, the routine disconnects the line and ends the command.

The 2740 basic test consists of transmitting a BID message to the terminal. The message, sent in both EBCD and correspondence code, prints at the terminal and indicates that the operator is to enter the MTA sign-on sequence (covered later in this manual).

When the terminal type (but not the transmission code) has been identified, the control pointer from the original line group table (LGT) can be updated to the stand-alone line group table (LGT). The two tables were created by GROUP macros defined for each MTA terminal type.

As an example, consider a telecommunication subsystem with three MTA lines. One line has all five terminal types; the second has 2741 and 1050 terminals; and the third has TWX, 2740 transmit control, and 1050 terminals. The MTA list for this NCP has the format shown in Figure 10.9.

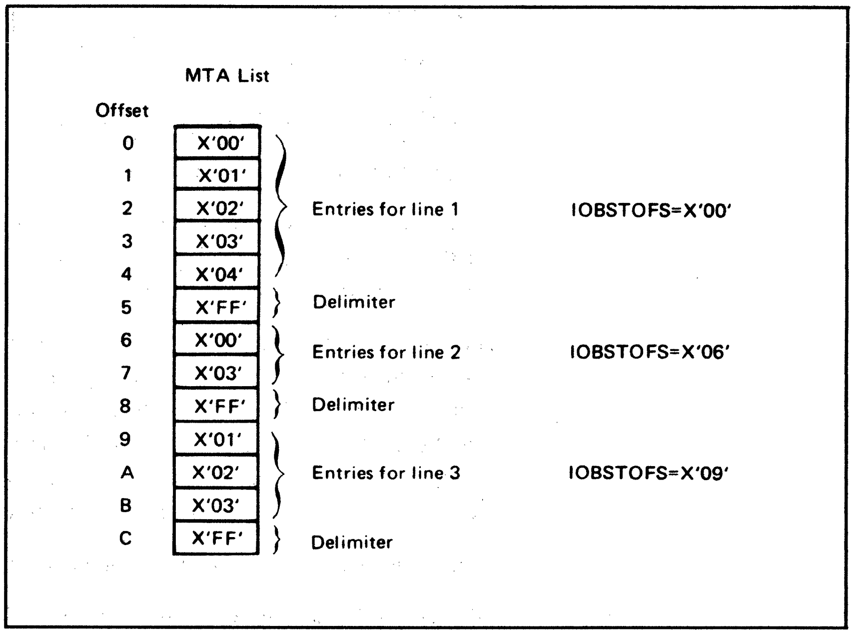


Figure 10.9. MTA List Format

The MTA list is defined by MTALIST macros coded for NCP generation.

MTA Line Group Table (LGT)

A GROUP macro, creating a line group table (LGT), is defined for the MTA line, plus one line-group table per MTA terminal type which calls in. The MTA identification routine initially uses the group definition associated with the MTA line. As soon as the terminal type and transmission code are identified, the pointer is changed to the specific LGT for the terminal type.

Line Control Selection Table (LCST)

The line control selection table (one per NCP) is used by the multiple terminal access (MTA) identification routine to initialize the line's character control block (CCB), once the routine has identified the type of terminal calling in. The table is also used to establish CCB parameters when the NCP calls a device on an MTA line.

The LCST may contain up to sixty-three 16-byte entries, each representing a particular set of operating parameters for some MTA device (or devices) in the telecommunication subsystem. The first entry in the LCST is used by the MTA identification routine during the identification process and does not represent a particular type of device.

The parameters in an LCST entry are those that can vary for terminal type, transmission code, or individual device. These parameters include such variables as line speed, carriage return rate, translate table addresses, size of print line, and error retry limits.

A series of MTALCST macros is coded for NCP generation to define the LCST entries.

Once the MTA identification routine has identified the terminal type calling in, the routine determines which LCST entry to use by referring to a list of valid LCSTs for that terminal type. One list exists for each possible combination of terminal type and transmission code. The terminal operator must enter a sign-on sequence to identify the correct terminal/code list and entry within the list. Each list contains up to ten halfword pointers to valid LCSTs for the combinations which that list represents. The sign-on sequence may include two identical digits, representing the number of the list entry to be used, relative to the beginning of the list for this terminal/code type. If the number is omitted, the routine assumes the first entry is to be used.

As an example, assume that the terminal has been identified as a 1050. The terminal operator enters sign-on sequence, /"44 CR EOB. The /" is unique for each type code, which identifies the code as BCD. Now that the terminal type and code are known, the digits '44' indicate that the MTA identification routine is to use entry 4 in the list of LCST pointers for 1050 BCD terminals. All 1050 terminals are checked terminals; however, the EOB, rather than EOT, provides the definition of a terminal with checking verses nonchecking features. Once the appropriate pointer to an LCST entry is located, the control block fields can be filled in. The relationship is shown in Figure 10.5.

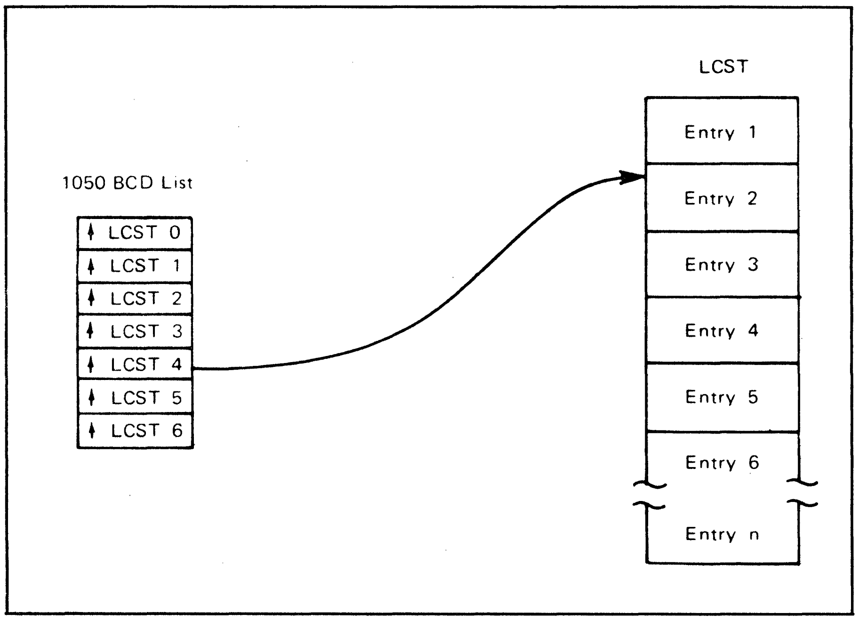


Figure 10.10. LCST Entry Relationships

The relationship of the pointers in the list to the LCST entries which the pointers represent is established during NCP generation, according to the parameters specified for the MTA lines and devices. MTATABL macros are used to define these lists for NCP generation.

1050 Polling List

When testing for a 1050 terminal, the MTA identification routine must poll the terminal. For this purpose, a single polling list exists in the NCP for all 1050 terminals on all MTA lines. The polling list contains a halfword pointer to the polling characters for each such device. The MTA identification routine goes through each entry in the polling list until it receives a positive response from the device or until it exhausts the list. In the latter case, the routine assumes that the device is not a 1050 and goes on to test for the next terminal type. Each polling attempt which is not successful requires a polling timeout; if many sets of 1050 polling characters are in the list, with a one-minute timeout per polling attempt, the sign-on could take excessively long.

The entries in the 1050 polling list are specified with the MTAPOLL NCP generation macro.

BSC/SS Processor Summary

The BSC/SS processor is that part of the NCP that processes requests for BSC or SS resources. The first BSC/SS processor component to receive control is the BSC/SS system router.

When INN path control passes a PIU to the BSC/SS system router, the system router branches to the PIU/BTU converter to convert the PIU to a BCU. From this point the BSC/SS processor component flow is as follows:

Upon receiving the BCU from the converter, the system router enqueues the BCU for the device command processor or nondevice command processor for line-oriented control commands. The device command processor passes control to the line work scheduler. The line work scheduler selects a chain of subtasks and passes control to the line I/O task. The communications control program gets control and sets up for the start of level 2 activity. Data transfer is by the character-service program for each byte with the type 2 scanner or by cycle-steal for the length of an NCP buffer with the type 3 scanner. Upon completion of the level 2 activity, control returns to the CCP and then to the line I/O task. The BCU is converted to a FID0 PIU and passed to INN path control for routing.

Chapter 11:

Programmed Resources and User Line Control

Introduction

An attachment facility for user coding is provided in ACF/NCP release 2. This facility allows separate program products, such as the Network Terminal Option (NTO) and the airline line control (ALC), to be included with ACF/NCP.

There are two major areas of support:

- Programmed resources: provide network addresses and control blocks with the user providing level 5 programming.
- User line control: allows the user to provide communications scanner support with level 2 routines, level 3 routines, scanner timer routines, and XIO level support.

These topics are not related.

CAUTION: It is likely that such user routines will require the use of NCP functions such as buffer management, the scheduling of input and output operations, and interaction with the SNA network (for example, the NCP and the access methods expect certain sequences to occur in order to start a session). Therefore, an intimate knowledge and understanding of the NCP and of SNA protocols and formats is required. Adding such routines must be approached with caution to avoid disrupting the operation of the NCP. You should also realize that additional work may be required on your part when you install later releases of the NCP.

Programmed Resources

The programmed resources facility provides control blocks in ACF/NCP protected storage to point to user control blocks and code in storage which has a separate protect key. User code using this facility is written in the IBM 3705 Communications Controllers assembler language. Before considering user code, the user should have a detailed knowledge of the ACF/NCP logic flow, control blocks, and IBM 3705 Communications Controllers hardware.

The definition of programmed (virtual) SNA resources provides an attachment facility for user control blocks and programming to be executed in level 5 (background). The user code can process commands and/or data for NCP supported line controls.

The user control blocks and user code are coded and assembled separately from the NCP generation. The control blocks and entry points must be defined in user code as entry labels for merging with the NCP generation defined external references.

User Line Control

User line control provides an attachment facility for user control blocks and programming to be executed in level 2 and level 3 (scanner support) using standard NCP processing or programmed resources in level 5.

The user line control that is added must be compatible with the type 2 or type 3 communications scanners (see *IBM 3704 and 3705 Communications Controller Principles of Operation).* Interrupts for specified lines are passed in level 2 to a user interrupt handler. Your code in level 2 can create an interrupt into user-written level 3 interrupt handler. All level 2 and level 3 functions related to user-controlled lines are the user responsibility. User level 2 and level 3 routines can have a serious impact on standard support lines.

The XIO macro is used to pass control from level 5 routines to user level 3 routines. Control may be passed back to level 5 from level 3 code by issuing NCP supervisor macros. You may utilize the existing BSC, SDLC, or start-stop level 5 interface, or use the programmed resource capability that was described previously as virtual resources.

User Coding

A text on user coding for programmed resources and user line control is available. Refer to *IBM 3705 NCP Instructions and Supervisor Macros* (SR20-4512).

The user control blocks and user code are coded and assembled separately from the NCP generation. The control blocks and entry points must be defined in user code as entry labels for merging with the NCP generation defined external references.

The following are prefixes to avoid in user code:

$	DRS	PCB
ACB	DVB	PIU
ATB	DVI	RH
ATP	DVQ	RCV
BCB	ICW	RG
BCH	IOB	RHN
BCO	IRN	RH#
BCT	LCB	RUN
BCU	LCS	SCB
BOQ	LCW	STQ
CCB	LGT	SYS
CM	LKB	TH
CRP	MDR	TVS
CX	OLL	UNASGN
CY	PAD	U1
DAE		X

Additional information on the following macros and operands are defined in:

ACF/NCP/VS Network Control Program System Support Programs Installation SC30-3142

ACF/NCP/VS Network Control Program Logic LY30-3041

Programmed Resources

Introduction

Programmed resources provide the user with an attachment facility for user control blocks and code in level 5. The definition of programmed resources by the user is by (1) the NCPNAU macro or (2) a GROUP, LINE, PU, and

LU macros with the GROUP operands of LNCTL=SDLC and VIRTUAL=YES.

The NCPNAU macro generates a control block and task pointers to user code. The NCPNAU TYPE=SSCP resource resolution table entry is generated in segment 6. The resource resolution table segment 6 is ignored by the host access methods and therefore is not addressable from the host.

The NCPNAU macro can be used to define an SSCP internal to NCP. The user-written initialization routine should schedule the NCPNAU task to issue commands to obtain ownership of the NCP and optionally issue commands to obtain links, etc.

The virtual definitions of GROUP, LINE, PU and LU generate control blocks and task pointers to user code. The resource resolution table entries are available to the host access method. The host access method can issue commands to the virtual link, physical unit, and logical unit. The commands are received by the virtual device, triggering the user task. The triggered user task can then take whatever programming action is desired.

The following are the virtual resource control blocks which are referenced in this material:

- (VLB) programmed resource virtual link block
- (NPB) programmed resource physical unit block
- (NLB) programmed resource logical unit block
- (NLX) programmed resource logical unit block extension

The following are the user defined control blocks which are required and separately coded and assembled in user code.

- (FVT) function vector table (FVTABLE macro)
- user-defined control block are:
    1. optional
    2. assembled with IBM control blocks (see GENEND SRCHI)

User Code Generation

The user code is assembled separately from the NCP stage 1 generation. All of the machine instructions, assembler instruction, and assembler extended mnemonics can be used in user coding. The user code may use the macros defined in *ACF/NCP/VS Network Control Program Logic* (LY30-3041).

A text on user coding for programmed resources and user line control is *IBM 3705 NCP Instructions and Supervisor Macros* (SR20-4512).

Programmed Resources

Function Vector Table

The FVTABLE macro instruction allows the user to construct a list of tasks which may be associated with each programmed resource. This list is called a function vector table (FVT). Its name is specified with the resource at program generation time. The FVTABLE macro is generated in user code.

The function vector table (FVT) control block is illustrated in Figure 11.1.

Offset	FVT	
0	X'01'	INITIAL TASK ADDRESS
4	X'02'	NOTIFY TASK ADDRESS
8	RESERVED	
24	X'0A'	USER TASK ADDRESS
28	X'0B'	USER TASK ADDRESS
n	X'00'	NEXT FVTABLE ADDRESS/ 00 00 00

Figure 11.1 Function Vector Table (FVT) Control Block

The function vector table (FVT) contains the addresses of user tasks. The user defined control blocks identify a FVT of tasks to executed.

The leftmost byte of each four-byte entry contains a task identifier. The address of a task (or zeros) is contained in each task entry.

At offset X'00' is a four-byte address of the user task which is inserted into the QCB to initialize the resource at NCP initialization. This task always has an index value of X'01' in the FVT. Each task address must be defined with an ENTRY statement.

At offset X'04' is a four-byte address of the user task which is inserted into the QCB by the NCP whenever a condition exists which directly affects the resource. This task is triggered by automatic network shutdown when the SSCP owner of a programmed resource fails. This task always has an index value of X'02' in the FVT. The task should take whatever action is required by SNA and any other action desired by the user within the constraints of SNA. The task address must be defined as with an ENTRY statement.

The four-byte addresses from offset X'08 to X'20' are reserved. More than one user task is supported by the FVT. The first or only task is the initial task (offset X'00'). Additional user task addresses may be defined beginning at offset X'28', task offset of X'0A'. The NCHNG macro is used to change the task address from the current task to one of the FVT tasks.

When a task is to be switched using the NCHNG macro, the NCP looks in the first FVT for the task offset number that is associated with the task. If the desired code is not found, the chained FVTs are searched until the code is found. Common code can be associated with a single FVT and be chained from other FVTs.

The last entry in a FVT contains an identifier of X'00' in the leftmost byte. A nonzero address in this entry chains this FVT to the FVT identified by the address.

NCPNAU Programmed Resource

The NCPNAU macro (Network Addressable Unit) can be used to define a special type of network addressable unit. The network addressable unit (NAU) in this case is part of the NCP itself and does not have a group, line, or physical unit defined for it. Because there is no associated line or physical unit, a host access method would not consider this logical unit a normal NCP LU and would not initiate a session.

The NCPNAU macro immediately follows the SYSCNTRL macro in the NCP generation. All other programmed resources are defined in the boundary network node (BNN).

The NAU employs the programmed resource logical unit block (NLB) and programmed resource logical unit block extension (NLX) for definition purposes. The purpose of the NAU is to provide a mechanism for defining programs to run in the NCP, as opposed to virtual resources.

The NCPNAU stage 1 macro instruction defines the names of the user control blocks and function vector tables associated with this network addressable unit, and assigns a specific name to the unit.

Each network addressable unit must be represented by a separate NCPNAU macro. The NCPNAU macros must immediately follow the SYSCNTRL macro.

The NCPNAU generated control blocks are the program resource logical unit block (NLB) and programmed resource logical unit block extension (NLX). The quantity of NLXs is specified via the NAUCB operand. The format of the NLB is shown in Figure 11.4. The format of the NLX is shown in Figure 11.5.

The NCPNAU label is required and specifies the resource name for the network addressable unit. A single network address is associated with the NLB/NLX generated group.

The NCPNAU NAUFVT operand specifies the names of the functional vector tables (FVT) associated with this network addressable unit. An FVT is generated in user code using the FVTABLE macro. At least one function vector table is required. The names in this operand are positionally related to the names in the NAUCB operand.

The NAUCB operand is optional. The NAUCB operand specifies the names of user control blocks associated with this network addressable unit. Each name generates an NLX.

If the NCPNAU macro is used as an SSCP the MAXSSCP operand must be coded 2 or greater to allow the NCP SSCP ownership of the NCP. The

NCPNAU TYPE=SSCP is not addressable from a host. The user initialization routine (or other user code) must trigger the NCPNAU task.

Boundary Network Node Programmed Resources

The GROUP operands of LNCTL=SDLC and VIRTUAL=YES results in the generation of programmed resource control blocks from this GROUP macro and the following LINE, PU and LU macros.

When programmed resources are selected (LNCTL=SDLC, VIRTUAL=YES), the GROUP macro does not generate a control block. LINE, PU, and LU macros which follow the GROUP provide the programmed resource control blocks.

Programmed Resource Virtual Link Block

The LINE macro generates a programmed resource virtual link block (VLB) which contains addresses of a user control block and user function vector tables. The VLB destination corresponds to the SDLC link control block (LKB).

The format of the VLB is illustrated in Figure 11.2.

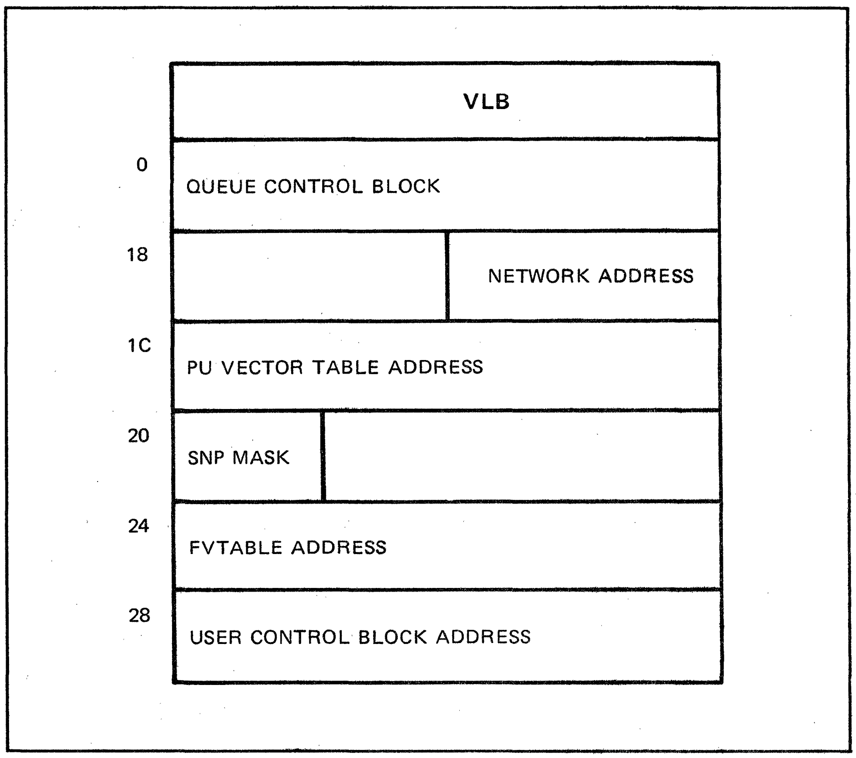


Figure 11.2 Programmed Resource Virtual Link Block (VLB)

The VLB contains a queue control block at offset X'00'. This task is triggered by (1) NCP physical services when an activate link or deactivate link command is received, (2) a permanent error occurs on a link, (3) a control command completes, or (4) user code dispatches the task.

The VLB network address is at offset X'1A'. The PUV vector table address is at offset X'1C'; the PUV is the same format as BNN PUV. The function vector table identified at offset X'24' contains user-written task addresses. The function vector table (FVT) is generated in user-written code using the FVTABLE macro. The user control block address is at offset X'28'.

Programmed Resource Physical Unit Block

The programmed resource physical unit block (NPB) is generated by a PU macro which follows the GROUP macro with operands of LNCTL=SDLC and VIRTUAL=YES.

The NPB contains addresses of a user control block and user function vector tables. The NPB destination corresponds to the SDLC common physical unit block (CUB).

The format of the NPB is illustrated in Figure 11.3.

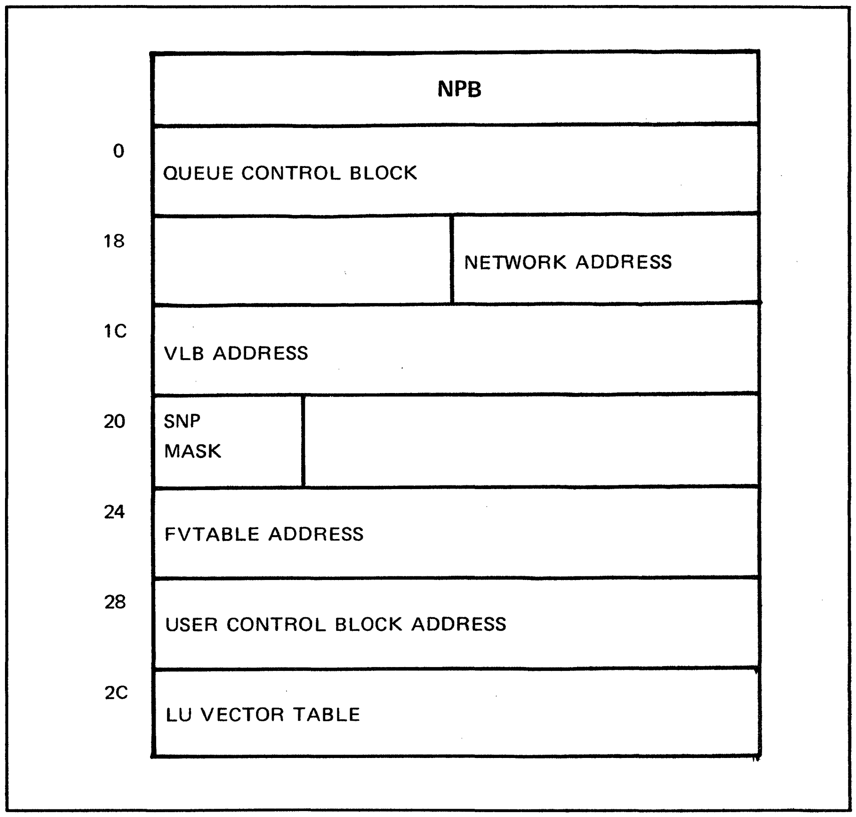


Figure 11.3 Programmed Resource Physical Unit Block (NPB)

The NPB contains a queue control block at offset X'00'. This task is triggered by a TRIGGER macro or PIUs queued with ACTV=YES.

The NPB network address is at offset X'1A'. The function vector table (FVT) identified at offset X'24' contains user-written task addresses. The function vector table (FVT) is generated in user-written code using the FVTABLE macro. The user control block address is at offset X'28'. The LU vector table address is at offset X'2C'.

If DIAL=YES is coded on the GROUP macro the LINE macro is followed by only one PU macro coded with the MAXLU operand. MAXLU determines the number of NLBs. The LUFVT determines the number of NLXs. The LUFVT and LUCB operands will pertain to all NLBs and NLXs generated. An NLB and its NLXs all have the same network address.

Programmed Resource Logical Unit Block

The programmed resource logical unit block (NLB) is generated by an LU macro which follows the GROUP macro with operands of LNCTL=SDLC and VIRTUAL=YES. The LU macro generates a programmed resource logical unit block (NLB) and zero to eight (one per LUCB operand labels) programmed resource logical unit block extensions (NLX).

The NLB contains addresses of a user control block and user function vector tables. The NLB destination corresponds to the SSCP/LU session of the SDLC logical unit block (LUB).

The format of the NLB is illustrated in Figure 11.4.

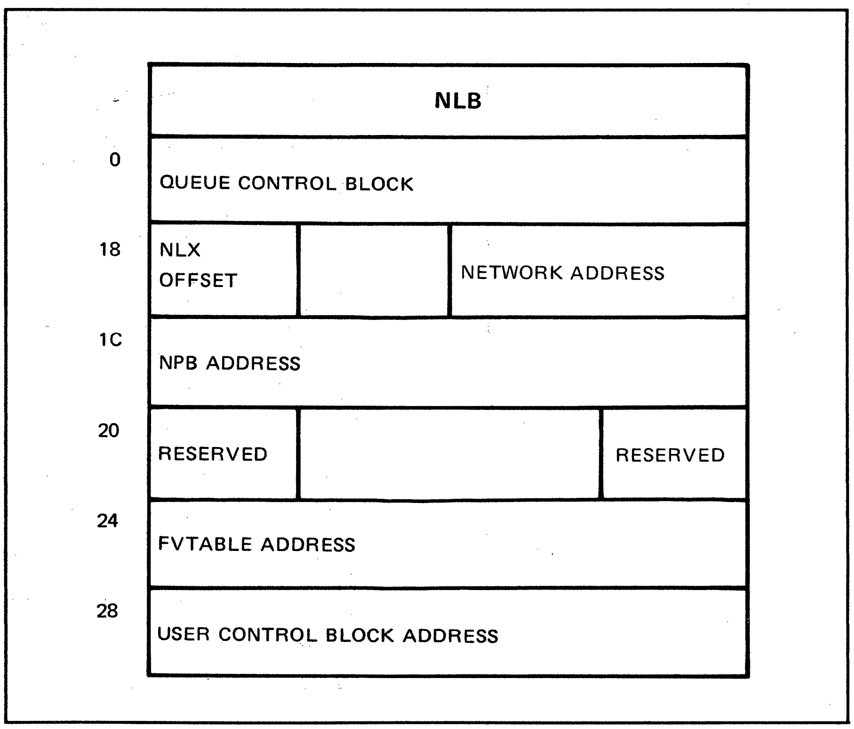


Figure 11.4 Programmed Resource Logical Unit Block (NLB)

The NLB contains a queue control block at offset X'00'. This task is triggered by a TRIGGER macro or PIUs queued with ACTV=YES.

The offset to the first programmed resource logical unit block extension (NLX) is at offset X'18'. The NLB network address is at offset X'1A'. Offset X'1C' contains the address of the NPB control block. The function vector table (FVT) identified at offset X'24' contains user-written task addresses. The function vector table (FVT) is generated in user-written code using the FVTABLE macro. The user control block address is at offset X'28'.

The user code is triggered by TRIGGER macro or a PIU queued with ACTV=YES queued on the NLB.

Programmed Resource Logical Unit Block Extension

The programmed resource logical unit block extension (NLX) is generated by an LU macro which follows the GROUP macro with operands of LNCTL=SDLC and VIRTUAL=YES. The LU macro generates a programmed resource logical unit block (NLB) and zero to eight (one per LUCB operand labels) programmed resource logical unit block extensions (NLX).

The NLX contains addresses of a user control block and user function vector tables. The NLX destination corresponds to the LU/LU session of the SDLC logical unit block (LUB).

The format of the NLX is illustrated in Figure 11.5.

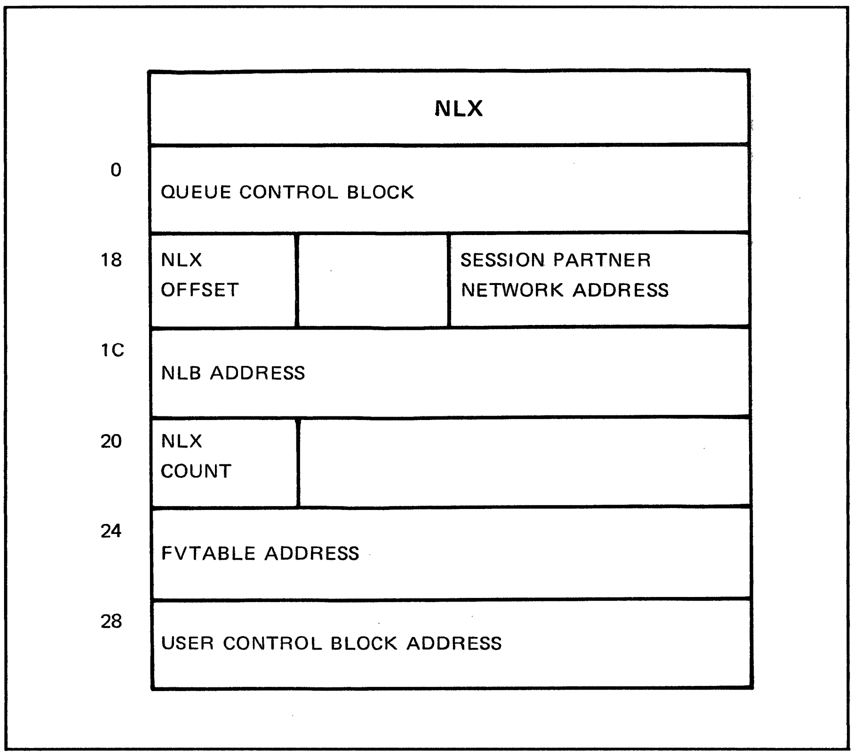


Figure 11.5 Programmed Resource Logical Unit Block Extension (NLX)

The NLX contains a queue control block at offset X'00'. This task is triggered by a TRIGGER macro or PIUs queued with ACTV=YES.

The offset to the next programmed resource logical unit block extension (NLX) is at offset X'18'. The NLX network address of a session partner is at offset X'1A'. Offset X'1C' contains the address of the NLB control block. Offset X'20' identifies which NLX (1 to 8) is represented by this control block. The function vector table (FVT) identified at offset X'24' contains user-written task addresses. The function vector table (FVT) is generated in user-written code using the FVTABLE macro. The user control block address is at offset X'28'.

The user code is triggered by a TRIGGER macro or PIU queued with ACTV=YES.

Programmed Resource Stage 2 Generation

The GENEND operands identify the storage location (low 64K storage or above 64K storage) and/or the following types of code or control blocks:

- INIT entry points of user initialization routine. When a user's initialization routine is given control, register 6 contains the address within the NCP to which the routine must return when it has completed its processing. This routine triggers the NCPNAU task for initial execution.
- SRCHI user source control blocks and tables for high storage.
- TMRTICK user timer tick service routine. The NCP checks the status of up to 10 lines at each tick of the 100-millisecond clock. The user's timer-tick routine is considered as one of these lines. Accordingly, processing in this routine should be kept to a minimum so as not to delay or interfere with proper and timely servicing of interrupts by the NCP.
- INCHI user object modules for level 5.
- INCINIT initialization routine object modules INCLUDE statements.

User Line Control

Introduction

The user can use the stable attachment facility to provide user line control. The line control that is added must be compatible with the type 2 or type 3 communications scanners (see *IBM 3704 and 3705 Communications Controller Principles of Operation)*. Interrupts for specified lines are passed in level 2 to a user interrupt handler. Your code in level 2 can create an interrupt into user-written level 3 interrupt handler. All level 2 and level 3 functions related to user-controlled lines are the user responsibility. User level 2 and level 3 routines can have a serious impact on standard support lines.

The XIO macro is used to pass control from level 5 routines to user level 3 routines. Control may be passed back to level 5 from level 3 code by issuing NCP supervisor macros. You may use the existing BSC, SDLC, or start-stop level 5 interface, or you may use the programmed resource capability that was described previously as virtual resources.

User line control (LEVEL2=, LEVEL3=) coding uses the GROUP, LINE, and SERVICE macros. Any macros coded following the user line control are (1) programmed resource (VIRTUAL=YES) or (2) standard NCP definitions.

The following are the control blocks for user line control which are referenced in this material:

- (GCB) group control block. In user code this control block in combination with the UACB functionally replaces the IBM ACB. The UACBs are chained from the GCB for user processing.
- (UACB) user adapter control block
- (ULVT) user line vector table

The MACLIB and USERLIB operands of the NCP BUILD macro provide the source of macros and object modules of preassembled user routines if they are not in the NCP libraries.

User Line Control Definitions

User line control generates control blocks for level 2 and level 3. Level 5 support (LINE, PU, LU, CLUSTER, TERMINAL, COMP) must be provided in support of user line control. The variations of coding user line control are:

1. User line control only (LNCTL=USER,LEVEL5=USER). No level 5 code is generated in this GROUP. A separate group of programmed resources is required.
2. Programmed resources (LNCTL=SDLC,VIRTUAL=YES)
3. Standard SDLC (LNCTL=SDLC,VIRTUAL=NO,LEVEL5=NCP)
4. Standard BSC (LNCTL=BSC,DIAL=NO,LEVEL5=NCP)
5. Standard SS (LNCTL=SS,LEVEL5=NCP)

The following are examples of the groups of macros and operands which may be selected for user line control.

User Line Control (No level 5)

```
GROUP    LNCTL=USER,
         LEVEL2=symbol,
         LEVEL3=symbol,
         LEVEL5=USER,
         TIMER=(error,ras,stap,lstap),
         XIO=(line,setmode,immed,link),
         VIRTUAL=NO,
         LEVEL5=USER
LINE     AUTUACB=symbol,
         UACB=(symbol1,symbol2)
SERVICE
```

Note: No PU, LU, CLUSTER, TERMINAL or COMP macros may follow the GROUP, LINE, and SERVICE macros when LNCTL=USER.

User Line Control and Programmed Resources

```
GROUP    LNCTL=SDLC,
         VIRTUAL=YES,
         LEVEL2=symbol,
         LEVEL3=symbol,
         LEVEL5=NCP,
         TIMER=(error,ras,stap,lstap),
         XIO=(line,setmode,immed,link)
LINE     AUTUACB=symbol,
         UACB=(symbol1,symbol2),
```

```
        LINECB=symbol,
        LINEFVT=symbol
SERVICE
PU      (standard programmed resource definitions)
LU      (standard programmed resource definitions)
```

User Line Control and SDLC Resources

```
GROUP   LNCTL=SDLC,
        VIRTUAL=NO,
        LEVEL2=symbol,
        LEVEL3=symbol,
        LEVEL5=NCP,
        TIMER=(error,ras,stap,lstap),
        XIO=(line,setmode,immed,link)
LINE    AUTUACB=symbol,
        UACB=(symbol1,symbol2),
        LINECB=symbol,
        LINEFVT=symbol
SERVICE
PU      (standard PU resource definitions)
LU      (standard PU resource definitions)
```

User Line Control and BSC Resources

```
GROUP   LNCTL=BSC,
        DIAL=NO,
        LEVEL2=symbol,
        LEVEL3=symbol,
        LEVEL5=NCP,
        TIMER=(error,ras,stap,lstap),
        VIRTUAL=YES,
        XIO=(line,setmode,immed,link)
LINE    LINECB=symbol,
        UACB=(symbol1,symbol2),
        LINEFVT=symbol
SERVICE
CLUSTER   (standard operands)
TERMINAL  (standard operands)
COMP      (standard operands)
```

User Line Control and SS Resources

```
GROUP   LNCTL=SS,
        LEVEL2=symbol,
        LEVEL3=symbol,
        LEVEL5=NCP,
        TIMER=(error,ras,stap,lstap),
        XIO=(line,setmode,immed,link)
LINE    LINECB=symbol,
        UACB=(symbol1,symbol2),
        LINEFVT=symbol
SERVICE
TERMINAL  (standard operands)
COMP      (standard operands)
```

User Line Vector Table

Any definition of a GROUP macro with the operands of LEVEL2= and LEVEL3= generates a user line vector table (ULVT). The ULVT has the same format as the standard line vector table (LNVT). The ULVT contains a halfword address for each line address of a communcations scanner defined by a CSB macro. User line control and undefined line definitions are indicated as invalid in the LNVT (rightmost bit of 1). Standard line control and undefined line definitions are indicated as invalid in the ULVT.

The ULVT is located in a linkage editor map at label CXTULVT.

Figure 11.6 illustrates the user line vector table (ULVT).

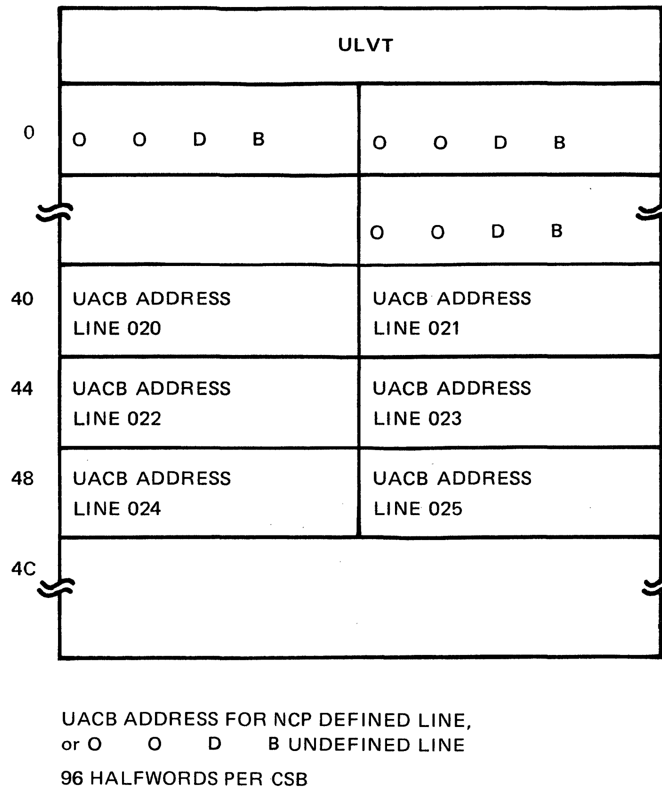


Figure 11.6 User Line Vector Table (ULVT)

Group Control Block

The GROUP macro defined with operands of LEVEL2= and LEVEL= generates a group control block (GCB). The GCB and user adapter control block (UACB) combined corresponds to the adapter control block (ACB) of SDLC, BSC, or SS definitions.

Figure 11.7 illustrates the group control block (GCB).

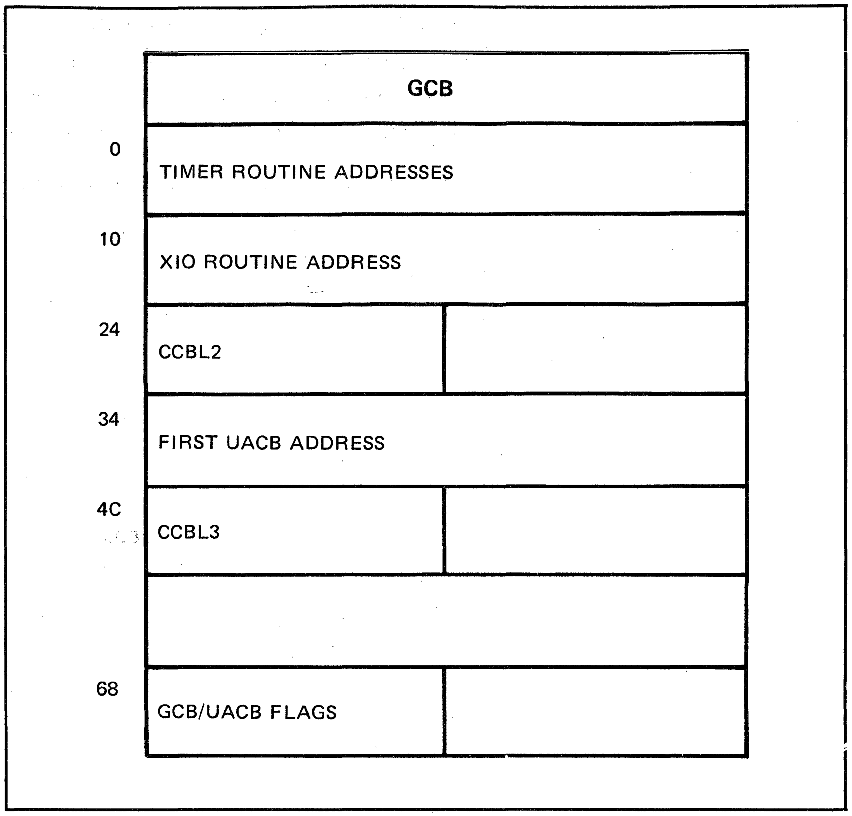


Figure 11.7 Group Control Block (GCB)

At GCB offset X'00' are the addresses of the user-written timer routines. The GROUP macro TIMER operand defines entry points of user routines for (1) timer error service routine, (2) ras service routine, (3) shoulder tap service routine, and (4) lagging shoulder tap routine.

At offset X'10' are the user-written XIO routines. The GROUP macro XIO operand defines entry points of user routines for (1) line I/O service routine, (2) I/O set mode service routine, (3) I/O immediate service routine, and (4) XIO LINK command routine.

Offset X'24' contains the address of the routine to be executed for a level 2 interrupt, the CCBL2 routine address. Offset X'4C' contains the CCBL3 level 3 routine address. The CCBL2 and CCBL3 routines are entry points for level 2 and level 3 interrupts for all UACBs associated with this GCB. The UACB for the interrupt is located in level 2 with the FINDUACB macro (see SR20-4512). The UACB for the interrupt is provided in level 3 in register 2. The specific interrupt requirement must be determined by the user.

The address of the first LINE user adapter control block (UACB) is at offset X'34'. Status flags are at offset X'68'.

User Adapter Control Block (UACB)

The user adapter control block (UACB) is generated by a LINE macro following a GROUP macro coded with operands of LEVEL2= and

LEVEL3=.

Figure 11.8 illustrates the user adapter control block (UACB).

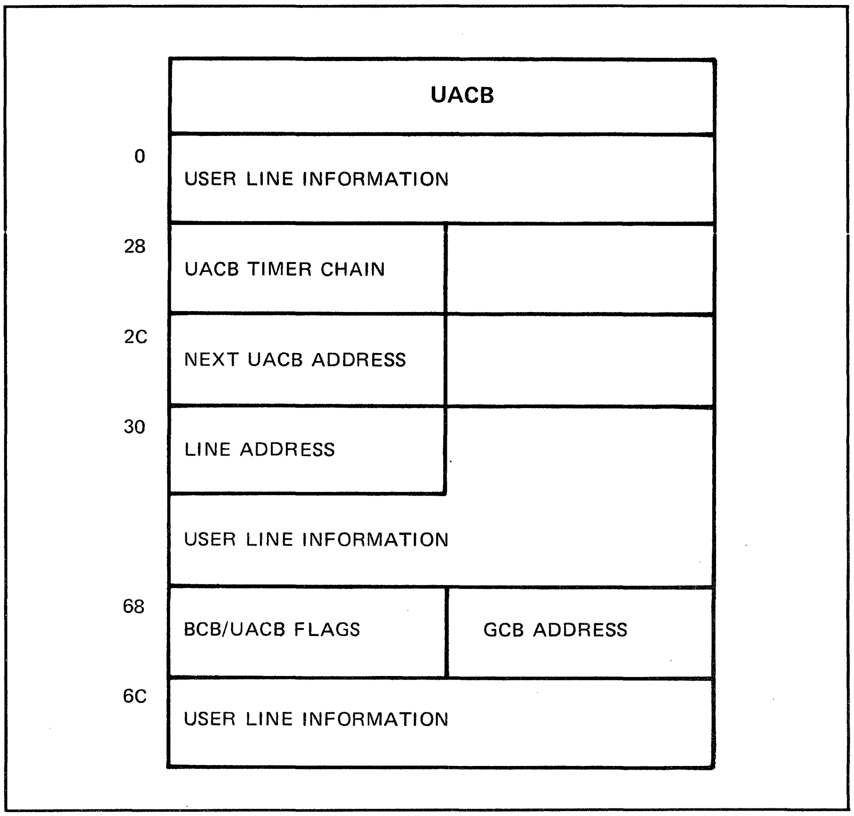


Figure 11.8 User Adapter Control Block (UACB)

User line information may be defined for UACB offsets X'00' to X'27', X'32' to X'67', and X'6C' and above. Other UACB fields are required at the specified offsets.

Offset X'28' contains the UACB timer chain address. Offset X'2C' contains the address of the next UACB or zero. The line (scanner) address is at offset X'30'. The GCB/UACB flags are at offset X'68', and the address of the GCB at offset X'6A'.

Service Order Table

The existing SERVICE macro support is unchanged by the inclusion of user written line control. See the appropriate service order table (SOT) control blocks in the boundary network node (BNN) or BSC/SS processor section of this book.

User Line Control Stage 2 Generation

The GENEND operands identify the storage location (low 64K storage or above 64K storage) and/or the following types of code or control blocks:

- INIT entry points of user initialization routine. When a user's initialization routine is given control, register 6 contains the address within the NCP to which the routine must return when it has completed its processing.

- SRCLO user source control blocks and tables for low storage.
- TMRTICK user timer tick service routine. This NCP checks the status of up to 10 lines at each tick of the 100-millisecond clock. The user's timer-tick routine is considered as one of these lines. Accordingly, processing in this routine should be kept to a minimum so as not to delay or interfere with proper and timely servicing of interrupts by the NCP.
- INCLO user object modules for levels 2 and 3.
- INCINIT initialization routine object modules INCLUDE statements.
- INCL2HI object library level 2 and 3 code for high storage.
- INCL2LO object library level 2 and 3 code for low storage.
- ORDLO object library ORDER statements level 2 and 3 code.
- ORDINIT initialization routine object library ORDER statements.
- ORDL2HI object library level 2 and 3 ORDER for high storage.
- ORDL2LO object library level 2 and 3 ORDER for low storage.

User-routine Macros for User Line Control

All of the machine instructions, assembler instruction, and assembler extended mnemonics can be used in user coding. Most of the macros in *ACF/NCP/VS Network Control Program Logic* (LY30-3041) can be used in user coding.

For additional information on coding user line control see *IBM 3705 Instructions and Supervisor Macros* (SR4512).

Chapter 12:

Partitioned Emulation Program Data Flow

Introduction

The partitioned emulation program allows concurrent NCP mode and emulation of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control Unit, an IBM 2703 Transmission Control Unit, or any combination of the three. The emulation program co-resides with NCP in the communications controller and is defined by a programmer, using emulation generation macros. BSC and/or SS lines may be defined as NCP only, EP only, or may be switched (PEP) between NCP and EP mode.

The emulation program consists of IBM-supplied modules. The routines within these modules provide the programming functions which cause the communications controller to appear to the host processor access methods as a 2701, 2702 or 2703.

When the host processor initiates an input or output operation for a device attached to the communications controller, the type 1 or type 4 channel adapter interrupts the emulation program. The emulation program routines verify that the input or output command is valid for the 2701, 2702, or 2703 being emulated and for the line addressed. The command is also checked for improper sequencing (for example, a Write command directed to a line that has been receiving).

For a Transmit type command, the emulation program requests data from the host processor for transmission to the device addressed. The host generally transfers 4, 8, 16, or 32 bytes of data at a time. The type 1 channel adapter has a four-byte transfer. The type 4 channel adapter has a user-defined transfer length of any of the four values. The user may specify a buffer length of 4, 8, 16, 32 or multiples of 32 up to 224 bytes. The channel can provide multiple transfers up to a buffer length.

The emulation program requests two buffer transfers from the host before beginning transmission to the device addressed. The emulation program initializes the communications scanner with the first character (type 2 scanner) or first buffer (type 3 scanner) to be transmitted. Thereafter, as data is sent out, the scanner requests more data. This sequence continues until the host has transferred all of the data to the emulation program. The emulation program continues to honor the scanner's requests for data until the last data is transmitted. The program then causes the scanner to stop transmitting and returns the ending status to the host processor.

For a Receive type command, the emulation program sets the scanner to receive mode, then continues with other tasks. When a byte (type 2 scanner) or buffer (type 3 scanner) has been received, the scanner interrupts the emulation program. The type 3 scanner cycle-steals data into a buffer, at two bytes per cycle steal, for the length of a buffer. For the type 2 scanners, the emulation program places the data in a buffer. When a buffer has been assembled, the data is transferred to the host. As the data arrives from the device addressed, it is monitored for an ending control character. When the type 3 scanner or emulation program detects an ending condition, the controller sends the accumulated characters and the ending status to the host processor.

The management of queues is basic to the flow of data within the emulation program. An emulation queue is a chain of character control blocks (CCBs) in communications controller storage. One CCB is associated with each line in the network and holds the information necessary to control that line. One CCB is chained to the next by a control field that contains a pointer to the succeeding CCB in the queue. In the last CCB in a queue, this control field contains zeros. The type 1 channel emulation queues are in the emulation queue control block (QCB). The type 4 channel emulation queues are in the channel control block (CHCB). The emulation queues provide pointers to the first and last CCBs in each queue.

Queues are manipulated within the emulation program by a combination of queue priorities and first-in first-out (FIFO) enqueuing. The priority of a queue is determined by the direction of data transfers associated with the queue, with output queues given the highest priority. Within a given queue, the requests are handled on a FIFO basis. The queues are scanned starting with the highest priority queue and removing the oldest (first-in) entry for that priority.

As previously stated, the emulation program causes the 3705 to perform most of the functions of the 2701, 2702, and 2703. These functions are controlled by the two basic parts of the emulation program: the interface control program (ICP) and the line control program (LCP).

The interface control program provides support for a type 1 or type 4 channel adapter and controls data transfers between the host processor and the line control program (LCP). The ICP executes in program level 3 and is interrupt-driven. The interrupts handled by the ICP are:

- Channel adapter interrupts
- Program-controlled interrupts (PCI) from levels 2 or 3 requesting data transfers to or from the host processor
- Program-controlled interrupts (PCI) from levels 2 or 3 requesting status transfers to the host processor

The line control program (LCP) operates in program level 2. The LCP processes all level 2 interrupts for character, CCB buffer full, and end-of-message service as well as certain error indications.

When a level 2 interrupt occurs (signifying a data service request), control is passed to the portion of the LCP that handles requests for a particular terminal type. The line control characters direct the flow of data between the line interface base and communications controller storage, as well as signaling the LCP to perform such functions as LRC accumulation. Ongoing checking procedures within the LCP recognize the presence of the line-control characters necessary for the completion of an operation.

When an operation ends, the LCP calls for ICP processing by placing the line's CCB in the appropriate queue and causing a program level 3 interrupt (PCI). The LCP passes the status of the line and pertinent pointers to the ICP through fields in the line's CCB.

Introduction Summary

The emulation program is controlled either by interrupts from the host, which drives the interface control program (ICP), or from the scanner, which drives the line control program (LCP). The ICP and LCP use the character control block (CCB) to provide control and data buffers for transfers between the host and terminals. The ICP provides the output command to initiate scanner interrupts and a CCB to define the action to be performed when that interrupt occurs. The line control program communicates with the interface control program by placing a character control block in a queue and, if necessary, causing a program-controlled interrupt (PCI) at level 3 to schedule the ICP code.

Work is scheduled by first-in first-out (FIFO) queuing, with priority for output from the controller before input to the controller. As each line requires servicing, the CCB representing that line is placed at the end of the appropriate emulation dispatching queue.

Emulation Program and 3705 Hardware

The IBM 3705 Communications Controllers have four hardware interrupt levels to provide priority program control for the emulation program. The four interrupt levels initiate program execution at fixed storage addresses. In priority sequence, the interrupt levels are:

- Level 1: Address X'10' provides the entry point for hardware errors and programming errors. Hardware errors include such things as the failure of a channel adapter, scanner, or other 3705 hardware component. Line errors recovery in emulation mode is controlled by the host system. Programming errors include such things as an invalid instruction, an invalid address, or storage protection exception.

- Level 2: Address X'80' provides the entry point for communication scanner interrupts. This address is the entry point for the emulation line control program (LCP). A line interface is enabled for interrupts to send or receive by the interface control program (ICP); but once interrupts occur, all scanner interrupts are processed by the level 2 LCP code.

- Level 3: Address X'100' provides the entry point for the interface control program (ICP). The ICP is initiated by one of two sources: (1) the channel adapter, or (2) a program-controlled interrupt (PCI) from the level 2 line control program (LCP).

The channel adapter interrupt is originally initiated by a command from the host. Once a read or write is being processed over the channel, an interrupt occurs at the end of each defined length of transfer.

The PCI interrupt from the emulation line control program is a request for the emulation interface control program to search the emulation queues for a CCB requiring processing. The LCP places a CCB in a queue when a buffer has been sent or received or when line status is to be passed to the host.

- Level 4: Address X'180' provides a low-priority processing of emulation panel commands.

Figure 12.1 provides a functional overview of the emulation program and the associated hardware. The figure illustrates the major functions that occur during the operation of the emulation program. The detailed functions of the program are described in the *IBM 3704 and 3705 Communications Controllers Emulation Program Program Logic Manual* (For 3705 with Type 1 Channel Adapter (SY30-3001), Section 2: Method of Operations Diagrams or *IBM 3704 and 3705 Communications Controllers Emulation Program Program Logic Manual* (For 3705 with Type 4 Channel Adapter (SY30-3031), Section 2: Method of Operations Diagrams.

The 3705 is attached to a host byte-multiplexor channel by means of (1) a type 1 channel adapter, (2) a type 4 channel adapter, or (3) two type 4 channel adapters. The appearance of the 3705 channel adapter so attached is that of a control unit with one native subchannel address (NSC) for network control program (NCP) and up to 255 emulation subchannel addresses (ESC) for emulation program operation. The channel adapter, with program assistance, enables the 3705 to establish and maintain asynchronous communication with the multiplexor channel. If multiple type 4 channel adapters are installed, a subchannel from each may identify a common line; the first subchannel to enable the line controls the line to a disable.

Data transfers normally occur in bursts of 4, 8, 16, or 32 bytes and are initiated by output instructions. The channel adapter disconnects from the channel at the end of each data transfer; reconnection occurs when the next appropriate output instruction is executed. Data transfer and control operations between the channel adapter and the emulation program are handled by the input and output instructions. An interrupt mechanism is invoked by the channel adapter to inform the emulation program of the receipt of a channel command initiating data transfer or the completion of a data transfer.

The specific functions performed by the emulation program are best described by a discussion of the components of the program: the emulation queues, control blocks, interface control program (ICP), the line control program (LCP), emulation control program (LCP).

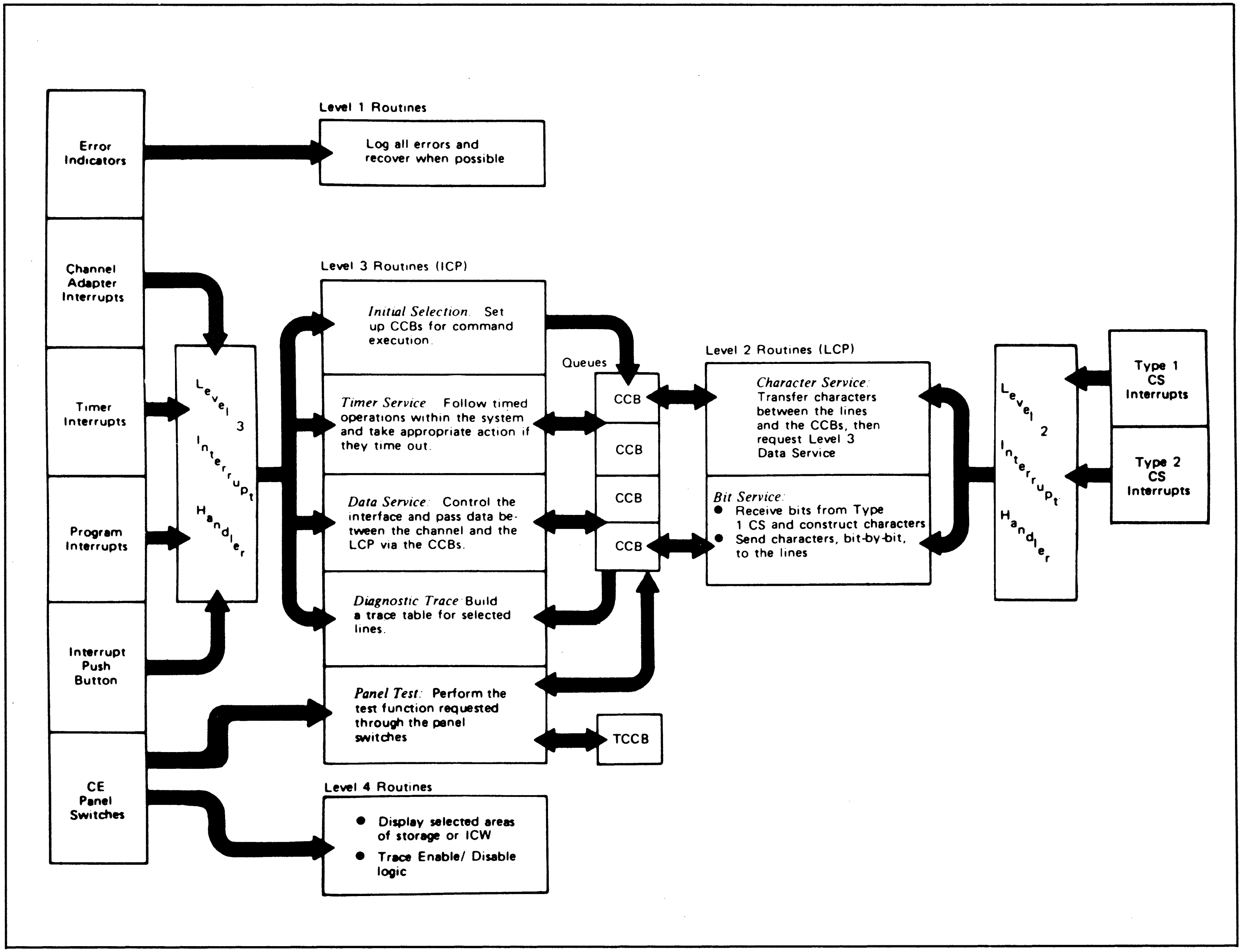


Figure 12.1 Emulation Program Overview

Emulation Queues

Queues are the means by which the interface control program (ICP) and line control program (LCP) communicate in operating communication lines and controlling the scanners and line interface bases. Some of the following queues only apply to type 4 channel adapter channel control block (CHCB), not the type 1 channel adapter queue control block (QCB). The ICP and LCP use the following queues, listed in the order of their service priority, as their interface with each other.

PDSOQ	Priority data service out queue
PEDSOQ	Priority extended data service out queue
DSOQ	Data service out queue
EDSOQ	Extended data service out queue
DSIQ	Data service in queue
EDSIQ	Extended data service in queue
SOQ	Status out queue
PSIQ	Poll service initiator queue
SNOQ	Sense out queue
SSQ	Stacked status queue

The LCP requests ICP processing by entering the CCB for the line into the appropriate queue and causing a level 3 program interrupt (PCI). The ICP scans the queues in order of priority, looking for work. Within each queue, the order in which the requests are enqueued determines the order in which they are serviced; that is, the queues are structured on a first-in first-out (FIFO) basis.

Type 2 scanner lines use only the PDSOQ, DSOQ, and DSIQ for data service transfers. Type 3 scanner lines use the PEDSO, EDSO, EDSI, and PSI for data service transfers. All lines use the SOQ, SNOQ, and SSQ for status and sense transfers.

The location, first and last pointers, of the queues can be found at address X'700' for type 1 channel adapters (QCB) or within the CHCB for type 4 channel adapters. The format is given in LY30-3043.

Priority Data Service Out Queue (PDSOQ) and Data Service Out Queue (DSOQ)

The PDSOQ and the DSOQ are the interfaces through which data is transferred from a type 2 scanner line to the host processor. The functional operation of the PDSOQ is the same as the DSOQ. However, when the ICP scans the queues, it services the PDSOQ first. A BSC line with LINE macros containing the operand CHNPRI=HIGH at emulation program generation time has a '1' placed in bit zero of the CCBFLGB1 field of the CCB. This '1' causes the line control program to enqueue the CCB on the PDSOQ rather than on the DSOQ.

As each data byte arrives from the line the LCP places the byte in one of the buffers located in the line's CCB. When the buffer is full, or when the LCP detects an end-of-data condition from the line, the LCP attaches the CCB for that line to the PDSOQ or the DSOQ. Then the LCP notifies the ICP by a level 3 PCI interrupt that service is requested. When the ICP has honored the request by successfully transferring the data to the host processor, the ICP detaches the CCB from the PDSOQ or DSOQ and indicates that the buffer is again available for use by the LCP.

Priority Extended Data Service Out Queue (PEDSO) and Extended Data Service Out Queue (EDSOQ)

The PEDSOQ and the EDSOQ are the interfaces through which data is transferred from a line attached to a type 3 communication scanner to the host. The functional operation of the PEDSOQ is the same as the EDSOQ; however, when the interface control program (ICP) scans the queues, it services the PEDSOQ first. Type 3 scanner lines whose LINE macros contain the operand CHNPRI=HIGH at generation time have a 1 placed in the bit 0 of the CCBFLGB1 field in their respective CCB's. This 1 causes the line control program (LCP) to enqueue the CCB on the CCBFLGB1 field in their respective CCB's. This 1 causes the line control program (LCP) to enqueue the CCB on the PEDSOQ rather than the EDSOQ.

On each level 2 interrupt that indicates a CCB buffer has been filled or that an ending condition has been detected, the LCP attaches the CCB for that line to the PEDSOQ or EDSOQ. The LCP then notifies the ICP by a level 3 interrupt that service is requested. When the ICP has honored the request by successfully transferring the data to the channel adapter, it detaches the CCB from the PEDSOQ or EDWOQ and indicates that the buffer is again available for use by the LCP.

The PEDSOQ and the EDSOQ are used for channel data transfer requests for airline control (ALC) WRITE commands as well as for READ commands. These queues are used because ALC SYN fill characters are not allowed in ALC messages; therefore, once a transmission has begun, a higher priority queue than the EDSIQ is required.

Data Service In Queue (DSIQ) and Extended Data Service In Queue (EDSIQ)

The DSIQ and EDSIQ are is the interfaces through which data is transferred from the host processor to the line. When the LCP empties a buffer in transmitting data on a line, the LCP indicates that it needs data from the host by attaching the CCB for the line to the DSIQ or EDSIQ. The LCP then signals the ICP by a level 3 program interrupt (PCI) that service is needed. When the data arrives from the host to satisfy the request, the ICP stores the data in the buffer, sets an indication that the buffer contains data, and detaches the CCB from the DSIQ or EDSIQ.

The EDSIQ is use by airline control lines for only the first buffer request of a transmission. Subsequent requests are queued on the PEDSOQ or the EDSOQ.

Status Out Queue (SOQ)

The SOQ is the interface through which ending status conditions are transferred from the LCP or the ICP to the host processor. For commands that require no line-control action, the ICP sets the appropriate ending status -- usually channel end (CE) and device end (DE) -in a byte in the CCB of the associated line. For other commands, the LCP detects an ending condition and places the appropriate ending status condition in the CCB. In all cases, the CCB is then attached to the SOQ. When the status has been sent successfully to the host processor, the associated CCB is detached from the SOQ. If the host processor does not accept the status transfer, a stacked status condition exists. In this case, the ICP detaches the CCB from the SOQ and attaches the CCB to the stacked status queue (SSQ).

Poll Service Initiation Queue (PSIQ)

The PSIQ is the interface through which polling sequences are obtained from the host for poll command processing on lines attached to a type 3 communications scanner. It is also the mechanism by which the frequency of polling operations is controlled. The PSIQ is assigned a priority lower than all the data service queues and the status out queue to provide and automatic governor on polling. The poll command initial selection routine enqueues the CCB on the PSIQ with buffer controls indicating both buffers should be filled. The PSIQ initiator, after having filled both buffers with polling sequences from the host, initiates the transmission of the first polling sequence and dequeues the CCB from PSIQ. The Poll ICP receives the response to the polling sequence and if negative, enqueues the CCB on the PSIQ again. The PSIQ initiator, when it gets control, will set up the transfer of more polling sequences from the host when necessary and will again initiate the transmission of the next polling list entry. The sequence continues until a positive response to the POLL is received or until the polling list has been exhausted.

Sense Out Queue (SNOQ)

The SNOQ is used by the ICP to present one byte of sense data to the host processor. The data is produced as the result of the execution of the last command and is transferred to the host processor when a host Sense command is executed.

Stacked Status Queue (SSQ)

When the ICP attempts to present an ending status to the host and finds that Suppress Out is active, the ICP removes the associated CCB from the SOQ and attaches the CCB to the SSQ. When the type 4 channel adapter indicates by an interrupt that Suppress Out has become inactive, or if Suppress Out is inactive when the SSQ is scanned, the ICP attempts to present the status from the CCBs on the SSQ.

Emulation Control Blocks

Direct Addressables

The direct addressables support the emulation program within a partitioned emulation program. Use of these areas in an emulation program provides the following pointers:

- X'68B' identifies a dump as including NCP and EP (PEP) with a value of:

X'x3' for a type 1 channel adapter

X'xB' for a type 4 channel adapter

Type 1 Channel Adapter

- X'700' to X'733' contains the emulation queue control block (QCB) which contains the EP queue pointers.
- Extended Halfword Direct Addressables (HWE) plus X'30' is the halfword address of the PEP channel vector table (CHVT).

Type 4 Channel Adapter(s)

- X'710' is the halfword address of the first channel control block (CHCB).
- X'712' is the halfword address of the second channel control block (CHCB).
- The CHCB contains the PEP queue pointers.
- The CHCB contains the channel vector table (CHVT) at offset X'68'.
- Extended Halfword Direct Addressables (HWE) plus X'30' is the halfword address of the first PEP channel vector table (CHVT). If a second PEP channel is defined the address of the second CHVT is in the last halfword of the first CHVT.

Character Control Block (CCB)

The character control block (CCB) contains current information on the physical operation of the line and the data that is being transferred to or from the line. This information is needed to control processing of the lines and to connect the processing from one level 2 interrupt to the next.

During emulation generation, one CCB is generated for each line specified (LINE macro) and the fixed parameter values that are included in a CCB also are determined. For each line (and its associated subchannel address) within the high-low range of the specified subchannel addresses, but not itself specified, the emulation program generation creates a ten-byte dummy CCB.

The CCB is the source for the definition of line information for the ICW. If the line definition is in error, resulting in a CCB definition error, the initialization of the ICW will also be in error.

Each CCB is pointed directly to by an entry in the line vector table (LNVT) and indirectly by an entry in the channel vector table (CHVT). These tables are covered later in this manual.

The emulation character control block (CCB) is illustrated in LY30-3043, Section 2: Data Area Layouts, Character Control Block CCB (PEP).

The current command (X'10') values are in LY30-3043, Section 7: PEP Command Codes.

Queue Control Block (QCB) (type 1 channel adapter)

The emulation queues for a type 1 channel adapter are within the emulation queue control block (QCB). The QCB is generated at X'700'. The emulation queues for a type 4 channel adapter are within the channel control block (CHCB).

The emulation queue addresses provide pointers to the first and last CCBs on all queues. The queue pointers are created during emulation generation and is modified by both the interface control program (ICP) and the line control program (LCP).

The fields of the emulation queues are two-byte address pointers, with each queue using two fields. The first field used by a queue is a pointer to the first CCB in that queue. The second field is a pointer to the last CCB in that queue. The types of queues are covered later.

The format of the emulation QCB is described in LY30-3043, Section 3: Data Area Layouts, Queue Control Block QCB (PEP).

Channel Control Block

The channel control block (CHCB) is generated for each type 4 channel adapter defined. The CHCB contains the queue pointers, native subchannel CCB, and channel vector table. The key fields are at the following offsets:

- X'00' Channel select bits and PEP flags
- X'0A' to X'30' Emulation queue pointers
- X'3E' Native subchannel CCB (42 bytes)
- X'68' Channel vector table (CHVT)

The format of the CHCB is in LY30-3043, Section 2: Data Area Layouts, Channel Control Block CHCB (PEP).

When type 4 channel adapters are defined, address pointers to the first type 4 PEP channel adapter CHCB are found at X'710' and to the second type 4 PEP channel adapter CHCB at X'712'.

Channel Vector Table (CHVT)

The channel vector table (CHVT) enables level 3 routines to find the CCB for a line via the LNVT when only the subchannel address is known.

The primary fields of the CHVT are:

- X'00' Lowest subchannel address
- X'01' Highest subchannel address
- X'02' to X'nn' Two-byte entries for each possible subchannel as defined by the first two entries: if the low-order bit is 0, an entry is the address of an entry in the line vector table. If the low-order bit is 1, the entry is an undefined and invalid entry.
- X'nn' plus 1 X'0001' delimiter entry

The format of the CHVT is shown in LY30-3043, Section 2: Data Area Layouts, Channel Vector Table CHVT (PEP). The 'old base' CHVT is for a type 1 channel adapter. The 'new base' CHVT is for a type 4 channel adapter.

In a PEP system, the CHVT pointer is in the extended halfword direct addressables (HWE) at offset X'30'. If a type 4 channel adapter is defined, the CHVT is within the channel control block (CHCB) at offset X'68'. Each type 4 channel adapter has a CHCB and imbedded CHVT. Address pointers to the CHCB are at X'710' and X'712'.

Line Vector Table (LNVT)

The line vector table (LNVT) allows routines to find the CCB for a line when only the line address is known. The logic for emulation mode is the same as NCP mode. Information on the LNVT was provided under the BSC/SS Processor and SDLC Link Scheduler.

Line Group Table (LGT)

One line group table (LGT) is created for each GROUP macro coded during emulation generation. At least one GROUP macro must be coded, containing the parameters common to a group of lines. Refer to (LY30-3043), Section 2: Data Area Layouts, Line Group Table (LGT).

Translate/Decode Table (TDT) (start-stop lines only)

For start-stop terminals on a type 2 scanner, the high-order bit of each data byte received from the host must be the first bit sent out over the line to the terminal. Since the type 2 scanner sends the low-order bit of the serial data field (SDF) of the ICW to the line first, the order of the bits in the data byte received from the host must be reversed and right-justified before being placed in the parallel data field (PDF) of the ICW. A similar situation exists when the scanner is receiving from the line; the character is assembled in the serial data field (SDF) in reverse order. Once again, the data byte must be reversed and right-justified before it is transmitted to the host processor. The translate/decode table allows the reversal to be done by table look-up and the justification to be done in code.

A cross reference is provided in the (LY30-3043), Section 15: Line Character Codes. For each line code, the charts illustrate the PDF code (reversed line code), the EBCDIC value, and the line code. An analysis of line traces, ICW displays on the panel, or the TDT requires this reference.

For IBM Type III start-stop lines, translation is done by code alone, and the TDT is not used.

WU Translate Table

The WU translation table is 64 bytes of translation data within the CYANUC control section. The table, which is used by data service routines for start-stop terminals to assist in translating Western Union code, is located in the CYAL3H routine of the CYANUC control section. The CYANUC control section and CYAL3H entry are identified in the linkage editor cross-reference table of the generation. The WU translation table location can be determined from the microfiche.

ICE Routine Address Table

The ICE routine address table is in module CYASVC in routine CYAIS. The 128-byte table contains pointers to the routines which provide normal execution of commands from the host. The routines are identified by a prefix of ICE for valid entries or CMDERROR for error conditions.

Command Table

The command table is in module CYASVC in routine CYAIS. The entry can be located in a linkage editor cross-reference table of the generation. The 48-byte table contains the CCB command codes used for translating the eight-bit command code into the five-bit CCB command code.

Interface Disconnect Dispatcher Table

The interface disconnect dispatcher table is in module CYASVC in routine CYAIS. The 64-byte table provides address pointers to interface disconnect routines with routine prefix names of IFD and CAEC. The routines are identified by two-byte addresses.

EBCDIC Character Decode Displacement Table

The EBCDIC character decode displacement table provides an offset into a branch table for proper control character processing. The 64-byte table is located in module CYABL. The offsets are used by CYATADA0 and CYARAPH1.

USASCII Character Decode Displacement Table

The USASCII character decode displacement table provides the same function for USASCII code as the EBCDIC table provides for EBCDIC code. The 32-byte table is also located in module CYABL, however, the offsets are used by PARTYCK and ASCXMT.

Test Control Block

The test control block is located in module CYATST. The 30-byte table provides current information about the line being tested, as specified by the panel test routine.

Line Trace Table

The emulation trace table is generated by coding the BUILD macro operand of LINETRC=(YES,n,m); n is the number of concurrent emulation line traces, and m the number of 8-byte table entries. The trace control table is 16-bytes, with address pointers to the current entry, beginning of the table, end of the table, and four bytes of indicators. The PEP trace table entries are covered in LY30-3043 with educational material in *Advanced Function NCP and Related Host Traces* (SR20-4510-4).

Functions of the Interface Control Program (ICP)

The functions of the ICP are:

- To meet the channel adapter hardware interface
- To control initial selection
- To control data and status transfers to and from the host program

When more than one type 4 channel adapter is being used, ICP can be communicating with several host processors. ICP executes in program level 3 and is interrupt driven. Interrupts handled by the ICP are:

- Channel adapter hardware interrupts
- Program-controlled interrupts (PCI) from level 2, requesting data transfers to and from the host processor
- Program-controlled interrupts (PCI) from level 2 requesting status transfers to the host processor

Upon receipt of a level 3 interrupt, the ICP causes the 3705 to select the type 4 channel adapter with the highest priority request pending. The level 3 interrupt will then to be dedicated to processing with that channel adapter.

The channel adapter operates in three active states:

- Initial selection
- Data servicing
- Final status servicing

These states are defined and controlled by the external control registers in the channel adapter. For example, the initial selection command and address data are in register X'61'. For an explanation of these states, refer to the *IBM 3704 and 3705 Principles of Operation* (GC30-3004).

Initial Selection State Processing

Upon receipt of a level 3 interrupt, the ICP recognizes the initial selection state by examining the results of an IN X'77' instruction (adapter interrupt requests group 2). The ICP identifies the command and subchannel address by examining the initial selection address and command register. The ICP gains access to this information by executing input instructions for the appropriate external registers.

Since the channel adapter accepts all possible command bytes as valid, the ICP must validate the command received. The ICP rejects all commands that are not valid for a 2701, 2702, or 2703 and also determines whether the command is valid for the line addressed.

If the command is issued to a MSLA subchannel, the ICP determines if the line is available for use by this subchannel. If the line is not available, error status and sense are generated to so inform the host program.

Data Transfer State Processing

The ICP initiates the data transfer state to prepare the channel adapter for a data transfer operation requested by the host processor. The data transfer state is activated by an output instruction for the data/status control register.

For write operations from the host processor, the ICP places the subchannel address of the associated communication line in the address/ESC status register. Next, the ICP places the number of bytes (up to 32 bytes) to be transferred in the data/status control register (DSCR) and instructs the channel adapter to request data from the host by setting the inbound data transfer sequence bit in the DSCR. When selected, the channel adapter reconnects to the channel, accepts the data, and disconnects from the channel. After placing the data in the data buffer registers, the channel adapter requests a level 3 interrupt for the ICP. The ICP fetches the data by executing input instructions and stores the data in the buffer. Finally, the ICP scans the emulation service and status queues for additional service requests for other lines.

For Read operations from the host processor, the ICP takes the data to be transferred from the data buffer, places the data in the data buffer registers, and places the subchannel address of the line from which the data came in the address/ESC status register. Next, the byte count and outbound data transfer sequence bit are then set in the data/status control register. The channel adapter then connects to the channel, transfers the data, disconnects from the channel, and notifies the ICP with a level 3 interrupt request that the data was transferred. Finally, as in Write operations, the ICP scans the emulation service and status queues looking for additional requests.

The processing of the channel interface sense command is identical to that of a Read command, except that only one byte of data in transferred.

Final Status State Processing

When the ICP recognizes an ending condition such as end-of-transmission, interface disconnect, etc., it instructs the channel adapter to transfer ending status. The ICP places the ending status in the address/ESC status register along with the subchannel address of the line and sets the ESC final status transfer sequence bit in the data/status control register.

If the host channel does not accept the status byte, the channel adapter indicates this fact to the ICP by a level 3 interrupt. The ICP then attaches the corresponding CCB to the stacked status queue. When the ending status is transferred normally, the channel adapter causes a level 3 interrupt and the ICP scans the service and status queues for outstanding service requests.

Interface Control Program (ICP) Interaction with the Line Control Program (LCP)

The ICP and LCP work together to operate communication lines and to control the communication scanners and line interface bases. The ICP initiates and terminates all communication line operations performed by the emulation program. An ICP operation begins when one of the following occurs:

- The host channel adapter issues a level 3 interrupt because the host processor has issued a new channel command.
- The LCP requests data service from or to the host processor.
- The LCP requests that the ICP notify the host processor of an ending status condition.

Interface Control Program (ICP) Interaction with the Host Channel Interface

The ICP manages both input and output channel operations in conjunction with the host processor. In the emulation program, channel management is unique, since it manages a resource over which it has limited physical control. Although the host has absolute control of the initiation of channel command operations, the emulation program controls the data-transfer and status-transfer operations. The host can write only the data that the emulation program requests and can read only the data that the emulation program transfers.

The logical unit of data transfer over the channel is the byte. Data is transferred in either 4, 8, 16 or 32 bytes, depending upon the buffer size. That is, the channel adapter connects to and disconnects from the channel for each burst of data, up to the buffer length or 32 bytes maximum. If the transfer of data requires the use of multiple CCWs in the host, the CCWs must be chained together to prevent status interrupts to the host and to prevent 'channel stop' from being presented to the channel adapter during command execution.

Data Transfer

The ICP provides the program assistance required by the channel adapter for data transfer. Since the channel adapter does not have access to 3705 storage, the ICP must fetch data from the buffers and place it in the channel adapters data buffer registers for outbound transfer. For inbound transfers, the ICP retrieves data from the channel adapters data buffer registers and stores it in the buffers.

Normally, channel stop from the channel or an end-of-transmission character indicates the end of a Transmit operation to the ICP. The end-of-transmission character from a terminal normally indicates the end of a Receive operation. The emulation program puts no limits on the number of bytes of data that may be transferred for a single command. Therefore, the ICP continues to transfer data until one of the ending conditions occurs.

The data transferred to the host processor is that data collected by the LCP. When the LCP has data ready for transfer to the host, the LCP enqueues the CCB of the associated line on the DSOQ. If all the service and status queues are empty, the LCP requests a program level 3 interrupt (PCI) to alert the ICP. The ICP then conditions the channel adapter to transfer the data to the host processor. If there are already entries on the service queues when the CCB is added, no level 3 interrupt is requested by the LCP. As each CCB processing completes, the emulation queues are searched for additional CCBs in the queues for processing.

Functions of the Line Control Program (LCP)

The principle function of the LCP is to transfer data units. The data units are bytes between the ICW and the buffer for a type 2 or type 3 scanner. The LCP has exclusive use of program level 2 for processing scanner requests for service, as well as certain error and special conditions, such as autoanswering and disconnect on switched lines. The LCP normally relies on the ICP to initiate operations by preparing the CCB for transfer and enabling the line interface for interrupts. Once an operation is initiated, the LCP processes the resulting level 2 interrupts. When the LCP detects the end of an operation, it signals the scanner to terminate its current operation.

When an operation ends, the LCP calls for ICP processing by enqueuing the CCB for the line in the appropriate queue and requesting a level PCI interrupt. The LCP passes the status of the line and pertinent pointers to the ICP through fields in the line's CCB. The LCP then initiates the next operation on the line or signals the ICP that the operation has ended.

When a level 2 interrupt occurs, signifying a data-service request, control is passed to the routine in the LCP that handles requests for a particular terminal type. The routine monitors the characters received from or transmitted to a line. The characters are either data characters or control characters. If the character is a data character, the LCP acts only as a transfer stage for the data. Otherwise, an operation is initiated as required for the particular control character (such as end-of-transmission sequences for an EOT character).

Line Control Program (LCP) Transmit Functions

The LCP performs the following functions when in transmit mode (that is, when transferring data from storage to a communications line):

- Checks for certain hardware-detected error conditions indicated in the secondary control field (SCF) of the ICW of the type 2 or type 3 scanner.
- Maintains the current state of the line in terms of line-control procedures, type of terminal, and operation in progress. The state of the line determines what functions are performed during the current program level 2 interrupt, how data characters are decoded and/or interpreted, and the condition that causes the state to be changed. Examples of transmit states are transmit text, transmit control, transmit LRC, transmit transparent text, and end transmit. Each line-control procedure has its own set of states.
- Adds a parity bit where required to transmitted characters in BSC-ASCII code. The LCP accumulates the LRC or CRC block-check and transmits it at the end of the block (or intermediate block).
- For BSC transmission, inserts synchronous idle characters into transmitted blocks at one-second intervals to comply with BSC operating procedures.
- For start-stop transmission, if appropriate, interprets a 'case' bit (upper or lower) appended to each translated character and transmits the required shift character with each change in case.

Line Control Program (LCP) Receive Functions

When receiving data from a line, the LCP performs the following functions:

- Checks for certain hardware-detected error conditions, such as character overrun, which are indicated in the secondary control field (SCF) of the ICW of the type 2 or type 3 scanner.
- Maintains the current state of line in terms of the line-control procedures, type of terminal, and operation in progress. The state of the line determines what functions are performed during the current level 2 interrupt, how data characters are decoded and/or interpreted, and the conditions that cause the state to be changed. Examples of receive states are receive control, receive text, receive ITB, receive transparent text, and hunt for sync.

- Decodes and acts upon both communication line-control characters and certain device-control characters, according to the current state. Generally, a decoded control character acts to change the state and, consequently, the subsequent operation on that line. Characters that are decoded in one state may be ignored or have a different effect in other states.

- Deletes control characters (up-shift, down-shift, sync idles, CRC accumulation, etc.) from the incoming message.

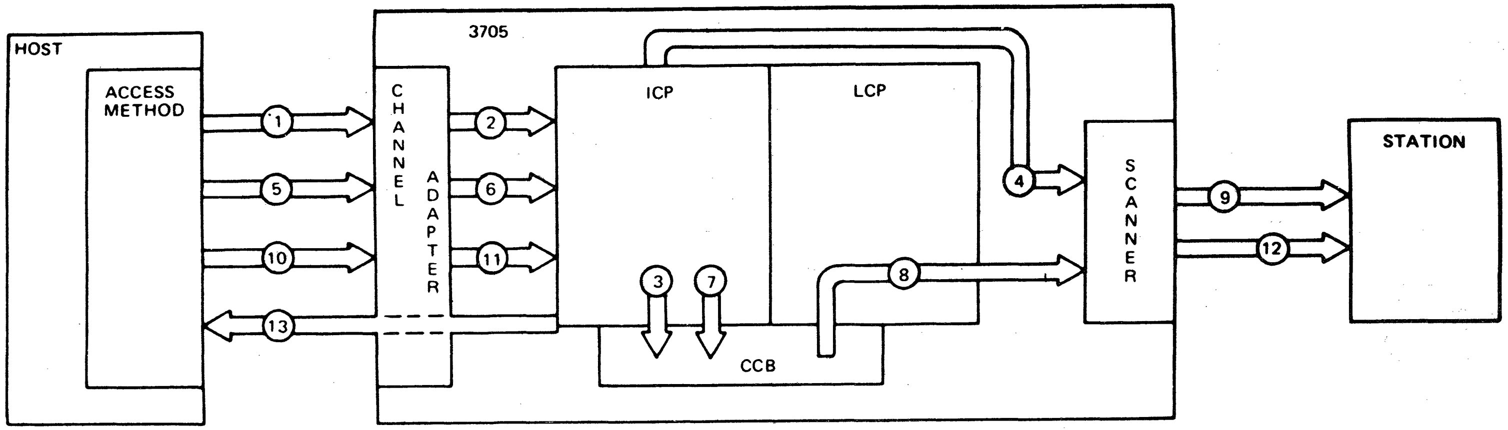
- May translate the characters in storage from scanner format to host processor format. This step is not a true code translation, but merely a reordering of bits. (For example, a C1248ABX character is received, but a XBA8421C character is transferred to the host processor.) *ACF/NCP/VS Network Control Program Program Reference Summary* (LY30-3043), Section 15: Line Character Codes, illustrates the C1248ABX character under the PDF code column, and the XBA8421C under the line code column.

Line Control Program (LCP) Interaction with the Interface Control Program (ICP)

While the LCP is in transmit mode, the DSIQ is used to fill the buffers for transmission onto the communication line. Each time the LCP empties one of the line's buffers, it attaches the CCB for the line to the DSIQ in order to notify the ICP of a request for service. If the DSIQ and all other queues are empty and the channel adapter is not servicing data, a level 3 PCI interrupt signals the ICP that data service is required.

While in Receive mode, the LCP places the characters from the line in one of its buffers (type 2 scanner) or sets up a buffer for buffer for cycle-steal mode (type 3 scanner). When the buffer is full or an ending condition has been detected, the ICP attaches the CCB for the line onto the DSOQ to request a data transfer to the host processor. If the DSOQ was previously empty and the channel adapter is not servicing data, a level 3 PCI interrupt signals the ICP that service is needed.

The LCP sets end-of-transmission, end-of-block, or other ending conditions in the CCB, then places the CCB on the SOQ. A level 3 PCI signals the ICP that service is required if all of the queues are empty.



1. The Host issues a WRITE command.
2. The Channel Adapter interrupts the ICP.
3. The ICP validates the command and prepares the CCB.
4. The ICP prepares the line.
5. The Host sends the data (up to 4 bytes).
6. The Channel Adapter interrupts the ICP.
7. The ICP places the data in the CCB.
8. The LCP removes the data from the CCB and sends it to the Scanner.
9. The Scanner sends the data to the Station via the LIB.
10. The Host completes its data transmission.
11. The Channel Adapter interrupts the ICP.
12. The Scanner completes its data transmission.
13. The ICP sends the ending status to the Host via the Channel Adapter.

Figure 12.2. Data Flow from Host to Station.

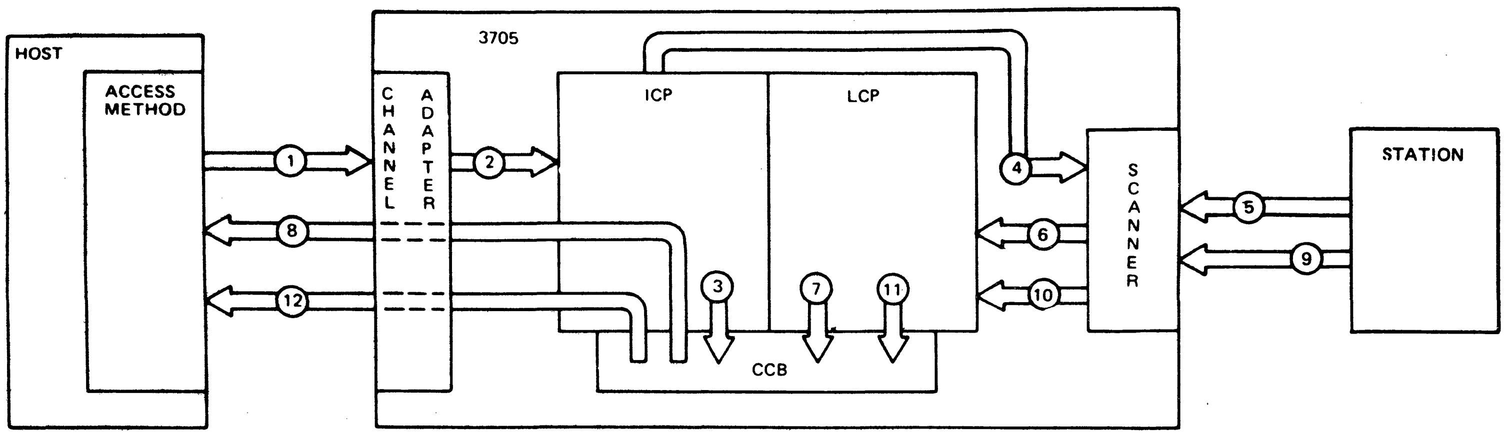


Figure 12.3. Data Flow from Station to Host.

1. The Host issues a READ command.
2. The Channel Adapter interrupts the ICP.
3. The ICP validates the command and prepares the CCB.
4. The ICP prepares the line.
5. The Station sends the data.
6. The Scanner interrupts the LCP.
7. The LCP places the data in the CCB and interrupts the ICP
8. The ICP removes the data from the CCB and sends it to the Host via the Channel Adapter.
9. The Station completes its data transmission.
10. The Scanner interrupts the LCP.
11. The LCP stores the ending status in the CCB and interrupts the ICP.
12. The ICP removes the ending status from the CCB and sends it to the Host via the Channel Adapter.

PEP LINE Switching

There are two LINE definitions in partition emulation support:

1. LINE TYPE=EP; emulation lines are dedicated to emulation only
2. LINE TYPE=PEP; the line is switchable between emulation and network control program mode.

A line which is defined for emulation mode only has an emulation line group table and an emulation CCB. A line defined for both emulation and network control program generates a line group table and CCB for each mode of support. The address pointers in the control blocks indicate which of the modes is currently active.

There are two directions of traffic, and two modes of processing. The conditions for a lines defined as both emulation and network control mode are as follows:

EP Mode on a Switchable Mode Line

Line Input

The line control block (LCB) has a pointer to the ACB. The ACB contains the NCP mode CCB with the CCBBAR field with an address value of less than X'800' for an emulation mode line. The LCB also has an address pointer to the channel vector table (CHVT) entry for this line. The CHVT points to a line vector table (LNVT) which contains the address of the emulation CCB.

Channel Input

Channel input is associated with a subchannel address which provides an entry to the channel vector table (CHVT). The CHVT entry contains an address pointer to the line vector table (LNVT) entry for this subchannel. The LNVT entry points to the emulation mode CCB.

NCP Mode on a Switchable Mode Line

Line Input

The line control block (LCB) contains an address pointer to the adapter control block (ACB). The ACB contains the NCP mode character control block (CCB). The CCB field of CCBBAR contains an address of X'800' or greater for NCP mode. This CCBBAR address points to an entry in the Line Vector Table (LNVT). The LNVT entry address points to the original ACB.

The LCB also contains an address of an entry in the channel vector table (CHVT). In NCP mode the CHVT entry points to an emulation dummy CCB rather than an entry in the LNVT.

Channel Input

Channel input over the emulation subchannel causes the channel vector table (CHVT) entry associated with this line to be referenced. When the line is in NCP mode the CHVT entry points to an emulation dummy CCB rather than an entry in the LNVT. A subchannel communication for a line in NCP mode causes the NCP to reject the subchannel command.

PEP Line Mode Switching

The original definition of a line for both emulation and network control mode is specified on the LINE macro as TYPE=PEP. The original mode of the line is specified on the LINE macro as USE=NCP or USE=EP. The mode switch is requested by a FID1 request PIU with X'52' (switch to EP) or X'51' (switch to NCP) request code. Lines that are designated as 'switchable-mode' at NCP generation have both an NCP adapter block (ACB) and emulation character control block (CCB) generated for them. The system router from NCP physical services performs the mode switch in the manner described below. Refer to figure 5, NCP Pointers to the CCB, in LY30-3043.

Switching from NCP Mode to EP Mode

The system router checks the pointer (CCBBAR) to the line vector table (LNVT) to determine what is the current line mode. If the value in the pointer is less than X'800', the line is already in EP mode, and a response is returned to the host. If the pointer's value is equal to or greater than X'800', the system router uses the subchannel address in the LCB to locate the channel vector table (CHVT) entry containing the EP CCB pointer. It puts the CCB pointer into the line vector table (LNVT) and puts the address of the LNVT entry into the CHVT entry. The system router then decrements CCBBAR by X'800' to flag the line as being in emulation mode.

Switching from NCP mode only occurs if there are no active sessions on the line in NCP mode.

Switching from EP mode to NCP Mode

The system router checks CCBBAR to ensure that the line is in emulation mode. If CCBBAR is equal to or greater than X'800', the line is already in NCP mode and a response is returned to the host indicating a user error. If CCBBAR is less than X'800', the system router adds X'800' to CCBBAR value, making it point to the LNVT entry for this line. The router then puts the ACB address into the LNVT and puts a dummy CCB pointer into the CHVT entry for the subchannel.

Switching from EP mode occurs whenever there is no current EP command. This may be between valid EP commands in a BTAM sequence, and not just at the end of a BTAM sequence. The user should ensure that the emulation subchannel is not active before switching from NCP mode.

Emulation Program Functions Summary

The interface control program provides support for the channel adapter and controls data transfers between the host processor and the line control program (LCP). The ICP executes in program level 3 (entry point X'100') and is interrupt-driven. The interrupts handled by the ICP are channel adapter interrupts, PCI interrupts from levels 2 or 3 requesting data transfers to or from the host processor, and PCI interrupts from levels 2 or 3 requesting status transfers to the host processor.

The line control program operates in program level 2. The LCP processes all level 2 interrupts for character service and buffer service, as well as certain error conditions. A level 2 interrupt (entry point X'80') signifies a data service request and control is passed to terminal type dependent code. Control functions are performed by this code. When an operation ends, the LCP places the CCB for the line in one of the emulation queues and PCIs level 3 if required. Line status is passed in CCB control fields.

The flow of data in an emulation program is illustrated in Figures 12.2, showing data flow from the host processor to the station through the emulation program, and 12.3, data flow from the station to the host processor through the emulation program.

Chapter 13:

Service Aids and Diagnostics

Purpose of Service Aids and Diagnostics

Service aid facilities and panel support routines provide the means for isolating and/or interrogating NCP failures. The NCP supports several functions to aid in problem determination and diagnostics. This topic explores the aids and diagnostic facilities available with the NCP, describes their implementation, and discusses their output.

For additional information, refer to *IBM Advanced Function NCP and Related Host Traces* (SR20-4510).

Dynamic Panel Display

The NCP allows you to display and change dynamically the following types of information on the 3705 control panel:

- Communication scanner interface control word (ICW)
- Contents of external registers
- Contents of a halfword of 3705 controller storage

Dynamic display functions are selected by setting the display/function select and storage address/register data switches on the panel and pressing the interrupt key.

NCP uses a group of routines to process level 3 interrupts from the panel. The panel control block (PCB) is the common data area for all panel routines.

In addition to display, ACF/NCP allows 3705 storage to be modified dynamically. The modified storage address can be a byte, two bytes, or fullword.

The panel routines are provided in the following:

Guide to Using the IBM 3705 Communications Controller Control Panel (GA27-3087)

Line Test

The line test facility allows the user to address, poll, dial, and transmit to or receive from a terminal. Testing is initiated by entering variables through the 3705 control panel. The status of the line resulting from the test is displayed in the panel lights. The line test control block (LTS) contains control information for panel test operations. See the control panel guide mentioned above (under 'Dynamic Panel Display') for operating instructions.

The line test facility is included in the NCP whenever SS or BSC devices are defined for the generation.

Address Trace

The address trace facility allows the user to select any combination of up to four registers and storage halfwords, contents of which are to be recorded each time data is loaded from or stored into a specified 3705 storage address at a specified program level. The NCP records the trace data in a trace table within controller storage. The contents of the trace table can be displayed on the control panel or examined in a dump listing. The address trace control block (ATB) has the address trace control information within it.

Operating procedures are given in the control panel guide identified above (see the section on 'Dynamic Panel Display').

Suspected errors within the network control program can be monitored by recording external registers and changes to storage addresses. This tool is used to identify NCP programming errors or hardware errors which can be identified in external registers.

The TRACE operand of the BUILD macro specifies whether the address trace facility is to be included in the NCP and specifies the size of the trace table.

Channel Adapter Trace

Channel adapter trace is an optional diagnostic and debugging aid that stores certain fields from the channel control block (CAB) in a trace table. An entry is made for channel adapter spurious interrupts, channel adapter level 3 interrupts, and level 1 interrupts caused by channel adapter errors.

The trace is included in the NCP by coding CATRACE=(YES,n) on the BUILD macro. The n value defines the number of 32 byte trace entries are reserved in storage.

This trace can be activated and deactivated from the 3705 panel. The trace involves significant overhead, especially with a type 1 or type 4 channel adapter. The trace should not be activated except in cases where a suspected or known channel error must be isolated.

Line Trace

The line trace facility is a diagnostic and debugging aid that stores certain fields from the ICW each time a level 2 interrupt occurs on a designated communication line. Line trace is activated and deactivated by network control commands from the host. The number of lines which can be traced concurrently is coded on the BUILD macro operand of LTRACE (1 to 8, default of 2). If the line is duplex, both legs are traced. The fields traced are the line control definer (LCD), primary control field (PCF), secondary control field (SCF), and the parallel data field (PDF). A timer field is also included. The line trace control block (LTCB) contains pertinent information about the trace.

An explanation of the line trace fields is available in *Advanced Function NCP and Related Host Traces* (SR20-4510).

Error and Statistic Recording

The NCP sends unsolicited Record Maintenance Service (RECMS) PIUs to the access method to record the following conditions:

- BSC/SS device or line errors
- BSC/SS station statistics
- Channel adapter errors
- Communications scanner errors
- I/O instruction exceptions
- Unresolved program level 1 and level 3 interrupts
- SNA link errors
- BSC 3270 status/sense

- SNA statistics
- Intensive-mode recording for SNA station errors

The Miscellaneous Data Recorder (MDR) record is a subset of the RECMS record. Previous documentation refers to the subset as MDR records.

These records are transferred to the host either when permanent errors occur or when temporary error counters overflow. These records can be analyzed for intermittent and permanent errors of 3705 hardware and line errors. A permanent failure of the network control program may not allow the RECMS records to be sent to the host. These records or record counters are then available in a dump of the NCP.

If you want to analyse the types of temporary line errors, the RECMS records can provide intensive-mode records of each temporary error. If a permanent error occurs, you may find the solution using RECMS records, or the error may require a line trace.

Online Tests

The online tests (a user selected option) provide the IBM customer engineer with online maintenance capability. Testing is performed by one or two routines depending on the type of resource to be tested. The online line tests (OLLT) check BSC/SS lines and SDLC links. The online terminal test (OLTT) checks BSC/SS devices. Both tests are controlled by the terminal online test executive program (TOLTEP) which resides in the host.

To include this facility in NCP, code the OLT=YES operand in the BUILD macro.

Abend

Programming errors detected during execution of supervisory and nonsupervisory code of the NCP cause an abnormal end of program execution. The examination of abend codes within an NCP dump can help in locating the error. The optional abend service aid extends detection of programming errors to the NCP supervisor, thus causing the program to terminate before a supervisor error can be propagated into nonsupervisory portions of the program. The abend service aid stores an abend code at X'760' in controller storage and the controller is hard-stopped.

To include the abend service aid for programming levels 1 through 4 of the NCP, code ABEND=YES in the BUILD macro.

Appendix A:

PIU Command Sequence

PIU Command Sequences

The following SRL references may be of assistance:

ACF/NCP/VS Network Control Program Logic (LY30-3041), Appendix A: Network Commands

ACF/NCP/VS Network Control Program - Program Reference Summary (LY30-3043), Section 5: NCP Network Commands

This section describes the command sequence to be followed for activation and session initiation for switched SDLC. Each entry in the 'switched' sequence is marked with an asterisk; you can determine the 'nonswitched' SDLC sequence by skipping those entries. In addition, this section identifies the general processing within the NCP and specifies the NCP control block changes made to record command processing.

If the correct operation takes place, the following command sequence occurs on a PIU trace:

1. Activate physical -- from SSCP to NCP physical services

This command is enqueued on the physical services block (PSB) connection point manager (CPM-OUT) queue. The vector table of SNPs (VTS) is used to obtain a SSCP-NCP session control block (SNP). The 'session established' bit in the SNP (X'02' bit 0) is set to 1.

The activate physical command to NCP physical services initiates an initialization complete PIU (see command 2) to be sent to the origin of the activate physical command. The initialization complete request is passed to INN path control before the activate physical response.

A configuration restart is transparent to the NCP. Configuration restart specified in the host selects a warm or cold restart.

The cold restart results in a new initial program load or an initialization of a loaded NCP which has not been previously activated. Cold restart results in commands being directed only to the network addresses which are to be activated (ISTATUS=ACT). A warm restart to a previously activated NCP has a network command addressed to every network resource: an 'activate' to network addresses to be initially active, and a 'deactivate' to each network address to be inactive. The active or inactive status of each network addressable resource is maintained in the disk configuration data set of VTAM. A warm restart allows an NCP with partitioned emulation program (PEP) to be restarted without affecting the emulation lines by reloading.

The response in the request/response unit (RU) contains the name of the NCP. This name is the NEWNAME operand value from the BUILD macro, which was obtained from the physical services block (PSB) at PSBLDID.

NOTE: This command completes the first level of sessions between SSCP and NCP.

2. Initialization complete -- from NCP physical services to SSCP

This message is generated during processing of an activate physical command to NCP physical services. The initialization complete is sent to the OAF of the activate physical. The request is passed to INN path control for routing. No response is requested. This request flows before the response to the activate physical.

3. Start data traffic -- from SSCP to NCP physical services

This command is enqueued on the PSB CPM-OUT queue. The 'data flow enabled' bit and 'data flow active' bits of the SNP at offset X'02'. A response is provided to SSCP from PSB CPM-IN and passed to INN path control for routing.

4. Set control vector -- from SSCP to NCP physical services

This command is optional (from NCP requirements) during activation. In the request unit (RU) byte 5, a value of 01 identifies this as the command which provides the time and date to be stored in the time and date control block. A response from PSB CPM-IN is sent to INN path control for routing.

5. Set control vector -- from SSCP to NCP physical services

This command is optional (from NCP requirements) during activation. In the request unit (RU) byte 5, a value of 05 identifies this as the command which changes the channel delay from the user-coded value of zero for the duration of the bring-up sequence. A response from NCP CPM-IN is sent to the OAF.

The same command is used to change the channel delay back to the original value after initialization is complete.

6. Activate link -- from SSCP to NCP physical services

A command for each defined link is sent to NCP physical services CPM-OUT. Each command identifies the network address of the link to be activated in the request unit (RU) field of RU1NA. The PSB CPM-OUT triggers the link control block (LKB) which defines the link.

The activate link command obtains ownership of the link for the SSCP which issued the command. Nonswitched SDLC links may have concurrent owners. All other links are owned serially.

The processing initiates the link scheduler code (level 3 NCP code) to search the service order table for that link for work to be done. For a nonswitched link, the modem is enabled. For a switched link, the modem is not enabled until a connect out (dial) or connect in (answer) command is processed. The active status of the link is provided in the LKB in LKBSTAT. The task address at LKB LKWTSKEP is changed from the run initiator to the address of the run terminator. (The change of the task address is referred to later under connect out, connect in, and contact commands.) Finally, when processing is complete, the PSB CPM-IN is triggered to send a response to the OAF.

The link scheduler is now active for this line definition. A timer queue is initiated for the PAUSE operand value before the link scheduler searches the service order table (SOT) for work to be performed. If the link scheduler completes a pass through the service order table in less than PAUSE value time, the LKB is placed on the timer queue. Service is suspended on this link until the time value expires or outbound PIU is enqueued for this link.

A negative response to an 'activate link' command normally indicates a modem problem, as only the 'enable' is processed on the link interface.

An activate line command to a BSC/SS line is followed by a set mode command for each device on the line. If the line is not multipoint, a buffer is allocated for input to the DVB work QCB. The BSC/SS BTU commands of invite, contact, read, write, and discontact are valid.

7. Set control vector - NCP subarea -- SSCP to physical services CPM-OUT (local/local or local/remote link only)

On a local/local or local/remote link the activated link can be either the non-backup link or backup link for this path. The host access method sends this command for both backup and non-backup. This command initiates checking of the link status. If the link is the non-backup no additional processing is required.

On a backup link the following processing occurs:

- Checks to ensure the primary link is not active.
- Checks that the primary link and backup link are not the same.
- Checks that the primary/secondary bit is set in both the current and backup SCB.
- Checks that the backup SCB is not in the 'contacted' state.
- Copies from the non-backup SCB to the backup SCB the SDLC address, network address, maximum outstanding limit (MAXOUT), pass limit (PASSLIM), and station type (PUTYPE=(4,LOCAL) or PUTYPE=(4,REMOTE)).

Once the link is active and the 'set control vector - NCP subarea' provides a positive response, the next command on a local/local or local/remote link is the 'contact' command.

8. * Connect out (dial) or Connect in (answer) -- SSCP to physical services CPM-OUT (switched line only)

On a switched link the line must be enabled for answering incoming calls or provided with a telephone number for outgoing calls. Either command is logical following an 'activate link' command. The 'connect out' or 'connect in' is addressed to physical services with the link identified in the request unit (RU) field of RU1NA. The 'connect out' or 'connect in' request is acknowledged by physical services CPM-IN and a connection is then attempted.

The 'connect in' enables the link for an incoming call. The 'connect out' consists of using the autocall unit to dial the telephone number provided in the connect out command request unit (RU), starting at RU byte 9. When the connection is established, an SDLC command of 'exchange ID' (XID) is transmitted to the physical unit with an address of FF (general poll). The XID response provides the terminal ID, and the LKB task at LKWTSKEP (run terminator task) is triggered. The task determines that a connection has been made, triggers physical services CPM-IN to send a request contact command to SSCP, and restarts the link scheduler run initiator.

The terminal ID is a 48-bit value with the following fields:

```
0       Reserved
x       Physical unit type (1 or 2)
00      Reserved
xxx     ID block, hardware by device type
          (example:  3790 is 006)
xxxxx   ID number, hardware or control
          program specified
```

The 'connect out' or 'connect in' status is indicated by bit settings in the LKB field of LKBSWST.

If a failure occurs during a callin or dial callout, the NCP physical services creates an 'inoperative' command with the failing link network address identified in the request unit (RU) field of RU1NA. An explanation of the 'inoperative' command is covered later in this appendix.

9. * Request contact -- physical services to SSCP (switched link only)

'Request contact' informs the SSCP that a physical connection has been established as a result of a connect out or connect in command. The network address of the link is carried in the request unit (RU) at RU1NA, and the terminal ID received by the SDLC XID response is sent in the RU in bytes 5-10. No response is requested by physical services.

10. * Set control vector PU -- SSCP to physical services (switched link only)

The original definition of the physical unit on this switched link was given to provide an unformatted control block for any switched physical unit calling in or being called. The 'set control vector PU' (RU byte 5 with a value of 3) provides the values for the CUB control block. The data provided to initialize the control block starting at RU byte 7 is as follows (see Appendix A of PLM):

```
byte 7       Physical unit type
byte 8       Reserved
byte 9       MAXOUT value
byte 10      PASSLIM value
byte 11      Immediate or deferred
               error recovery
byte 12-13   Reserved
byte 14-15   MAXDATA value
```

Physical services initializes the CUB control block and sends a response to INN path control for routing.

11. Contact -- from SSCP to NCP physical services

The 'contact' command is addressed to NCP physical services PSB CPM-OUT. The network address to be contacted is provided in the request unit (RU) at RU1NA. The common unit physical block (CUB) or station control block (SCB) is located and the 'set normal response mode' bit at CUBSSCP is set to 1. The 'contact' command obtains ownership of the CUB or SCB for the OAF. The PSB CPM-IN passes the response to INN path control for routing. The 'contact' response acknowledges the 'contact' PIU, but not that the device was contacted.

The link scheduler, which was started for this link by the 'activate link' command, placed the LKB on a timer queue. When the timer interrupt occurs, the link scheduler searches the CUBs or SCBs on that link for work. Each CUB and SCB is initially defined as being in the disconnect mode (X'1F') and the poll skip flag (X'1E') is 'on'. Normal servicing is still indicated as being disconnected; however, the link scheduler looks for command processing after the normal servicing sequence.

When the 'set normal response mode' bit is found, the link scheduler sends an SDLC command of SNRM on the link to the device defined by the CUB or SCB. If a timeout occurs before a response, the 'set normal response' bit is left 'on'. With an SNRM bit 'on', an attempt is made to contact one of the CUBs or SCBs on this link each time this link is serviced for 'contact poll' commands. If a response is received to the SDLC SNRM, the LKB task at LKWTSKEP of the LKB is triggered. This task is the run terminator task set up by the 'activate link' command. The 'run terminator' triggers the PSB CPM-IN to send in the 'contacted' command from NCP PSB to the CUB or SCB owner. Also, the link scheduler is restarted (as if a new 'activate link' were issued), and again the task address of the run terminator is in the LKB task address.

NOTE: If the physical unit contacted is type 4 (SCB), the bring-up sequence depends upon the response to the SNRM SDLC command.

Remote

If 'request initialization mode' (RIM) was received, the 'load initial', 'load data' (repeated), 'load final' take place. The 'contact' command is retried until an SNRM response of 'unnumbered acknowledgement' (UA) is received. When an UA is received, initialization of the remote begins with the first item of this list ('activate physical' to the remote NCP physical services).

Local secondary

If 'diconnect mode' (DM) was received, the secondary has been activated with an 'activate link' but has not received a 'contact' command. When the secondary receives and processes the 'contact' command a 'unnumbered acknowledgement' (UA) is sent.

12. Contacted -- NCP physical services to SSCP

The 'contacted' command was initiated by an 'unnumbered acknowledgement' (UA) SDLC reply from a physical unit as a response to the 'set normal response mode' (SNRM). This command provides the network address of the physical device contacted in the request unit (RU) at RU1NA.

This information sent to the owner of the CUB or SCB allows the physical unit to be sent the next command (an 'activate physical') to establish the next level of session.

An SCB would receive an 'activate physical' covered in item 1. A CUB would receive an 'activate physical' covered in item 12.

13. Activate physical -- SSCP to CUB physical unit process queue

The 'activate physical' command is enqueued to the CUB physical unit process queue. This command is the first command not addressed to NCP physical services and is the first which may be sent on a link to a physical unit. If the device is a type 2 physical unit, the command is transmitted to the physical unit. If the device is a type 1 physical unit, the command is processed by the NCP and is not sent to the physical unit.

The processing of the command results in setting the 'processing session initiating request' bit of CUBSTAT in the CUB control block to 1. The command format is modified from FID1 to FID2 and placed on the CUB link outbound queue (CUBLOBH); this is the queue searched by the link scheduler (started by an 'activate link' command) for data to be transmitted to the physical unit.

When the command has been processed by the physical unit and the response received by the link scheduler polling the physical unit, the response is enqueued on the CUB link inbound queue, triggering the CUB link inbound task. The task converts the PIU from FID2 to FID1 and checks for the type of response. If a positive response is received, the 'processing session initiation request' bit is set to 0, and the 'session established' bit is set to 1 in CUBSTAT of the CUB. If a negative response is received, the 'processing session initiation request' bit is set to 0. The response is passed to INN path control for routing.

The response request/response unit (RU) contains the name of the control program generation name for type 2 physical units.

For the type 1 PU, the 'session established' bit is set to 1 by the physical unit processing queue task, and the response to the OAF is created by the NCP.

NOTE: This command completes the second level of session (SSCP/CUB).

14. * Request Network Address Assignment - SSCP to physical services (switched link only)

The logical units are not assigned to any switched link or physical unit at generation time. The common physical unit control block (CUB) contains an address of a logical unit vector table (LUV) and the maximum number of entries in the LUV. The definition on the PU macro of MAXLU defines the maximum LUV entries.

The logical unit control blocks (LUBs) for a switched SDLC link are created by a LUDRPOOL macro. (The LUDRPOOL is used for switched LUs and dynamic reconfiguration LUs.) The request network address assignment (RNAA) command requests logical units for a switched connection.

The fields in the RNAA request unit are:

```
bytes 0-2    X'410210'
bytes 3-4    Network address of the
                 physical unit
byte  5      X'44' (LUDRPOOL)
byte  6      Number of LUs to be assigned
bytes 7-n    Request: halfword per LU with
                      LU local address
             Response: Halfword per LU with
                       LU network address
```

The number of addresses to be assigned may not exceed the entries in the LUV table (MAXLU operand of PU), and the addresses of LUs assigned are allocated from available entries in the LUDRPOOL.

Note: Host releases prior to ACF release 2 require the 'assign network address' command which allocates LUBs from the LUPOOL defined logical unit pool.

15. * Set control vector LU -- SSCP to physical services (switched link only)

The 'set control vector LU' command to NCP physical services provides the LU network address in the request unit (RU). A separate 'set control vector LU' command must be processed for each logical unit (LUB) to be used during a switched connection.

The command provides the following data:

```
byte 6 - LUB network address
byte 7 - n pacing count
byte 8 - m pacing count
byte 9 - Dispatching priority of
         APPL/LU CPM-OUT
```

The logical unit block (LUB) is now initialized with appropriate definitions and pointers which are generated for nonswitched LUBs. A response is passed to INN path control to be routed.

16. Activate logical -- SSCP to LU/SSCP process queue

The 'activate logical' command is enqueued on the LU/SSCP process queue of the logical unit control block (LUB). The LUB CPM-OUT task checks the command type, turns the 'processing activate logical' bit to 1 in LUBCPSET of the LUB, converts the command from FID1 to FID2 (or FID3), and places the command on the CUB link outbound queue for the link scheduler to find and transmit. Except for type 1 SDLC 3270, an 'activate logical' is processed by the link-attached physical unit. All commands for the type 1 SDLC 3270 are processed by the network control program.

The PIU response to polling the physical unit is enqueued to the CUB link inbound queue. The CUB link inbound task dequeues the FID2 (or FID3), converts it to a FID1, and branches to the CPM-IN task of LU/SSCP to process the input. A positive response requires the 'processing activate logical' bit to be set to 0 and the 'session established' bit to be set to 1 in LUBCPSET of the LUB. The response is then passed to INN path control for routing.

NOTE: This command completes the third level of sessions (SSCP/LU). No additional session is started until an application program is connected to be logical unit.

17. Initiate self (formatted) or unsolicited data (unformatted) -from LU to SSCP (Logical unit initiated logon only)

An inbound PIU in the SSCP/LU session may be a formatted 'initiate self' command or unformatted text to be resolved by the SSCP. The inbound PIU is received from the polled physical unit and placed on the CUB link inbound queue. The CUB link inbound task dequeues the PIU, converts the FID2 or

FID3 to FID1, and determines whether the PIU is from a defined LU which has a LU/SSCP session. If an SSCP/LU session does not exists a negative response is sent to the OAF. If an SSCP/LU session exists the PIU is passed to INN path control to be routed. The host receives the PIU and processes the request.

The host converts an unformatted PIU into an 'initiate self' format. The request unit (RU) contains (1) the name of the application (VTAM APPL statement label, TCAM message handler label) to which logical unit wants to be connected, or (2) text used as an entry to the interpret table.

The 'initiate self' is required only if the connection is initiated from the network logical unit. A host application initiates the connection with a 'bind' command.

18. Bind command -- host application to LU

The 'bind' command is sent from the host application to the APPL/LU process queue of the logical unit block (LUB). The APPL/LU process queue task dequeues the request, sets to 1 the 'processing bind' bit of LUBAPSET of the LUB, converts the FID1 to FID2 or FID3, and places the PIU on the CUB link outbound queue for the link scheduler to transmit.

The response to 'bind' command is received and queued on the CUB link inbound queue. The CUB link inbound task dequeues the FID2 or FID3, converts it to a FID1, and branches to the APPL/LU CPM-IN for processing. If the response is positive, the 'processing bind' is set to 0 and 'session established' bit is set to 1 in the LUBAPSET field of the LUB. The response is passed to INN path control for routing.

NOTE: This command completes the fourth and last level of sessions (application/LU).

19. Start data traffic -- from host application to LU

The 'start data traffic' command is optional. The option is selected in the 'bind' command profiles. If 'start data traffic' is not required, data and subsequent commands immediately follow the 'bind' command.

The 'start data traffic' and all subsequent commands and data transfers are placed on the LUB APPL/LU process queue for converting from FID1 to FID2 or FID3 and placed on the CUB link outbound queue for transmission to the SDLC terminal. If the device is a type 1 physical unit, the sequence number processing is performed by NCP. If the PIU is text, the PIU is checked for APPL/BNN and BNN/LU pacing. A response is checked for LU/APPL pacing response. Data traffic is also segmented as required by the MAXDATA operand of the PU.

All text and data from the logical unit are received and placed on the CUB link inbound queue, converted to FID1, processed to identify which logical unit (or the CUB) the FID1 is from to locate the LUB control block, and processed by type.

This command completes the initialization of the session. The last level of session could be ended by a 'terminate self' from the network logical unit followed by an 'unbind' or an 'unbind' initiated by the host application. A new 'bind' from a different or the same host application could initiate a new fourth level session without ending other levels. The switched support requires a full sequence of 'unbind', 'deactivate logical', 'free network addresses' (free LUBs to LUPOOL and clear LUV pointers), 'deactivate physical', 'discontact' (which sends SDLC 'request disconnect' (RD) for a secondary SCB and an SDLC 'disconnect' (DISC) for primary SCB and CUB), and 'abandon connection'; and then a new 'connect out' or 'connect in' command may be issued for that switched interface.

A PIU trace provides the above sequence, and a formatted control block dump of NCP provides the bit settings to identify the levels of commands in process or completed.

20. Inoperative -- from NCP physical services to SSCP

The 'inoperative' command may be required at any point in the command sequences after the 'activate link' command. After the 'activate link' command, a 'connect out' or 'connect in' is issued by the owner of a switched link. A 'contact' command obtains ownership of a nonswitched SCB or CUB for the OAF of the command. An immediate response is returned for a 'contact' command. An actual physical communication (SNRM) is indicated by a 'contacted' command sent to the owner of the CUB or SCB.

The method of indicating an abnormal end or break in the processing on a link is for NCP physical services to send an 'inoperative' command to all owners of the link. The 'inoperative' command identifies the network address in the request unit (RU) field of RU1NA of the link or resource. If the current command is to the link, the link address is carried in the request unit (RU) field of RU1NA and byte 5 of the RU contains a value of X'02'. If the current command is to a resource on the link, the resource network address is in RU1NA and byte 5 of the RU contains a X'01'.

No response is requested from owner; however, the owner is expected to provide a sequence of commands to terminate, retry, or alternate path alternatives to the failing resource.

Abbreviations

ACB	Adapter control block
ACU	Auto call unit
ANS	Automatic network shutdown
APPL	Application
ATB	Address trace block
BCU	Block control unit
BH	Buffer prefix
BHD	Block handler driver table
BHR	Block handler routine extension to DVB
BHS	Block handler set
BLU	Basic link unit
BSC	Binary synchronous
BST	Block handler set table
BTU	Basic transmission unit
BNN	Boundary network node
BUE	Switched backup extension to DVB
CAB	Channel control block
CCB	Character control block
CCP	Communications control program
CSP	Character service program
CCW	Channel control word
CDRM	Cross-domain resource manager
CDS	Configuration data set
CE	Channel end
CGP	Channel control block for EP, PEP
CHCB	Channel control block for EP, PEP
CHVT	Channel vector table for EP, PEP
CIC	Channel input chain
CIE	Call-in extension to DVB
COC	Channel output chain
COE	Call-out extension to DVB
CPM	Connection point manager
CPT	Channel parameter table
CRP	Check record pool
CSP	Character service program
CTB	Communications line timer and RAS control table
CUB	Common unit physical block
CUCR	Cycle utilization counter register
CW	Channel word
DAE	Device addressing extension to DVB
DAF	Destination address field
DDB	Dummy data buffer - performance measurement facility
DIA	Device input area extension to DVB
DLC	Data link control
DM	Disconnect mode (was ROL)
DRS	Display/refresh/select table
DVB	Device base control block
ECB	Event control block

EP	Emulation program
FID	Format identification
FIFO	First in/first out
FM	Function management
FVT	Function vector table
GCB	Group control block
HWE	Extended halfword direct addressables
HWX	Halfword extended extension
IAR	Instruction address register
ICP	Interface control program
ICW	Interface control word
IDE	Identification list entry
IDL	Identification list header
INN	Intermediate network node
IOB	Input/output block
IPL	Initial program load
LCB	Line control block
LCP	Line control program
LCST	Line control selection table
LGT	Line group table
LKB	Link control block
LNVT	Line vector table
LOBQ	Link outbound queue
LOSQ	Link outstanding queue
LSA	Lost subarea
LTCB	Line trace control block
LTCT	Line type command table
LTS	Line test control block
LTVT	Line trace vector table
LU	Logical unit
LUB	Logical unit control block
LUV	Logical unit vector table
LXB	Link XIO block
NAU	Network addressable unit
NCP	Network control program
NLB	Programmed resource logical unit block
NLX	Programmed resource logical unit block extension
NPB	Programmed resource physical unit block
NSA	Nonsequenced acknowledgement (see UA)
OAF	Origin address field
OLLT	Online line test
OLLTCB	Online line test control block
OLLTLAB	Online line test lookahead buffer
OLLTQCB	Online line test QCB control block
OLTT	Online terminal test
OLTTCB	Online terminal test control block
PC	Path control
PCB	Panel control block
PCI	Program-controlled interrupt
PEP	Partitioned emulation program
PIU	Path information unit

PMF	Peformance measurement facility
PSB	Physical services block
PUV	Physical unit vector table
QCB	Queue control block
RD	Request disconnect (was RQD)
RECMS	Record maintenance statistics
RH	Request/response header
RIM	Request initialization mode (was RQI)
ROL	Request online (see DM)
RQD	Request disconnect (see RD)
RQI	Request initialization (see RIM)
RR	Receive ready
RS	Read start (RS0, RS1)
RU	Request/response unit
RVT	Resource vector table
SC	System control
SCB	Station control block
SCRT	System control router table
SDLC	Synchronous data link control
SGE	Switched line group entry
SGT	Switched line group table
SID	Send identification
SIT	Subarea index table
SM	Status modifier
SNP	SSCP-NCP session control block
SNRM	Set normal response mode
SOT	Service order table
SPB	SDLC/BSC path control block
SS	Start-stop
SSCP	System services control point
STE	Selection table entry
SVT	Subarea vector table
SVC	Supervisor call
SVT	Subarea vector table
TDT	Translate/decode table
TH	Transmission header
TND	Time and date control block
UA	Unnumbered acknowledgement
UACB	User adapter control block
ULVT	User line vector table
UC	Unit check
UE	Unit exception
VLB	Programmed resource virtual link block
VTS	Vector table of SNPs
WS	Write start (WS0, WS1)
WU	Western union translate table
XDA	Word direct addressables
XDB	Byte direct addressables
XDH	Halfword direct addressables

Index

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

SR20-4621-1

Advanced Communications Function NCP Rel. 2 Data Flow Printed in U.S.A. SR20-4621-1