Comm/Exec Internals: DECnet Data Structures Manual

July 1983

The Comm/Exec Internals: DECnet Data Structures manual provides reference information on the permanent and volatile data stuctures used by the RSX DECnet product. These data structures include CETAB macros and various control blocks that are set up when DECnet is loaded. Also included in this manual are two appendices dealing with the data link level protocol modules used by DECnet: the DDCMP Process (DCP) and the Ethernet Protocol Manager (EPM).

UPDATE INFORMATION: This is a new manual.

OPERATING SYSTEM: RSX-11M DECNET V4.0
RSX-11S V3.0
RSX-11M-PLUS V2.0

SOFTWARE VERSION: COMM/EXEC VERSION 3.0

VERSION HISTORY

Version numbers without specific dates are target versions.

02-Aug-83 Version 1.00C	First Draft Revision C Manual divided into a manual set Cross references changed
xx-aug-83 Version 2.00	Final Draft - Completed Manual Set

First Printing, Sept 1983

All comments on this manual should be sent to the Distributed Systems Group in Newksbury, Mass using the form found on the last page of this document. Internal sites may also use ENET mail (Mailstop: ELROND::COMMEX.ECDOC).

TABLE OF CONTENTS

DECNET DATA STRUCTURES MANUAL
Table of Contents

PREFACE

MANUAL OBJECTIVES

The goal of this manual is to provide the information necessary for you to understand the internal data structures used by the RSX DECnet product.

MANUAL AUDIENCE

This manual assumes that you are systems level programmer interested in the internals of the RSX DECnet product. You are assumed to be familiar with the PDP-11 family of processors, and be an advanced MACRO-11 programmer who has a firm knowledge of Communications Executive environment.

MANUAL STRUCTURE

This manual is organized in a reference style.

Chapter 1 contains a description of the CETAB macros used by DECnet.

Chapter 2 contains a description of the data structures used by RSX DECnet.

Chapter 3 contains a description of the CEDump switches used to dump the DECnet data structures of a running DECnet system.

Appendix A covers the data structures used by the Ethernet Protocol Manager (EPM) process.

Appendix B covers the data structures that are used in the DDCMP process (DCP).

ASSOCIATED DOCUMENTS

This manual is part of the Comm/Exec Internals manual set describing the internals of the RSX Communications Executive. Although this manual is intended to be a free standing document, references are made to the other manuals in the Comm/Exec Internals manual set.

Familiarity with the Communications Executive from the point of view of a system user can be obtained from the RSX-11 DECnet manual set (Order No. xxxxx). Network Management topics (especially counter and event logging issues) are covered in the DNA Network Management Specification (Order No. xxxxxx). Source listings for the RSX DECnet products are not included but could improve your understanding of the

RSX DECnet internals. These listings must be obtained directly from the Distributed Systems Group at Tewksbury, Massachusetts.

CHAPTER 1

DECNET CETAB MACROS

There are several CETAB macros that are used to define the data structures needed to support DECnet. They are not present in the configuration file if DECnet is not requested at network generation time.

The macros described in the following sections are only for the DECnet product. For other CETAB macro descriptions see the Guide to Writing a CommExec Process Manual and the ESI Data Structures Manual.

The macro descriptions presented in this chapter are presented in the same order that they are likely to be encountered in the CETAB source file.

****
* ROU$DF *
****

1.1 ROU$DF - Define Routing Task Parameters

Format: ROU$DF nn,maxc,maxh,routm,maxa,rtadj,enadj,amxc,amxh,rttim,rtpri

where:

- **nn** The maximum node address allowed in the network (1.-255.)
- **maxc** The maximum cost allowed per line (1.-1022.)
- **maxh** The maximum hops allowed in the network (1.-32.)
- **routm** The routing timer interval in seconds (0-65535.)
- **maxa** Maximum number of areas in network (1.-255.) (must be zero if areas are not supported)
- **rtadj** Number of broadcast router adjacencies (1.-255.)
- **enadj** Number of broadcast endnode adjacencies (1-255.)
- **amxc** Area maximum cost (1.-1022.)
- **amxh** Area maximum hops (1.-32.)
- **rttim** Broadcast router timer (1.-255.)
- **rtpri** Broadcast router priority (1.-255.)

Description: This macro is used to define the routing task parameters.

Restrictions: None

Example:

```
ROU$DF    300.,1022.,15.,30.,0.,5.,255.,0.,0.,10.,0.
```

This example sets the maximum node address to 300, the maximum cost to 1022, and the maximum hops to 15. The routing timer is set to 30. The network defined does not support areas. The routing task supports up to 5 broadcast router adjacencies and 255 broadcast end-node adjacenecies. The broadcast router timer is 10.

```
**********
*        *
*  NOD$DF*
*        *
**********
```

1.2 NOD$DF - Define Local Node Parameters

Format: NOD$DF <nam>,<id>,larea,laddr[,harea,haddr]

where:

- **nam** The name of the local node (must be six characters in length - including blanks).
- **id** The local node ID (Maximum of 32 characters including blanks).
- **larea** The local node area number (1.-255.) (must be 1 in single area network)
- **laddr** The local node address (1.-255.)
- **harea** The host node area number (1.-255.) (must be 1 in single area network)
- **haddr** The host node address. If blank the local node address is used. (1.-255.)

Description: This macro is used to define the parameters that characterize the local node.

Restrictions: None

Example:

```
NOD$DF    <ELROND>,<DECNET-11M-PLUS-BL7>,1.,19.,1.,19.
```

This example sets the node name to ELROND and the node ID to DECNET-11M-PLUS-BL7. The node is in a single area network and has an address of 19. The host node address information is set the same as the local node.

****
*   *
* REM$DF *
*   *
****

1.3 REM$DF - Define Remote Node Parameters

Format: REM$DF <name>,area,address

where:

name The node name (1-6 alpha-numeric)

area The node area number (must be 1 in single area network) (1.-255.)

address The node address (1.-255.)

Description: This macro describes the affinity between a node name and a node number. Each defined remote is appended to the remote list.

NOTE

The node name is a 'local phenomenon' and in no way does the local node's perception of a remote node's name have to agree with what the remote node perceives his own name to be. It is however, recommended that node names agree for the sake of consistency across the network.

Restrictions: None

Example: REM$DF <BERGIL>,1.,6.

The example shown defines a remote node name BERGIL and maps that name to area 1, node 6.

```
****************
*              *
*    SER$DF    *
*              *
****************
```

1.4 SER$DF - Define Service Parameters

Format: SER$DF narea,naddr,harea,haddr,scir,sdev,[pass],[daddr],[dct][,hrdaddr]

where:

narea	The node area address (1.-255.)
naddr	The node address (1.-255.)
narea	The host area address (1.-255.)
haddr	The host address (1.-255.)
scir	The service circuit (System line number)
sdev	The service device (ASCII line name)
pass	The password (1-16 hexidecimal digits)
daddr	The dump address (1-16 hexideciaml digits)
dct	The dump count (1.-255.)
hrdaddr	The hardware address (1-16 hexidecimal digits)

Description: This macro define the parameters needed for down-line load and up-line dump of RSX-11S and Server Base software.

Restrictions: Must be followed by a FIL$DF macro.

Example:

```
SER$DF  1.,43.,1.,23.,3,UNA-0,0,,,AA00300003B421
```

This example declares node 23 in area 1 to be node 43's host node. The service circuit associated with node 43 is circuit 3 (UNA-0). The hardware address of node 43 is AA00300003B421.

****
*   *
* FIL$DF *
*   *
****

1.5 FIL$DF - Define File Name Parameters

Format: FIL$DF [secldr],[terldr],[ldfil],[diagfil][,dumpfil]

where:

secldr	The secondary loader file name
terldr	The tertiary loader file name
ldfil	The load file name
diagfil	The diagnostics file name
dumpfil	The dump file name

Description: This macro define the various file names needed for servicing the RSX-11S or Server Base software. All filenames are specified as ASCII strings and should conform to legal RSX-11 filename syntax (device:[uic]filename.type). The macro sets up the file names needed as a result of a SER$DF macro.

Restrictions: Must immediately follow a SER$DF macro. Although all fields are optional, at least one must be present.

Example:

```
FIL$DF    PLUTO2.SYS,PLUTO3.SYS,RSX11.SYS,,PLUTO.DMP
```

This example sets up PLUTO2.SYS as the secondary loader file name and PLUTO3.SYS as the tertiary loader file name. The desired load file is RSX11S.SYS. There is no diagnostics file specified. The dump file name is set to PLUTO.DMP.

CHAPTER 2

DECNET VOLATILE DATA STRUCTURES

2.1 INTRODUCTION

This chapter describes the DECnet-RSX Phase IV DEC volatile data base. Its primary purpose is to provide more detailed descriptions of the volatile data structures of the DECnet-RSX products. The first chapter in this manual contains the description of the permanent data base and the other manuals in the Comm/Exec Internals series describe other components of the volatile data base.

The volatile data base structures are initially allocated at network load time based on the permanent data base. These structures may then be modified by the running network software or by use of Network Management commands. Any changes, additions or deletions to this data only affect the running system and are not permanent (I.E., they are lost when the network is re-loaded).

These data structures reside in the Communications Executive (Comm/Exec), in system pool (DSR), in the various processes' space, in network pool, and in the user's task.

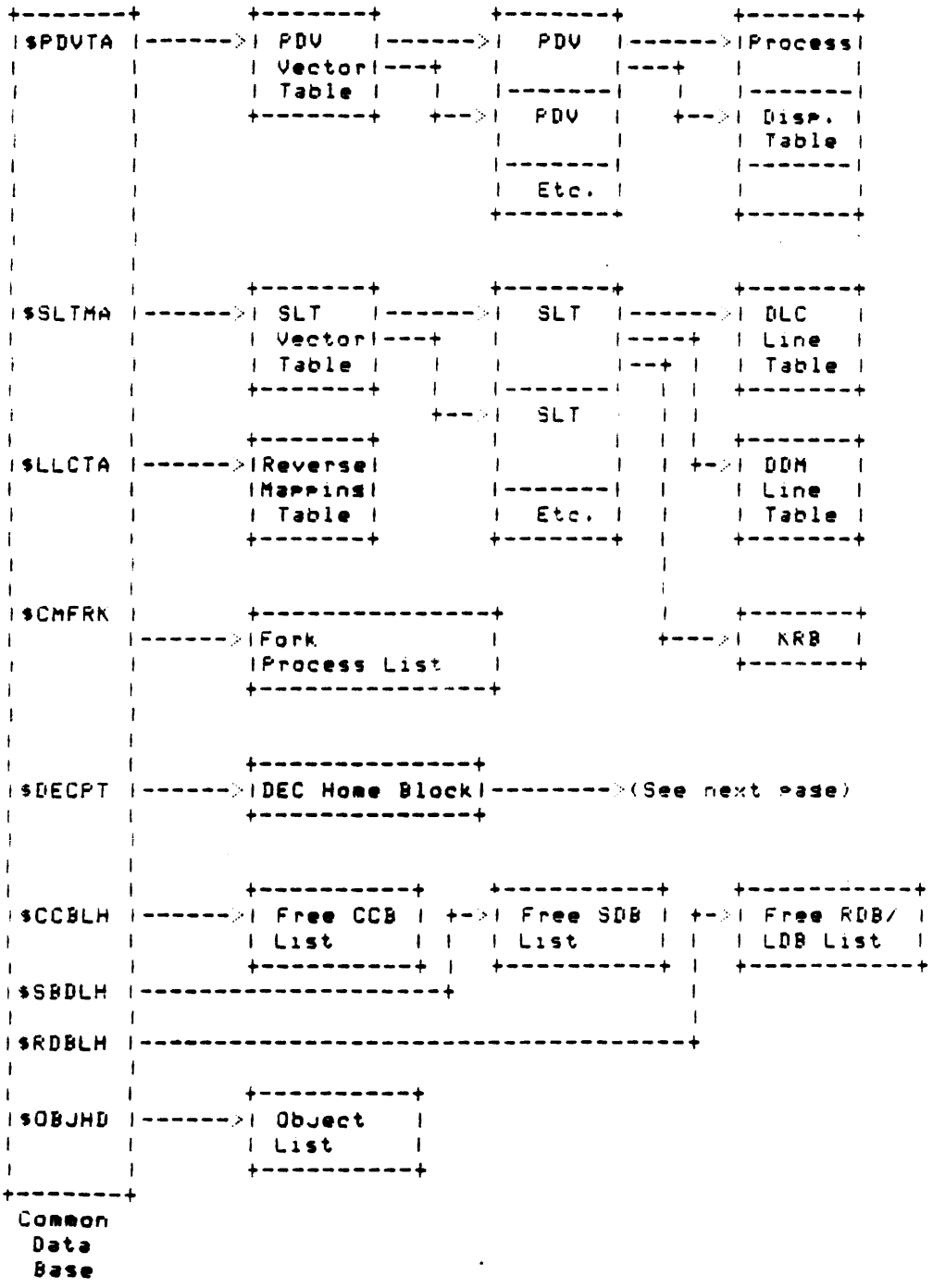
NOTE

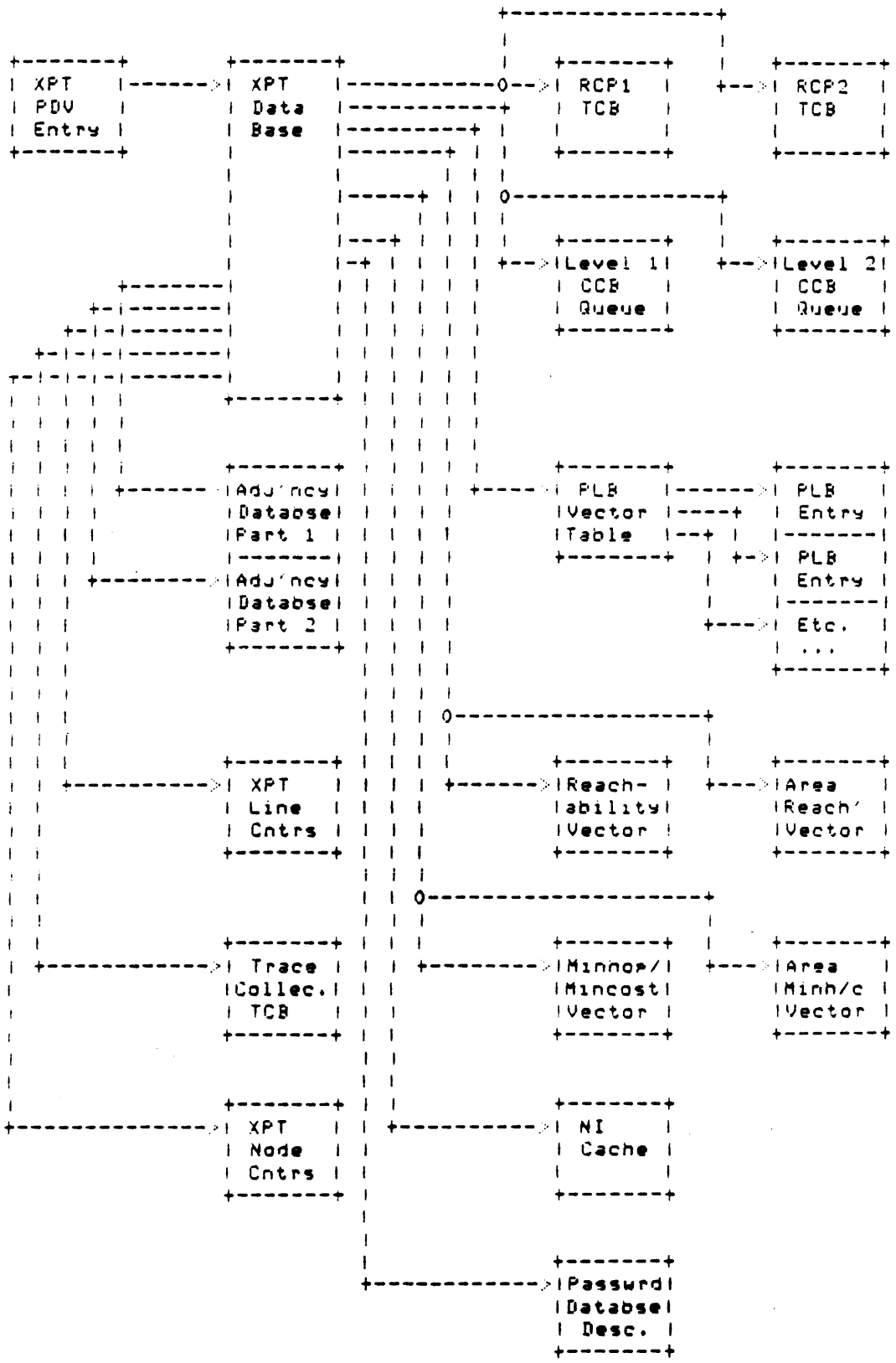
It cannot be assumed that a type of data structure will always be allocated from the same place. If there is insufficient memory for the allocation of a data structure in one pool, then another pool will be used to allocate that data structure. Therefore, the data structures in the rest of this document are put under the sections that they would appear if no allocation failures result. If a data structure is normally allocated from Comm/Exec pool, except when there is insufficient memory, then that data structure can be found in the Comm/Exec pool section of this document.

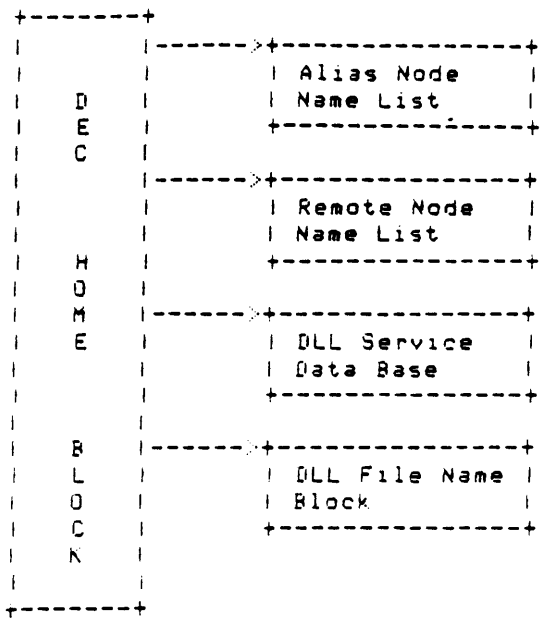
Also, several data structures are allocated from different places depending on the operating system type (M or M-Plus). In this document all data structures of this type will be categorized as DECnet-11M data structures.

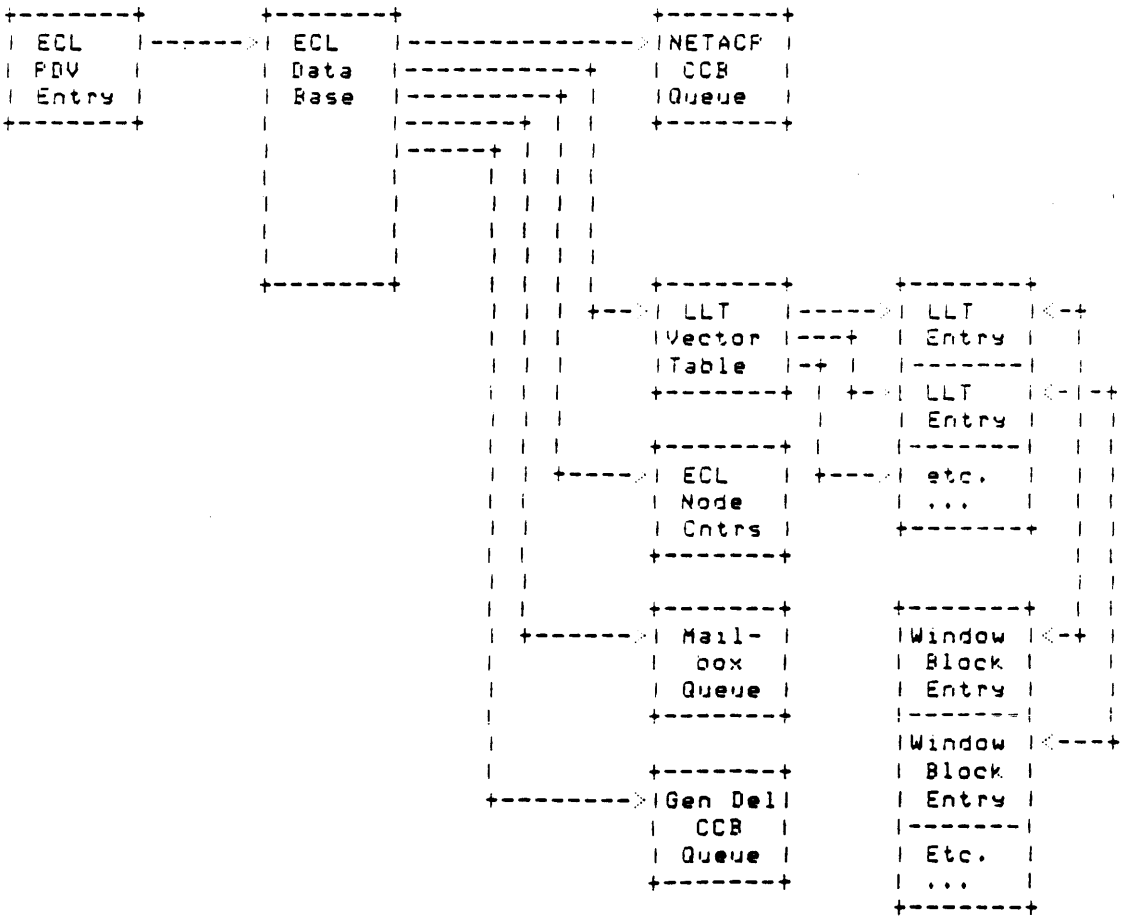
The relationship of the major volatile data base components is

illustrated in the following 4 figures. The specific structure of each component (excluding the Comm/Exec data structures) follows on subsequent pages.




XPT PDV Entry
XPT Data Base
RCP1 TCB
RCP2 TCB
Level 1 CCB Queue
Level 2 CCB Queue
Adj'ncy Database Part 1
Adj'ncy Database Part 2
PLB Vector Table
PLB Entry
PLB Entry
Etc. ...
XPT Line Cntrs
Reach-ability Vector
Area Reach' Vector
Trace Collec. TCB
Minhop/Mincost Vector
Area Minh/c Vector
XPT Node Cntrs
NI Cache
Password Database Desc.


DECNET HOME BLOCK
Alias Node Name List
Remote Node Name List
DLL Service Data Base
DLL File Name Block


ECL PDV Entry
ECL Data Base
NETACP CCB Queue
LLT Vector Table
LLT Entry
LLT Entry
ECL Node Cntrs
etc. ...
Mail-box Queue
Window Block Entry
Window Block Entry
Gen Dll CCB Queue
Etc. ...

2.2 DATA IN Comm/Exec / RSX POOL

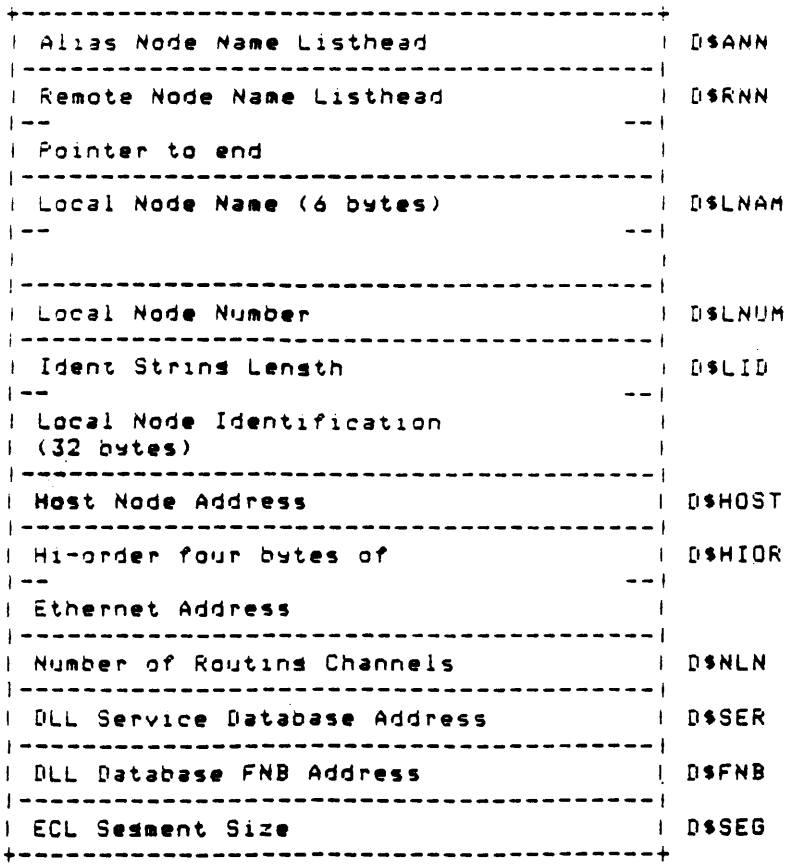
Within the Comm/Exec partition is the Comm/Exec pool. The size of the Comm/Exec pool is determined at Network Generation time. The following data structures are located in either the Comm/Exec pool or the system pool (DSR). Both are mapped in the Executive virtual address space.

NOTE

The Comm/Exec pool is not available for data structures in RSX-11M-PLUS systems with I/D space. The Comm/Exec resides in I space provided by the RSX executive.

2.2.1 DHBDF$ - DECnet Home Block Format

The DECnet Home Block contains DECnet specific information which is accessed by multiple DECnet processes and/or tasks. The DECnet Home Block is allocated from RSX system pool (DSR) and is pointed to via $DECPT in the Comm/Exec Common. The DECnet home block can be examined in a running system by using the CEDump /DH switch (See Section 3.1).

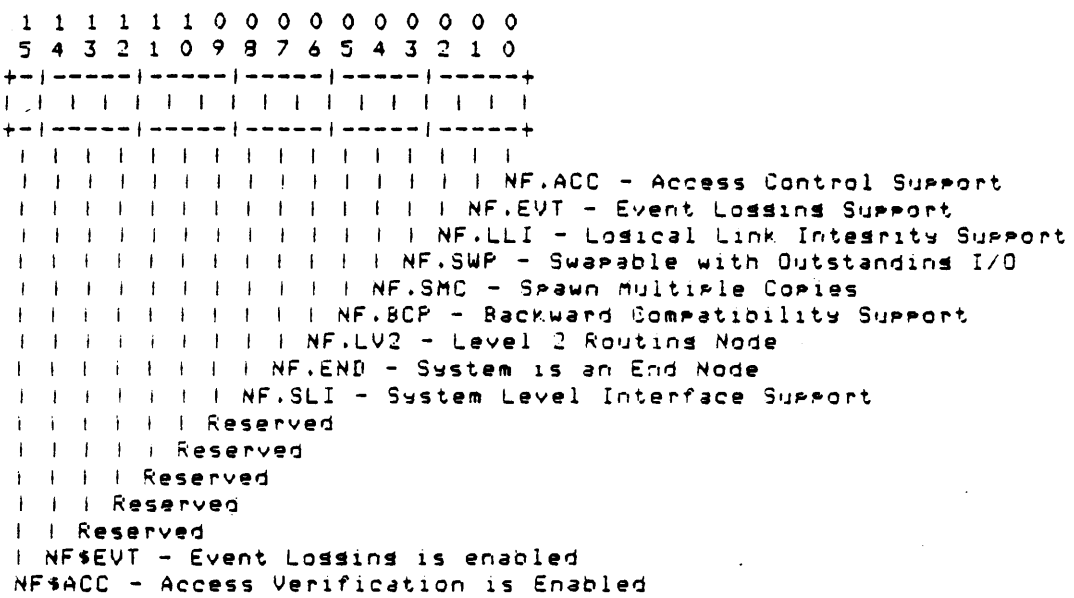


The following Home Block cells are present for routing nodes only:

Number of Broadcast Router Adjacencies	D$NBRA
Number of Broadcast Endnode Adjacen's	D$NBEA
Number of Nodes in Network	D$NN
Number of Areas in Network	D$NA
Maximum Cost	D$MAXC
Maximum Hops	D$MAXH
Maximum Visitation Count	D$MAXV
Area Maximum Cost	D$AMXC
Area Maximum Hops	D$AMXH
Square Root Limiting Factor	D$SQRL
Routing Timer (seconds)	D$RTMR
Broadcast Routing Timer (seconds)	D$BRTM
Broadcast Router Priority	D$BRPR

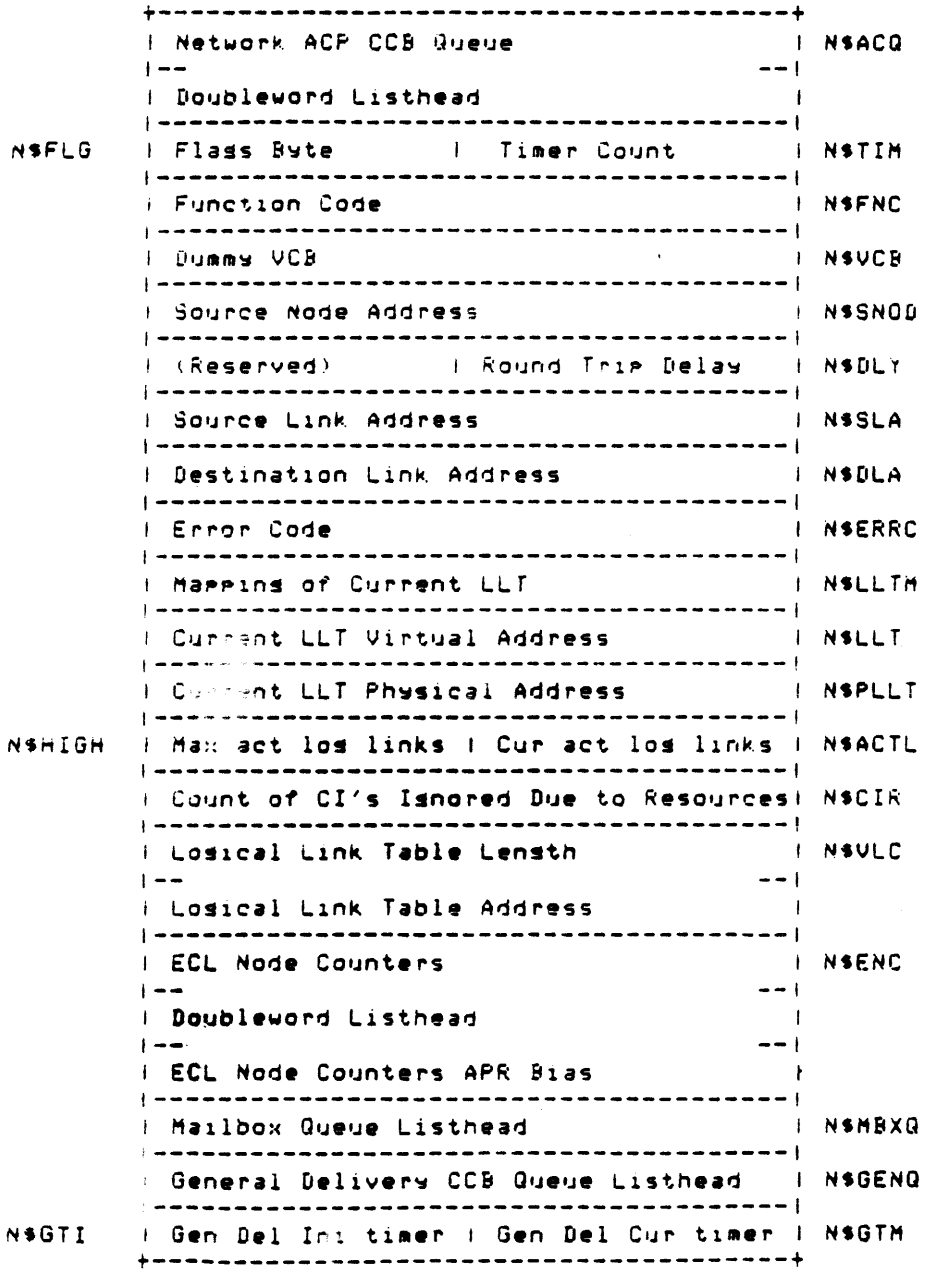
2.2.2 NSFDF$ - Features Word For DECnet

The following flag word is found in the UCB for NS:. The offset for this word is U.CW3.

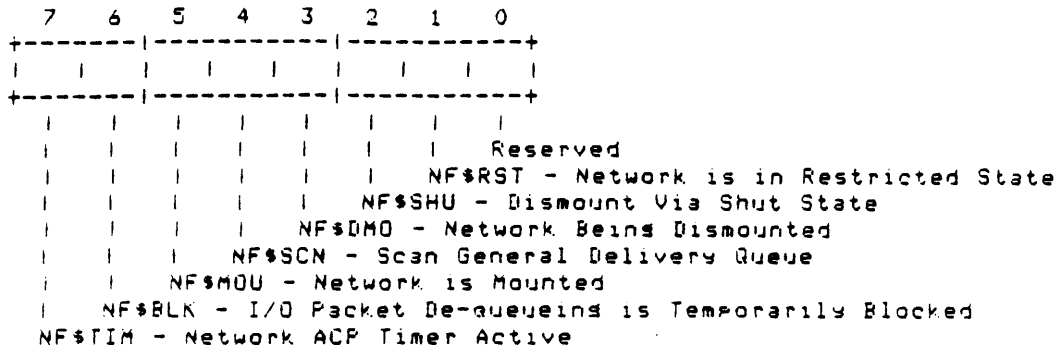


2.2.3 ECDDB$ - ECL Data Base <DSR>

The data shared between the ECL process and the NETACP task is contained in this structure. The pointer to the data is contained in the Process Descriptor LLC data base address cell (Z.DAT) for the ECL process.

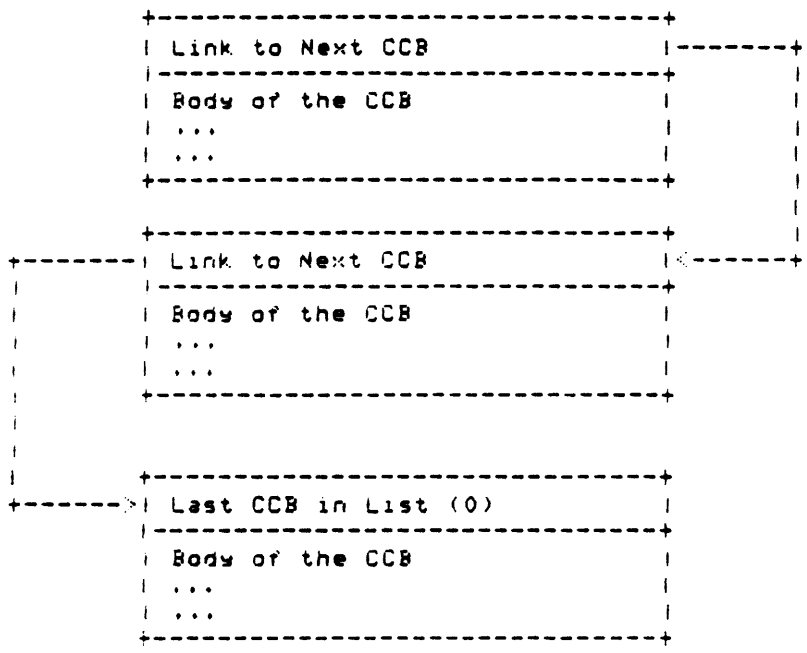


N$FLG Flags Byte Definitions



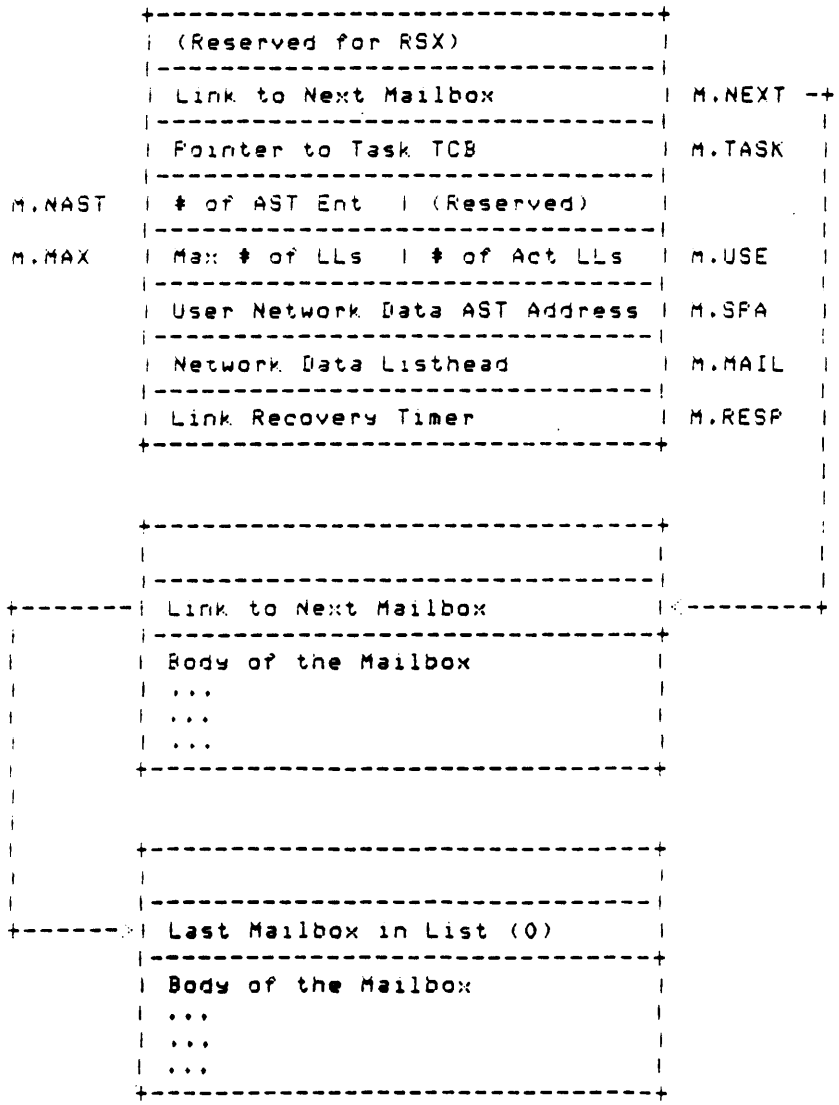
2.2.4 Network ACP CCB Queue <DSR>

CCBs to be processed by the Network ACP are placed on this queue and the ACP is UNSTOPed to initiate their processing. Both transmit complete and receive complete CCBs may be placed on this queue.



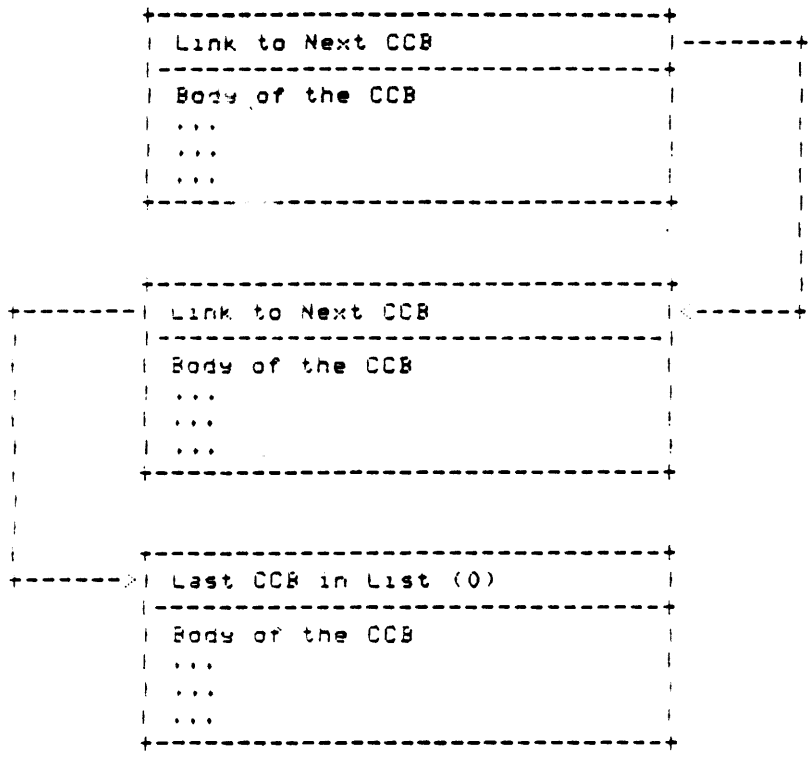
2.2.5 MBXDF$ - Mailbox Queue <Comm/Exec Pool>

A mailbox is created when a user task indicates that it is an active network user by issuing an OPN$ function. The mailbox holds common information about the user task for use by the network software. The CEDump /MB switch can be used to dump the mailbox queue of a running system (See Section 3.7).



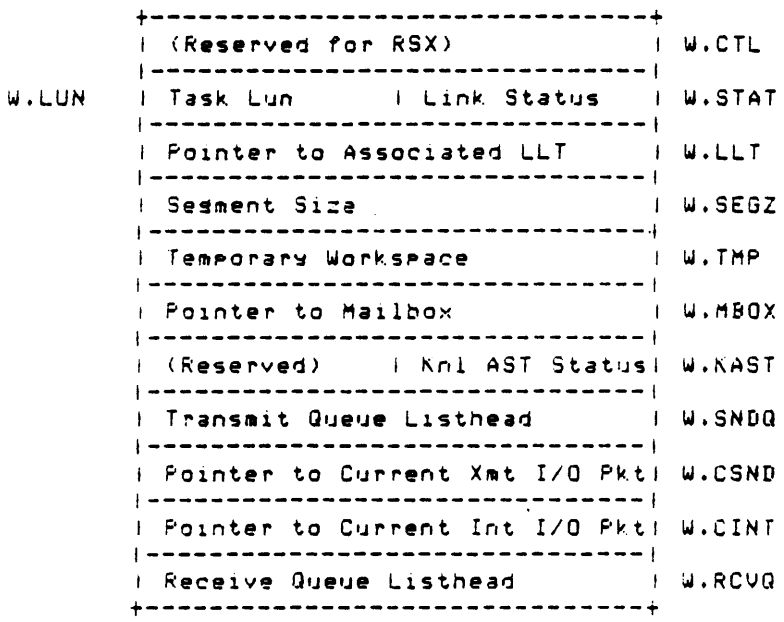
2.2.6 General Delivery CCB Queue

The general delivery queue contains incoming connect and verification requests for tasks which have not yet been requested or tasks which have not been able to issue an OPN$ function. The CEDUMP /GE switch can be used to dump the general delivery queue of a running system (See Section 3.2).

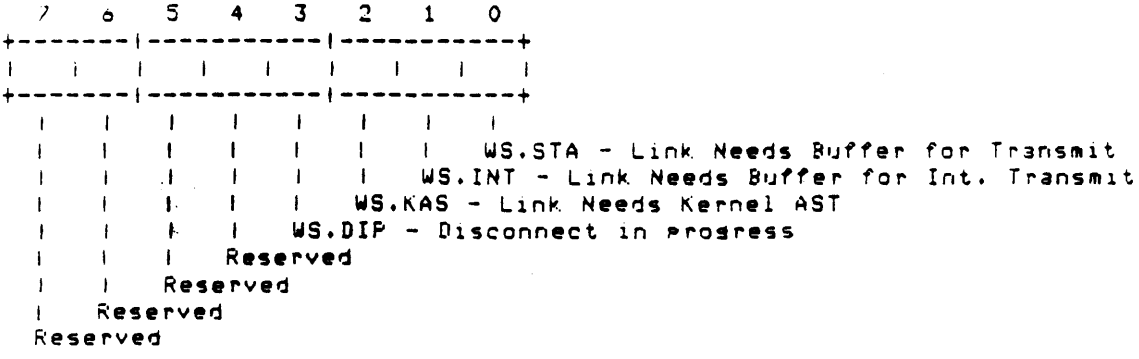


2.2.7 LLWDF$ - Window Block <Comm/Exec Pool>

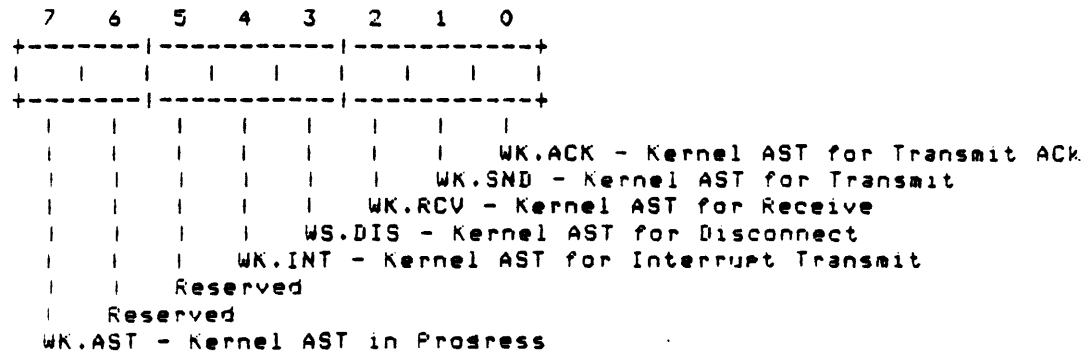
The window block holds the current state of a logical link with respect to the QIO's issued by the user task. It is pointed to by the second LUN word in the task header. The window blocks of a running system may be examined by using the CEDump /LL switch (See Section 3.5).



W.STAT Status Byte Definitions

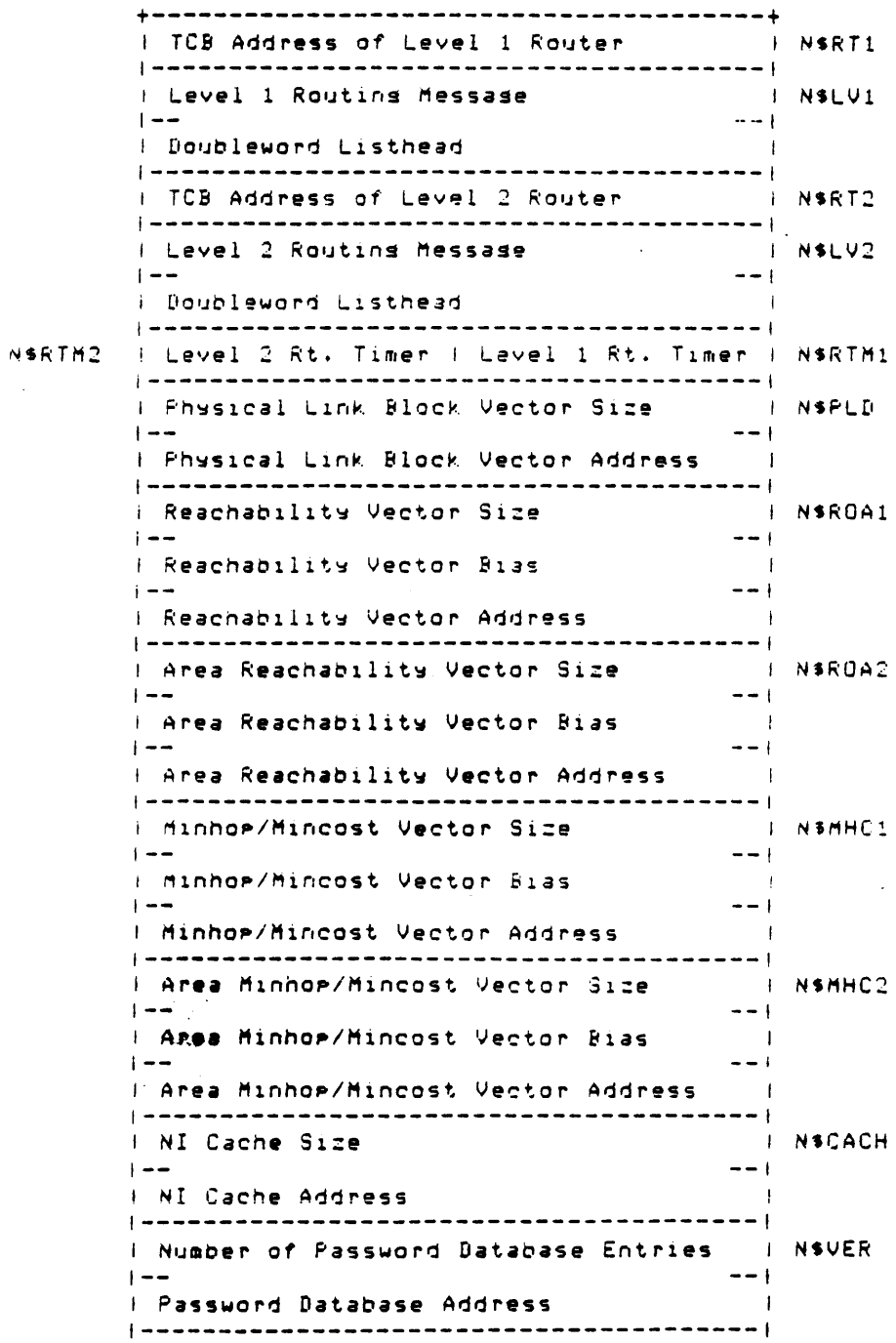


W.KAST Kernel AST Status Flags



2.2.8 XPDDB$ - XPT Database

The data shared between the XPT process and the Routing tasks (RCP1/RCP2) is contained in this structure. The pointer to the data is contained in the Process Descriptor LLC data base address cell (Z.DAT) for the XPT process.



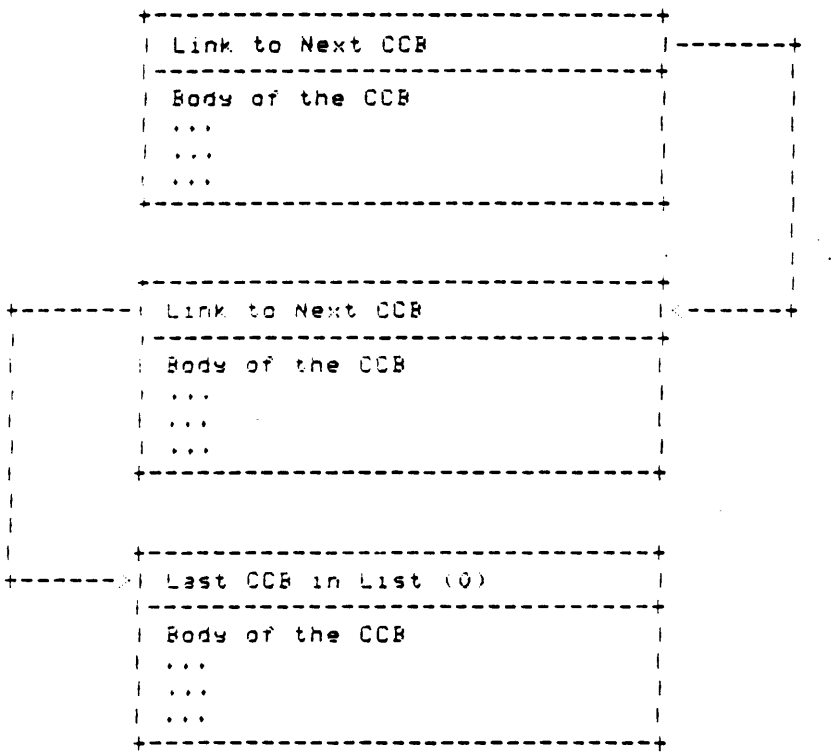
Bias of Adjacency Database Part 1	N$ADJ1
Bias of Adjacency Database Part 2	N$ADJ2
Broadcast Router Priority Table Addr.	N$PRI
Number of Transport Line Cntr Blocks	N$TLC
Transport Line Counter Bias	
Transport Line Counters Address	
Trace Control Word	N$TRC
Trace Collector TCB Address	N$TTCB
Trace Control Block Address	N$TCTL
Transport Node Counter Size	N$TNC
Transport Node Counters Address	

The following additional cell is present only on DLM nodes:

$TCRC Library Pointer	N$CRC

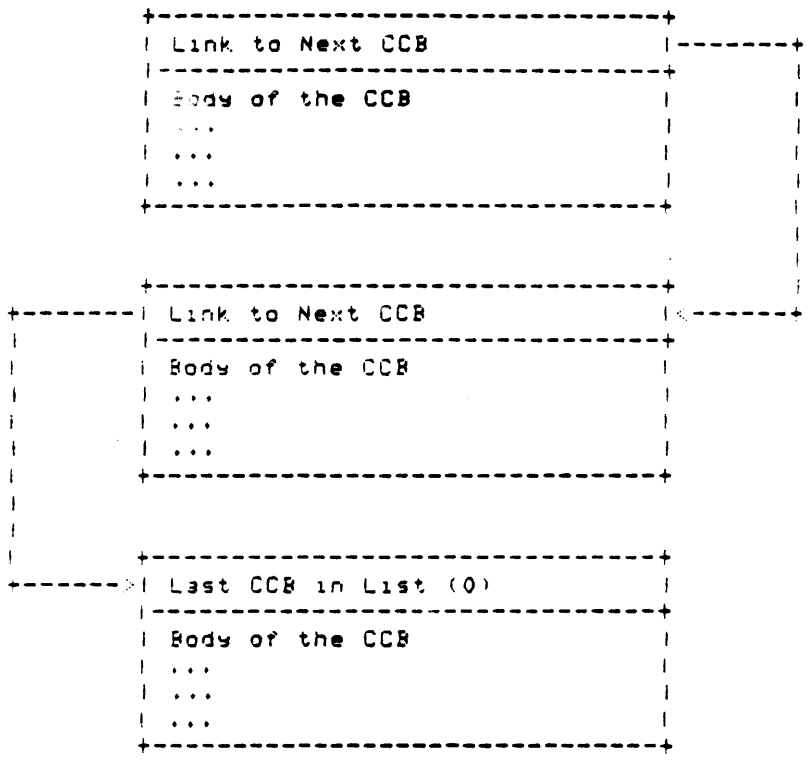
2.2.9 Level 1 Router CCB Queue <DSR>

This is linked list of CCBs and associated LDBs containing level 1 routing messages. Buffers are placed on this list by XPT and taken off by RCP1. This list will always be empty for non-routing (end) nodes.



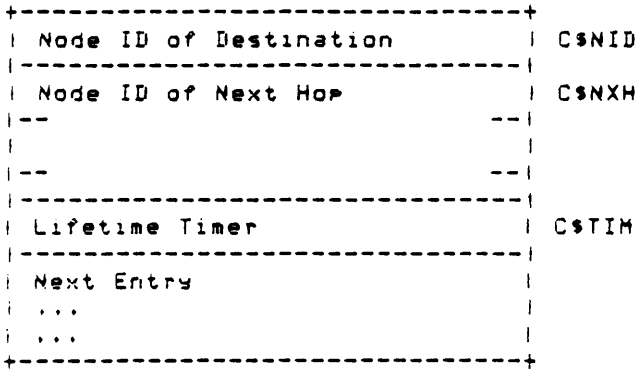
2.2.10 Level 2 Router CCB Queue <DSR>

This is a linked list of CCBs and associated LDBs containing level 2 routing messages. Buffers are placed on this list by XPT and taken off by RCP2. This list will always be empty for non-routing (end) nodes and non-level 2 routing nodes.



2.2.11 CACDF$ - NI Cache <DSR>

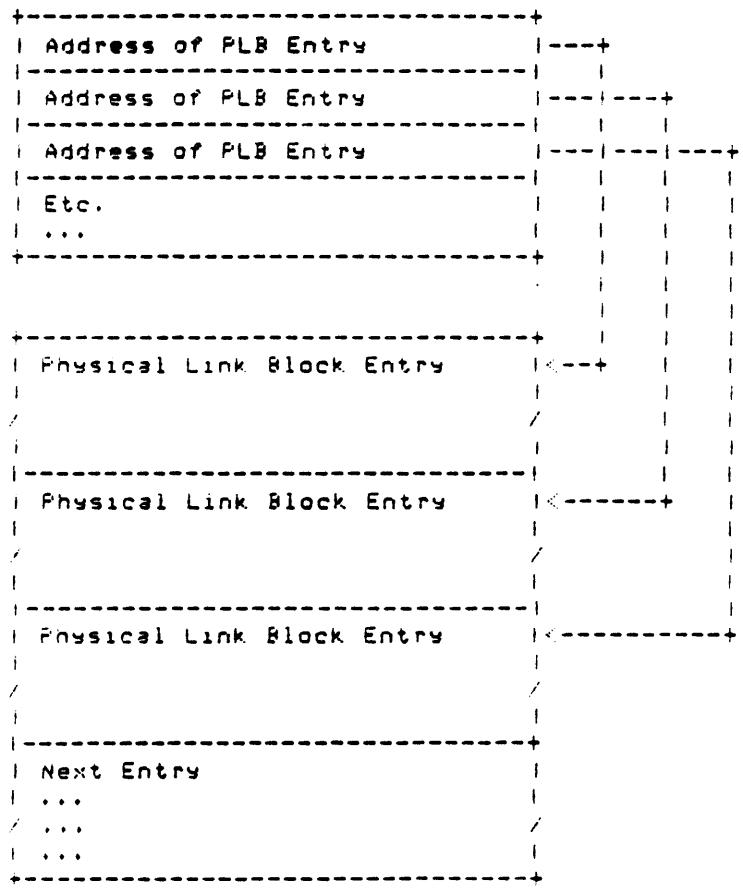
The NI Cache is only present for non-routing (end) Ethernet nodes. It contains information about the next hop to get to certain nodes. Entries are made in the cache when a logical link is set up to a node and removed when a time interval has elapsed without any communications with that node. The pointer to the NI Cache is contained in the XPT database.



Note: The Node ID of Destination (C$NID) is a 16 bit node address and the Node ID of Next Hop (C$NXH) is a 48 bit node address.

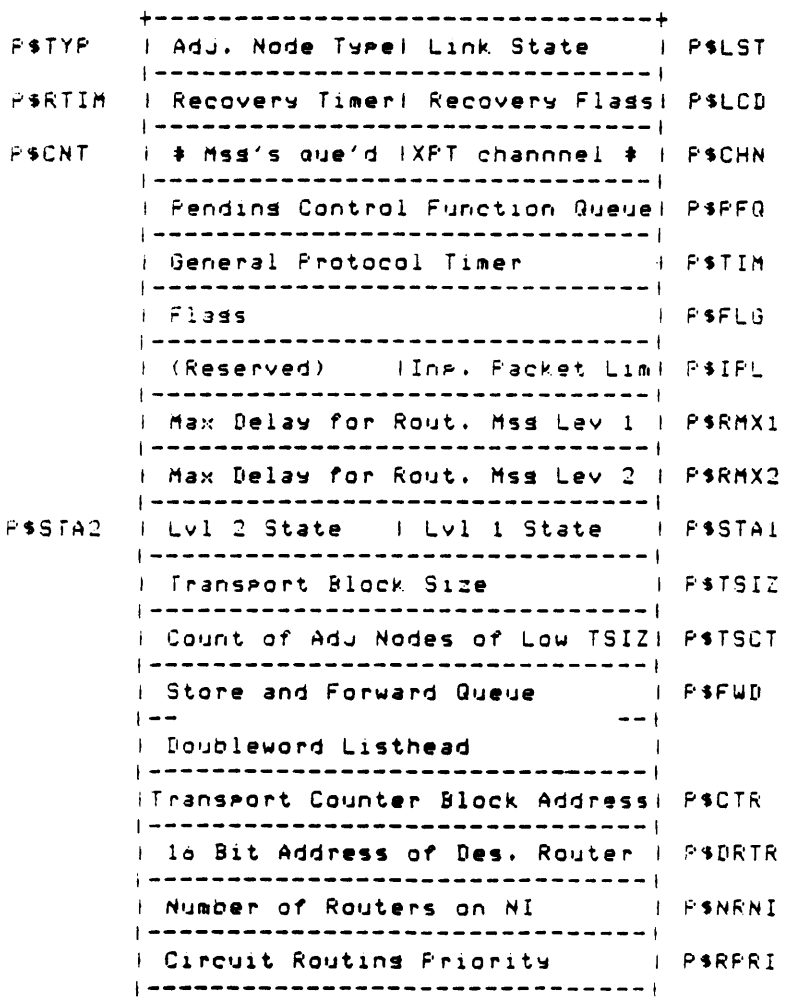
2.2.12 Physical Link Block Vector Table <DSR>

Each Physical Link Block entry is addressed by the appropriate offset into the PLB Vector Table. The pointer to the PLB Vector Table resides in the XPT database.

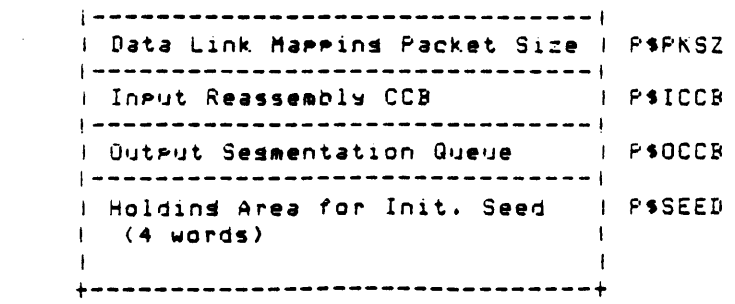


2.2.13 PLBDF$ - Physical Link Block (PLB) <DSR>

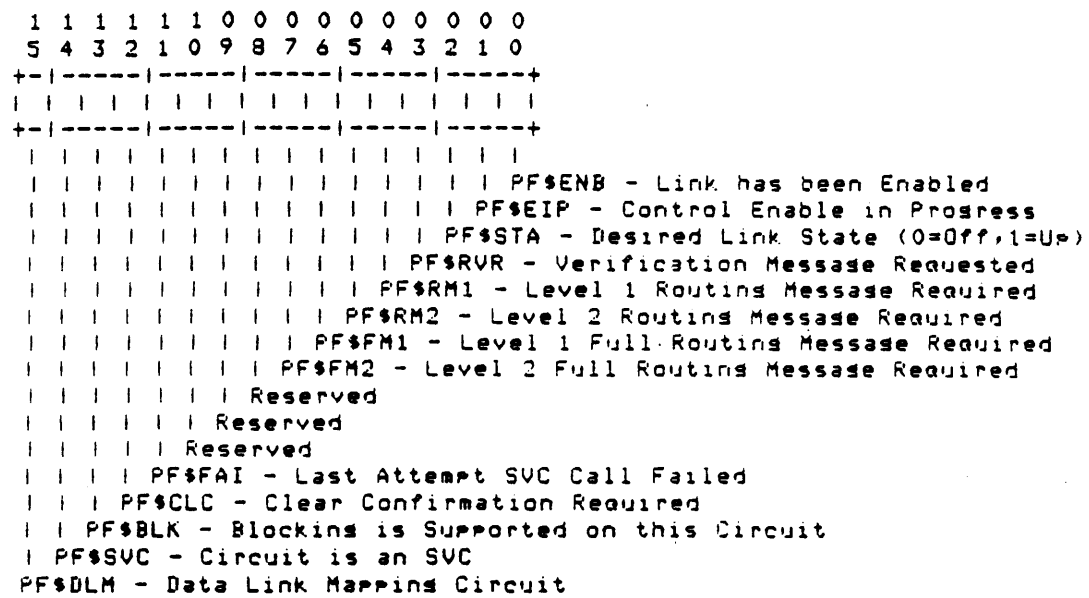
The Physical Link Block is used by XPT to maintain the current status and control/monitor each physical connection to an adjacent node. Pointers to each PLB entry reside in the Physical Link Block Vector Table.



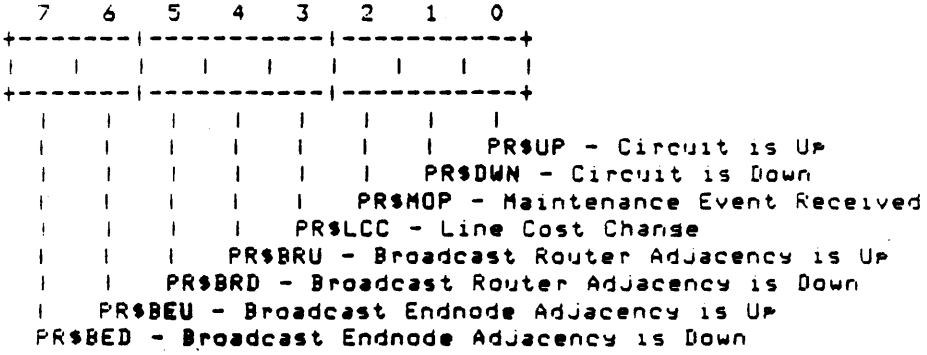
The following cells are only used for DLM circiuts:



P$FLG Physical Link Flag Definitions



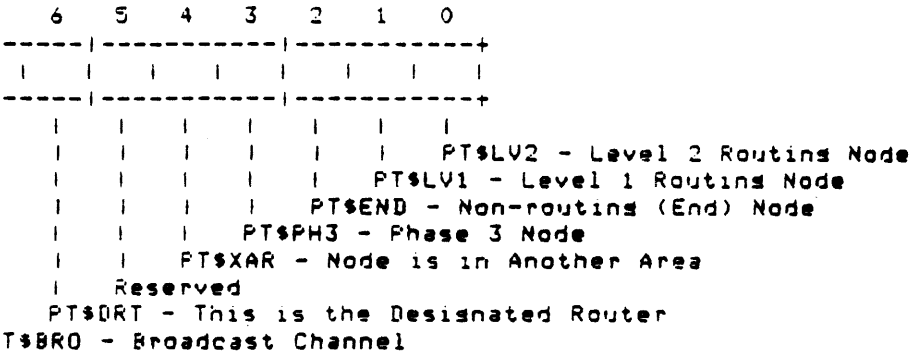
P$STA1/2 Routing State Definitions



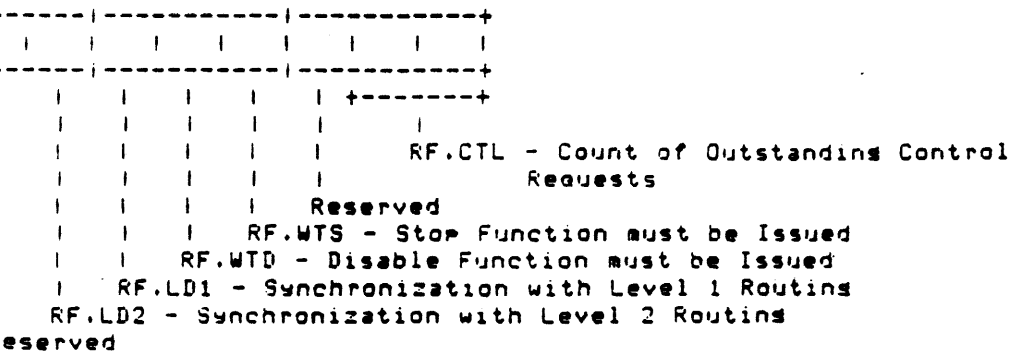
PSLST Link State Definitions

```
PS$OFF   = 0     Off
PS$STR   = 2     Starting
PS$WT    = 4     Waiting
PS$NTI   = 6     Not Initialized
PS$VER   = 10    Verification Wait
PS$UP    = 12    Up
PS$FAI   = 14    Failed (On-failed for DLM)
PS$CHR   = 16    Set Characteristics Failed
```

PSTYP Adjacent Node Type Definitions

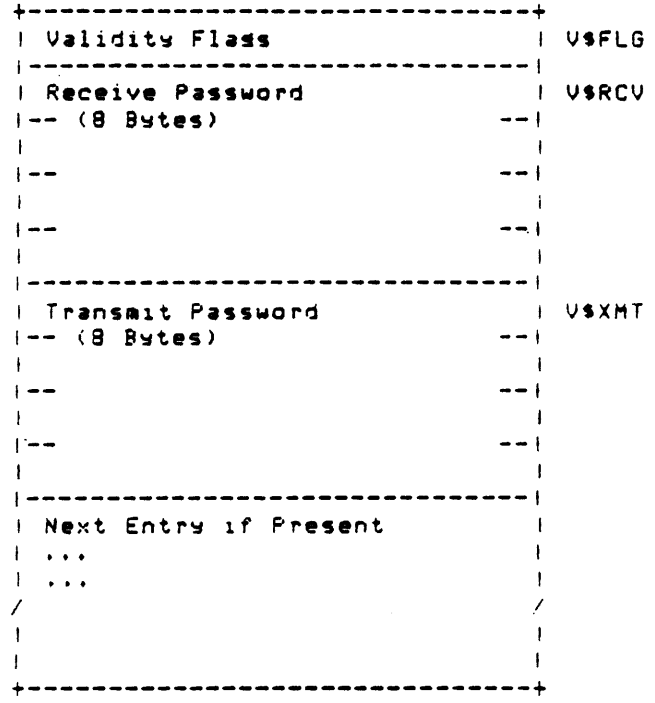


PSLCD Link Recovery Flag Definitions

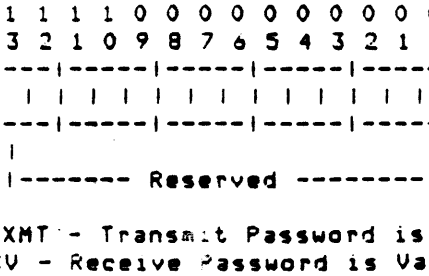


2.2.14 PLBDF$ - Password Descriptors Table <DSR>

The password descriptors are used by transport initialization to provide a level of security. Presence of a receive password notifies XPT that node verification must be requested from the adjacent node. The transmit password is transmitted in the transport verification message when verification is requested from the adjacent node.

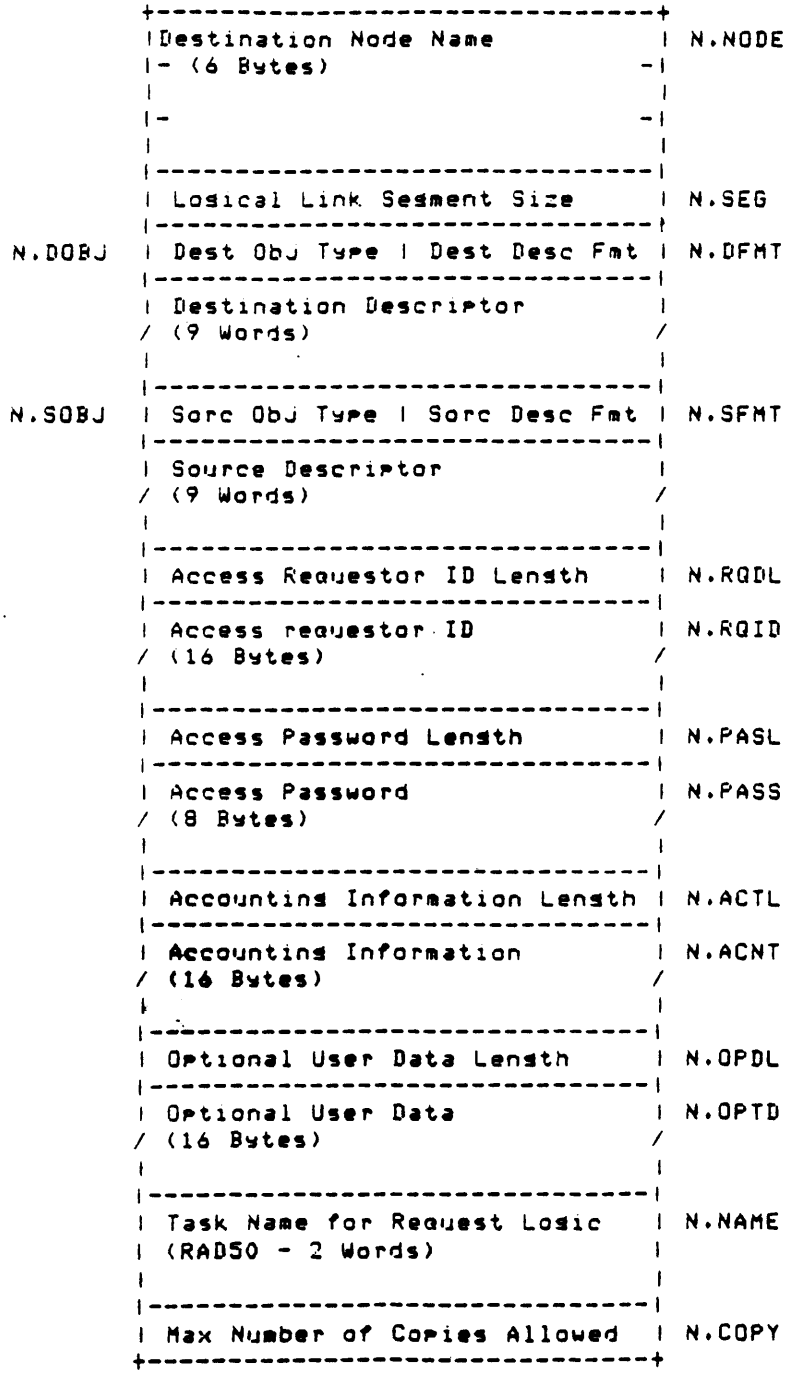


V$FLG Password Validity Flag Definitions



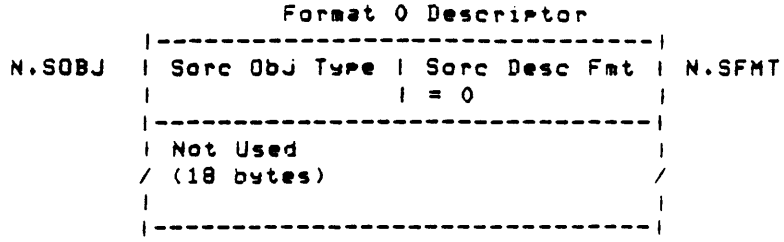
2.2.15 NSPIF$ - Connect Initiate Block <DSR>

The Session Control information for outgoing connects is obtained from the user task space to the Connect Initiate Block for use in building the Connect Initiate Message. When a Connect Initiate Message is received, the information is saved in the Connect Pending Block prior to notifying the user task.

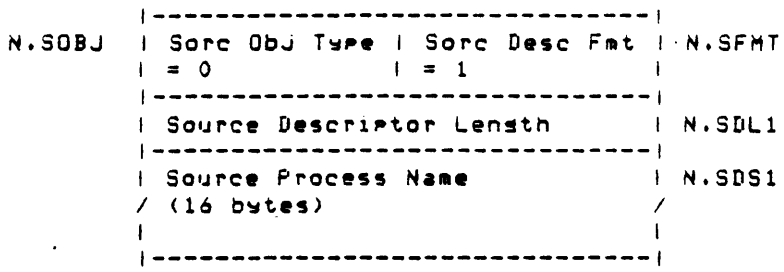


Source Descriptor Formats

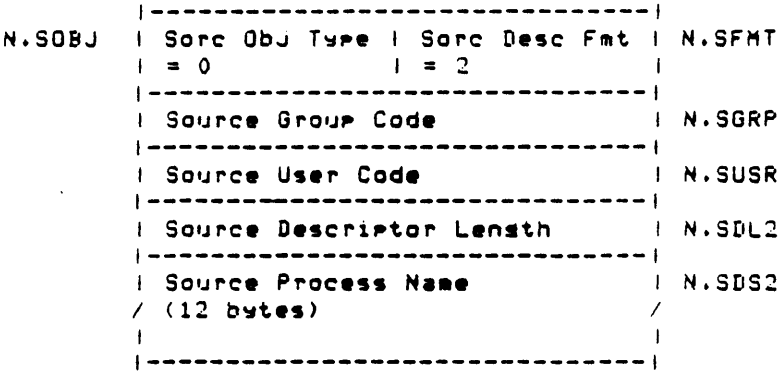
Format 0 Descriptor



Format 1 Descriptor

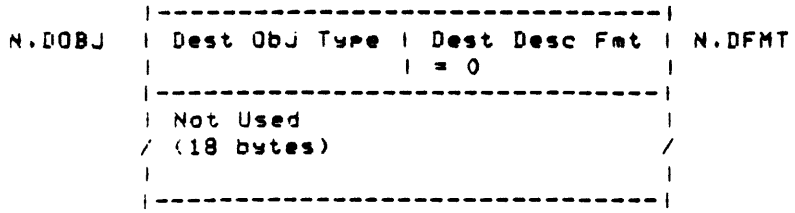


Format 2 Descriptor

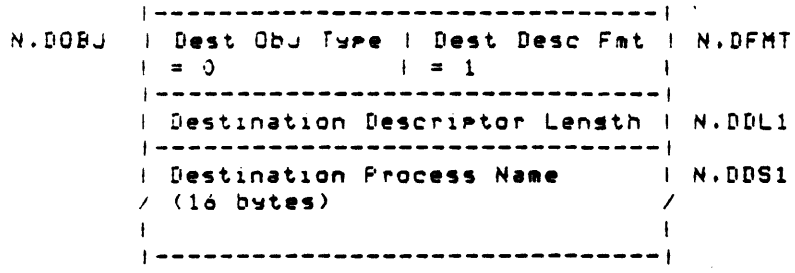


Destination Descriptor Formats

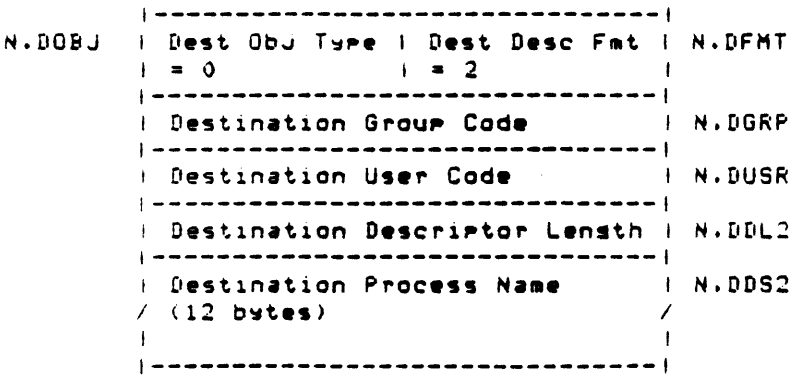
Format 0 Descriptor



Format 1 Descriptor



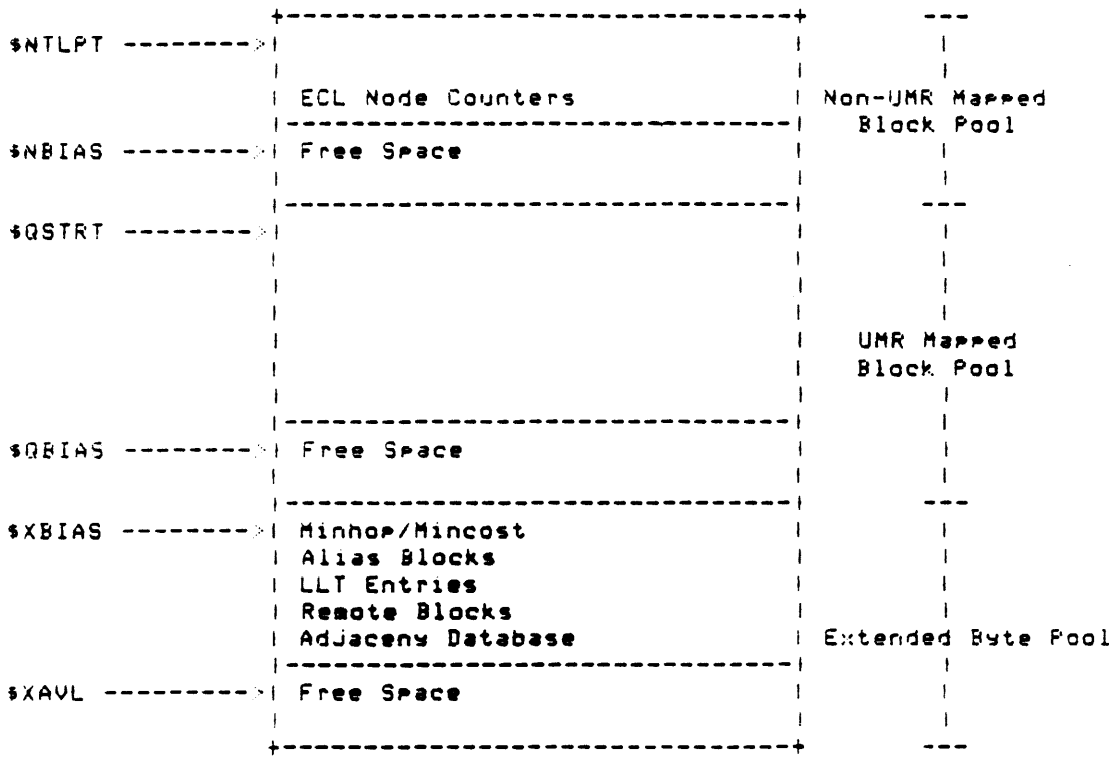
Format 2 Descriptor



2.3 DATA IN NETWORK POOL

In addition to the Comm/Exec / RSX Pool, DECnet uses another pool for the location of some of its data structures. The primary reason for an additional pool (not in Exec space) is the limited amount of DSR. Any additions to the Comm/Exec are taken from the system pool (DSR). The additional pool (Network Pool) is not in the network task's virtual address space, and therefore the data structure must be mapped before they can be used.

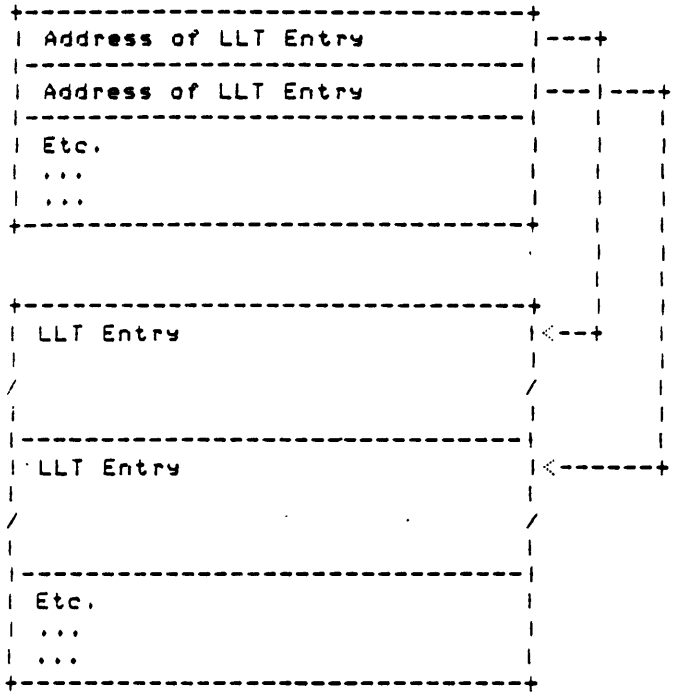
The general layout for the network buffer pool is as follows:



Note: $NBIAS is 0 if no extended memory

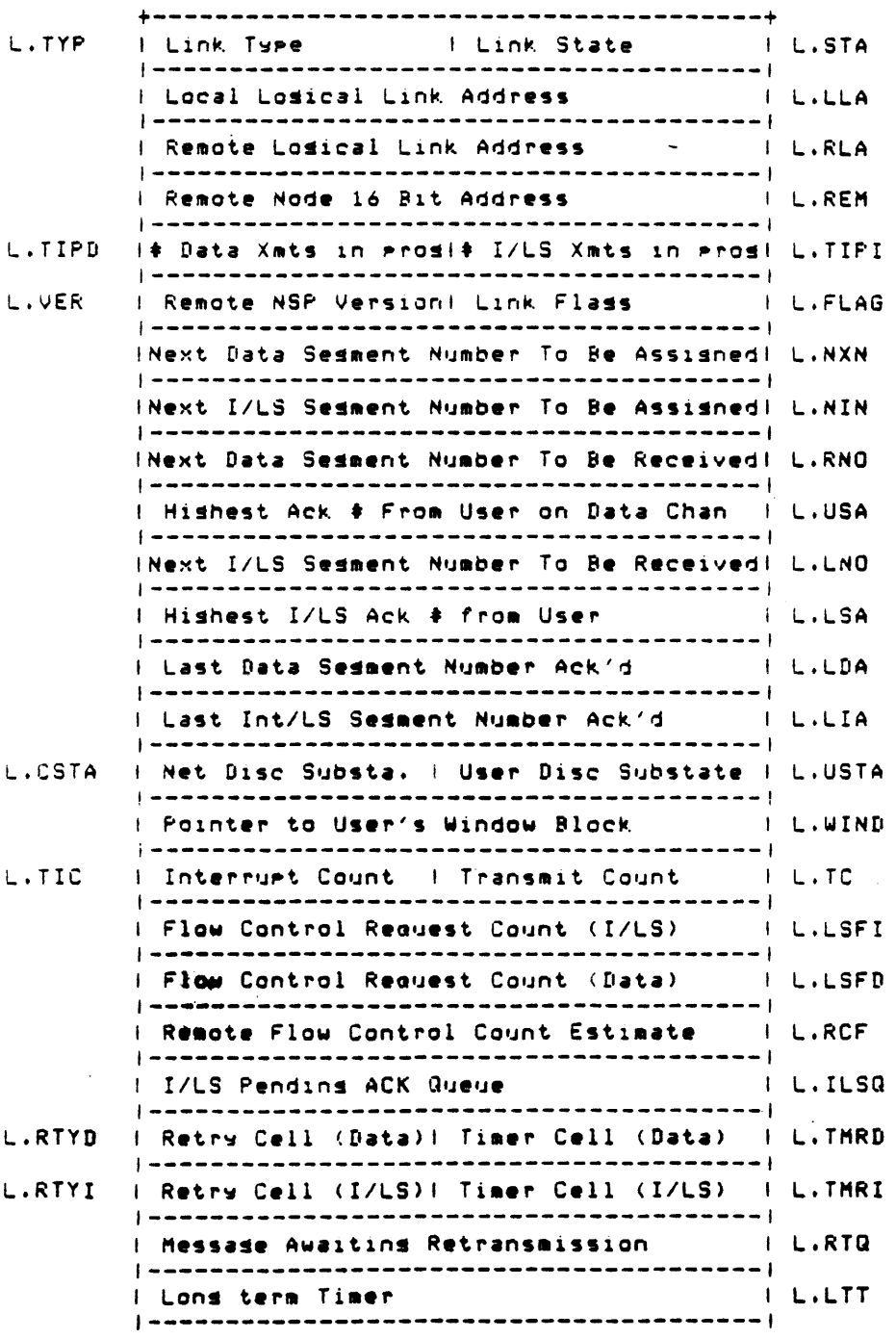
2.3.1 Logical Link Table (LLT)

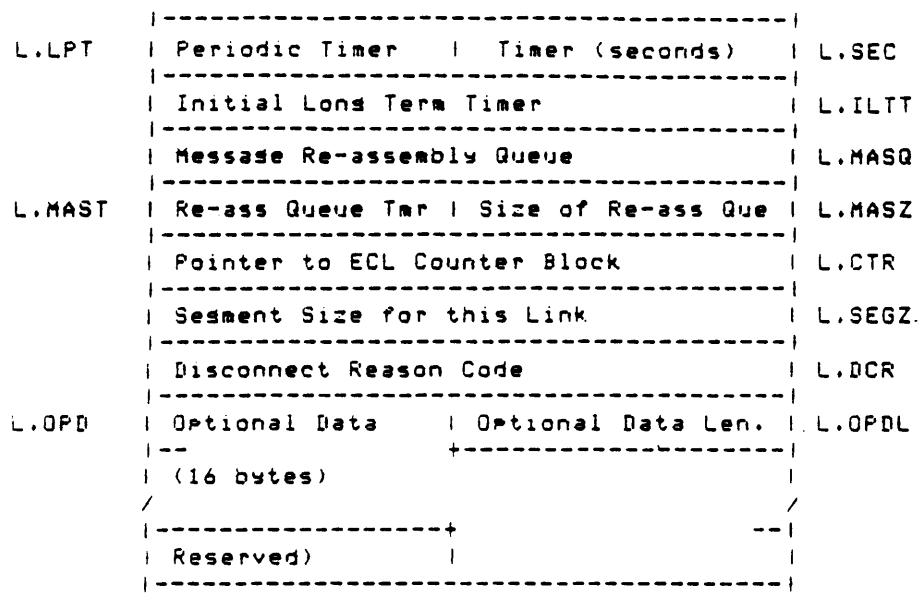
Each Logical Link Table entry is addressed by the appropriate offset into the LLT table. This offset is referred to as the the LLT index. The pointer to the Logical Link Table resides in the ECL database. Note that the LLT vector table is allocated from DSR.



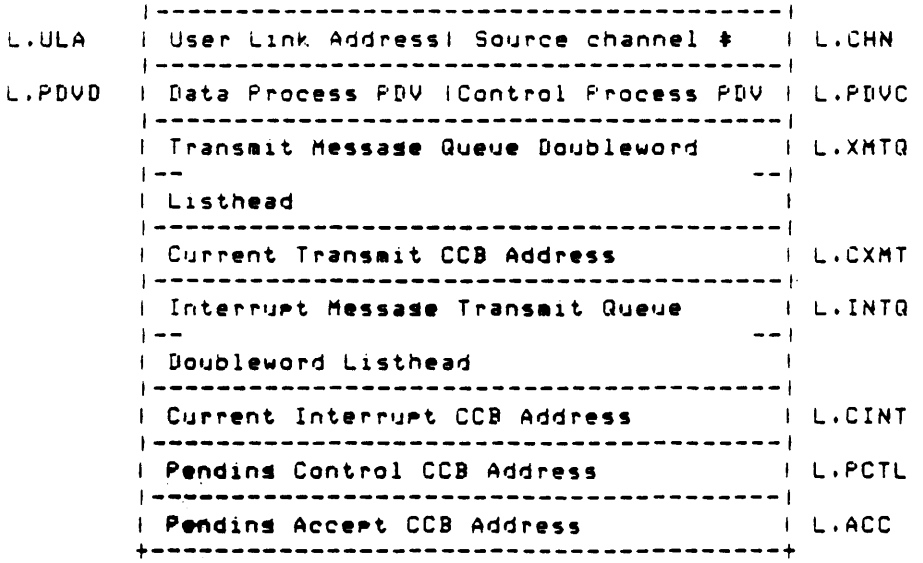
2.3.2 LLTDF$ - Logical Link Table Entry

Each LLT entry contains the current information for a single logical link. The LLT is used by the ECL process and the NETACP task to service logical links. The LLT entries of a running DECnet system can be dumped using the CEDump /LL switch (See Section 3.5).





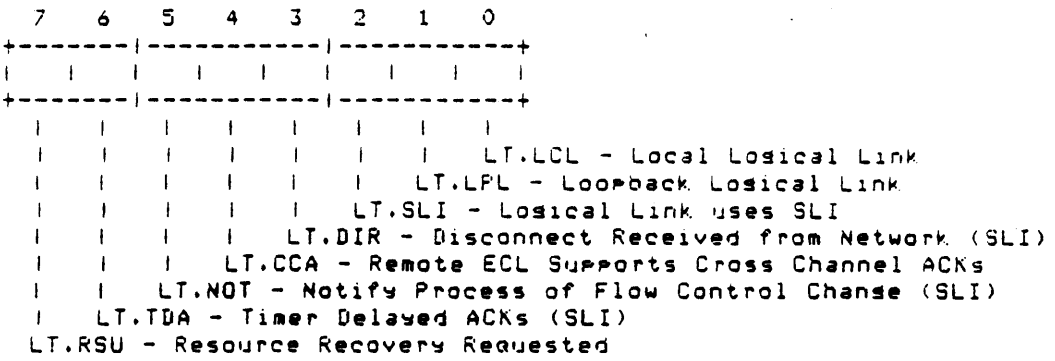
The following additional fields are only included if System Level Interface (SLI) is present.



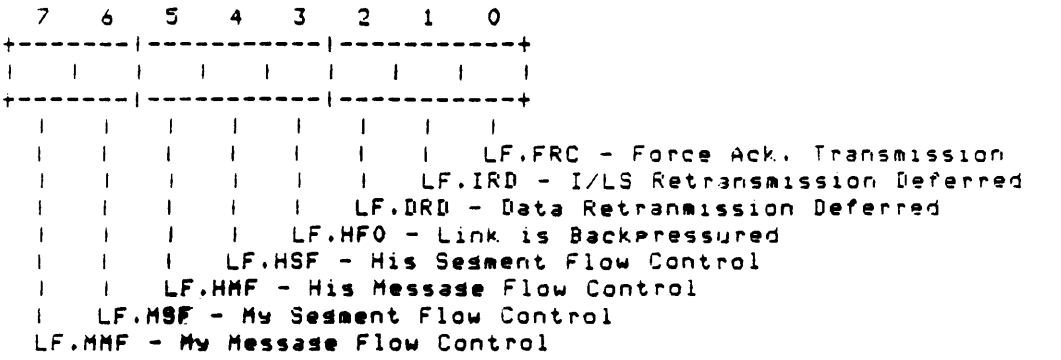
L.STA Logical Link State Definitions

```
ST$CIS  = 2     Connect Initiate Sent
ST$CC   = 4     Received CI, Sent CC
ST$CIR  = 6     Connect Initiate Received
ST$DAT  = 10    Normal Data Transfer
ST$DIP  = 12    Disconnect in Progress
ST$PND  = 14    Disconnect Pending (SLI)
```

L.TYP Link Type Definitions



L.FLAG Logical Link Flag Definitions



L.USTA Disconnect Substate for User

US$DON	= 0	Disconnect Complete
US$CNF	= 2	Connect Failure
US$DSC	= 4	User Disconnect Pending
US$DIS	= 6	Inform user of Disconnect
US$WDS	= 10	Wait for Disconnect Acknowledge
US$EAC	= 12	Error Response to Connect Accept

L.CSTA Disconnect Substate for Network

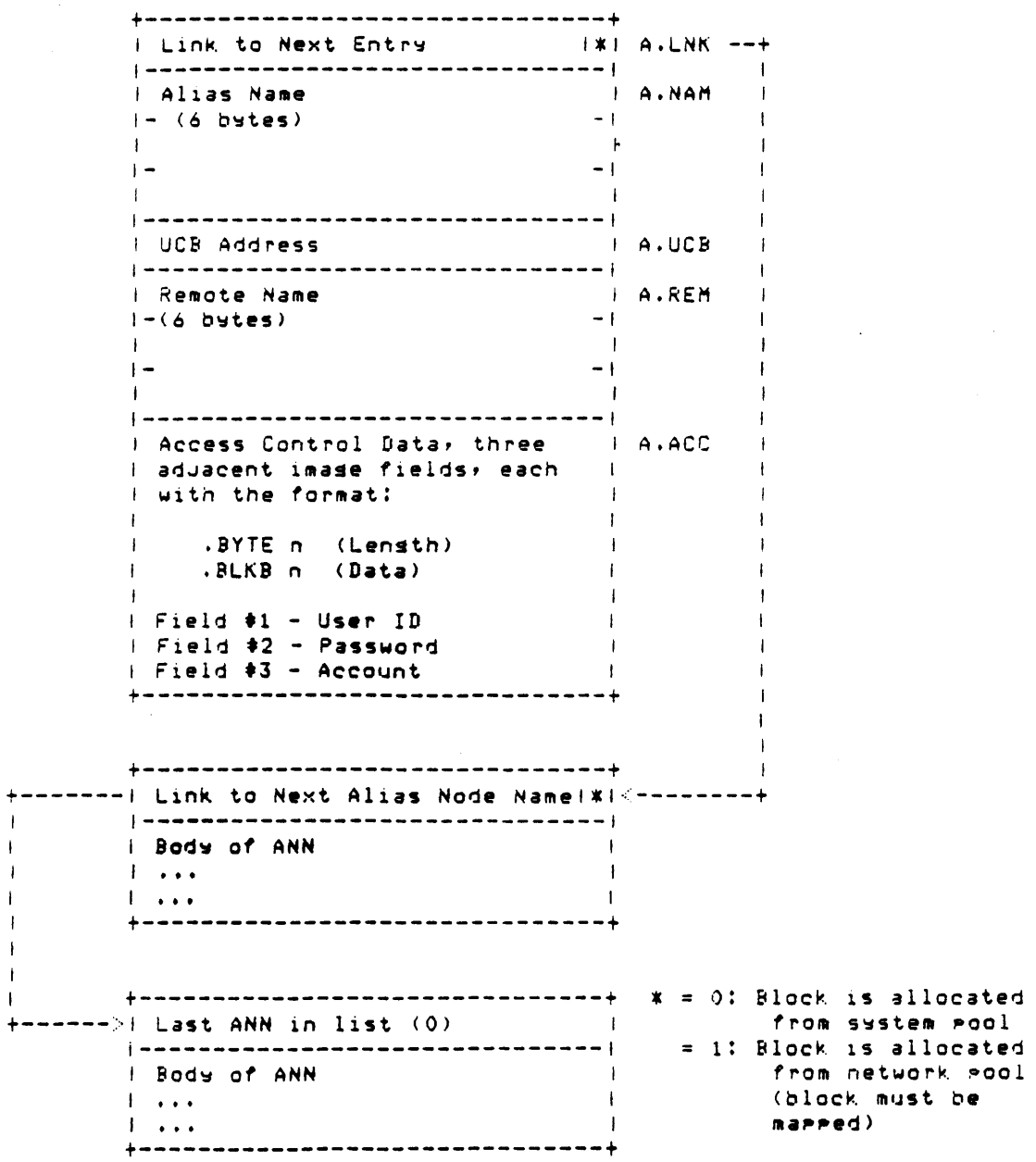
NS$DON	= 0	Disconnect Complete
NS$SDI	= 2	Send Disconnect Initiate Message
NS$WDC	= 4	Wait for Disconnect Confirm Message

Note that the high bit (Bit 15) of the L.LSFI and L.LSFD flow control request counts are used to flag that a Link Service message is outstanding for the sub-channel. The low order 15 bits (Bits 00-14) are the actual count.

The high bit (Bit 15) of L.USA and L.LSA indicate that an acknowledgement should be sent to the remote network services process.

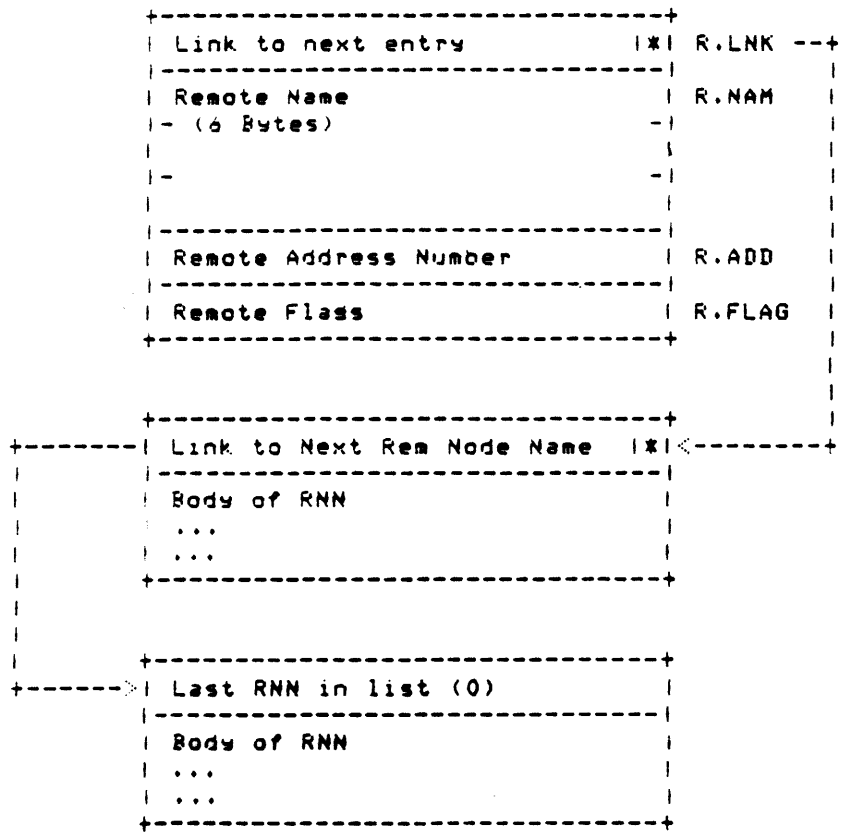
2.3.3 ANBDF$ - Alias Node Name (ANN) List

The mapping bias and virtual address of the first Alias Name Block in the Alias list is found in the D$ANN cell of the DECnet Home Block. The alias blocks may either be allocated from the Comm/Exec / RSX pool or Network Pool. If bit 0 of the alias block pointer is clear then the pointer is the virtual address of the alias block in Comm/Exec / RSX pool. If bit 0 is set then bits 1 through 15 of the pointer are the virtual address of the object block in network pool ($XAVL contains the bias).



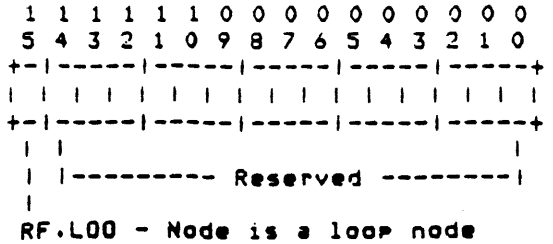
2.3.4 RNBDF$ - Remote Node Name (RNN) List

The mapping bias and virtual address of the first Remote Name Block in the Remote Name List is found in $RNNHD of the Comm/Exec. The remote name blocks may either be allocated from the Comm/Exec / RSX pool or Network Pool. If bit 0 of the remote name block pointer is clear then the pointer is the virtual address of the remote name block in Comm/Exec / RSX pool. If bit 0 is set then bits 1 through 15 of the pointer are the virtual address of the object block in network pool ($XAVL contains the bias).



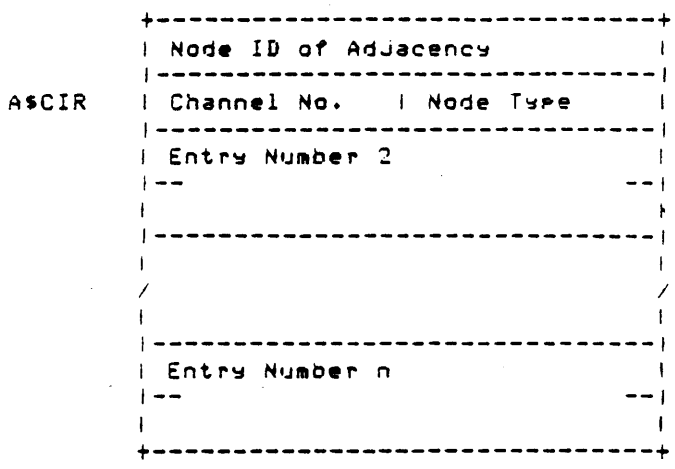
* = 0 - Block is allocated from system pool
= 1 - Block is allocated from network pool (block must be mapped)

R.FLAG Remote Flag Definitions

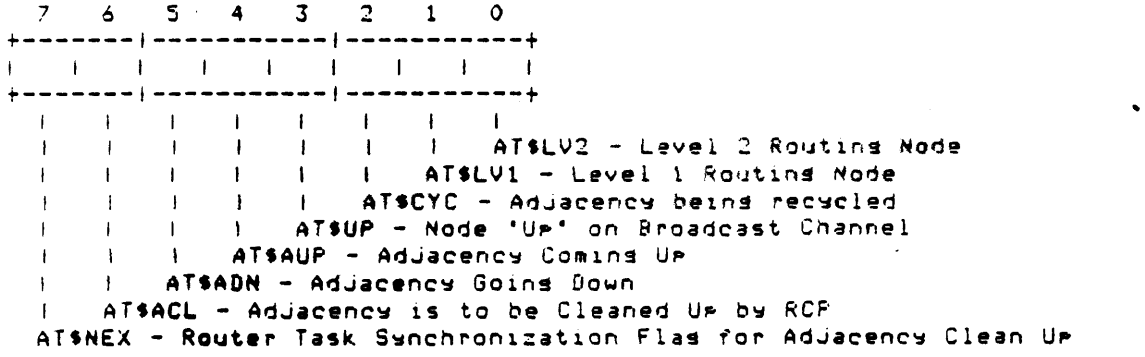


2.3.5 ADJDF$ - Adjacency Database Part 1

The Adjacency database contains information about each adjacent node (point to point neighbors and NI neighbors). Due to the size and attendant mapping problems, the database is split into 2 sections (ADJ1 and ADJ2). Adjacency Database Part 1 can be mapped by the bias contained in N$ADJ1 of the XPT database. The Adjacency Database is not present for non-routing (end) nodes.

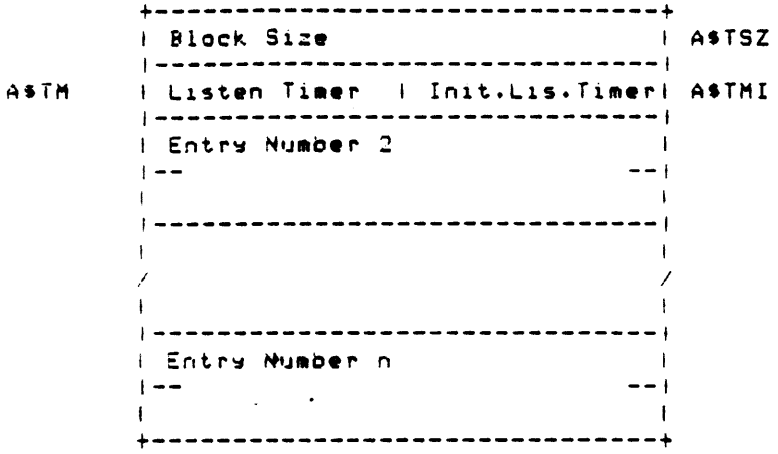


A$TYP Node Type Flag Definitions



2.3.6 ADJDF$ - Adjacency Database Part 2

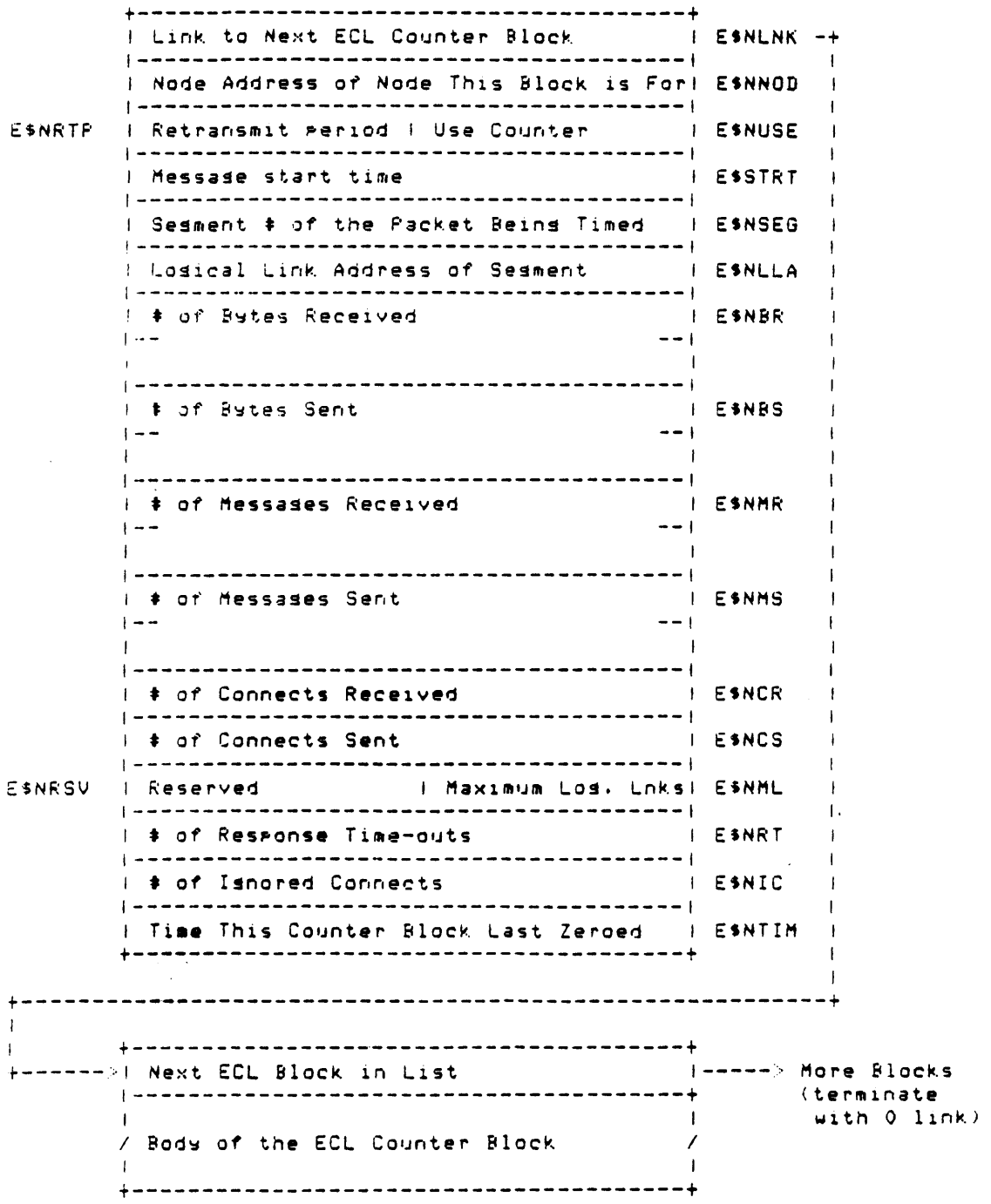
The Adjacency database contains information about each adjacent node (point to point neighbors and NI neighbors). Due to the size and attendant mapping problems, the database is split into 2 sections (ADJ1 and ADJ2). Adjacency Database Part 2 can be mapped by the bias contained in N$ADJ2 of the XPT database. The Adjacency Database is not present for non-routing (end) nodes.



Note: The Listen Timer and Initial Timer Values are kept as multiples of 4 second intervals. This allows for a maximum value of 1020 seconds (4 255.).

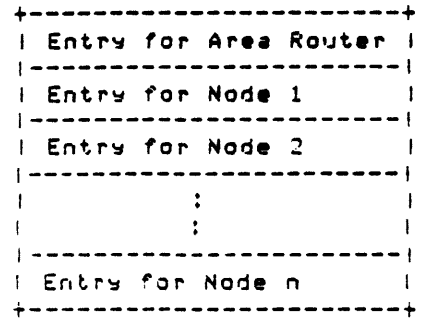
2.3.7 CTRDF$ - ECL Node Counters List

These counters are maintained by ECL and read/zeroed by Network Management.

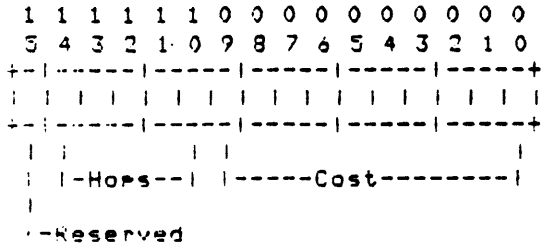


2.3.8 Node MinHop/MinCost Table

The pointer to the minhop/mincost table resides is NSMHC1 inthe XPT database. The contents of the database represent the minimum hops and cost along a path to a target node. This database is not present in non-routing (end) nodes.

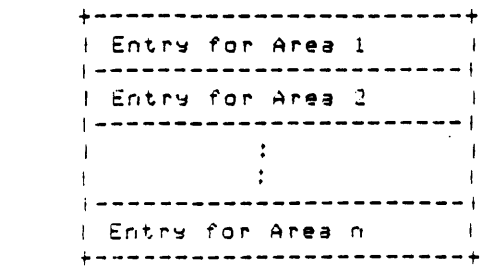


Entry in Hop / Cost Table

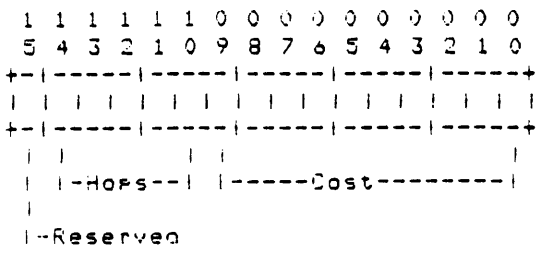


2.3.9 Area MinHop / MinCost Table

The pointer to the area minhop/mincost table resides is N$MHC2 in the XPT database. The contents of the database represent the minimum hops and cost along a path to a target area routing node. This database is not present in non-routing (end) nodes.



Entry in Area Hop / Cost Table

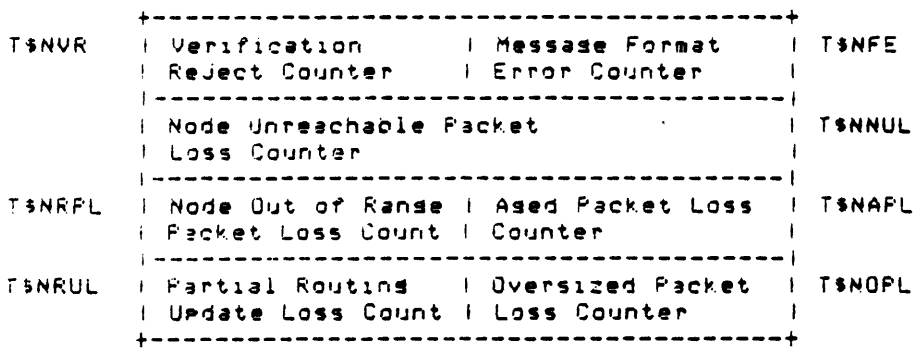


2.4 DATA IN PROCESS SPACE

The following data structures are located in process space.

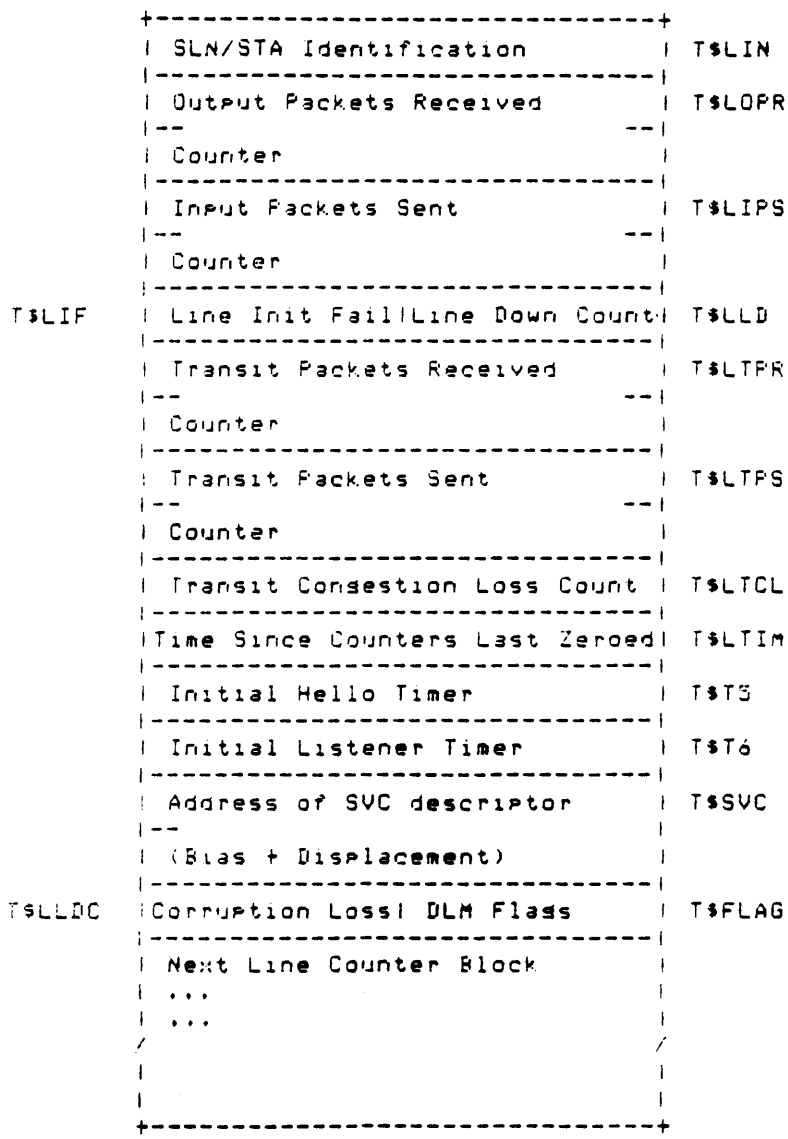
2.4.1 CTRDF$ - Transport Node Counters Table

These counters are maintained by XPT and read/zeroed by Network Management.



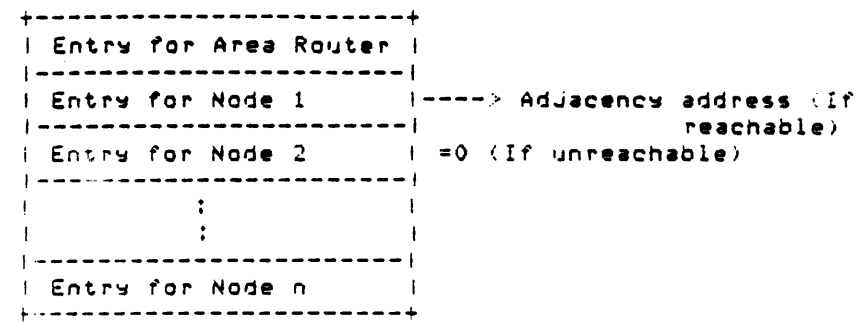
2.4.2 CTRDF$ - Transport Line Counters Table

These counters are maintained by XPT and read/zeroed by Network Management.



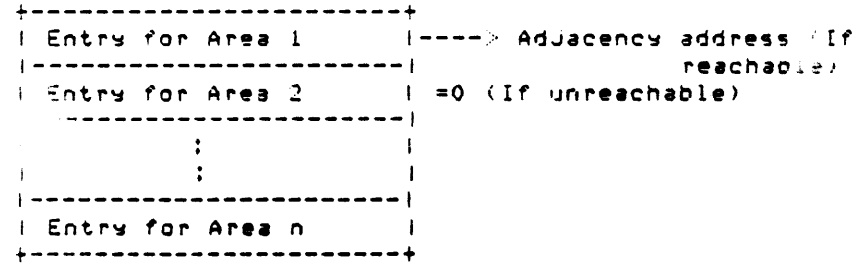
2.4.3 Node Reachability Vector

??? Insert structure explanation here ???



2.4.4 Area Reachability Vector

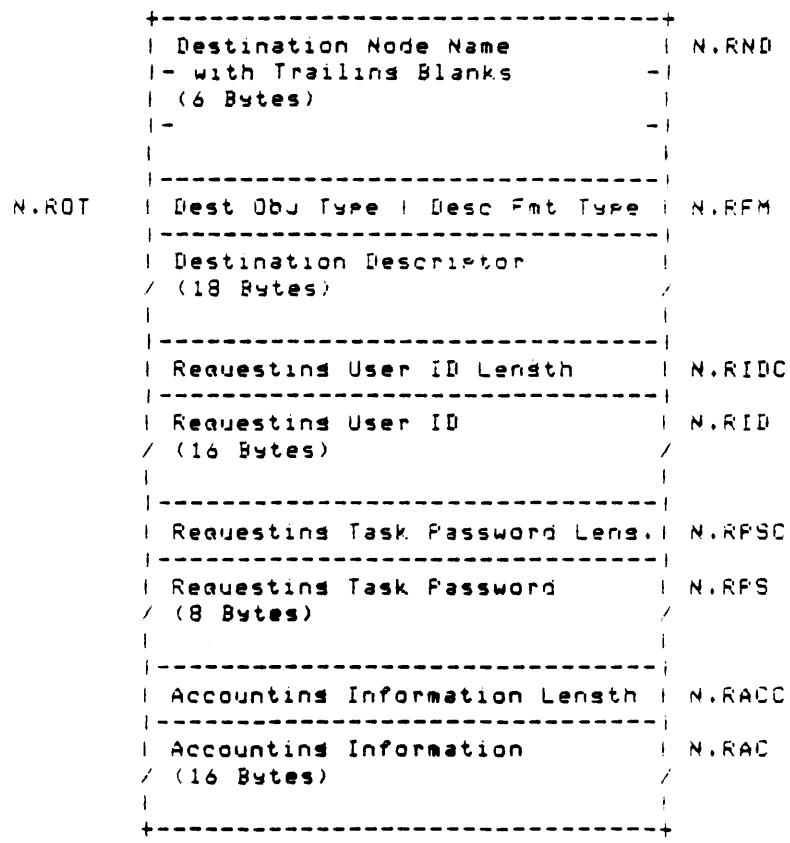
??? Insert structure explanation here ???



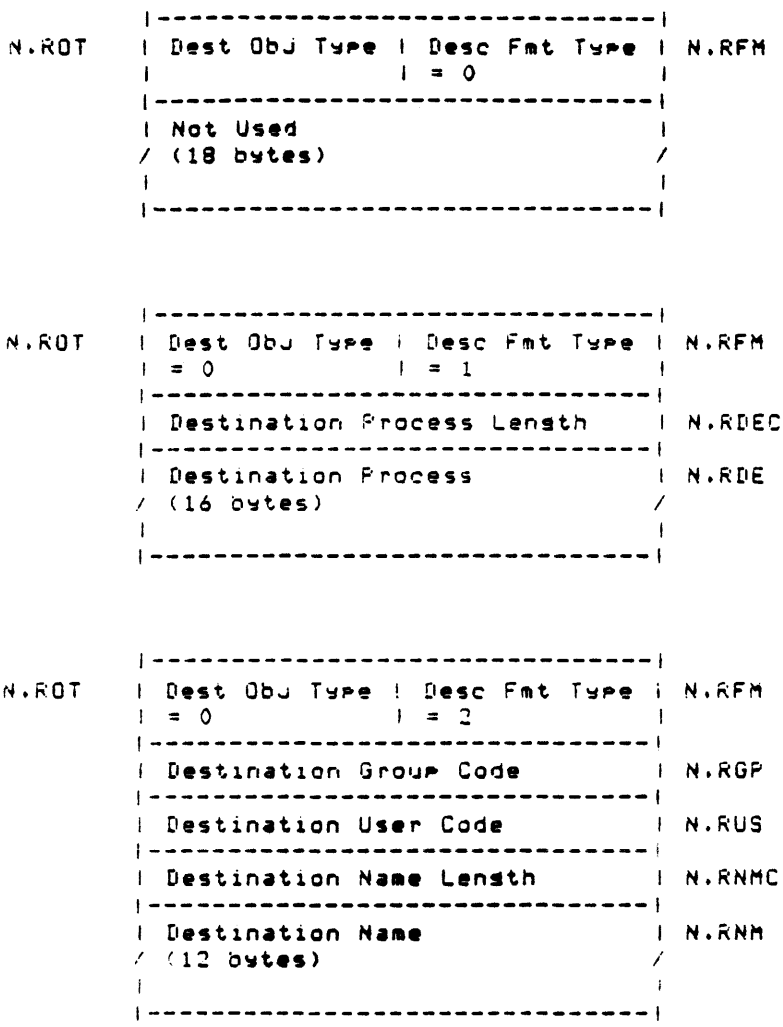
2.5 DATA IN USER TASK

The following data structures are used by the user. They are all in user space.

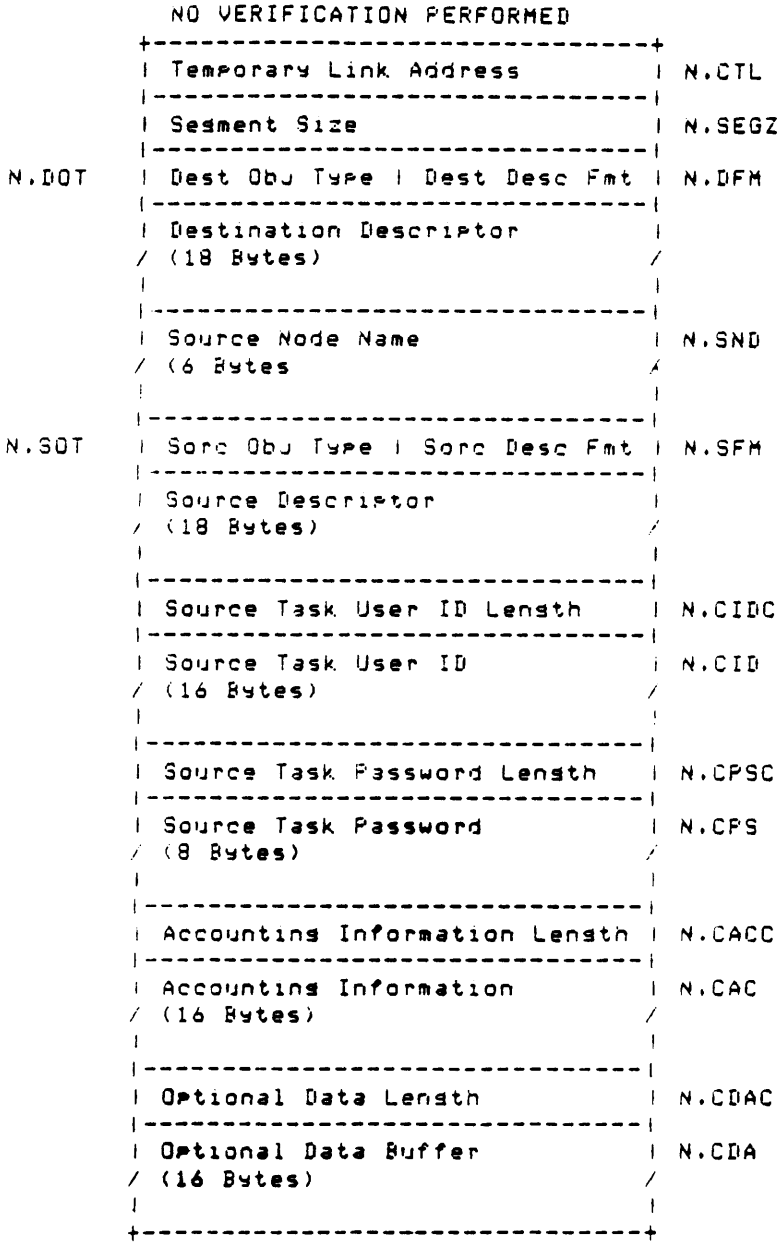
2.5.1 Request Connect Block

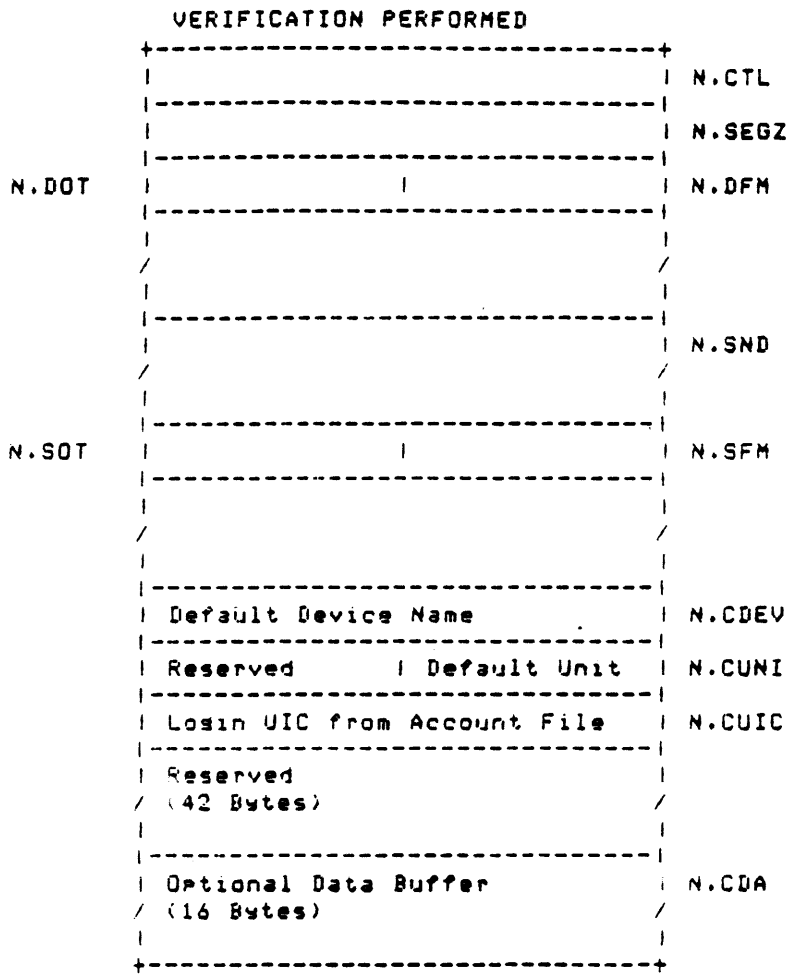


Descriptor Field Formats

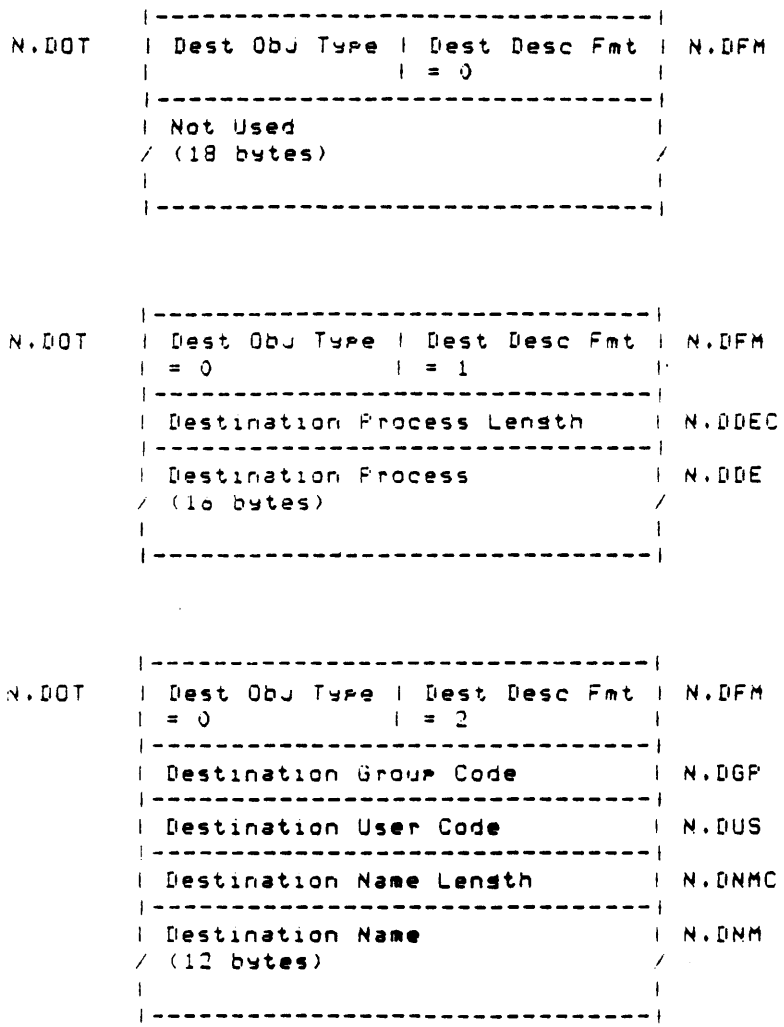


2.5.2 CNBDF$ - Received Connect Block

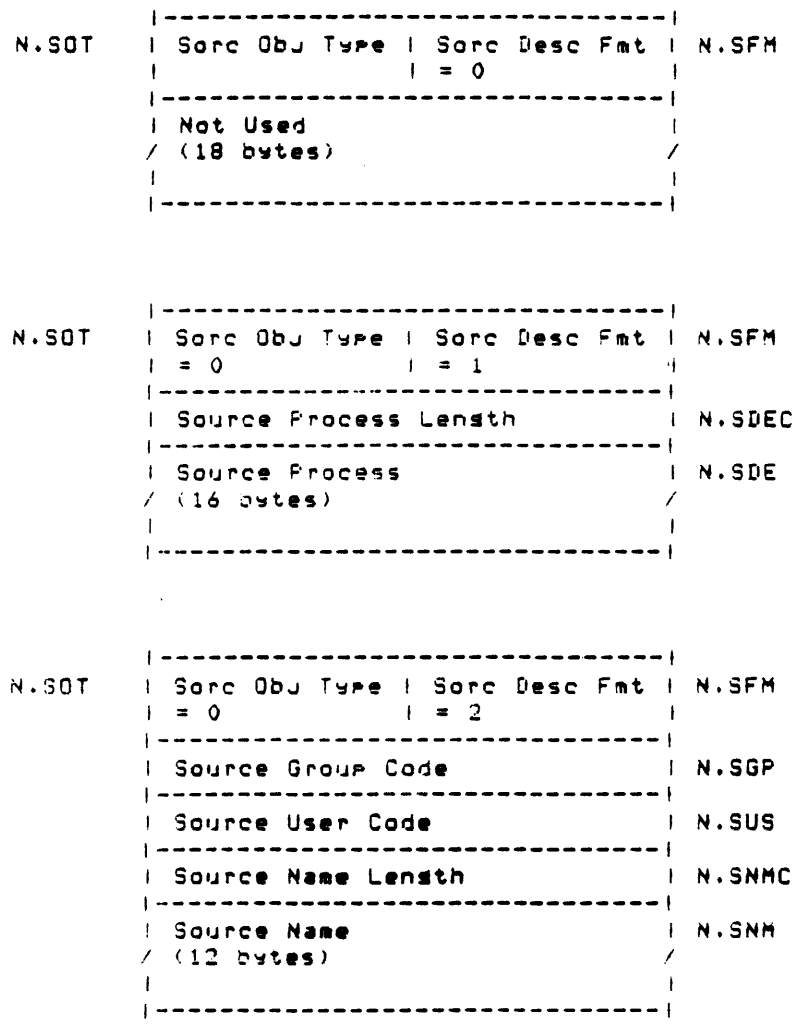



VERIFICATION PERFORMED
N.CTL
N.SEGZ
N.DOT
N.DFM
N.SND
N.SOT
N.SFM
Default Device Name
N.CDEV
Reserved
Default Unit
N.CUNI
Login UIC from Account File
N.CUIC
Reserved
(42 Bytes)
Optional Data Buffer
(16 Bytes)
N.CDA

Destination Descriptor Formats



Source Descriptor Formats



CHAPTER 3

DECNET CEDUMP SWITCHES

This chapter describes the switches available to dump the internal data structures of a running DECnet system. For a description of the CEDump utility see the Guide to Writing a Comm/Exec Process Manual. The switches described in this chapter will only function if the DECnet product is installed in the system and loaded. The /AL switch described in the Guide to Writing a Comm/Exec Process Manual will also dump all the DECnet switches if DECnet is present.

****
*   *
* /DH   *
*   *
****

3.1 The /DH Switch - Dump DECnet Home Block

Format: /DH

Description: This switch will cause CEDUMP to dump the DECnet home block. This block holds many of the constants used by DECnet. For a description of the DECnet home block layout see Section 2.2.1.

Output Example:

```
Dump of DECnet home block for:

ELROND(4.19) DISTRIBUTED SYSTEMS

4.19            Host node address
000252          Local Ethernet high-order address
000004
9.              Number of routing channels
1022.           Number of broadcast router adjacencies
12.             Number of broadcast endnode adjacencies
1022.           Maximum node number
24.             Maximum cost
21.             Maximum hops
0.              Maximum area number
10.             Maximum visitation count
64.             Square root limiting factor
15149.          Routing timer
16133.          Broadcast routing timer
0.              Broadcast routing priority
100427          Features flags
                - Access verification support
                - Access verification enabled
                - Event logging support
                - Logical link integrity support
                - Multi-copy task support
                - System level interface support
```

****
*   *
* /GE   *
*   *
****

3.2 The /GE Switch - Dump General Delivery Queue

Format: /GE

Description: This switch will cause CEDUMP to dump the general delivery queue. The general delivery queue is explained in section 2.2.6.

Output Example:

```
NETACP general delivery queue:

CCB @070136     type code: 201 (NT.CON)
CCB @060124     type code: 201 (NT.CON)
```

DHBDF

```
**************
*            *
*    /IO     *
*            *
**************
```

3.3 The /IO Switch - Dump I/O Queues

Format: /IO

Description: This switch will cause CEDUMP to dump the QIO I/O request queue associated with the NETACP.

Output Example:

```
NETACP pending I/O queue:

I/O request from task-TASKAB /TI=TT14:
Function code- 123        Device-NS:
Parameters- 000001 000002 000003 000004 000005 000006 000007 000008
   .
   .
   .
 (repeated for all items on the I/O queue)
   .
   .
```

****
*   *
* /LI   *
*   *
****

3.4 The /LI Switch - Dump Active Link Counts

Format: /LI

Description: This switch will cause CEDUMP to dump the logical links information for DECnet. This includes the maximum and current link counts.

Output Example:

```
Active link counts:

Current-3.      Maximum-9.
```

****
*   *
* /LL   *
*   *
****

3.5 The /LL Switch - Dump Logical Link Tables

Format: /LL

Description: This switch will cause CEDUMP to dump the logical link tables. This data structure is explained in section 2.3.2. Also included with the dump is the associated window block. The window block structure is described in section 2.2.7.

Output Example:

```
Logical link vector table:

LLT entry @ 3735514
Logical link state - DAT (Normal Data Transfer)
Link type-20
  Link type flags set:
  Supports cross channel acks
Local link address-44401 Remote link address-32076
Remote node address-4.24 Remote flow control estimate-1.
Link flags-200 Remote NSP version-2.
  Link flags set:
  Local message flow control
                                DATA I/LS
                                ---- ----
Transmits in progress-          0. 0.
Next segment to be assigned-    2. 4.
Next segment to be received-    2. 2.
Highest ack from user-          1. 1.
Highest ack from network-       1. 3.
Segments/msgs requested-        0. 1.
Flow control count-             0. 0.
Timer cell-                     0. 0.
Retry cell-                     5. 5.
Long term timer-0.        Periodic timer-20.
Counter block address-142020  Segment size-561.
Window block @53150
Link status-0  Task lun-4.
  Status flags set:
  None
Segment size-561.         Mailbox address-31320
  Kernel AST status-0
  Kernel AST flags set:
  None
Current transmit I/O packet address-0
Current interrupt I/O packet address-0
  .
  .
  (repeated for all LLTs in the system)
  .
  .
```

```
**************
*            *
*    /LN     *
*            *
**************
```

3.6 The /LN Switch - Dump Physical Link Data Base

Format: /LN

Description: This switch will cause CEDUMP to dump the physical link data base. These blocks, called physical link blocks (PLBs), are described in section 2.2.13.

Output Example:

```
Physical link data base:

Physical Link Block @ 55122
State - Up
Adjacent node type-12
  Adjacent node type flags set:
  Level 1 router
  Phase 3 node
Outstanding control msgs-0.   Pending control requests-None
Recovery timer-0.
Channel-1.     Number of msgs queued-0.
General protocol timer-13.
Link flags-5
  Link flags set:
  Enabled
  Up state desired
Input packet limiter-0.
Maximum delay for routing msgs-145.
Routing state-0
  Flags set:
  None
Transport block size-576.
Number of nodes with lowest block size-1.
Transport counter block address-136460
Designated router address-0

                .
                . (repeated for all PLBs)
                .
```

****
*   *
* /MB   *
*   *
****

3.7 The /MB Switch - Dump Mailbox Queue

Format: /MB

Description: This switch will cause CEDUMP to dump the mailbox queues associated with all active network tasks. The mailbox queue and mailbox structures are described in section 2.2.5.

Output Example:

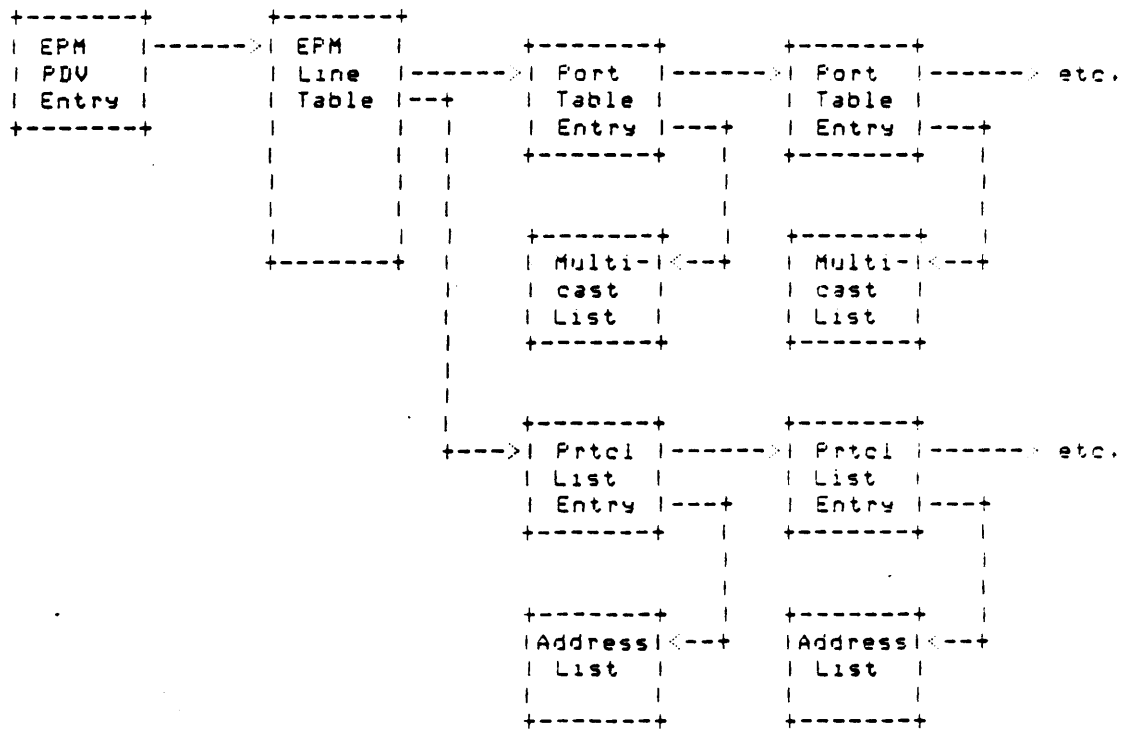
```
DECnet Mailbox queue:

Mailbox @ 53430 for task-NIC.1 TI=C00
# of ASTs needed-0.
# of active logical links-1.  Max # of links allowed-1.
Network data AST virtual address-121250
# of items on mailbox queue-0.
Logical link recovery timer-0.

Mailbox @ 55760 for task-RVTT14 TI=TT14
# of ASTs needed-0.
# of active logical links-1.  Max # of links allowed-0.
Network data AST virtual address-121250
# of items on mailbox queue-0.
Logical link recovery timer-0.

                .
                . (repeated for all mailboxes)
                .
```

APPENDIX A

EPM PROCESS DATA STRUCTURES

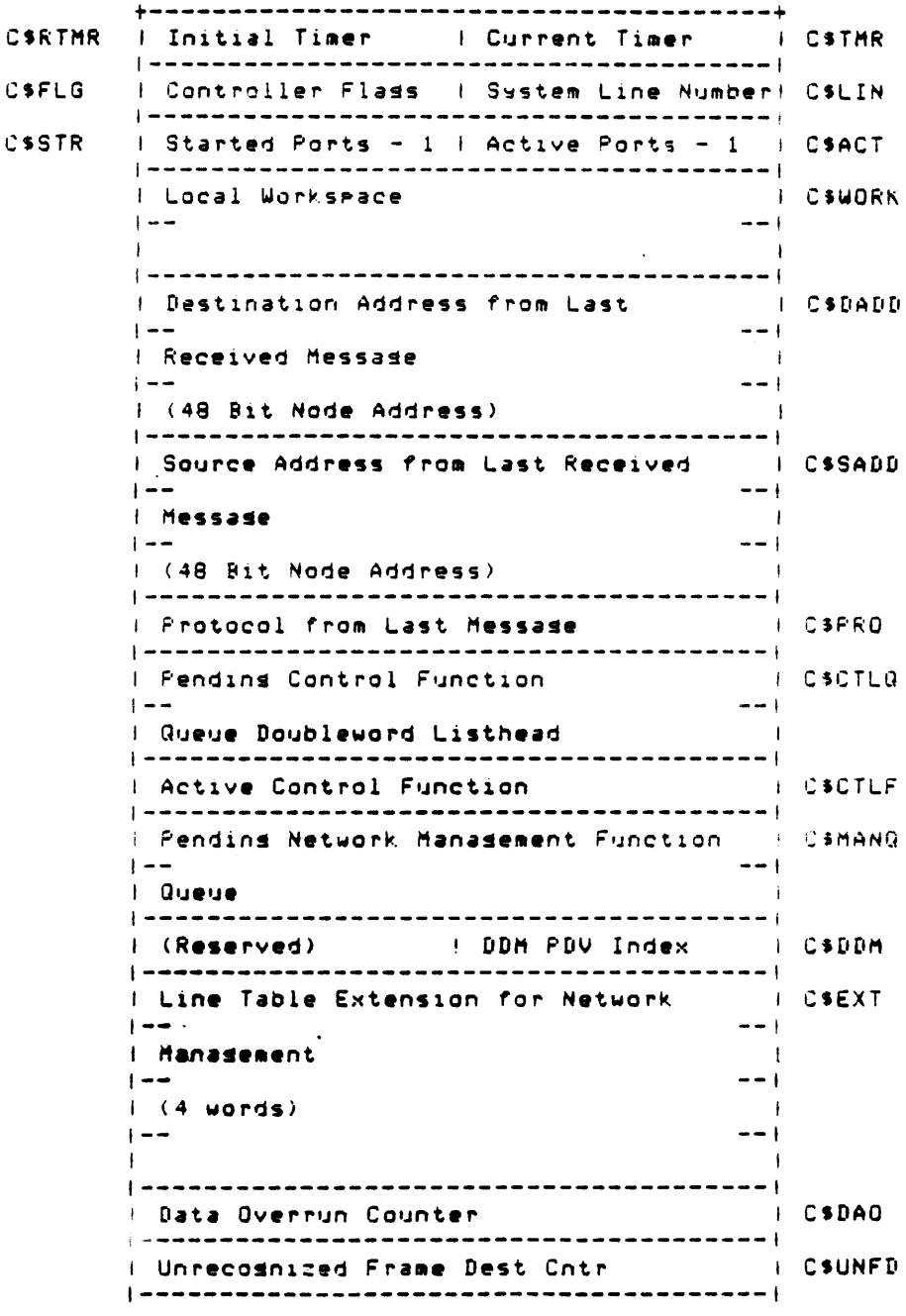
A.1 Introduction

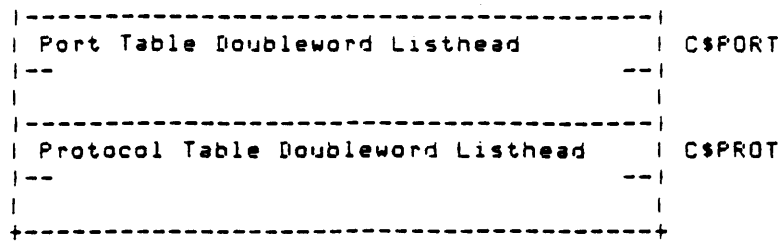
This appendix describes the RSX-DECnet V4.0/2.0 Ethernet Protocol Manager (EPM) data base. Its primary purpose is to provide more detailed descriptions of the EPM data structures.



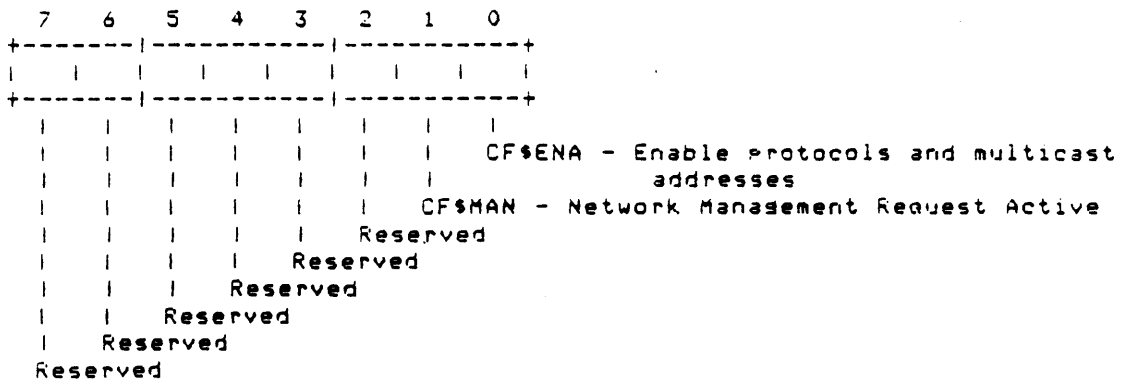
A.2 EPMDF$ - EPM LINE TABLE

The Ethernet Protocol Manager (EPM) Line Table contains the information for managing broadcast channels (e.g., UNA and QNA). The Line Table contains listheads for each port (port table listhead) and protocol (protocol listhead) which is assigned for the broadcast channel.



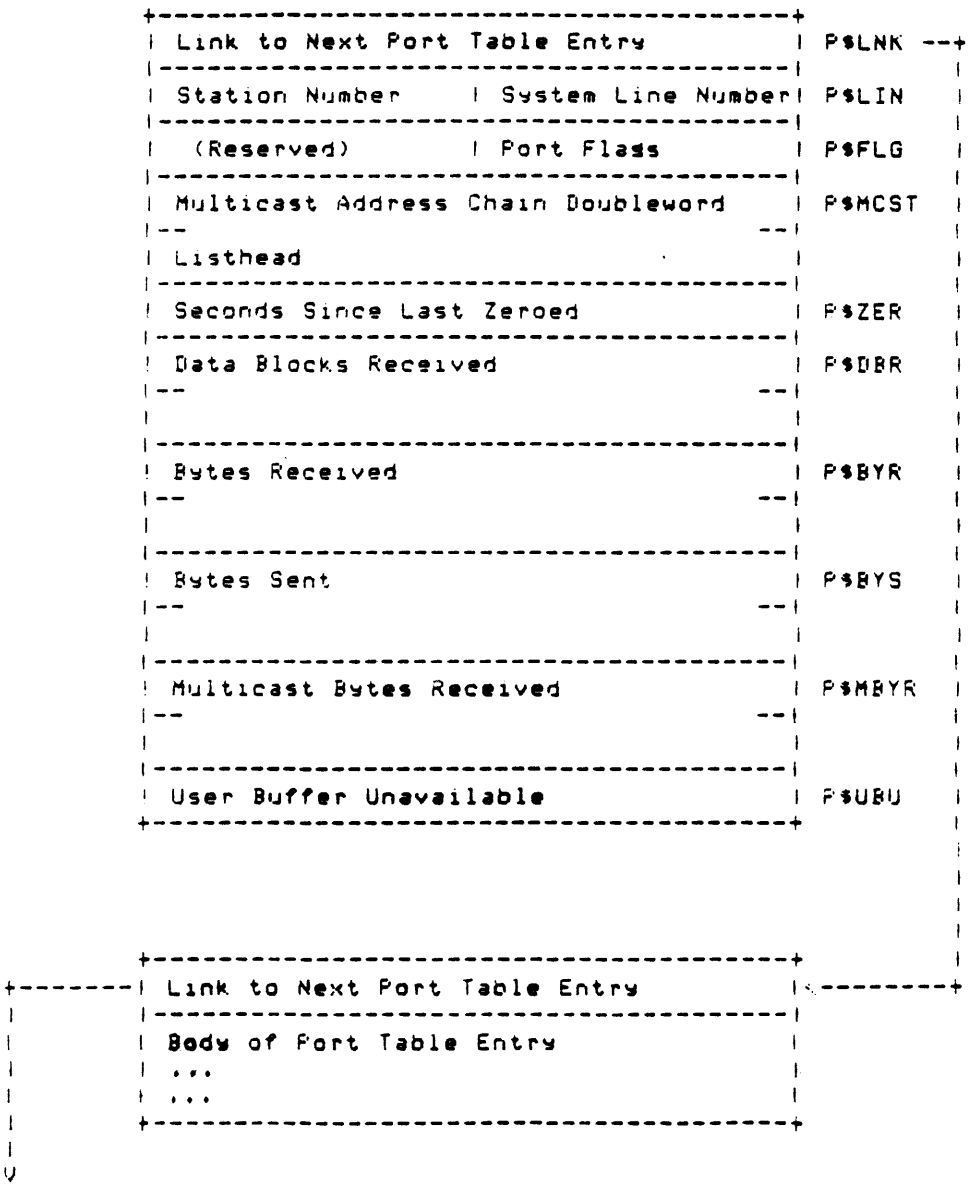


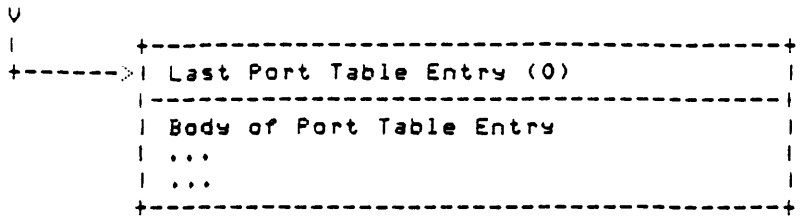
C$FLG Controller Flag Definitions



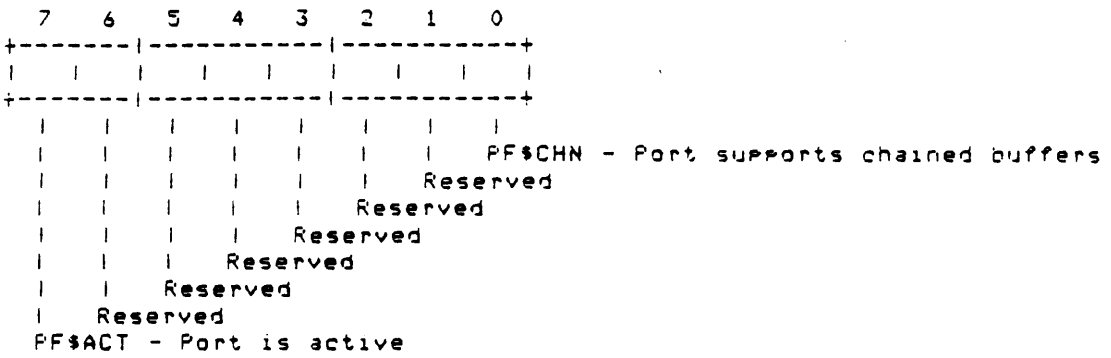
A.3 EPM PORT TABLE

The EPM port table contains information for each portal assigned to a broadcast channel. The port table listhead is contained in the EPM Line Table (C$PORT). Each port entry contains a list of multicast addresses which have been assigned (multicast listhead) to that port.



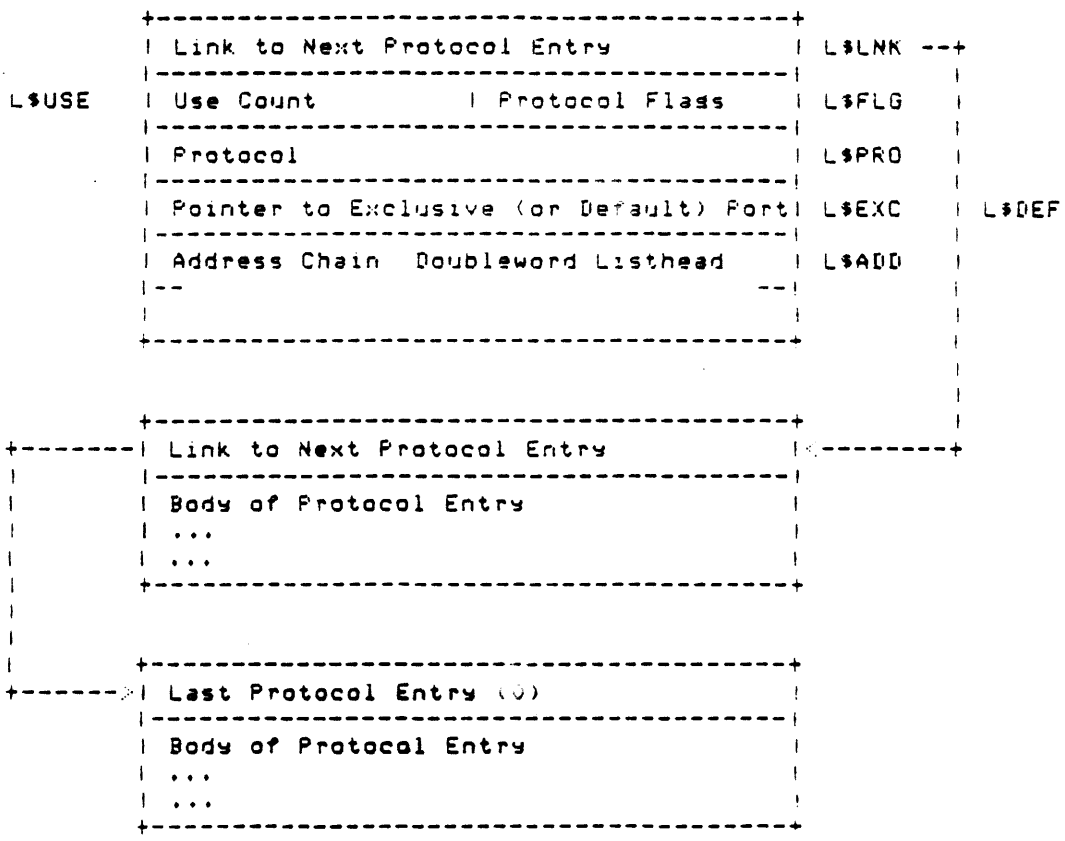


P$FLG Port Flags

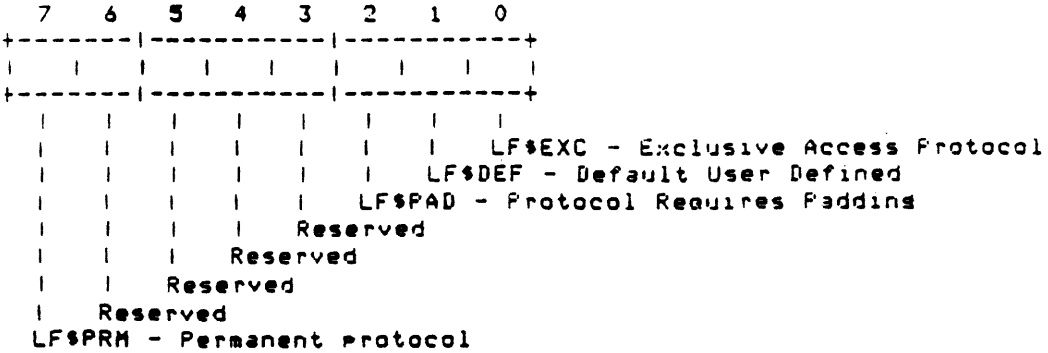


A.4 EPM PROTOCOL TABLE (LIST)

The Protocol Table is a list of protocols which are enabled for a broadcast channel. The Protocol Table listhead is contained in the EPM Line Table (C$PROT). Each protocol type defined has an associated list of Ethernet addresses which are enabled for that protocol (the address chain list).

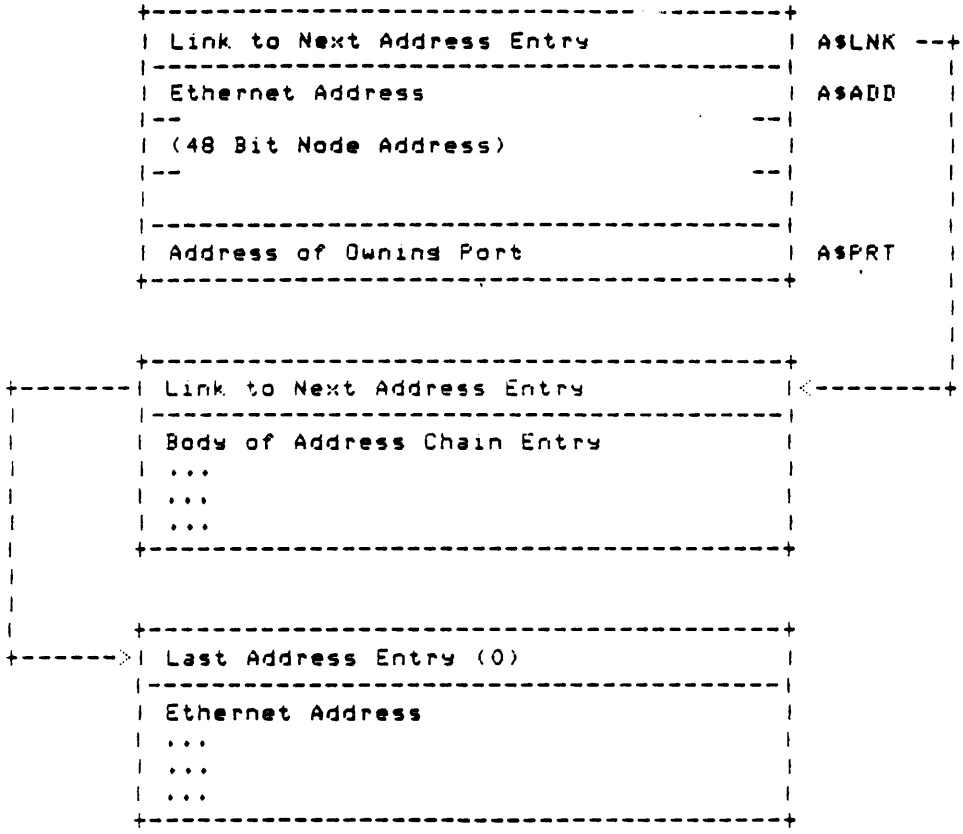


L$FLG Protocol Flags



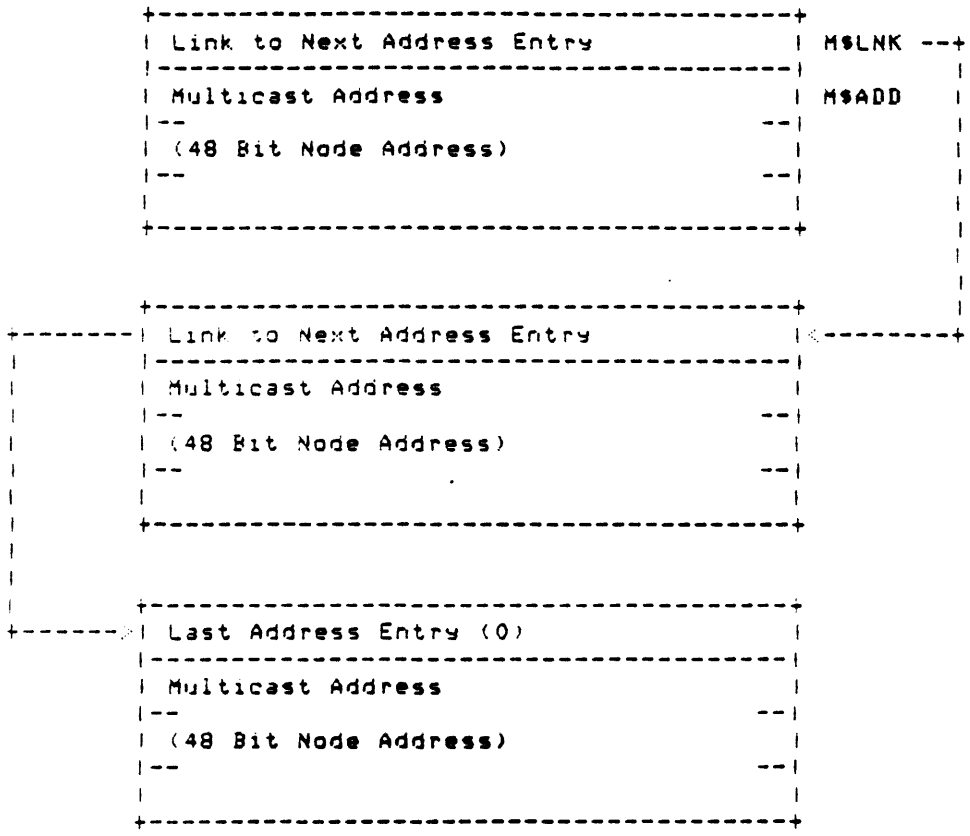
A.5 EPM ETHERNET ADDRESS CHAIN

The address chain entry contains the Ethernet address which is enabled for a particular protocol type. The listhead is contained in the EPM Protocol entry.



A.6 EPM MULTICAST ADDRESS CHAIN

The Multicast Address Chain entry contains the multicast address which is enabled for a broadcast channel portal. The listhead is contained in the EPM Port Table entry (P$MCST).

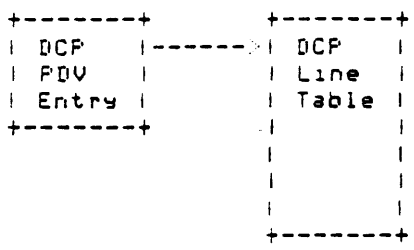


APPENDIX B

DCP PROCESS DATA STRUCTURES

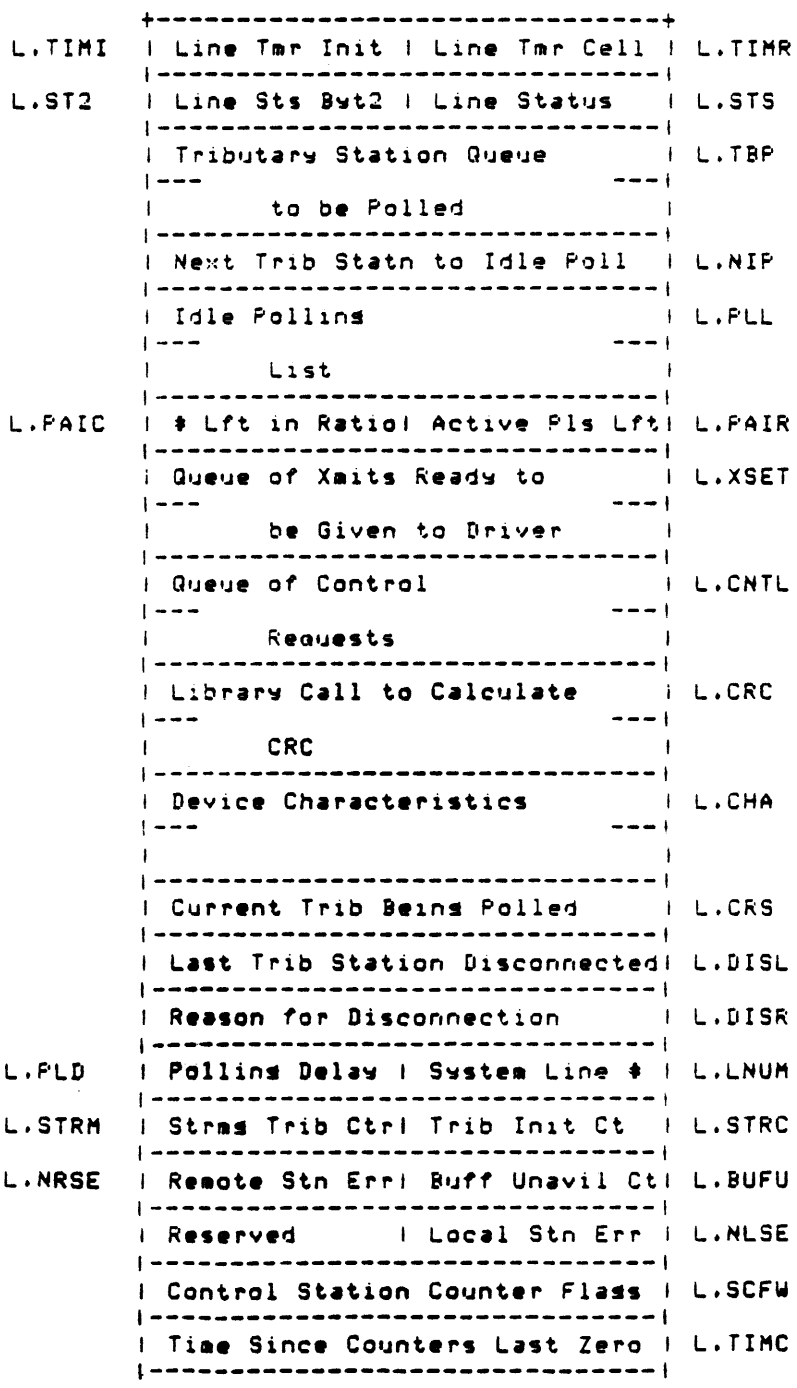
B.1 Introduction

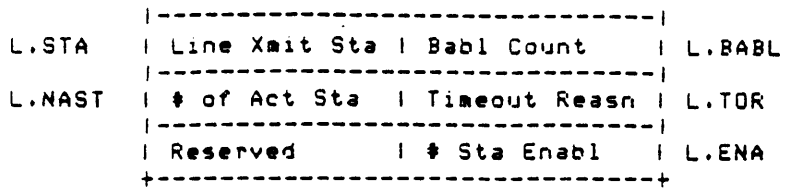
This appendix describes the RSX-DECnet V4.0/2.0 DDCMP process (DCP) data base. Its primary purpose is to provide more detailed descriptions of the DCP data structures.



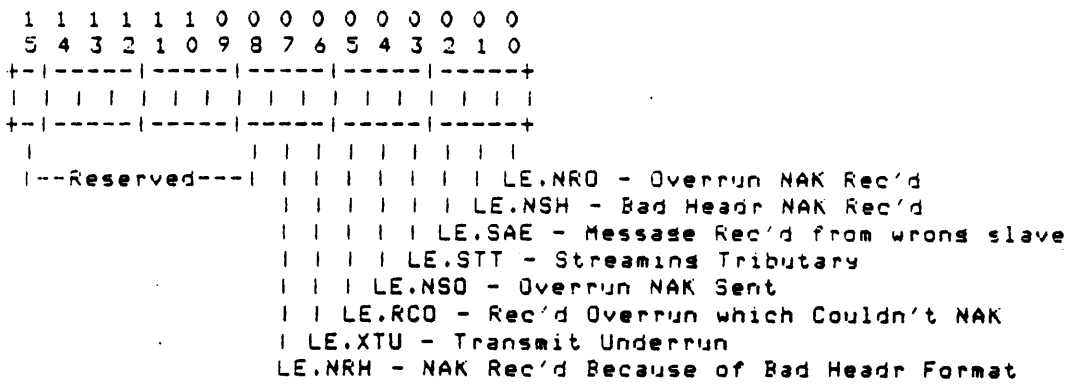
B.2 DCPDF$ - DCP LINE TABLE

The software DDCMP process (DCP) Line Table contains information for control of system lines which which use DCP as the Data Link Control process. Point-to-point lines require only the line table; whereas, multi-point masters require both the line table and tributary station tables.

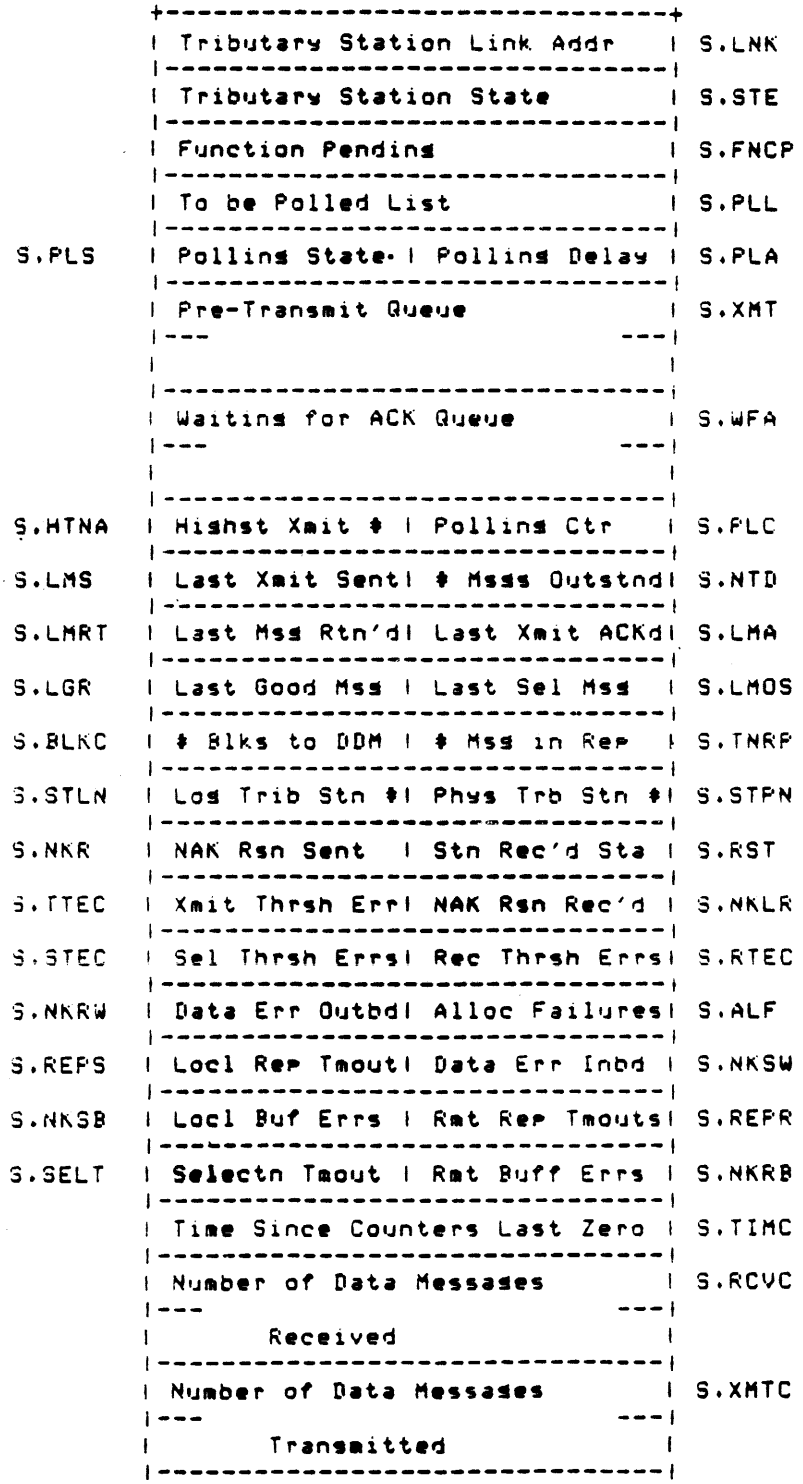


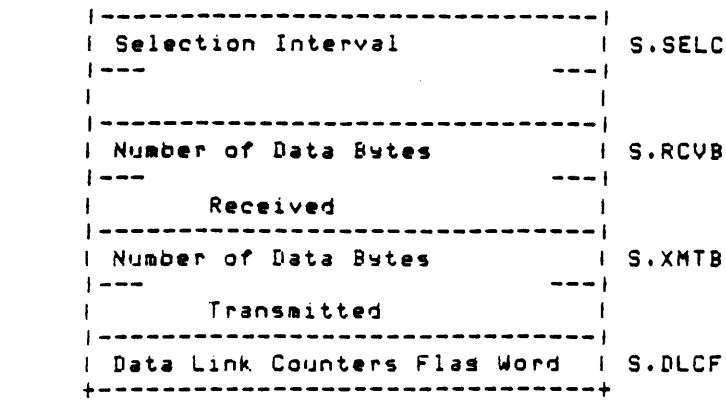


Control Station Counters Flag Word



B.3 DCP TRIBUTARY STATION TABLE





S.DLCF Data Link Counters Flag Word