



Preside Multiservice Data Manager

MDP Data Formats for DPN Reference

241-6001-806

Preside Multiservice Data Manager

MDP Data Formats for DPN

Reference

Publication: 241-6001-806

Document status: Standard

Document version: 15.1RSUP

Document date: August 2004

Copyright © 2004 Nortel Networks.

All Rights Reserved.

Printed in Canada

NORTEL, NORTEL NETWORKS, the globemark design, the NORTEL NETWORKS corporate logo, PRESIDE, DPN, and PASSPORT are trademarks of Nortel Networks. UNIX is a trademark licensed exclusively through X/Open Company Ltd. SUN is a trademark of Sun Microsystems Inc. IBM is a trademark of International Business Machines Corporation. VAX is a trademark of Digital Equipment Corporation.

Publication history

August 2004

15.5RSUP Standard

Commercial availability except for MPE support which will be available in a future release.

Contents

About this document **13**

Who should read this document and why 14

What you need to know 14

How this document is organized 15

What's new in this document 15

Text conventions 15

Related documents 17

Chapter 1

Published Format accounting records **19**

Published Format overview 19

File-naming conventions 20

 Backward compatibility option enabled 20

 Backward compatibility option disabled 21

Field formats 22

Common fields 23

 Common field descriptions 26

Time change records 43

 Time change record field descriptions 44

Optional fields 45

 NUI option 47

 Inter-network option 50

 Original called address option 54

 GAS option 55

 National CUG option 55

 International CUG option 57

- Sensor identifier option 60
- X.75 interface identifier option 61
- Gateway option 61
- Videotex option 62
- Routing class of service option 64
- Extended national address option 65
- Frame Relay option 66
- Accounting meter reference points 79

Chapter 2

DPN-100 BDF accounting records

81

- BDF file names 81
- Common fields 83
 - Common field descriptions 84
- Peak water mark fields 100
- Time change records 104
 - DPN time change record field descriptions 104
- Optional fields 105
 - NUI option 106
 - Free-format NUI option 106
 - Inter-network option 106
 - Original Called Address option 108
 - GAS option 108
 - National CUG option 109
 - International CUG option 110
 - Sensor identifier option 112
 - X.75 interface identifier option 112
 - Gateway option 112
 - Videotex option 112
 - Routing class of service option 113
 - Extended national address option 113
 - Frame Relay options 114
 - Customized accounting information option 123

Chapter 3**DPN statistics records****125**

DPN statistics types quick reference	126
3270HostPAD (subtype 22)	128
3270TermPAD (subtype 20)	131
APIDevice (subtype 19)	134
APILink (subtype 11)	135
BOLDTermPAD (subtype 45)	138
ACPPAD (subtype 76)	140
CallRedirSvr (subtype 32)	141
CallRedirSvrRSI (subtype 33)	142
CallRouterDst (subtype 29)	142
CallRouterGateDst (subtype 40)	143
CallRouterGateSrc (subtype 39)	143
CallRouterSrc (subtype 28)	144
ClusterRemoteDU (subtype 51)	145
DialOutRouting (subtype 36)	148
FrameRelay (subtype 59)	149
FrameRelayDLCIEnhanced (subtype 66)	155
FrameRelayDLCIG33 (subtype 69)	158
FrameRelayDLCIG34 (subtype 73)	162
GASTrtxtUnivGate (subtype 44)	168
ISDNDLCI (subtype 24)	170
ISRB Phase 2 (subtype 68)	172
ITI (subtype 12)	182
ITIEnhanced (subtype 57)	184
LAPDPhysical (subtype 23)	187
LAPDSignalling (subtype 47)	190
LAPDE_Physical (subtype 77)	193
NCSOpAgent (subtype 42)	196
NLLAPBEhanced (subtype 58)	197
NUIDatabaseInt (subtype 43)	207
NUIValidRemSvrRSI (subtype 34)	210
OfficeMaster (subtype 18)	211

HighPerfPE (subtype 75)	211
PE286386 (subtype 30)	214
SABRE (subtype 46)	216
SNALink (subtype 26)	219
SNALogicalUnit (subtype 27)	221
TokenRing (subtype 53)	223
TokenRingDevice (subtype 52)	226
TrunkRMUTP (subtype 25)	229
TrunkRMUTPEnhanced (subtype 64)	233
UTP (subtype 35)	238
UTPEnhanced (subtype 65)	242
NLOverFR (subtype 72)	247
VC (subtype 67)	253
GVCG34 (subtype 74)	257
X25Port (subtype 63)	262
X25MLAgent (subtype 61)	274
X25MLSerPort (subtype 62)	282
X75MLAgentG32 (subtype 71)	293
X75PortG32 (subtype 70)	300

Chapter 4

DPN alarm records 313

Alarm output format 314

Chapter 5

DPN operator command logs 317

Log record format 318

Chapter 6

Outage records 319

DPN outage records 319

Chapter 7	
Availability report records	321
Chapter 8	
File processing audit records	323
Index	327

About this document

This guide provides network operating personnel with a description of the following data types for the specified Nortel Networks switch products:

- DPN-100 accounting and performance
- DPN-100 and Passport outage
- DPN-100 and Passport availability

This data is generated by Nortel Networks switches and transferred to a Management Data Provider (MDP) which converts the data to one, or more, of the following formats:

- Published Format (PF)
- Bulk Data Format (BDF)
- ASCII format

The MDP transfers converted data to customer hosts, as specified during switch configuration on the MDP.

“Supported data conversion formats” (page 13) shows the data conversion format supported for each switch data type.

Table 1
Supported data conversion formats

Data Type	DPN
accounting	PF/BDF
statistics	BDF
logs	BDF

DataType	DPN
alarms	BDF
SCN	
outage	BDF
availability	BDF/ASCII
SRS	

Gen refers to data that is not switch-type specific.

The following topics are discussed in this section:

- “Who should read this document and why” (page 14)
- “What you need to know” (page 14)
- “How this document is organized” (page 15)
- “What’s new in this document” (page 15)
- “Text conventions” (page 15)
- “Related documents” (page 17)

Who should read this document and why

This document is useful for personnel that process accounting and performance data from DPN-100 switches. Network administrators can use this document to locate useful switch and network diagnostic data.

What you need to know

You must have had some exposure to both computers and network concepts. Basic computer literacy is required to operate a Management Data Provider. You must know how to use a keyboard and a mouse, what a menu is, and how to start applications from a menu using a mouse.

A Management Data Provider (MDP) resides on a UNIX workstation. You should be familiar with a UNIX editing facility so that you are able to modify files. You should be familiar with SUN workstations, the UNIX operating system, and X.25 network communications.

Network management knowledge of networks that include Nortel Networks switches is required.

How this document is organized

This document contains the following sections:

- “Published Format accounting records” (page 19) describes the Published Format of DPN accounting records.
- “DPN-100 BDF accounting records” (page 81) describes Bulk Data Format (BDF) DPN accounting records.
- “DPN statistics records” (page 125) describes the Bulk Data Format of DPN statistics records.
- “DPN alarm records” (page 313) describes the Bulk Data Format of DPN alarm records.
- “DPN operator command logs” (page 317) describes the Bulk Data Format of DPN operator command logs.
- “Outage records” (page 319) describes the Bulk Data Format of outage records.
- “Availability report records” (page 321) describes the Bulk Data Format of availability records.
- “File processing audit records” (page 323) describes the fields contained by file processing audit report records.

What’s new in this document

The DPN performance metrics have been documented according to the records generated by the DPNs.

In this release, this document has been modified to contain only DPN-specific information.

Text conventions

This document uses the following text conventions:

- `nonproportional spaced plain type`
Nonproportional spaced plain type represents system generated text or text that appears on your screen.

- **nonproportional spaced bold type**

Nonproportional spaced bold type represents words that you should type or that you should select on the screen.

- *italics*

Statements that appear in italics in a procedure explain the results of a particular step and appear immediately following the step.

Words that appear in italics in text are for naming.

- [optional_parameter]

Words in square brackets represent optional parameters. The command can be entered with or without the words in the square brackets.

- <general_term>

Words in angle brackets represent variables which are to be replaced with specific values.

- UPPERCASE,lowercase

In MDM, uppercase and lowercase letters that appear in UNIX commands and parameters must be matched exactly. The system matches upper and lowercase characters differently.

- |

This symbol separates items from which you may select one; for example, ON|OFF indicates that you may specify ON or OFF. If you do not make a choice, a default ON is assumed.

- ...

Three dots in a command indicate that the parameter may be repeated more than once in succession.

The term absolute pathname refers to the full specification of a path starting from the root directory. Absolute pathnames always begin with the slash (/) symbol. A relative pathname takes the current directory as its starting point, and starts with any alphanumeric character (other than /).

Related documents

This section lists documents to which you can refer, including the following subjects:

- “General” (page 17)
- “DPN” (page 17)

General

241-6001-101 *Preside MDM Engineering Guide*

241-6001-303 *Preside MDM Administrator Guide*

241-6001-309 *Preside MDM Management Data Provider User Guide*

This NTP also contains references to *SunSoft* applications publications, copies of which can be obtained by contacting Sun Microsystems at the following address:

Sun Microsystems Inc.
2550 Garcia Avenue,
Mountain View, CA 94043
United States of America

The SunSoft applications publications referred to in this document are as follows:

- *SPARC, Installing Solaris Software*
- *SunLink HSI/P x.x User's Guide*

DPN

241-1001-100 *DPN-100 Product Overview*

241-1001-115 *DPN-100 Call Redirection System User Guide*

241-1001-320 *DPN-100 Accounting User Guide*

241-2001-332, *DPN-100 Data Spooling Description*

Chapter 1

Published Format accounting records

A Management Data Provider (MDP) can convert DPN-100 accounting records to *Published Format (PF)* or *Bulk Data Format (BDF)* records before transferring them to a Billing customer host. This section describes the Published Format and includes:

- “Published Format overview” (page 19)
- “File-naming conventions” (page 20)
- “Field formats” (page 22)
- “Common fields” (page 23)
- “Time change records” (page 43)
- “Optional fields” (page 45)

Published Format overview

Each record has accounting information of both the calling and called ends. Each record has a variable length format that contains a *common (fixed)* portion, followed by an *optional* portion.

The *optional* portion may contain none, one, or more optional facility sections. An optional facility section contains information (fields) of an optional facility used on a call. An optional facility section for a particular optional facility does not appear in the optional portion of a Published Format accounting record if that optional facility is not used on the call. If none of these optional facilities are used, then the Published Format accounting record contains no optional sections following the fixed portion of the accounting record.

File-naming conventions

Published Format files are DPN-100 accounting files stored in the dump directory. The following sections describe the file-naming conventions for DPN-100 files.

File names for Published Format files are different according to the selected setting of the Published Format backward compatibility option. For more information about the Published Format backward compatibility option, see 241-6001-309 *Preside MDM Management Data Provider User Guide*.

- “Backward compatibility option enabled” (page 20)
- “Backward compatibility option disabled” (page 21)

Backward compatibility option enabled

If the Published Format backward compatibility option is enabled:

- DPN hourly files contain data already appended to the daily file

`Dpn_TPACT<?>.mmddHHMM (opened file)`

`Dpn_TPACT<?>.mmddHH (closed file)`

- the daily file

`TPACT<?>.mmdd`

Note: The daily Published Format file contains DPN accounting records if DPN switches exist in the same network.

where:

mm indicates the month the file was received

dd indicates the day the file was received

HH indicates the hour the file was received

MM indicates the minute the file was received

O indicates the file is open

? is a variable indicating one of

- N a non-duplicate file—the accounting record is unique. This typically occurs for single-ended accounting.
- D a duplicate file—a matching accounting record exists from both ends of the call. This typically occurs for double-ended accounting.
- E a file containing Published Format records from a DPN accounting file that can only be partially converted. View this file using the *pfview* command.
- G a file containing corrupted raw accounting records for DPN. These files contain records that cannot be processed further and have to be analyzed by experts in DPN accounting.

Backward compatibility option disabled

If the Published Format backward compatibility option is disabled the Published Format filenames are as follows:

```
<switch_type>_<dataType>_<date_time>_<nodeID>_<seq#>.  
<?>.<ext>
```

where:

<switch_type> switch_type is *dpn* for DPN-100.

<dataType> data_type is *acc* for accounting.

<date_time> is the date and time the file was opened on the switch, as follows;

- for DPN-100, this field contains YYMMDDhhmmss (YY=year, MM=month, DD=day, hh=hour, mm=minute, ss=seconds).

The MDP published format converter uses the following algorithm to determine 'century' for DPN-100, whose raw switch records contain 2-digit years:

year = YY

YY >= 50 (indicates 20th century (0YYMMDD))

YY < 50 (indicates 21st century (1YYMMDD))

<nodeID> is the *NAMSid* of the DPN-100 switch that generated the data.

<seq#> is a file sequence number assigned by a switch.

? is a variable indicating one of

- N a non-duplicate file—the accounting record is unique. This typically occurs for single-ended accounting.
- D a duplicate file—a matching accounting record exists from both ends of the call. This typically occurs for double-ended accounting.
- E a file containing Published Format records from a DPN accounting file that can only be partially converted. View this file using the *pfview* command.
- G a file containing corrupted raw accounting records for DPN. These files contain records that cannot be processed further and have to be analyzed by experts in DPN accounting.

<ext> is *pf_err* for a file that stores records that cannot be converted successfully (the error file is stored in the appropriate dump directory). This value is *pf* for a Published Format file.

Field formats

Accounting record fields are in one of the following COBOL-readable formats:

PIC X()	Character
COMP	Binary
COMP-3	Packed decimal with two BCD digits per byte and the sign is contained in the right-most four bits of the right-most byte. The valid BCD digits are digits 0 through 9. In the Published Format accounting record, the COMP-3 fields have value H.C (that is, positive) or value H.D (that is, negative) for the sign nibble.

Note: The COBOL COMP field format does not indicate the possible value range for that field. For example, a 9(5) COMP field does not mean that the field can have values up to 99999, and a 9(4) or 9999 COMP field does not mean that the field can have values up to 9999. As used in COBOL programs, these COBOL formats are used to indicate the number of bytes of data to be read for that field or the storage necessary

to store all the possible values of the field. Refer to the descriptions of the fields in this document or other documents, and functional specifications for the valid ranges or formats of the fields.

Common fields

This section describes the *common (fixed) fields* in a Published Format accounting record.

The terms *calling* and *called* are used as part of some of the field names. The term *calling* end refers to that part of the switch interface that initiated the call request. The term *called* end refers to that part of the switch interface that received the call request. For permanent virtual circuits (PVCs), the Master end initiated the call. For switched virtual circuits (SVCs), the calling device initiated the call.

Table 2
Common portion of a Published Format accounting record

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
0	REC_LEN	2	PIC 9999 COMP
2	ACCT_REC_ TYPE	2	PIC 9999 COMP
4	VINTAGE #	1	PIC S9 COMP-3
5	START_DATE	4	PIC S9(7) COMP-3
9	START_TIME	4	PIC S9(7) COMP-3
13	END_TIME	4	PIC S9(7) COMP-3
17	TIME_FLAG	2	PIC 9999 COMP
(Sheet 1 of 4)			

Table 2 (Continued)
Common portion of a Published Format accounting record

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
19	CLG_NAMS_ID	4	PIC 9(5) COMP
23	CLG_PORT_ ID	2	PIC X(2)
25	CLG_NPI	1	PIC S9 COMP-3
26	CLG_ADDR_ LEN	2	PIC 9999 COMP
28	CLG_ADDR	8	PIC S9(15)COMP -3
36	CLG_LCN	2	PIC 9999 COMP
38	CLG_DSE	2	PIC 9999 COMP
40	CLG_SERV_ TYPE	2	PIC 9999 COMP
42	CLG_XMIT_ RECV_WDWS	1	PIC 9999 COMP
		1	
44	CLG_CUST_ID	2	PIC X(2)
46	CLD_NAMS_ID	4	PIC 9(5) COMP
50	CLD_PORT_ID	2	PIC X(2)
52	CLD_NPI	1	PIC S9 COMP-3
53	CLD_ADDR_ LEN	2	PIC 9999 COMP
(Sheet 2 of 4)			

Table 2 (Continued)
Common portion of a Published Format accounting record

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
55	CLD_ADDR	8	PIC S9(15)COMP -3
63	CLD_LCN	2	PIC 9999 COMP
65	CLD_DSE	2	PIC 9999 COMP
67	CLD_SERV_ TYPE	2	PIC 9999 COMP
69	CLD_XMIT_ RECV_WDWS	1	PIC 9999 COMP
		1	
71	CLD_CUST_ID	2	PIC X(2)
73	CALL_ID	4	PIC 9(8) COMP
77	COLLECT_ FLAGS	2	PIC 9999 COMP
79	FACILITY_ FLAGS	2	PIC 9999 COMP
81	PKT_SIZES	1	PIC 9999 COMP
		1	
83	THRPT_ CLASSES	1	PIC 9999 COMP
		1	
85	CLEAR_ CAUSE	2	PIC X(2)

(Sheet 3 of 4)

Table 2 (Continued)
Common portion of a Published Format accounting record

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
87	DIAGNOSTIC	2	PIC 9999 COMP
89	TRANSIT_ DELAY	2	PIC 9999 COMP
91	SEG_SIZE	2	PIC 9999 COMP
93	TDA_RATES	2	PIC 9999 COMP
95	RESETS	4	PIC 9(5) COMP
99	SEG_REC'D	4	PIC 9(8) COMP
103	SEG_XMIT	4	PIC 9(8) COMP
107	DURATION	4	PIC 9(8) COMP
(Sheet 4 of 4)			

Common field descriptions

This section describes the *common (fixed) fields* of a Published Format accounting record:

recLen The length of the accounting record in bytes, which includes the optional facility portion but excludes this *rec_len* field.

recordType This field defines the type of accounting record.

1 = Accounting record

2 = Time change record

vintageNumber This number indicates the version of the Published Format accounting record used. The value of this field is set to 4 for a Published Format accounting record.

BCD Char.	Item	Hex Value
1	Vintage Number	4
2	Sign	C or D

startDate: This is the start date of the period accounted for in the record. It is in the format 0YYMDD. The first character is either 0 or 1; 0 indicates the 20th century (19xx) and 1 indicates the 21st century (20xx).

The MDP published format converter uses the following algorithm to determine 'century' for DPN-100, whose raw switch records contain 2-digit years:

year = YY

YY >= 50 (indicates 20th century (0YYMMDD))

YY < 50 (indicates 21st century (1YYMMDD))

BCD Char.	Item	Hex Value
1	Century	0 or 1
2-3	Year (YY)	00 - 99
4-5	Month (MM)	01 - 12
6-7	Day (DD)	01 - 31
8	Sign	C

startTime: Represents the starting time of the period accounted for in the record. For SVC, *start time* is the time when the call request packet was created. For PVC, *start time* is set by the Master VC when the master call request is formatted. On subsequent accounting records, *start time* corresponds to the end time of the previous record. If Time of Day Accounting (TODA) is used, *start time* (and *end time*) correspond to times set in the schedule.

It is in the format *hmmss0*. The last digit is set to zero.

BCD Char.	Item	Hex Value
1-2	Hour (hh)	00 - 24
3-4	Minute (mm)	00 - 59
5-6	Second (ss)	00 - 59
7	Padding	0
8	Sign	C

endTime: Represents the end of the period accounted for in the record. If Time of Day Accounting (TODA) is used, *end time* in non-final records is the time of the TODA period. For 12-hour accounting, *end time* in non-final records is the time when the 12-hour timer expires at the charged end. On final accounting records, *end time* is the time when the disconnect packet is first received. If Time of Day Accounting (TODA) is used, *end time* (and *start time*) correspond to times set in the schedule.

The *end time* field is in the format *hhmmss0*, similar to *start time*.

timeFlag: This field is currently not used and the value is set to zero.

callingNodeIdentifier: This number uniquely identifies the node, in an autonomous network, where the call originates.

For DPN, this field is the NAMSID. If the switch is NM equipped, the NAMSID is the RID and can have a value in the range 1 to 126. If the switch is AM or RM equipped, this value can be any integer between 256 and 49151.

callingPortIdentifier: A 16-bit field that identifies the physical port number where the call originates.

If the physical port is on a DPN-100, the Port Number will be identified by a PE Number, PI Number, and a Port Number. In the case of X.75 multi-link, only the PE Number is specified (the PI and PO will be set to zero). For details, refer to “Port number (DPN-100)” (page 29).

Figure 1
Port number (DPN-100)

15 PO Number					9 PI Number					4 PE Number					0

callingNpi: This field is the Numbering Plan Indicator that indicates the numbering plan used in the calling address.

0 = X.121 numbering plan

1 = E.164 numbering plan

other = undefined

BCD Char.	Item	Hex Value
1	Numbering plan index	0 - 1
2	Sign	C

callingAddressLength: The length of the callingAddress field in bytes.

callingAddress: The number of digits in the calling address (DNA) field. The maximum is 15. It also contains the full address field of the calling address (DNA), including the DNIC (first four digits) for X.121 or Country Code for E.164 address. The field is left-justified with trailing zeroes.

BCD Char.	Item	Hex Value
1-15	Digits	0 - 9
16	Sign	C

callingDlci: The Logical Channel Number of the calling end. The value ranges from 0 to 4095. For frame relay service, the valid range is 16 to 1007.

callingDse: This is the Data Service Exchange (DSE) of the calling end. The value ranges from 0 to 255. For transit/incoming X.75 SVC calls, the calling DSE information is valid only if the calling tariff is absent from the accounting record. DSE is defined in the DNA component on DPN-100.

callingServiceType: The service supported by the port used by the calling end-user. The value ranges from 0 to 255. For transit/incoming X.75 SVC calls, the calling service type information is valid only if the calling tariff is absent from the accounting record. This number is defined as accountClass in the DNA component on DPN-100.

callingReceiveWindow and **callingSendWindow:** This two-byte field consists of the values for the calling end local transmit and receive window sizes, arithmetically combined. The low-order byte contains the value for the calling end local transmit window size and the high-order byte contains the value for the calling end local receive window size. Both the calling end local transmit and receive window sizes have values ranging from 1 to 7. For SVC, this field is only populated for X.25 and X.75 services. All other services have the value zero. For PVC, this field is populated as provided in the PVC envelope, for all services.

Note: For frame relay service, these values are not relevant and should be ignored. These fields do not apply because frame relay uses the lightweight virtual circuit instead of the flow-controlled virtual circuit.

callingCustomerIdentifier: This field is used as an optional identifier for the calling end of the call in the Customer Network Management (CNM) and Virtual Private Network environment. The value in this field is taken from the Customer Network Management Identifier (CID) field in the DNA-CUG

envelope for the access line. The value zero is the default value reserved for all ports that are not associated with a Virtual Private Network and for which CNM is not provided.

calledNodeIdentifier: This number uniquely identifies the node, in an autonomous network, where the call terminates.

For DPN, this field is the NAMSID. If the switch is NM equipped, the NAMSID is the RID and can have a value in the range 1 to 126. If the switch is AM or RM equipped, this value can be any integer between 256 and 49151.

calledPortIdentifier: A 16-bit field that identifies the physical port number where the call terminates.

If the physical port is on a DPN-100, the Port Number will be identified by a PE Number, PI Number, and a Port Number. In the case of X.75 multi-link, only the PE Number is specified (the PI and PO will be set to zero). For details, refer to “Port number (DPN-100)” (page 29).

calledNpi: This field is the Numbering Plan Indicator that indicates the numbering plan used in the called address. The field format is similar to the *calling NPI* field.

0 = X.121 numbering plan

1 = E.164 numbering plan

other = undefined

calledAddressLength: This is the length of the calledAddress field in bytes.

calledAddress: The number of digits in the called address (DNA) field. The maximum is 15. It also provides the full address field of the called address (DNA), including the DNIC (first four digits) for X.121 or

Country Code for E.164 address. The field is left-justified with trailing zeroes. The field has a maximum of 15 digits. The field format is similar to the *calling address* field.

calledDlci: The Logical Channel Number of the called end. The value ranges from 0 to 4095. For frame relay service, the range is 16 - 1007.

calledDse: This is the Data Service Exchange (DSE) of the called end. The value ranges from 0 to 255. For outgoing/transit X.75 SVC calls, the called DSE information is valid only if the called tariff is absent from the accounting record. DSE is defined in the DNA component on DPN-100.

calledServiceType: The service supported by the port used by the called end-user. The value ranges from 0 to 255. For outgoing/transit X.75 SVC calls, the called service type information is valid only if the called tariff is absent from the accounting record. This number is defined in the DNA_CUG envelope as accountClass.

calledReceiveWindow and **calledSendWindow:** This two-byte field consists of the values for the called end local transmit and receive window sizes, arithmetically combined. The low-order byte contains the value for the called end local transmit window size and the high-order byte contains the value for the called end local receive window size. Both the called end local transmit and receive window sizes have values that range from 1 to 7. For SVC, this field is only populated for X.25 and X.75 services. All other services have the value zero. For PVC, this field is populated as provided in the PVC envelope, for all services.

Note: For frame relay service, these values are not relevant and should be ignored. These fields do not apply because frame relay uses the lightweight virtual circuit instead of the flow-controlled virtual circuit.

calledCustomerIdentifier: This field is used as an optional identifier for the called end of the call in the Customer Network Management (CNM) and Virtual Private Network environment. The value in this field is taken from the provisioning data for the access line, or more specifically, from the Customer Network Management Identifier (CID) field in the DNA-

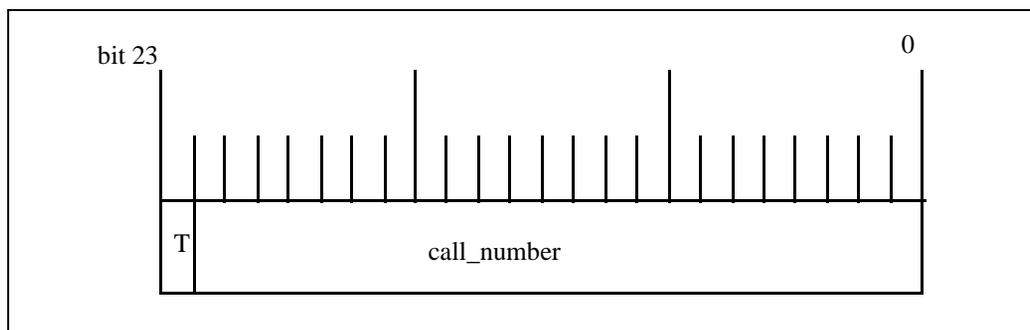
CUG envelope for the access line. The value zero is the default value reserved for all ports that are not associated with a Virtual Private Network and for which CNM is not provided.

callReferenceNumber: This is a three-byte number generated by the switch, or taken from the call identifier utility in the X.75 call request packet. The same call reference number is stored in the Published Format accounting record at both ends of the call. It is used as one of the fields in matching duplicate records generated at each end of the call or for reconciling records between networks. See “Call reference number generated by a PE on RM/AM and processor (DPN-100 only)” (page 33) for the format of the call reference number.

- T = 0 for SVC or 1 for PVC.
- call_number is initialized to a semi-random value (based on the time/date) when the service PE/processor first comes up as follows:
call_number = CENTURY/YEAR XOR MONTH/DAY XOR HOUR/MINUTES.

The most significant byte of call_number is set to the SECONDS of the current time/date on the PE/processor. The call number is incremented by one for every outgoing call.

Figure 2
Call reference number generated by a PE on RM/AM and processor (DPN-100 only)



collectFlags: This is a field containing 15 one-bit flags. The flags, starting from bit 0 (the least significant bit), are as follows:

1. BIT 0 - WHO CLEARED (flags_calledCleared)
2. BIT 1 - WHO GENERATED (flags_calledGenerated)
3. BIT 2 - UNIQUE (flags_unique)
4. BIT 3 - CALL REACHED DATA TRANSFER (flags_reachedDataTransfer)
5. BIT 4 - FIRST RECORD FOR THIS CALL (accountingRecord_first)
6. BIT 5 - LAST RECORD FOR THIS CALL (accountingRecord_last)
7. BIT 6 - TERMINATOR IS GATEWAY (gateWay_terminator)
8. BIT 7 - ORIGINATOR IS GATEWAY (gateWay_terminator)
9. BIT 8 - COLLECTION FOR BILLING (CALLING END) (callingCollectReason_bill)
10. BIT 9 - COLLECTION FOR TEST (CALLING END) (callingCollectReason_test)
11. BIT 10 - COLLECTION FOR STUDY (CALLING END) (callingCollectReason_study)
12. BIT 11 - COLLECTION FOR AUDIT (CALLING END) (callingCollectReason_audit)
13. BIT 12 - COLLECTION FORCED (CALLING END) (callingCollectReason_force)
14. BIT 13 - CALL DEFLECTION
15. BIT 14 - SPARE
16. BIT 15 - (not used - always set to zero)

These one-bit flags have the values 1 or 0 depending whether they are turned on or off. The values of the 15 one-bit flags are explained as follows:

1 WHO CLEARED:

- 0 - Cleared from calling end
- 1 - Cleared from called end

The callCleared bit is initialized to 0 on the calling end and initialized to 1 on the called end. If the call is cleared on the local switch, that switch retains its value for the callCleared bit and sends a disconnect message to the other end. When a switch receives a disconnect message, the switch takes the value for the callCleared flag from the remote end.

2 WHO GENERATED:

For SVC:

- 0 - Generated from calling end
- 1 - Generated from called end

For PVC:

- 0 - Generated from master end
- 1 - Generated from slave end

3 UNIQUE:

- 0 - Possible duplicate, both ends may have generated the accounting record
- 1 - No duplicates, only one end has generated the accounting record

4 CALL REACHED DATA TRANSFER:

- 0 - Call does not reach data transfer at both ends
- 1 - Call reached data transfer at both ends

5 FIRST RECORD FOR THIS CALL:

- 0 - Not first record for the call
- 1 - First record for the call

6 LAST RECORD FOR THIS CALL:

- 0 - Not last record for the call
- 1 - Last record for the call

- 7** TERMINATOR IS GATEWAY:
 - 0 - Terminator is not Gateway
 - 1 - Terminator is Gateway

- 8** ORIGINATOR IS GATEWAY:
 - 0 - Originator is not Gateway
 - 1 - Originator is Gateway

- 9** COLLECTION FOR BILLING (CALLING END):
 - 0 - Not collected for billing purposes
 - 1 - Collected for billing purposes

- 10** COLLECTION FOR TEST (CALLING END):
 - 0 - Not collected for test purposes
 - 1 - Collected for test purposes

- 11** COLLECTION FOR STUDY (CALLING END):
 - 0 - Not collected for study purposes
 - 1 - Collected for study purposes

- 12** COLLECTION FOR AUDIT (CALLING END):
 - 0 - Not collected for audit purposes
 - 1 - Collected for audit purposes

- 13** COLLECTION FORCED (CALLING END):
 - 0 - collection not forced by calling end
 - 1 - collection forced by calling end

- 14** CALL DEFLECTION
 - 0 - Call not deflected
 - 1 - Call deflected

- 15** SPARE:
 - Bit 14 is a spare reserved for future use. It is set to zero.

facilityFlags: This is a field containing 15 one-bit flags. The flags, starting from bit 0 (the least significant bit), are as follows:

1. BIT 0 - REVERSE CHARGING (flags_calledEndCharging)
2. BIT 1 - FAST SELECT
3. BIT 2 - RESTRICTED RESPONSE
4. BIT 3 - PRIORITY(flags_highPriority)
5. BIT 4 - PERMANENT VIRTUAL CIRCUIT(flags_pvcType)
6. BIT 5 - USER DATA ON CALL
REQUEST(flagsuserDataInCallReq)
7. BIT 6 - HUNT GROUP(flags_hunted)
8. BIT 7 - CALL REDIRECTION(flags_redirected)
9. BIT 8 - COLLECTION FOR BILLING (CALLED
END)(calledCollectReason_bill)
10. BIT 9 - COLLECTION FOR TEST (CALLED
END)(calledCollectReason_test)
11. BIT 10 - COLLECTION FOR STUDY (CALLED
END)(calledCollectReason_study)
12. BIT 11 - COLLECTION FOR AUDIT (CALLED
END)(calledCollectReason_audit)
13. BIT 12 - CALLED END COLLECTION REASON
VALID(flags_collReasonValid)
14. BIT 13 - COLLECTION FORCED (CALLED
END)(calledCollectReason_force)
15. BIT 14 - SPARE
16. BIT 15 - (Not used - always set to zero)

These one-bit flags have the values 1 or 0 depending whether they are turned on or off. The values of the 15 one-bit flags are as follows:

- 1 REVERSE CHARGING:
 - 0 - Charged to calling end
 - 1 - Charged to called end

- 2 FAST SELECT:**
 - 0 - Not a fast select call
 - 1 - A fast select call

- 3 RESTRICTED RESPONSE:**
 - 0 - Restricted response option not used
 - 1 - Restricted response option used

- 4 PRIORITY:**
 - 0 - Normal priority
 - 1 - High priority

- 5 PERMANENT VIRTUAL CIRCUIT:**
 - 0 - SVC call
 - 1 - PVC call

- 6 USER DATA ON CALL REQUEST:**
 - 0 - No user data on call request
 - 1 - User data on call request

- 7 HUNT GROUP:**
 - 0 - Call has not been hunted
 - 1 - Call has been hunted

- 8 CALL REDIRECTION:**
 - 0 - Call not redirected
 - 1 - Call redirected

- 9 COLLECTION FOR BILLING (CALLED END):**
 - 0 - Not collected for billing purposes
 - 1 - Collected for billing purposes

- 10 COLLECTION FOR TEST (CALLED END):**
 - 0 - Not collected for test purposes
 - 1 - Collected for test purposes

11 COLLECTION FOR STUDY (CALLED END):

- 0 - Not collected for study purposes
- 1 - Collected for study purposes

12 COLLECTION FOR AUDIT (CALLED END):

- 0 - Not collected for audit purposes
- 1 - Collected for audit purposes

13 CALLED END COLLECTION REASON VALID:

- 0 - Called end collection reasons are not valid
- 1 - Called end collection reasons are valid

14 COLLECTION FORCED (CALLED END):

- 0 - collection not forced by called end
- 1 - collection forced by called end

15 SPARES

Bit 14 is a spare reserved for future use. It is set to zero.

callingCalledPktSize and **calledCallingPktSize**: This two-byte field consists of the values of the packet size sent from the calling DTE to the called end and packet size sent from the called DTE to the calling end, arithmetically combined. The low-order byte contains the value of the packet size sent from the calling DTE to the called end. The value is the base 2 logarithm of the actual packet size. The high-order byte contains the value of the packet size sent from the called DTE to the calling end. The value is the base 2 logarithm of the actual packet size.

Note: For frame relay service, these values are not relevant and should be ignored. These fields do not apply because frame relay uses the lightweight virtual circuit instead of the flow-controlled virtual circuit.

callingCalledThruput and **calledCallingThruput**: This field contains the throughput classes of packets sent from the calling DTE to the called end, and vice versa, arithmetically combined. The low-order byte

contains the throughput class of the packet sent from the calling DTE to the called end and the high-order byte contains the throughput class of the packet sent from the called DTE to the calling end.

Note: For frame relay service, these values are not relevant and should be ignored. These fields do not apply because frame relay uses the lightweight virtual circuit instead of the flow-controlled virtual circuit.

Refer to “Throughput classes” (page 40) for throughput classes.

Table 3
Throughput classes

Value of low-order or high-order byte	Throughput class (bit/s)
0 - 2	Reserved
3	75
4	150
5	300
6	600
7	1200
8	2400
9	4800
10	9600
11	19200
12	48000
13	64000
14 - 15	Reserved

clearCause: This a clear cause indicator. It indicates the reason for a call termination. For additional details, refer to 241-1001-509 *DPN-100 Call Clear, Reset and Diagnostic Codes*.

diagnosticCode: This a clear diagnostic indicator. It is set to zero (0) on non-final records, and to the network diagnostic code of the call clear on the final record. For additional details, refer to 241-1001-509 *DPN-100 Call Clear, Reset and Diagnostic Codes*.

transitDelay: This field is to be used in the future to indicate transit delays. The value is currently set to zero.

segmentSizeIndex: This field gives the segment size used on the call. The value in this field is the base 2 logarithm of the actual segment size.

tdaRates: The field is currently not used and the value is set to zero.

localUserResets: Count of the customer-caused reset request packets, transmitted and received. The value ranges from 0 to 65535.

localEgressSegments/Frames: Number of Egress data segments/frames received from the network. The value ranges from 0 to 16777215. For all calls, the segments/frames received field in the accounting record is calculated for the data received in the data packets. For all calls, the segments/frames received field in the accounting record is calculated for the data received in the *interrupt* and *interrupt confirmation* packets. Only for Fast Select calls (both restricted and unrestricted response), the segments/frames received field in the accounting record is calculated for the data received in *call request*, *call confirm (accept)*, and *clear request* packets. Non-Fast Select calls do not count the segments/frames in these

three packets. For all the packets mentioned above, the calculation of segments/frames is performed in a common manner (based on the segment size index).

For *best effort delivery* services such as frame relay, the count is recorded at the entry and exit points to and from the network.

localIngressSegments/Frames: Number of Ingress data segments/frames sent to the network. The value ranges from 0 to 16777215. For all calls, the segments/frames sent field in the accounting record is calculated for the data sent in the data packets. For all calls, the segments/frames sent field in the accounting record is calculated for the data sent in the *interrupt* and *interrupt confirmation* packets. Only for Fast Select calls (both restricted and unrestricted response), the segments/frames sent field in the accounting record is calculated for the data sent in *call request*, *call confirm (accept)*, and *clear request* packets. Non-Fast Select calls do not count the segments/frames in these three packets. For all the packets mentioned above, the calculation of segments/frames is performed in a common manner (based on the segment size index).

For *best effort delivery* services such as frame relay, the count is recorded at the entry and exit points to and from the network.

elapsedTime: Duration of the call in 0.1 second increments. This time interval is started when the call request is processed. If the call does not reach the data transfer stage, *elapsed time* is set to zero. *Elapsed time* represents a true measure of the accounting period. It is measured independently of *start time* and *end time*. Since *start time* and *end time* can be changed externally by adjusting the clock, *elapsed time* can be shorter or longer than the difference between *start time* and *end time*. This difference can show as much as 60 seconds on accounting records for the VCs that are idle (carry no traffic) and when TODA is used and is caused, in this case, by an internal TODA detection mechanism. When the 12 hour accounting period is complete, *elapsed time* is exact or differs by a few seconds from 12 hour duration.

Time change records

Time change records contain fields that describe events caused by a change of time on the switch. When the time on the switch is adjusted by 2 seconds, or more, the time change record is issued and sent as part of the accounting stream. The time change record is used to keep track of clock changes on the switch and to explain potential inconsistencies in accounting records.

The *Time change record fields* in a Published Format accounting record have the following format:

Table 4
Time change records

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
0	REC_LEN	2	PIC 9999 COMP
2	ACCT_REC_TYPE	2	PIC 9999 COMP
4	VINTAGE #	1	PIC S9 COMP-3
5	OLD_DATE	4	PIC S9(7) COMP-3
9	OLD_TIME	4	PIC S9(7) COMP-3
13	NEW_DATE	4	PIC S9(7) COMP-3
17	NEW_TIME	4	PIC S9(7) COMP-3
21	NAMS_ID	4	PIC 9(5) COMP

Time change record field descriptions

This section describes the *Time change record fields* of a Published Format accounting record:

recLen: The length of the accounting record in bytes, which includes the optional facility portion but excludes this *rec_len* field.

recordType: This field defines the type of record. This field is set to 2 for time change records.

vintageNumber: The version of the Published Format accounting record used. The value of this field is set to 4 for a Published Format accounting record.

BCD Char.	Item	Hex Value
1	Vintage Number	4
2	Sign	C or D

oldDate: The date on the switch before the time was adjusted. It is in the format *OYYMDD*. The first character is either 0 or 1; 0 indicates the 20th century and 1 indicates the 21st century.

Note: The MDP published format converter uses the following algorithm to determine 'century' for DPN-100, whose raw switch records contain 2-digit years:

year = YY

YY >= 50 (indicates 20th century (*OYYMMDD*))

YY < 50 (indicates 21st century (*1YYMMDD*))

oldTime: The time on the switch before the time was adjusted.

newDate: The date on the switch after the time was adjusted. It is in the format *OYYMDD*. The first character is either 0 or 1; 0 indicates the 20th century and 1 indicates the 21st century.

Note: The MDP published format converter uses the following algorithm to determine 'century' for DPN-100, whose raw switch records contain 2-digit years:

year = YY

YY >= 50 (indicates 20th century (0YYMMDD))

YY < 50 (indicates 21st century (1YYMMDD))

newTime: The time on the switch after the time was adjusted.

Accounting records are generated if the time change skips over the time set in Time of Day Accounting (TODA). For example, if TODA is set to 15:00 hours and the time change goes from 14:59 to 15:05, accounting records are generated. The TODA time is defined in the *collectionTimes* field of the *Collector/Accounting* component.

namsId: The NAMS ID of the switch that issued the time change record. The value of the NAMS ID is provisioned on the switch.

Optional fields

If an *optional* facility is used on a call, the information on the optional facility is captured in the optional facility section of the accounting record. If the optional facility is not used on the call, then that particular optional facility section will not appear in the accounting record for that call.

Every optional facility has a two-byte *code/length* field in the first two bytes. The low-order byte of this field contains the *length* value for the optional facility section and the high-order byte contains the *code* value. The *length* value gives the length of the optional facility section, excluding this *code/length* field. The *code* value identifies the optional facility section.

The terms *calling* and *called* are used as part of some of the optional field names. The term *calling* end refers to that part of the switch interface that initiated the call request. The term *called* end refers to that part of the switch interface that received the call request. For PVCs, the Master end initiated the call. For SVCs, the calling device initiated the call.

“Optional facility section code values” (page 46) lists the optional facilities selections and the code values that identify the optional facility sections.

Table 5
Optional facility section code values

Optional facility section	Code value
NUI	1
Inter-network	2
Original Called Address	3
GAS	4
National CUG	5
International CUG	6
Sensor identifier	7
X.75 interface identifier	8
Gateway type	9
Videotex	10
Routing class of service (RCOS)	11
Extended national Address	12
Frame relay Accounting - Local	13
Frame relay Accounting - Remote	14

The optional facility sections may appear in the accounting record in any order. The user’s software that processes the accounting records can identify the optional facility sections that need to be processed using the *code* values. The user’s software should be coded to ignore the optional facility sections with unwanted or unknown *code* values, if they are present in the accounting record, by skipping over the number of bytes specified by the *length* value. In this way, new optional facility sections can be added to the accounting record in future, without impacting the user’s software.

NUI option

The *NUI* option contains the information on the Network User Identification used on the call. There are two types of NUI: DPN-100 NUI and Calling Card NUI. The DPN-100 NUI is a character field of variable length. Currently, the maximum length is 8. The length of the Calling Card NUI is 8. The NUI Optional Facility Section has the following format:

Table 6
NUI optional facility section

Offset (bytes)	DPN-100 field names	Field size (bytes)	COBOL format
0	CODE/LENGTH	2	PIC 9999 COMP
2	NUI_TYPE	1	PIC S9 COMP-3
3	NUI	n	PIC X(n)
			or
		8	PIC S9 (15) COMP-3
	NUI_TYPE		COBOL
	Value	Type of NUI	Format
	1	DPN-100 NUI	PIC X(n)
	2	Calling card NUI	PIC 9(15) COMP-3
	3	Suppressed NUI	(no NUI value)

Note: In the above table, n is the number of DPN-100 NUI characters, where $n = \text{LENGTH} - 1$.

code/length: This field contains a *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length* value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section. The value of LENGTH is:

1 + n (for DPN-100 NUI), or
9 (for Calling Card NUI), or
1 (for Suppressed NUI).

nuiType: This field identifies the type of Network User Identification field that follows this field:

1 - DPN-100 NUI
2 - Calling Card NUI
3 - Suppressed NUI

BCD Char.	Item	Hex Value
1	NUI Type	1, 2, 3
2	Sign	C

nui: This field contains a DPN-100 NUI or a Calling Card NUI. The DPN-100 NUI is a variable length character string containing the exact number of characters of the network user identifier. Currently, the maximum length for the DPN-100 NUI is 8. Suppressed NUI has a length of zero.

The Calling Card NUI has the following format. A value of H.00 is placed in the first byte of the NUI field in the accounting record. This is an invalid DPN-100 NUI character, and signals that the information in the remaining 7 bytes of the NUI field consists of the Calling Card NUI information.

Table 7
Calling Card NUI

BCD Char.	Item	Hex Value
1	Padding	0
2-15	Digits	0-9
16	Sign	C (all valid digits) D (has invalid digits)

Table 8
Nibble description

0-1	H.00	(signals Calling Card information in the following bytes)
2		Indicates whether NUIOP is in effect. Possible Values: 0 = not used 1 = NUIOP not in effect 2 = NUIOP in effect
3		Indicates what information is present in nibbles 4-5. Possible Values: 0 = not used 1 = CCSAN present 2 = Error Code present 3 = CSDI present 4 = PSDI present 5 = PIN restricted
4-5		Contents depend on the contents of nibble 3. See "Nibble 4-5 description" (page 50)
6-15		Indicates Billing. Value is a 10-digit billing number

Table 9
Nibble 4-5 description

Nibble 3 content	Nibble 4-5 content	Description
0	00	Padding (not used)
1	two digits	CCSAN
2	01	Unexpected Component Sequence
	02	Unexpected Data Value
	03	Unavailable Network Resource
	04	Missing Record
	05	Reply Overdue
	06	Data Unavailable
	3	01
03		Service Denial on CCAN
4	02	Service Denial on PIN due to threshold exceeded
	03	Service Denial on PIN due to nonpayment
5	01	PIN restricted

Inter-network option

The *Inter-network* option contains the additional accounting information related to inter-network calls. The information includes: Gateway Identifiers, Tariffs, Clear Network Identifier, and Transit Network Identifier.

The number of TNICs, Transit Network Identifiers, that appear in this section is variable. It will be the exact number specified in the TOTAL_TNICS field. The maximum number of TNIC values is 10, contained in TNIC(1) to TNIC(10). No TNIC fields are included in the accounting record if the TOTAL_TNICS has the value zero, which means that there are no TNICs.

Table 10
Inter-network optional facility section

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol Format
0	CODE/LENGTH	2	PIC 9999 COMP
2	INCOMING_GAT EWAY_ID	2	PIC 9999 COMP
4	OUTGOING_GAT EWAY_ID	2	PIC 9999 COMP
6	CLG_TARIFF	2	PIC S9(3) COMP-3
8	CLD_TARIFF	2	PIC S9(3) COMP-3
10	CNIC	3	PIC S9(4) COMP-3
13	TOTAL_TNICS	2	PIC 9999 COMP
15	TNIC(1)	3	PIC S9(4) COMP-3
.	.	.	.
.	.	.	.
.	.	.	.
	TNIC(n)	3	PIC S9(4) COMP-3

code/length: This field contains a *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length*

value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section. The value of $LENGTH = 13 + (3*N)$ bytes.

incomingGatewayId: This field identifies the incoming (calling) Gateway ID on an X.75 gateway call. The incoming (calling) gateway is where a call request packet is received from another network. The value ranges from 0 to 9999.

outgoingGatewayId: This field identifies the outgoing (called) Gateway ID on an X.75 gateway call. The outgoing (called) gateway is where a call request packet is sent to another network. The value ranges from 0 to 9999.

clgTariff: This field contains the value of the calling tariff utility supplied in the incoming call request packet from the X.75 link. This field has the following format:

BCD Char.	Item	Hex Value
1-2	Primary Subclass Code	00-31
3	Secondary Subclass Code	0-9
4	Sign indicator	C (Calling Tariff present) D (Calling Tariff not present)

cldTariff: This field contains the value of the called tariff utility supplied in the call accept packet or call clear packet (in direct response to the call packet) from the X.75 link. This field is in the same format as the *clgTariff* field.

cnic: This field contains the Clearing Network Id Code. The value ranges from 0000 to 9999.

BCD Char.	Item	Hex Value
1	Padding	0
2-5	Digits	0-9
6	Sign indicator	C (CNIC present) D (CNIC not present)

totalTnics: Indicates the total number of TNICs, Transit Network Identifiers, in an international call. The value ranges from 1 to 10.

tnic: The transit network identifier. There are 1 to 10 identifiers depending on the value of total TNICs. The value ranges from 0000 to 9999.

BCD Char.	Item	Hex Value
1	Padding	0
2-5	Digits	0-9
6	Sign	C

Original called address option

The *Original called address* option provides the original called address if the call was redirected or hunted. The Original called address has the following format:

Table 11
Original called address optional facility section

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
0	CODE/LENGTH	2	PIC 9999 COMP
2	ORIG_NPI	1	PIC S9 COMP-3
3	ORIG_ADDR_LE N	2	PIC 9999 COMP
5	ORIG_ADDR	8	PIC S9(15)COMP -3

code/length: This field contains a *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length* value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section. The value of LENGTH = 11.

origNpi: This field is the Numbering Plan Indicator to indicate the numbering plan used in the original called address. This field has the same format as the *callingNpi* field.

origAddrLen: This indicates the length of the address (DNA) in the *origAddr* field. The maximum is 15.

origAddr: Indicates the original called address (called address before the called DTE address was modified). The actual called DTE address is recorded in the *calledAddress* field. This field has the same format as the *calledAddress* field.

GAS option

The *GAS* option provides information on the *GAS*, Gateway Access Service, for the Transtext Universal Gateway used particularly by the BOC. The 52 bytes of *GAS* information in the *GAS* optional facility section of the Published Format accounting record are extracted directly from the 52 bytes of *GAS* optional facility data that appear in the Published Format accounting record when the facility is used.

Table 12
Optional gas facility section

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
0	CODE/LENGTH	2	PIC 9999 COMP
2	GAS data	52	PIC X(52)

Note: The value of LENGTH = 52.

code/length: This field contains a *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length* value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section.

gasData: A 52-byte data field that provides information on the Gateway Access Service (*GAS*) for the Transtext Universal Gateway.

National CUG option

The *National CUG* option contains the *CUG* Interlock code of the national *CUG* where the call is established.

Table 13
National CUG optional facility section

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
0	CODE/LENGTH	2	PIC 9999 COMP
2	CUG_ INTERLOCK_ CODE	4	PIC 9(5) COMP
6	CUG_FLAGS	2	PIC 9999 COMP

code/length: This field contains a *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length* value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section.

cugInterlockCode: This field contains the CUG Interlock Code of the CUG where the call is established. The value ranges from 0 to 65535.

cugFlags: This is a two-byte binary field that contains four one-bit flags as follows:

BIT 0	- Outgoing Access(cugFlags_outgoingAccess)	(OA)
BIT 1	- IA/MA Bits Valid(cugFlags_validIaMaBits)	(VA)
BIT 2	- Incoming Access(cugFlags_incomingAccess)	(IA)
BIT 3	- CUGs Matched(cugFlags_cugsMatch)	(MA)

Outgoing Access (OA) This field specifies if the originator specified Outgoing Access (OA) on a call from a Closed User Group. Values:

- 0 indicates OA is NOT used.
- 1 indicates OA is used.

IA/MA Bits Valid (VA) This field indicates if the IA/MA bits are valid or not. Values:

- 0 indicates IA/MA bits are NOT valid.
- 1 indicates IA/MA bits are valid.

Incoming Access (IA) This field specifies if the destination of a call from a Closed User Group has Incoming Access (IA). Values:

- 0 indicates destination does NOT have IA.
- 1 indicates destination has IA.

CUGs Matched (MA) This field specifies if the CUGs matched at the destination of a call from a Closed User Group. Values:

- 0 indicates CUGS do NOT match.
- 1 indicates CUGs do match.

Note 1: The value for CODE is 5 to indicate the NCUG optional facility section.

Note 2: The value for LENGTH is 6 to indicate that six bytes of data (*cugInterlockCode* and *cugFlags*) follow the CODE/LENGTH field.

Note 3: The value for *cugInterlockCode* is extracted from the same field in the switch national CUG optional facility section.

Note 4: The range for *cugInterlockCode* is 0 to 65535.

International CUG option

The *International CUG* option contains the CUG Interlock Code, along with the DNIC, of the international CUG where the call is established.

Table 14
International CUG Optional facility section

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
0	CODE/LENGTH	2	PIC 9999 COMP
2	CUG_DNIC	3	PIC S9(4) COMP-3
5	CUG_ INTERLOCK_ CODE	4	PIC 9(5) COMP
9	CUG_FLAGS	2	PIC 9999 COMP

code/length: This field contains a *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length* value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section.

cugDnic: This field contains the DNIC of the International CUG where the call is established. It has the same format and value ranges as TNIC.

cugInterlockCode: This field contains the CUG Interlock Code of the CUG where the call is established. The value ranges from 0 to 65535.

cugFlags: This is a two-byte binary field that contains four one-bit flags as follows:

- 1 BIT 0 - Outgoing Access (OA)
- 2 BIT 1 - IA/MA Bits Valid (VA)
- 3 BIT 2 - Incoming Access (IA)
- 4 BIT 3 - CUGs Matched (MA)

These one-bit flags have the values 1 or 0 as follows:

- 1 **Outgoing Access (OA)**(icugFlags_outgoingAccess)
This field indicates if the originator specified Outgoing Access (OA) on a call from a Closed User Group:
0 indicates OA is NOT used
1 indicates OA is used
- 2 **IA/MA Bits Valid (VA)**(icugFlags_validIaMaBits)
This field indicates if the IA/MA bits are valid:
0 indicates IA/MA bits are not valid
1 indicates IA/MA bits are valid
- 3 **Incoming Access (IA)**(icugFlags_incomingAccess)
This field specifies if the destination of a call from a Closed User Group has Incoming Access (IA):
0 indicates destination does NOT have IA
1 indicates destination has IA.
- 4 **CUGs Matched (MA)**(icugFlags_cugsMatch)
This field specifies if the CUGs matched at the destination of a call from a Closed User Group:
0 indicates CUGs do NOT match
1 indicates CUGs do match

Note 1: The value for CODE is 6 to indicate the ICUG optional facility section.

Note 2: The value for LENGTH is 9 to indicate that nine bytes of data (*cugDnic*, *cugInterlockCode* and *cugFlags*) follow the CODE/LENGTH field.

Note 3: The values for *cugDnic* and *cugInterlockCode* are extracted from the same fields in the switch international CUG optional facility section.

Note 4: The format of the *cugDnic* field is as follows.

BCD Char.	Item	Hex Value
1	Padding	0
2 - 5	Digits	0 - 9
6	Sign	C

Note 5: The range for *cugInterlockCode* is 0 to 65535.

Note 6: The *cugFlags* field is described previously in National CUG layout.

Sensor identifier option

The *Sensor identifier* option contains the Sensor identifiers of the calling and called DTEs. The ISDN/PPSN bit is indicated in the 23rd bit of the integer and is extracted by dividing the value by 8388608.

Table 15
Sensor identifier optional facility section

Offset (bytes)	DPN-100 field names	Field size (bytes)	COBOL format
0	CODE/LENGTH	2	PIC S9999 COMP
2	CALLING SENSOR ID	4	PIC S9(7) COMP
6	CALLED SENSOR ID	4	PIC S9(7) COMP

code/length: This field contains a *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length* value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section.

Note 1: The value of code is 0007.

Note 2: The value of length is 8.

callingSensorId: The sensor identifier of the calling DTE.

calledSensorId: The sensor identifier of the called DTE.

X.75 interface identifier option

The *X.75 interface identifier* option contains the incoming and outgoing X.75 interface identifiers. Zero is used to indicate that no value exists.

Table 16
X.75 interface identifier optional facility section

Offset (bytes)	DPN-100 field names	Field size (bytes)	COBOL format
0	CODE/LENGTH	2	PIC 9999 COMP
2	INCOMING X75II	6	PIC 9999999999 9 COMP-3
8	OUTGOING X75II	6	PIC 9999999999 9 COMP-3

code/length: This field contains a *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length* value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section.

Note 1: The value of code is 0008.

Note 2: The value of length is 12.

incomingX75ii: The incoming X.75 interface identifier.

outgoingX75ii: The outgoing X.75 interface identifier.

Gateway option

The *Gateway* option contains the type of gateways traversed.

Table 17
Gateway type optional facility section

Offset (bytes)	DPN-100 field names	Field size (bytes)	COBOL format
0	CODE/LENGTH	2	PIC 9999 COMP
2	INCOMING GTWY TYPE	2	PIC 9999 COMP
4	OUTGOING GTWY TYPE	2	PIC 9999 COMP

code/length: This field contains a *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length* value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section.

Note 1: The value of code is 0009.

Note 2: The value of length is 4.

incomingGtwyType: The incoming gateway type.

outgoingGtwyType: The outgoing gateway type.

Videotex option

The 90 bytes of Videotex information in the *Videotex* option are extracted directly from the 90 bytes of Videotex optional facility data that appear in the Published Format accounting record when the facility is used.

Table 18
Optional Videotex facility section

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
0	CODE/LENGTH	2	PIC 9999 COMP
2	Videotex data	90	PIC X(90)

code/length: This field contains a *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length* value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section.

Note 1: The value of code is 10.

Note 2: The value of length is 90.

videotexData: A 90-byte data field.

Routing class of service option

The *Routing class of service* option provides information about the Global Routing Class of Service (RCOS) settings on the called and calling ends of a call.

Table 19
Routing class of service (RCOS) section

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
0	CODE/LENGTH	2	PIC 9999 COMP
2	ACCT_CLG_ RCOS	1	PIC S9 COMP
3	ACCT_CLD_ RCOS	1	PIC S9 COMP

code/length: This field contains a *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length* value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section.

Note 1: The value of LENGTH is 2 bytes.

Note 2: The value of CODE is 11.

acctClgRcos: This field is two bits indicating the class of service of the calling end.

- Bit 1(callingRcos_normalReliability) is the normal reliability bit. Normal reliability RCOS is used if it is set; otherwise high reliability RCOS is used.

- Bit 0(callingRcos_minimumDelay) is the minimum delay bit. Delay RCOS is used if it is set; otherwise it uses the throughput RCOS.

acctCldRcos: This field is two bits indicating the class of service of the called end.

- Bit 0(calledRcos_minimumDelay) is the minimum delay bit. Delay RCOS is used if it is set; otherwise it uses the throughput RCOS.
- Bit 1(calledRcos_normalReliability) is the normal reliability bit. Normal reliability RCOS is used if set; otherwise high reliability RCOS is used.

Extended national address option

The *Extended national address* option contains the Extended national address digits for the calling, called, and original called addresses.

The Extended national address facility will be included in the accounting record only when Extended national addressing is enabled, and the given call uses an extended national address.

Table 20
Extended national address facility section

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
0	CODE/LENGTH	2	PIC 9999 COMP
2	Calling address length/Calling address extension	3	PIC S9 (6) COMP
5	Called address length/Called address extension	3	PIC S9 (6) COMP
8	Original called address/ Extension called address	3	PIC S9 (6) COMP

code/length: This field contains a *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length* value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section.

Note 1: The value of LENGTH is 9 bytes.

Note 2: The value of CODE is 12.

Note 3: For Calling address length, Called address length, and Original called address, the lengths are 4 bits.

callingAddLength: The calling address extension.

calledAddLength: The called address extension.

orgCalledAdd: The extension called address.

Frame Relay option

The Published Format for *Frame relay* option is the base part of the traditional Published Format, plus two frame relay-specific optional facilities, and one Original Called Address optional facility (when call redirection is supported).

All frame relay usage data is contained in two optional sections: the local (calling) section and the remote (called) section. The local section contains the last snapshot of frame relay data from the local (calling) end taken by the frame relay protocol. The remote section contains the last snapshot of frame relay data from the remote (called) end taken by the frame relay protocol.

The terms *calling* and *called* are used as part of some of the frame relay field names. The term *calling* refers to the calling end, which is that part of the switch interface that initiated the call request. The term *called* refers to the called end, which is that part of the switch interface that received the call request. For PVCs, the Master end initiated the call. For SVCs, the calling device initiated the call.

In normal conditions, the calling end will generate both the calling and the called records, and the called end will not generate any records. In unusual circumstances, such as when the trunk is down, the calling end will generate both calling and called records, and the called end will also generate both calling and called records. However, the remote section may be accurate only up to the last count exchange, which could take place up to 15 minutes prior.

For the byte counts described, the frame header, frame trailer, DLCI header, and CRC bytes are not included.

For further details on frame relay refer to one of the following documents:

- DPN-100
 - 241-1001-120 *DPN-100 Frame Relay Service User Guide and Specification*

Local (calling) part of Frame Relay option

“Remote (called) part of the overall frame relay snapshot” (page 73) shows the optional facility for the local (calling) part of the overall frame relay snapshot.

Table 21
Local (calling) part of the overall frame relay snapshot

Offset (bytes)	Field size (bytes)	Cobol format
0	2	PIC 9999 COMP
2	2	PIC 9999 COMP
4	4	PIC 9 (8) COMP
8	4	PIC 9(8) COMP
12	8	PIC 9(16) COMP
(Sheet 1 of 2)		

Table 21 (Continued)
Local (calling) part of the overall frame relay snapshot

Offset (bytes)	Field size (bytes)	Cobol format
20	8	PIC 9 (16) COMP
28	8	PIC 9(16) COMP
36	8	PIC 9(16) COMP
44	4	PIC 9(8) COMP
48	4	PIC 9(8) COMP
52	4	PIC 9(8) COMP
56	4	PIC 9(8) COMP
60	4	PIC 9(8) COMP
64	8	PIC 9(16) COMP
(Sheet 2 of 2)		

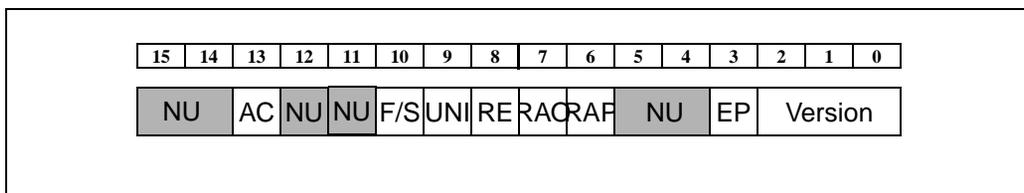
Local (calling) Frame Relay field descriptions. The following are the field descriptions for the local (calling) portion of the frame relay option.

code/length: The *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length* value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section.

Note: The value of CODE is 13.

flags: This 16-bit field contains values for call characteristics. The layout of the bits are explained below.

Figure 3
Layout of the “Flags” field



The acronyms shown in the layout of the flags field are defined as follows:

Note: For further details on any of the flags fields, refer to the frame relay documents listed in “Frame Relay option” (page 66).

Version: the version number of the accounting record. As new fields are added, the version number is incremented. That is, if a new release of software contains new fields, the version number will be incremented to indicate that new fields exist in the record. The range is 000 - 111.

EP (*callingClassOfService_emissionPri*): the emission priority. The emission priority lets you choose between two distinct classes of services. This value is provisioned. If set to 1, high emission priority is selected for low delay. If set to zero, normal emission priority is selected.

NU: This field is not used. It is not defined.

RAP (*callingRateAdaptation_provisioned*): The rate adaptation is a congestion control mechanism. This value is provisioned. If set to 1, the rate adaptation was provisioned “on”. If set to zero, the rate adaptation was provisioned “off”. Rate adaptation can only be turned on if rate enforcement is also turned on.

RAO (*callingRateAdaptation_occured*): If set to 1, rate adaptation occurred on the interface. If set to zero, rate adaptation did not occur on the interface during this accounting report interval.

RE (callingRateAdaptation_enforced): the rate enforcement. Rate Enforcement prevents one user from using an inequitable share of the network's resources by ensuring that user traffic follows the provisioned CIR and EIR rates. If set to 1, the rate enforcement was provisioned "on". If set to zero, the rate enforcement was provisioned "off".

UNI (flags_callingUni): If set to 1, the interface is UNI. If set to zero, the interface is NNI.

F/S: If set to 1, the accounting record contains frame counts. If set to zero, the accounting record contains segment counts.

AC (flags_callingAbit): If set to 1, the A-bit to or from the interface was changed on the interface during the current accounting period. If set to zero, the A-bit was not changed.

callingCir: The calling end provisioned CIR (Committed Information Rate). It is rounded up to the next 800 bits per second.

callingEir: The calling end provisioned EIR (Excess Information Rate). It is provisioned as excessive burst size and is rounded up to the next 800 bytes per second.

callingEgressBytes: The total number of bytes received at the calling end dlci. This is the number of bytes sent by frameRelay dlci to the application. With no congestion and no errors, this number will be equal to the total number of ingress bytes (calledIngressBytes) at the called end of vc connection. The value for this field is metered at point E as shown in "Accounting meter reference points" (page 79).

callingIngressBytes: The total number of bytes sent at the calling end dlci. This is the number of bytes sent by the application to frameRelay dlci. With no congestion and no errors, this number will be equal to the total number of egress bytes (calledEgressBytes) at the called end of vc connection. Only good frames are passed to dlci from the application. The frames discarded due to CRC error are not included in this count. It

does include the count *callingDiscardBytes*. The value for this field is metered at point A as shown in “Accounting meter reference points” (page 79).

callingEirEgressBytes: The total number of EIR bytes received from the network at the calling end with discard eligibility bit set. This total is included in the value of *callingEgressBytes* at the called end of the connection. Note that the discard eligibility bit can be set by application or by frameRelay. The value for this field is metered at point D as shown in “Accounting meter reference points” (page 79).

callingEirIngressBytes: The total number of EIR bytes sent to the network at the calling end from the application to frameRelay dlcI. This total is included in *callingIngressBytes* value. It is the number of bytes or frames that have the discard eligibility bit set by the application. The value for this field is metered at point C as shown in “Accounting meter reference points” (page 79).

callingEgressFrames and **callingEgressSegments:** The total number of frames/segments sent at the calling end. This is the total number of frames/segments sent by frameRelay dlcI to the application. Frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. A billing system uses the F/S flag to determine whether this count reports frames or segments. The value for this field is metered at point E as shown in “Accounting meter reference points” (page 79).

callingIngressFrames and **callingIngressSegments:** The total number of frames/segments sent at the calling end. This is the number of frames/segments sent by the application to frameRelay dlcI. The frames discarded due to CRC error are not included in this count. The frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. The value for this field is metered at point A as shown in “Accounting meter reference points” (page 79).

callingEirEgressFrames or **callingEirEgressSegments:** The total number of EIR frames/segments received at the calling end. This is the total number of frames/segments received by frameRelay dlcI from the

network with the discard eligibility bit set. This number is included in the *callingEgressFrames/callingEgressSegments*. Note that the discard eligibility bit can be set by the application or by frameRelay. The frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. A billing system uses the F/S flag to determine whether this count reports frames or segments. The value for this field is metered at point D as shown in “Accounting meter reference points” (page 79).

callingEirIngressFrames or **callingEirIngressSegments**: The total number of EIR frames/segments sent to the network at the calling end. This is the total number of frames/segments sent by the FrameRelay dlcI to the network with the discard eligibility bit set. Note that the discard eligibility bit can be set by the application or by frameRelay. The frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. A billing system uses the F/S flag to determine whether this count reports frames or segments. The value for this field is metered at point C as shown in “Accounting meter reference points” (page 79).

callingDiscardedFrames or **callingDiscardedSegments**: The number of Ingress frames/segments discarded at the calling end for the following reasons: A-Bit is off, Ingress frame is too long, Ingress frame is too short, or discarded due to rate enforcement. The value for this field is metered at point B as shown in “Accounting meter reference points” (page 79).

callingDiscardedBytes: The number of Ingress bytes discarded at the calling end for the following reasons: rate enforcement, A-Bit is off, Ingress frame is too long, or Ingress frame is too short. The value for this field is metered at point B as shown in “Accounting meter reference points” (page 79).

Note: **callingTransferPriority**: is not converted to Published Format.

Remote (called) part of Frame Relay option

“Remote (called) part of the overall frame relay snapshot” (page 73) shows the optional facility for the remote (called) part of the overall frame relay snapshot.

Note: The formats of the remote and local fields are identical, they differ only in the value of the code subfield.

Table 22**Remote (called) part of the overall frame relay snapshot**

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
0	CODE/LENGTH	2	PIC 9999 COMP
2	Flags	2	PIC 9999 COMP
4	Provisioned CIR	4	PIC 9 (8) COMP
8	Provisioned EIR	4	PIC 9(8) COMP
12	Bytes received	8	PIC 9(16) COMP
20	Bytes sent	8	PIC 9 (16) COMP
28	EIR bytes sent to the link	8	PIC 9(16) COMP
36	EIR bytes sent to network	8	PIC 9(16) COMP
44	frames/segments received	4	PIC 9(8) COMP
48	frames/segments sent	4	PIC 9(8) COMP
52	EIR frames/segments sent to the link	4	PIC 9(8) COMP
(Sheet 1 of 2)			

Table 22 (Continued)
Remote (called) part of the overall frame relay snapshot

Offset (bytes)	DPN-100 field names	Field size (bytes)	Cobol format
56	EIR frames/ segments sent to network	4	PIC 9(8) COMP
60	frames/segments discarded	4	PIC 9(8) COMP
64	Bytes discarded	8	PIC 9(16) COMP
(Sheet 2 of 2)			

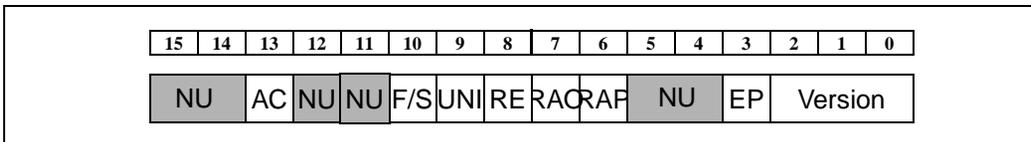
Remote (called) Frame Relay field descriptions. These are the field descriptions for the remote (called) portion of the frame relay option.

code/length: The *code* value and a *length* value, arithmetically combined. The *code* value identifies the optional facility section and the *length* value indicates the length of the optional facility section, excluding this field. The low-order byte contains the *length* value for the optional facility section and the high-order byte contains the *code* value that identifies the optional facility section.

Note: The value of CODE is 14.

flags: This 16-bit field contains values for call characteristics. The layout of the bits are explained as follows.

Figure 4
Layout of the “Flags” field



The acronyms shown in the layout of the flags field are defined as follows:

Note: For further details on any of the flags fields, refer to one of the frame relay documents listed in “Frame Relay option” (page 66).

Version: the version number of the accounting record. As new fields are added, the version number is incremented. That is, if a new release of software contains new fields, the version number will be incremented to indicate that new fields exist in the record. The range is 000 - 111.

EP (calledClassOfService_emissionPri): the emission priority. The emission priority lets you choose between two distinct classes of services. This value is provisioned. If set to 1, high emission priority is selected for low delay. If set to zero, normal emission priority is selected.

NU: This field is not used. It is not defined.

RAP (calledRateAdaptation_provisioned): The rate adaptation is a congestion control mechanism. This value is provisioned. If set to 1, the rate adaptation was provisioned “on”. If set to zero, the rate adaptation was provisioned “off”. Rate adaptation can only be turned on if rate enforcement is also turned on.

RAO (calledRateAdaptation_occured): If set to 1, rate adaptation occurred on the interface. If set to zero, rate adaptation did not occur on the interface during this accounting report interval.

RE (calledRateAdaptation_enforced): the rate enforcement. Rate Enforcement prevents one user from using an inequitable share of the network's resources by ensuring that user traffic follows the provisioned CIR and EIR rates. If set to 1, the rate enforcement was provisioned "on". If set to zero, the rate enforcement was provisioned "off".

UNI (flags_calledUni0): If set to 1, the interface is UNI. If set to zero, the interface is NNI.

F/S: If set to 1, the accounting record contains frame counts. If set to zero, the accounting record contains segment counts.

AC (flags_calledAbit): If set to 1, the A-bit to or from the interface was changed on the interface during the current accounting period. If set to zero, the A-bit was not changed.

calledCIR: The called end provisioned CIR (Committed Information Rate). It is rounded up to the next 800 bits per second.

calledEir: The called end provisioned EIR (Excess Information Rate). It is provisioned as excessive burst size and is rounded up to the next 800 bytes per second.

calledEgressBytes: The total number of bytes received at the called end dlci. This is the number of bytes sent by frameRelay dlci to the application. With no congestion and no errors, this number will be equal to the total number of ingress bytes (callingIngressBytes) at the calling end of vc connection. The value for this field is metered at point E as shown in "Accounting meter reference points" (page 79).

calledIngressBytes: The total number of bytes sent at the called end dlci. This is the number of bytes sent by the application to the frameRelay dlci. With no congestion and no errors, this number will be equal to the total number of egress bytes (callingEgressBytes) at the calling end of the virtual circuit connection. Only good frames are passed to dlci from the

application. The frames discarded due to CRC error are not included in this count. It does include the count *calledDiscardBytes*. The value for this field is metered at point A as shown in “Accounting meter reference points” (page 79).

calledEirEgressBytes: The total number of EIR bytes received from the network at the called end with the discard eligibility bit set. This total is included in the value of *calledEgressBytes* at the called end of the connection. Note that the discard eligibility bit can be set by the application or by frameRelay. The value for this field is metered at point D as shown in “Accounting meter reference points” (page 79).

calledEirIngressBytes: The total number of EIR bytes sent to the network at the called end from the application to frameRelay dlcI. This total is included in the *calledIngressBytes* value. It is the number of bytes or frames that have the discard eligibility bit set by the application. The value for this field is metered at point C as shown in “Accounting meter reference points” (page 79).

calledEgressFrames and **calledEgressSegments:** The total number of frames/segments sent at the called end. This is the total number of frames/segments sent by frameRelay dlcI to the application. The frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. A billing system uses the F/S flag to determine whether this count reports frames or segments. The value for this field is metered at point E as shown in “Accounting meter reference points” (page 79).

calledIngressFrames and **calledIngressSegments:** The total number of frames/segments sent at the called end. This is the number of frames/segments sent by the application to frameRelay dlcI. The frames discarded due to CRC error are not included in this count. The frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. The value for this field is metered at point A as shown in “Accounting meter reference points” (page 79).

calledEirEgressFrames or **calledEirEgressSegments**: The total number of EIR frames/segments received at the called end. This is the total number of frames/segments received by the frameRelay dlci from the network with the discard eligibility bit set. This number is included in the *calledEgressFrames/calledEgressSegments*. Note that the discard eligibility bit can be set by the application or by frameRelay. The frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. A billing system uses the F/S flag to determine whether this count reports frames or segments. The value for this field is metered at point D as shown in “Accounting meter reference points” (page 79).

calledEirIngressFrames or **calledEirIngressSegments**: The total number of EIR frames/segments sent to the network at the called end. This is the total number of frames/segments sent by the frameRelay dlci to the network with the discard eligibility bit set. Note that the discard eligibility bit can be set by the application or by frameRelay. The frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. A billing system uses the F/S flag to determine whether this count reports frames or segments. The value for this field is metered at point C as shown in “Accounting meter reference points” (page 79).

calledDiscardedFrames or **calledDiscardedSegments**: The number of Ingress frames/segments discarded at the called end for the following reasons: rate enforcement, A-Bit is off, Ingress frame is too long, or Ingress frame is too short. The value for this field is metered at point B as shown in “Accounting meter reference points” (page 79).

calledDiscardedBytes: The number of Ingress bytes discarded at the called end for the following reasons: A-Bit is off, Ingress frame is too long, Ingress frame is too short, or discarded due to rate enforcement. The value for this field is metered at point B as shown in “Accounting meter reference points” (page 79).

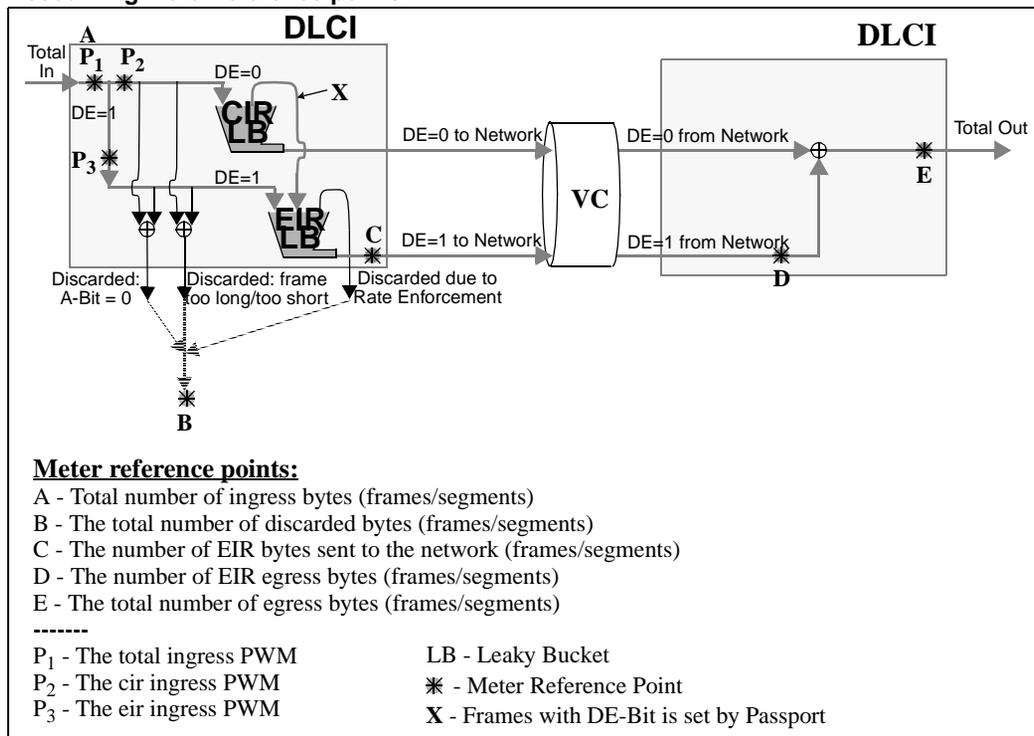
Note: calledTransferPriority: is not converted to Published Format.

Accounting meter reference points

The traffic at a DLCI or a virtual circuit (VC) is metered at reference points as shown in “Accounting meter reference points” (page 79). Although only traffic in one direction is shown, the same metering of the DLCI-VC-DLCI set is done in both directions.

“Accounting meter reference points” (page 79) depicts only those traffic metering reference points at which accounting data is collected. In addition to values collected by the accounting meter, real-time statistics contains other data, such as the number of bytes/frames discarded when A-Bit is off, the number of frames discarded due to the frame being too short, the number of bytes/frames discarded due to the frame being too long, or the number of bytes/frames discarded due to rate enforcement.

Figure 5
Accounting meter reference points



Chapter 2

DPN-100 BDF accounting records

This section describes DPN-100 BDF accounting records:

- “BDF file names” (page 81)
- “Common fields” (page 83)
- “Peak water mark fields” (page 100)
- “Time change records” (page 104)
- “Optional fields” (page 105)

The DPN-100 BDF accounting fields, and their sequence within a BDF accounting file, is determined by file `dpn_acc.rdf`.

BDF file names

Converted DPN accounting data files that are ready for transfer to a customer host are placed in directory `/opt/MagellanMDP/data/mdp/dump/accounting`.

BDF files without errors have a suffix of *bdf*. BDF files that contain errors have a suffix of *err*. Error messages that identify the records and fields with errors are written to the appropriate File Manager log file.

Note: Error files are not transferred to customer hosts.

BDF files use a normalized naming convention as follows:

```
<switch>_<dataType>_<date_time>_<nodeID>_<seq#>.<ext>
```

where:

switch

is *dpn* for DPN-100.

dataType

is *acc* for accounting.

date_time

is the date and time when the file was opened on the switch;

for DPN-100, this field contains YYMMDDhhmmss (YY=year, MM=month, DD=day, hh=hour, mm=minute, ss=seconds). The *date* fields of all records generated by DPN-100 switches contain two-digit years. This format requires that the MDP applies an algorithm in each of its data conversion processes to determine the appropriate century (19xx or 20xx); this algorithm is as follows:

```
year = YY
YY >= 50 (indicates 20th century)
YY < 50 (indicates 21st century)
```

nodeID

is the DPN-100 switch that generated the data.

seq#

is a file sequence number assigned by a switch.

ext

the file extension is *err* for a file that stores records that cannot be converted successfully (the error file is stored in the appropriate dump directory). This value is *bdf* for a BDF file.

Examples

```
dpn_acc_941101154905_2700_19.bdf
dpn_acc_941101154905_2700_19.err
```

Short BDF file names

A file name limitation exists for transferring BDF files to an IBM customer host; a maximum of 16 characters can be used. The use of the short file naming convention is specified during MDP configuration using the *Configure File Mover* option of the MDP Configuration Client *gmdpconfig*.

If multiple and different types of customer hosts (for example, IBM and VAX) are used, the same file naming convention must be used for all hosts; if one of the customer hosts is an IBM, the short file naming convention must be specified.

The File Mover process truncates the *switchID* to the first 4 characters of the full *switchID*. MDP does not check for unique *switch_id* naming. If short file names are used, it is your responsibility to ensure that the first 4 characters of the nodeID are unique.

The short file naming convention for open BDF files is:

```
<s><t>mmddhh.MM
```

The short file naming convention for closed BDF files is:

```
<s><t>mmddhh.MM<seq#><switchID>
```

where:

s=switch type (d=DPN); *t*=data type (a=accounting); *mm*=month; *dd*=day; *hh*=hour; *MM*=minutes; *seq#*=only the last 2 digits; *switchID*=only the first 4 digits.

If the long file naming convention is selected, the transfer of files to IBM customer hosts fails and the BDF file is renamed in the dump directory using the following naming format:

```
<switchType>_<dataType>_<date_time>_<nodeID>_<seq#>  
_bdf_#<failed host number>
```

Common fields

This section describes the *fixed fields* of a BDF accounting record for DPN.

The terms *calling* and *called* are used as part of some of the field names. The term *calling* end refers to that part of the switch interface that initiated the call request. The term *called* end refers to that part of the switch interface that

received the call request. For permanent virtual circuits (PVCs), the Master end initiated the call. For switched virtual circuits (SVCs), the calling device initiated the call.

Common field descriptions

This section describes the *fixed fields* of a DPN-100 BDF accounting record:

switchType: This field identifies the type of switch that generated the data record. This value is `dpn` for DPN.

dataType: This field identifies the data type of the record; with a value of `acc` for accounting data.

subDataType: This field identifies different record types (schemas) within each data type.

recordIdentifier: This field does not apply to DPN.

accountingType: This field defines the type of DPN accounting record.
1 = Accounting record
2 = Time change record

vintageNumber: This number indicates the version of the Bulk Data Format (BDF) accounting record used and is set to 4 for all BDF DPN accounting records.

namsCat: This DPN-100 field indicates the NAMS category.
4 = Vintage 4 Accounting
3 = Vintage 3 Accounting
2 = statistics
0 = alarms

startDateTime: This is the start time of the period accounted for in the record. It is in the format `YYYYMMDDThhmmss`, where: `YYYY` indicates the year, `MM` is for the month, `DD` is for the day, `T` indicates the beginning of the time data, `hh` is for the hour, `mm` is for the minute, and `ss` is for the seconds.

Note: Raw DPN records contain 2-digit years. The MDP BDF converter resolves these 2-digit years into 4-digit years using the following algorithm based on windows set by DPN:

year = YY

YY >= 50 (indicates 20th century (19xx))

YY < 50 (indicates 21st century (20xx))

A zero value in any of the time fields causes the field to be truncated. Even the T indicator will be dropped if the record begins at exactly midnight. For example, 19970627T120300 is displayed as 19970627T1203 (the 00 seconds have been truncated) and 19970627T000000 is displayed as 19970627 (all of the time fields have been truncated, including the T indicator).

For SVC, *startTime* is the time when the call request packet was created. For PVC, *startTime* is set by the Master VC when the master call request is formatted. On subsequent accounting records, *startTime* corresponds to the end time of the previous record. If Time of Day Accounting (TODA) is used, *startTime* (and *endTime*) correspond to times set in the schedule.

endTime: This is the end time of the period accounted for in the record. It is in the format YYYYMMDDThhmmss, where: YYYY indicates the year, MM is for the month, DD is for the day, T indicates the beginning of the time data, hh is for the hour, mm is for the minute, and ss is for the seconds.

Note: Raw DPN records contain 2-digit years. The MDP BDF converter resolves these 2-digit years into 4-digit years using the following algorithm based on windows set by DPN:

year = YY

YY >= 50 (indicates 20th century (19xx))

YY < 50 (indicates 21st century (20xx))

A zero value in any of the time fields causes the field to be truncated. Even the T indicator will be dropped if the record ends at exactly midnight. For example, 19970627T120300 is displayed as

19970627T1203 (the 00 seconds have been truncated) and 19970627T000000 is displayed as 19970627 (all of the time fields have been truncated, including the T indicator).

If Time of Day Accounting (TODA) is used, *endTime* in non-final records is the time of the TODA period. For 12-hour accounting, *endTime* in non-final records is the time when the 12-hour timer expires at the charged end. On final accounting records, *endTime* is the time when the disconnect packet is first received. If Time of Day Accounting (TODA) is used, *endTime* (and *startTime*) correspond to times set in the schedule.

callingNodeIdentifier: This field uniquely identifies the node, in an autonomous network, where the call originates.

This field is the NAMSID. If the switch is NM equipped, the NAMSID is the RID and can have a value in the range 1 to 126. If the switch is AM or RM equipped, this value can be any integer between 256 and 49151.

callingPortIdentifier: A 16-bit field that identifies the physical port number where the call originates.

If the physical port is on a DPN-100, the Port Number will be identified by a PE Number, PI Number, and a Port Number. In the case of X.75 multi-link, only the PE Number is specified (the PI and PO will be set to zero).

The Calling Port Number is 16 bits long and has the following format:

- PE Number is represented in the first 5 bits of the first byte of the calling port number.
- PI Number is represented in the next 5 bits of the calling port number.
- PO Number is represented in the last 6 bits of the second byte in the calling port number.

Note: If VirtualFramer is provisioned, the PO Number will always be set to 3F.

Information for the *callingPort Identifier* field may also appear in a record with the following three fields: *callingPEIdentifier*, *callingPIIdentifier*, and *callingPOIdentifier*.

callingNpi: This field is the Numbering Plan Indicator that indicates the numbering plan used in the calling address, as follows:

- 0 = X.121 numbering plan
- 1 = E.164 numbering plan
- other = undefined

callingAddressLength: This is the length of the callingAddress field, in bytes.

callingAddress: This indicates the number of digits in the calling address (DNA) field. The maximum is 15. It also contains the full address field of the calling address (DNA), including the DNIC (first four digits) for X.121 or Country Code for E.164 address. The field is left-justified with trailing zeroes.

callingDlci: The Logical Channel Number of the calling end. The value ranges from 0 to 4095. For frame relay service, the valid range is 16 to 1007.

callingDse: This is the Data Service Exchange (DSE) of the calling end. The value ranges from 0 to 255. For transit/incoming X.75 SVC calls, the calling DSE information is valid only if the calling tariff is absent from the accounting record. DSE is defined in the DNA component on DPN-100.

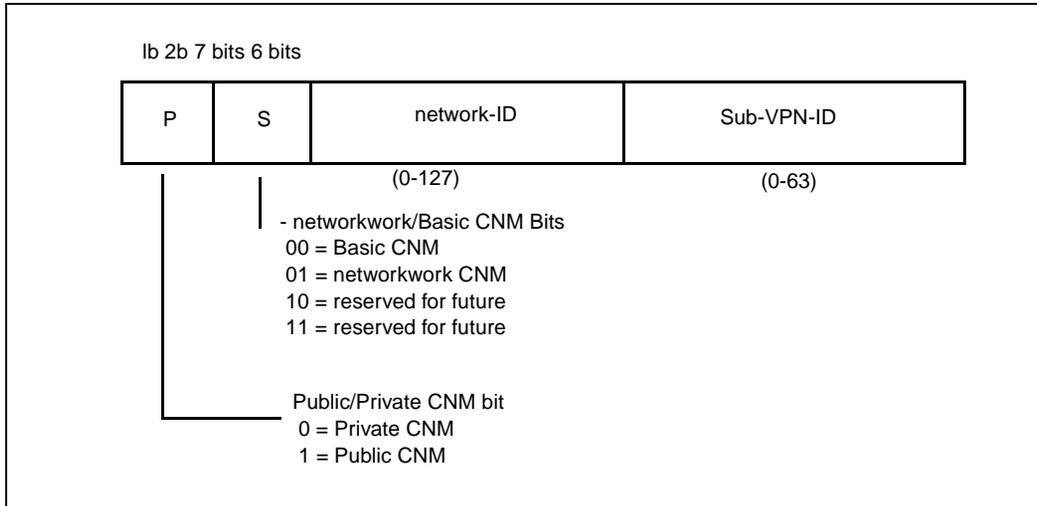
callingServiceType: Indicates the service supported by the port used by the calling end-user. The value ranges from 0 to 255. For transit/incoming X.75 SVC calls, the calling service type information is valid only if the calling tariff is absent from the accounting record. This number is defined as `accountClass` in the DNA component on DPN-100.

callingReceiveWindow and **callingSendWindow:** This two-byte field consists of the values for the calling end local transmit and receive window sizes, arithmetically combined. The low-order byte contains the value for the calling end local transmit window size and the high-order byte contains the value for the calling end local receive window size. Both the calling end local transmit and receive window sizes have values ranging from 1 to 7. For SVC, this field is only populated for X.25 and X.75 services. All other services have the value 0. For PVC, this field is populated as provided in the PVC envelope, for all services.

Note: For frame relay service, these values are not relevant and should be ignored. These fields do not apply because frame relay uses the lightweight virtual circuit instead of the flow-controlled virtual circuit.

callingCustomerIdentifier: This field is used as an optional identifier for the calling end of the call in the Customer Network Management (CNM) and Virtual Private Network environment. The value in this field is taken from the Customer Network Management Identifier (CID) field in the DNA-CUG envelope for the access line. The value 0 is the default value reserved for all ports that are not associated with a Virtual Private Network and for which CNM is not provided.

Figure 6
16 bit field format



calledNodeIdentifier: This field uniquely identifies the node, in an autonomous network, where the call terminates.

For DPN, this field is the NAMSID. If the switch is NM equipped, the NAMSID is the RID and can have a value in the range 1 to 126. If the switch is AM or RM equipped, this value can be any integer between 256 and 49151.

calledPortIdentifier: A 16-bit field that identifies the physical port number where the call terminates.

If the physical port is on a DPN-100, the Port Number will be identified by a PE Number, PI Number, and a Port Number. In the case of X.75 multi-link, only the PE Number is specified (the PI and PO will be set to zero).

The Called Port Number is 16 bits long and has the following format:

- PE Number is represented in the first 5 bits of the first byte of the called port number.

- PI Number is represented in the next 5 bits of the called port number.
- PO Number is represented in the last 6 bits of the second byte in the called port number.

Information for the *calledPort Identifier* field may also appear in a record with the following three fields: *calledPEIdentifier*, *calledPIIdentifier*, and *calledPOIdentifier*.

calledNpi: This field is the Numbering Plan Indicator that indicates the numbering plan used in the called address. The field format is similar to the *calling NPI* field.

0 = X.121 numbering plan

1 = E.164 numbering plan

other = undefined

calledAddressLength: This is the length of the calledAddress field in bytes.

calledAddress: The number of digits in the called address (DNA) field. The maximum is 15. It also provides the full address field of the called address (DNA), including the DNIC (first four digits) for X.121 or Country Code for E.164 address. The field is left-justified with trailing zeroes. The field has a maximum of 15 digits. The field format is similar to the *calling address* field.

calledDlci: The Logical Channel Number of the called end. The value ranges from 0 to 4095. For frame relay service, the range is 16 to 1007.

calledDse: This is the Data Service Exchange (DSE) of the called end. The value ranges from 0 to 255. For outgoing/transit X.75 SVC calls, the called DSE information is valid only if the called tariff is absent from the accounting record. DSE is defined in the DNA component on DPN-100.

calledServiceType: Indicates the service supported by the port used by the called end-user. The value ranges from 0 to 255. For outgoing/transit X.75 SVC calls, the called service type information is valid only if the called tariff is absent from the accounting record. This number is defined in the DNA_CUG envelope as accountClass.

calledReceiveWindow and **calledSendWindow:** This two-byte field consists of the values for the called end local transmit and receive window sizes, arithmetically combined. The low-order byte contains the value for the called end local transmit window size and the high-order byte contains the value for the called end local receive window size. Both the called end local transmit and receive window sizes have values that range from 1 to 7. For SVC, this field is only populated for X.25 and X.75 services. All other services have the value zero. For PVC, this field is populated as provided in the PVC envelope, for all services. For frame relay service, these values are not relevant and should be ignored. These fields do not apply because frame relay uses the lightweight virtual circuit instead of the flow-controlled virtual circuit.

calledCustomerIdentifier: This field is used as an optional identifier for the called end of the call in the Customer Network Management (CNM) and Virtual Private Network environment. The value in this field is taken from the provisioning data for the access line, or more specifically, from the Customer Network Management Identifier (CID) field in the DNA-CUG envelope for the access line. The value zero is the default value reserved for all ports that are not associated with a Virtual Private Network and for which CNM is not provided. The format of this 16-bit field can be found in “16 bit field format” (page 89).

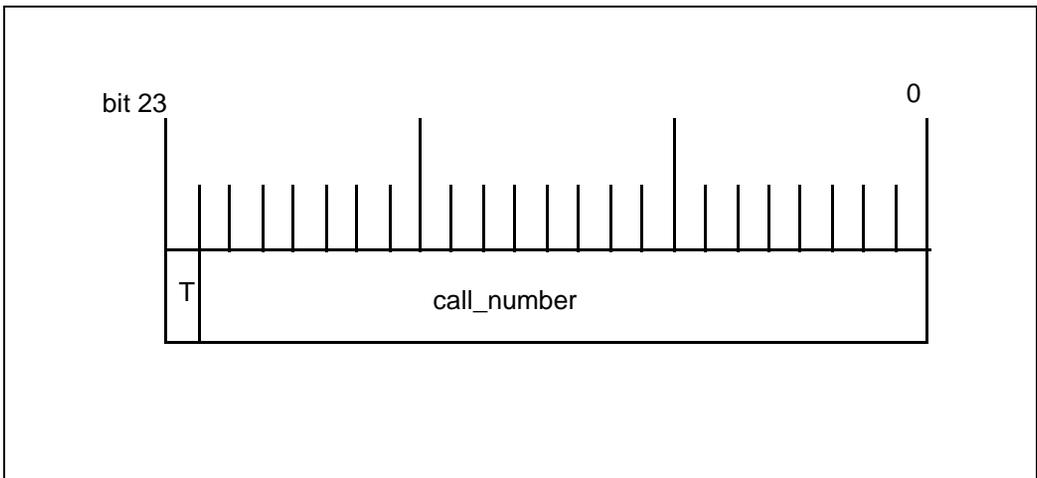
callReferenceNumber: This is a three-byte number generated by the switch, or taken from the call identifier utility in the X.75 call request packet. The same call reference number is stored in the BDF accounting record at both ends of the call. It is used as one of the fields in matching duplicate records generated at each end of the call or for reconciling

records between networks. See “Call reference number generated by a PE on RM/AM and processor” (page 92) for the format of the call reference number.

- T = 0 for SVC or 1 for PVC.
- The call_number is initialized to a semi-random value (based on the time/date) when the service PE/processor first comes up as follows:
 $\text{call_number} = \text{CENTURY/YEAR XOR MONTH/DAY XOR HOUR/MINUTES}$.

The most significant byte of call_number is set to the SECONDS of the current time/date on the PE/processor. The call number is incremented by one for every outgoing call.

Figure 7
Call reference number generated by a PE on RM/AM and processor



flags: These one-bit flags have the values 1 or 0, depending whether they are turned on or off. The values of the flags are as follows:

- 1 WHO CLEARED(flags_calledCleared)
- 0 - Cleared from calling end
1 - Cleared from called end
- The callCleared bit is initialized to 0 on the calling end and initialized to 1 on the called end. If the call is cleared on the local switch, that switch retains its value for the callCleared bit and sends a disconnect message to the other end. When a switch receives a disconnect message, the switch takes the value for the callCleared flag from the remote end.
- 2 WHO GENERATED(flags_calledGenerated)
- For SVC:
0 - Generated from calling end
1 - Generated from called end
- For PVC:
0 - Generated from master end
1 - Generated from slave end
- 3 UNIQUE(flags_unique)
- 0 - Possible duplicate, both ends may have generated the accounting record
1 - No duplicates, only one end has generated the accounting record
- 4 CALL REACHED DATA
TRANSFER(flags_reachedDataTransfer)
- 0 - Call does not reach data transfer at both ends
1 - Call reached data transfer at both ends
- 5 FIRST RECORD FOR THIS CALL(record_first)
- 0 - Not first record for the call
1 - First record for the call

- 6 LAST RECORD FOR THIS CALL(record_last)
 - 0 - Not last record for the call
 - 1 - Last record for the call
- 7 TERMINATOR IS GATEWAY(callType_gateway)
 - 0 - Terminator is not Gateway
 - 1 - Terminator is Gateway
- 8 ORIGINATOR IS GATEWAY (callType_localNetwork)
 - 0 - Originator is not Gateway
 - 1 - Originator is Gateway
- 9 COLLECTION FOR BILLING (CALLING END)
(callingCollectReason_bill)
 - 0 - Not collected for billing purposes
 - 1 - Collected for billing purposes
- 10 COLLECTION FOR TEST (CALLING END)
(callingCollectReason_test)
 - 0 - Not collected for test purposes
 - 1 - Collected for test purposes
- 11 COLLECTION FOR STUDY (CALLING END)
(callingCollectReason_study)
 - 0 - Not collected for study purposes
 - 1 - Collected for study purposes
- 12 COLLECTION FOR AUDIT (CALLING END)
(callingCollectReason_audit)
 - 0 - Not collected for audit purposes
 - 1 - Collected for audit purposes

- 13 COLLECTION FORCED (CALLING END)
(callingCollectReason_force)
- 0 - collection not forced by calling end
1 - collection forced by calling end
- 14 CALL DEFLECTION
- 0 - Call not deflected
1 - Call deflected
- 15 REVERSE CHARGING (flags_calledEndCharging)
- 0 - Charged to calling end
1 - Charged to called end
- 16 FAST SELECT (flags_fastSelect)
- 0 - Not a fast select call
1 - A fast select call
- 17 RESTRICTED RESPONSE (flags_restrictedResponse)
- 0 - Restricted response option not used
1 - Restricted response option used
- 18 PRIORITY(flags_highPriority)
- 0 - Normal priority
1 - High priority
- 19 PERMANENT VIRTUAL CIRCUIT(flags_pvcType)
- 0 - SVC call
1 - PVC call
- 20 USER DATA ON CALL
REQUEST(flagsuserDataInCallReq)
- 0 - No user data on call request
1 - User data on call request

- 21 HUNT GROUP(flags_hunted)
 - 0 - Call has not been hunted
 - 1 - Call has been hunted
- 22 CALL REDIRECTION(flags_redirected)
 - 0 - Call not redirected
 - 1 - Call redirected
- 23 COLLECTION FOR BILLING (CALLED
END)(calledCollectReason_bill)
 - 0 - Not collected for billing purposes
 - 1 - Collected for billing purposes
- 24 COLLECTION FOR TEST (CALLED
END)(calledCollectReason_test)
 - 0 - Not collected for test purposes
 - 1 - Collected for test purposes
- 25 COLLECTION FOR STUDY (CALLED
END)(calledCollectReason_study)
 - 0 - Not collected for study purposes
 - 1 - Collected for study purposes
- 26 COLLECTION FOR AUDIT (CALLED
END)(calledCollectReason_audit)
 - 0 - Not collected for audit purposes
 - 1 - Collected for audit purposes
- 27 CALLED END COLLECTION REASON
VALID(flags_collReasonValid)
 - 0 - Called end collection reasons are not valid
 - 1 - Called end collection reasons are valid

28 COLLECTION FORCED (CALLED
 END)(calledCollectReason_force)

0 - collection not forced by called end

1 - collection forced by called end

callingCalledPktSize and **calledCallingPktSize**: This two-byte field consists of the values of the packet size sent from the calling DTE to the called end and packet size sent from the called DTE to the calling end, arithmetically combined. The low-order byte contains the value of the packet size sent from the calling DTE to the called end. The value is the base 2 logarithm of the actual packet size. The high-order byte contains the value of the packet size sent from the called DTE to the calling end. The value is the base 2 logarithm of the actual packet size.

Note: For frame relay service, these values are not relevant and should be ignored. These fields do not apply because frame relay uses the lightweight virtual circuit instead of the flow-controlled virtual circuit.

callingCalledThruput and **calledCallingThruput**: This field contains the throughput classes of packets sent from the calling DTE to the called end, and vice versa, arithmetically combined. The low-order byte contains the throughput class of the packet sent from the calling DTE to the called end and the high-order byte contains the throughput class of the packet sent from the called DTE to the calling end.

Note: For frame relay service, these values are not relevant and should be ignored. These fields do not apply because frame relay uses the lightweight virtual circuit instead of the flow-controlled virtual circuit.

Refer to “Throughput classes” (page 98) for throughput classes.

Table 23
Throughput classes

Value Of low-order or high-order byte	Throughput class (bit/s)
0 - 2	Reserved
3	75
4	150
5	300
6	600
7	1200
8	2400
9	4800
10	9600
11	19200
12	48000
13	64000
14 - 15	Reserved

clearCause: This a clear cause indicator. It indicates the reason for a call termination. For additional details, refer to 241-1001-509 *DPN-100 Call Clear, Reset and Diagnostic Codes*.

diagnosticCode: This a clear diagnostic indicator. It is set to zero (0) on non-final records, and to the network diagnostic code of the call clear on the final record. For additional details, refer to 241-1001-509 *DPN-100 Call Clear, Reset and Diagnostic Codes*.

transitDelay: This field is to be used in the future to indicate transit delays. The value is currently set to zero.

segmentSizeIndex: This field gives the segment size used on the call. The value in this field is the base 2 logarithm of the actual segment size.

localUserResets: Count of the customer-caused reset request packets, transmitted and received. The value ranges from 0 to 65535.

callingEgressSegments/Frames: Number of Egress data segments/frames received from the network. The value ranges from 0 to 16777215. For all calls, the segments/frames received field in the accounting record is calculated for the data received in the data packets. For all calls, the segments/frames received field in the accounting record is calculated for the data received in the *interrupt* and *interrupt confirmation* packets. Only for Fast Select calls (both restricted and unrestricted response), the segments/frames received field in the accounting record is calculated for the data received in *call request*, *call confirm (accept)*, and *clear request* packets. Non-Fast Select calls do not count the segments/frames in these three packets. For all the packets mentioned above, the calculation of segments/frames is performed in a common manner (that is, based on the segment size index).

For *best effort delivery* services such as frame relay, the count is recorded at the entry and exit points to and from the network.

callingIngressSegments/Frames: Number of Ingress data segments/frames sent to the network. The value ranges from 0 to 16777215. For all calls, the segments/frames sent field in the accounting record is calculated for the data sent in the data packets. For all calls, the segments/frames sent field in the accounting record is calculated for the data sent in the *interrupt* and *interrupt confirmation* packets. Only for Fast Select calls (both restricted and unrestricted response), the segments/frames sent field in the accounting record is calculated for the data sent in *call request*, *call confirm (accept)*, and *clear request* packets. Non-Fast Select calls do not count the segments/frames in these three packets. For all the packets mentioned above, the calculation of segments/frames is performed in a common manner (that is, based on the segment size index).

For *best effort delivery* services such as frame relay, the count is recorded at the entry and exit points to and from the network.

elapsedTime: Duration of the call in 0.1 second increments. This time interval is started when the call request is processed. If the call does not reach the data transfer stage, *elapsed time* is set to zero. *Elapsed time* represents a true measure of the accounting period. It is measured independently of *startDateTime* and *endTime*. Since *startDateTime* and *endTime* can be changed externally by adjusting the clock, *elapsed time* can be shorter or longer than the difference between *startDateTime* and *endTime*. This difference can show as much as 60 seconds on accounting records for the VCs that are idle (carry no traffic) and when TODA is used and is caused, in this case, by an internal TODA detection mechanism. When the 12-hour accounting period is complete, *elapsed time* is exact or differs by a few seconds from 12 hour duration.

Peak water mark fields

This section contains descriptions of the peak water mark fields that are used to record the peak byte usage at the network side of the interface when Time of Day Accounting (TODA) is provisioned.

The following must be considered:

- If TODA is not running, peak water mark fields are not included in the accounting record.
- Peak water mark monitoring is available only for the frame relay service, and Egress Accounting must be enabled in order for PWM to be applied.
- The peak water mark monitoring interval period is provisioned in the frame relay DNA CUG envelope.
- For each DLCI, the peak byte usage for CIR, EIR, and total traffic, as well as the interval in which the peak occurs are recorded. The terms *calling* and *called* refer to the interface at the calling and called end on the connection.

- The peak water mark fields can be used to validate the provisioned value of CIR and EIR against actual usage, resulting, possibly, in the re-engineering of the provisioned parameters. The frame header, frame trailer, DLCI header and CRC bytes are not included. LMI frames are also not included.
- When you use DE=0 or DE=1 traffic, but not both, the total peak water mark byte counts are the same as the CIR byte counts or the EIR byte counts.

callingTotalPwmBytes: This attribute contains the calling end peak water mark for the total number of ingress bytes. This peak occurs at the interval *callingTotalPwmIntervalNumber*. If the value of this attribute is zero (no traffic), *callingTotalPwmIntervalNumber* is also zero.

callingEirPwmBytes: This attribute contains the calling end peak water mark for the number of ingress bytes carried on frames with DE-bit = 1. The DE-bit is set by the user's equipment. This peak occurs at the interval *callingEirPwmIntervalNumber*. If the value of this attribute is zero (no traffic), *callingEirPwmIntervalNumber* is also zero.

callingCirPwmBytes: This attribute contains the calling end peak water mark for the number of ingress bytes carried on frames with DE-bit = 0. The DE-bit is set by the user's equipment. This peak occurs at the interval *callingCirPwmIntervalNumber*. If the value of this attribute is zero (no traffic), *callingCirPwmIntervalNumber* is also zero.

calledTotalPwmBytes: This attribute contains the called end peak water mark for the total number of ingress bytes. This peak occurs at the interval *calledTotalPwmIntervalNumber*. If the value of this attribute is zero (no traffic), *calledTotalPwmIntervalNumber* is also zero.

calledEirPwmBytes: This attribute contains the called end peak water mark for the number of ingress bytes carried on frames with DE-bit = 1. The DE-bit is set by the user's equipment. This peak occurs at the interval *calledEirPwmIntervalNumber*. If the value of this attribute is zero (no traffic), *calledEirPwmIntervalNumber* is also zero.

calledCirPwmBytes: This attribute contains the called end peak water mark for the number of ingress bytes carried on frames with DE-bit = 0. The DE-bit is set by the user's equipment. This peak occurs at the interval *calledCirPwmIntervalNumber*. If the value of this attribute is zero (no traffic), *calledCirPwmIntervalNumber* is also zero.

callingTotalPwmIntervalNumber: This attribute contains the number of the pwm (peak water mark) interval in which the *callingTotalPwmBytes* occurred.

If no traffic is sent (*callingTotalPwmBytes* is zero), the value of this attribute is also zero. The maximum possible value is 720 if *peakWaterMarkInterval* is set to one minute and the accounting period measured is 12 hours.

Interval numbers count down from the current change-over, ending with interval number one, which is the interval just prior to the next TODA change-over. For example, for a TODA period between 16:00 and 17:00, with a pwm interval time set to 5 minutes, if the traffic peak is reported in interval 9, the peak occurred between 16:15 and 16:20.

callingEirPwmIntervalNumber: This attribute contains the number of the pwm (peak water mark) interval in which the *callingEirPwmBytes* occurred.

If no traffic is sent (*callingEirPwmBytes* is zero), the value of this attribute is also zero. The maximum possible value is 720 if *peakWaterMarkInterval* is set to one minute and the accounting period measured is 12 hours.

Interval numbers count down from the current change-over, ending with interval number one, which is the interval just prior to the next TODA change-over. For example, for a TODA period between 16:00 and 17:00, with a pwm interval time set to 5 minutes, if the traffic peak is reported in interval 9, the peak occurred between 16:15 and 16:20.

callingCirPwmIntervalNumber: This attribute contains the number of the pwm (peak water mark) interval in which the *callingCirPwmBytes* occurred.

If no traffic is sent (*callingCirPwmBytes* is zero), the value of this attribute is also zero. The maximum possible value is 720 if *peakWaterMarkInterval* is set to one minute and the accounting period measured is 12 hours.

Interval numbers count down from the current change-over, ending with interval number one, which is the interval just prior to the next TODA change-over. For example, for a TODA period between 16:00 and 17:00, with a pwm interval time set to 5 minutes, if the traffic peak is reported in interval 9, the peak occurred between 16:15 and 16:20.

calledTotalPwmIntervalNumber: This attribute contains the number of the pwm (peak water mark) interval in which the *calledTotalPwmBytes* occurred.

If no traffic is sent (*calledTotalPwmBytes* is zero), the value of this attribute is also zero. The maximum possible value is 720 if *peakWaterMarkInterval* is set to one minute and the accounting period measured is 12 hours.

Interval numbers count down from the current change-over, ending with interval number one, which is the interval just prior to the next TODA change-over. For example, for a TODA period between 16:00 and 17:00, with a pwm interval time set to 5 minutes, if the traffic peak is reported in interval 9, the peak occurred between 16:15 and 16:20.

calledEirPwmIntervalNumber: This attribute contains the number of the pwm (peak water mark) interval in which the *calledEirPwmBytes* occurred.

If no traffic is sent (*calledEirPwmBytes* is zero), the value of this attribute is also zero. The maximum possible value is 720 if *peakWaterMarkInterval* is set to one minute and the accounting period measured is 12 hours.

Interval numbers count down from the current change-over, ending with interval number one, which is the interval just prior to the next TODA change-over. For example, for a TODA period between 16:00 and 17:00, with a pwm interval time set to 5 minutes, if the traffic peak is reported in interval 9, the peak occurred between 16:15 and 16:20.

calledCirPwmIntervalNumber: This attribute contains the number of the pwm (peak water mark) interval in which the *calledCirPwmBytes* occurred.

If no traffic is sent (*calledCirPwmBytes* is zero), the value of this attribute is also zero. The maximum possible value is 720 if *peakWaterMarkInterval* is set to one minute and the accounting period measured is 12 hours.

Interval numbers count down from the current change-over, ending with interval number one, which is the interval just prior to the next TODA change-over. For example, for a TODA period between 16:00 and 17:00, with a pwm interval time set to 5 minutes, if the traffic peak is reported in interval 9, the peak occurred between 16:15 and 16:20.

Time change records

Time change records contain attributes that describe events caused by a change of time on the switch. When the time on the switch is adjusted by 2 seconds, or more, the time change record is issued and sent as part of the accounting stream. The time change record is used to keep track of clock changes on the switch and to explain potential inconsistencies in the accounting records.

DPN time change record field descriptions

The *Time change record fields* of the BDF accounting record for DPN are:

oldDateTime: The date and time on the switch before the time was adjusted.

newDateTime: The date and time on the switch after the time was adjusted.

The format is `YYYYMMDDThhmmss` (year, month, day, T=delimiter, hours, minutes, seconds).

Accounting records are generated if the time change skips over the time set in Time of Day Accounting (TODA). For example, if TODA is set to 15:00 hours and the time change goes from 14:59 to 15:05, accounting records are generated. The TODA time is defined in the *collectionTimes* attribute of the *Collector/Accounting* component.

Note: Raw DPN records contain 2-digit years. The MDP BDF converter resolves these 2-digit years into 4-digit years using the following algorithm based on windows set by DPN:

year = YY
YY >= 50 (indicates 20th century (19xx))
YY < 50 (indicates 21st century (20xx))

namsId: The NAMS ID of the switch that issued the time change record. The value of the NAMS ID is provisioned on the switch.

Optional fields

If an *optional* facility is used on a call, the information on the optional facility is captured in the optional facility section of the accounting record. If the optional facility is not used on the call, then that particular optional facility section will not appear in the BDF accounting record for that call.

The terms *calling* and *called* are used as part of some of the optional field names. The term *calling* end refers to that part of the switch interface that initiated the call request. The term *called* end refers to that part of the switch interface that received the call request. For PVCs, the Master end initiated the call. For SVCs, the calling device initiated the call.

The following optional facilities are available:

- “NUI option” (page 106)
- “Free-format NUI option” (page 106)
- “Inter-network option” (page 106)
- “Original Called Address option” (page 108)
- “GAS option” (page 108)
- “National CUG option” (page 109)
- “International CUG option” (page 110)
- “Sensor identifier option” (page 112)
- “X.75 interface identifier option” (page 112)
- “Gateway option” (page 112)
- “Videotex option” (page 112)
- “Routing class of service option” (page 113)
- “X.75 interface identifier option” (page 112)

- “Frame Relay options” (page 114)
- “Extended national address option” (page 113)
- “Customized accounting information option” (page 123)

NUI option

The *NUI* option contains information about the Network User Identifier (NUI) used on a DPN-100 call. This information exists in the *nuiData* field of the DPN-100 accounting record.

nuiType: The first portion of the *nuiData* field describes the type of NUI.

- 1 - DPN-100 NUI
- 2 - Calling Card NUI
- 3 - Suppressed NUI

nui: The second portion of the *nuiData* field contains a DPN-100 NUI or a Calling Card NUI. The DPN-100 NUI is a variable length character string, with a maximum of 8 characters. The Calling Card NUI has a fixed length of 8 characters. Suppressed NUI has a length of zero.

Free-format NUI option

If the *Free-format NUI* option is used for a DPN-100 connection, 2 BDF fields are populated with information about the free-format Network User Identifier (NUI) used on the call. If the Free-format NUI option is not used, these 2 fields are empty.

billingIdType: This field contains a decimal value of zero, as there is currently only one type of free-format NUI billing identifier available.

billingId: This field contains a DPN-100 free-format NUI. The DPN-100 free-format NUI is a variable length character string, with a maximum of 41 characters.

Inter-network option

The *Inter-network* option contains the additional accounting information related to inter-network calls. The information includes: Gateway Identifiers, Tariffs, Clear Network Identifier, and Transit Network Identifier.

The number of TNICs, Transit Network Identifiers, that appear in this section is variable. It will be the exact number specified in the TOTAL_TNICS field. The maximum number of TNIC values is 10, contained in TNIC(1) to TNIC(10). No TNIC fields are included in the accounting record if the TOTAL_TNICS has the value zero, which means that there are no TNICs.

incomingGatewayId: This field identifies the incoming (calling) Gateway ID on an X.75 gateway call. The incoming (calling) gateway is where a call request packet is received from another network. The value ranges from 0 to 9999.

outgoingGatewayId: This field identifies the outgoing (called) Gateway ID on an X.75 gateway call. The outgoing (called) gateway is where a call request packet is sent to another network. The value ranges from 0 to 9999.

clearNetworkId: This field contains the Clearing Network Id Code. The value ranges from 0000 to 9999. The initialization value FFFF can be inserted into the record to indicate the absence of the CNIC in the clear block.

totalTnics: This field indicates the number of inter-networks. The value ranges from 0 to 10.

callingTariffValid: This field indicates the validity of the *callingTariff* value. 1 indicates the *callingTariff* is valid, while 0 indicates the *callingTariff* is invalid.

calledTariffValid: This field indicates the validity of the *calledTariff* value. 1 indicates the *calledTariff* is valid, while 0 indicates the *calledTariff* is invalid.

callingTariff: This field contains the value of the calling tariff utility supplied in the incoming call request packet from the X.75 link.

calledTariff: This field contains the value of the called tariff utility supplied in the call accept packet or call clear packet (in direct response to the call packet) from the X.75 link. This field is in the same format as the *clgTariff* field.

transitNetworkId: The transit network identifier (tnic). There are 1 to 10 identifiers depending on the value of total TNICs. The value ranges from 0000 to 9999.

Original Called Address option

The *Original called address* option provides the original called address if the call was redirected or hunted.

originalNpi: This field is the Numbering Plan Indicator to indicate the numbering plan used in the original called address. This field has the same format as the *callingNpi* field.

origAddrLen: This portion of the *originalAddress* field indicates the length of the address (DNA) in the *origAddr* field. The maximum is 15.

origAddr: This portion of the *originalAddress* field indicates the original called address (that is, the called address before the called DTE address was modified). The actual called DTE address is recorded in the *calledAddress* field. This field has the same format as the *calledAddress* field.

GAS option

The *GAS* option provides information on the *GAS*, Gateway Access Service, for the Transtext Universal Gateway used particularly by the BOC. The 52 bytes of *GAS* information in the *GAS* optional facility section of the Bulk data format accounting record are extracted directly from the 52 bytes of *GAS* optional facility data that appear in the Bulk data format accounting record when the facility is used.

gasData: A 52-byte data field that provides information on the Gateway Access Service (*GAS*) for the Transtext Universal Gateway.

National CUG option

The *National CUG* option contains the CUG Interlock code of the national CUG where the call is established.

cugInterlockCode: This field contains the CUG Interlock Code of the CUG where the call is established. The value ranges from 0 to 65535.

cugFlags: This is a two-byte binary field that contains four one-bit flags as follows:

BIT 0	- Outgoing Access(cugFlags_outgoingAccess)	(OA)
BIT 1	- IA/MA Bits Valid(cugFlags_validIaMaBits)	(VA)
BIT 2	- Incoming Access(cugFlags_incomingAccess)	(IA)
BIT 3	- CUGs Matched(cugFlags_cugsMatch)	(MA)

These one-bit flags have the values 1 or 0 as follows.

- 1 Outgoing Access (OA)(cugFlags_outgoingAccess)
The field indicates if the originator specified Outgoing Access (OA) on a call from a Closed User Group:
0 indicates OA is NOT used
1 indicates OA is used
- 2 IA/MA Bits Valid (VA)(cugFlags_validIaMaBits)
This field indicates if the IA/MA bits are valid:
0 indicates IA/MA bits are not valid
1 indicates IA/MA bits are valid
- 3 Incoming Access (IA)(cugFlags_incomingAccess)
This field specifies if the destination of a call from a Closed User Group has Incoming Access (IA):
0 indicates destination does NOT have IA
1 indicates destination has IA.

4 CUGs Matched (MA)(cugFlags_cugsMatch)

This field specifies if the CUGs matched at the destination of a call from a Closed User Group:

0 indicates CUGs do NOT match

1 indicates CUGs do match

Note 1: The value for CODE is 5, which indicates the NCUG optional facility section.

Note 2: The value for LENGTH is 6 to indicate that six bytes of data (*cugInterlockCode* and *cugFlags*) follow the CODE/LENGTH field.

Note 3: The value for *cugInterlockCode* is extracted from the same field in the switch national CUG optional facility section.

Note 4: The range for *cugInterlockCode* is 0 to 65535.

International CUG option

The *International CUG* option contains the CUG Interlock Code, along with the DNIC, of the international CUG where the call is established.

cugDnic: This field contains the DNIC of the International CUG where the call is established. It has the same format and value ranges as TNIC.

cugInterlockCode: This field contains the CUG Interlock Code of the CUG where the call is established. The value ranges from 0 to 65535.

cugFlags: This is a two-byte binary field that contains four one-bit flags as follows:

BIT 0	- Outgoing Access(icugFlags_outgoingAccess)	(OA)
BIT 1	- IA/MA Bits Valid(icugFlags_validIaMaBits)	(VA)
BIT 2	- Incoming Access(icugFlags_incomingAccess)	(IA)
BIT 3	- CUGs Matched(icugFlags_cugsMatch)	(MA)

These one-bit flags have the values 1 or 0 as follows.

- 1 **Outgoing Access (OA)**(icugFlags_outgoingAccess)
The field indicates if the originator specified Outgoing Access (OA) on a call from a Closed User Group:
0 indicates OA is NOT used
1 indicates OA is used
- 2 **IA/MA Bits Valid (VA)**(icugFlags_validIaMaBits)
This field indicates if the IA/MA bits are valid:
0 indicates IA/MA bits are not valid
1 indicates IA/MA bits are valid
- 3 **Incoming Access (IA)**(icugFlags_incomingAccess)
This field specifies if the destination of a call from a Closed User Group has Incoming Access (IA):
0 indicates destination does NOT have IA
1 indicates destination has IA.
- 4 **CUGs Matched (MA)**(icugFlags_cugsMatch)
This field specifies if the CUGs matched at the destination of a call from a Closed User Group:
0 indicates CUGs do NOT match
1 indicates CUGs do match

Note 1: The value for CODE is 6 to indicate the ICUG optional facility section.

Note 2: The value for LENGTH is 9 to indicate that nine bytes of data (*cugDnic*, *cugInterlockCode* and *cugFlags*) follow the CODE/LENGTH field.

Note 3: The values for *cugDnic* and *cugInterlockCode* are extracted from the same fields in the switch international CUG optional facility section.

Note 4: The range for *cugInterlockCode* is 0 to 65535.

Note 5: The *cugFlags* field is described previously in National CUG layout.

Sensor identifier option

The *Sensor identifier* option contains the Sensor identifiers of the calling and called DTEs. The ISDN/PPSN bit is indicated in the 23rd bit of the integer and is extracted by dividing the value by 8388608.

callingIsdn: A value of 0 indicates the calling DTE is PPSN and a value of 1 indicates the calling DTE is ISDN.

callingSensorId: The sensor identifier of the calling DTE.

calledIsdn: A value of 0 indicates the called DTE is PPSN and a value of 1 indicates the called DTE is ISDN.

calledSensorId: The sensor identifier of the called DTE.

X.75 interface identifier option

The *X.75 interface identifier* option contains the incoming and outgoing X.75 interface identifiers. Zero is used to indicate that no value exists.

incomingX75ii: The incoming X.75 interface identifier.

outgoingX75ii: The outgoing X.75 interface identifier.

Gateway option

The *Gateway* option contains the type of gateways traversed.

incomingGtwyType: The incoming gateway type.

outgoingGtwyType: The outgoing gateway type.

Videotex option

The 90 bytes of Videotex information in the *Videotex* option are extracted directly from the 90 bytes of Videotex optional facility data that appear in the Bulk data format accounting record when the facility is used.

videotexData: A 90-byte data field.

Routing class of service option

The *Routing class of service* option provides information about the Global Routing Class of Service (RCOS) settings on the called and calling ends of a call.

acctClgRcos: This field is two bits indicating the class of service of the calling end.

- Bit 1(callingRcos_normalReliability) is the normal reliability bit. Normal reliability RCOS is used if it is set; otherwise high reliability RCOS is used.
- Bit 0(callingRcos_minimumDelay) is the minimum delay bit. Delay RCOS is used if it is set; otherwise it uses the throughput RCOS.

acctCldRcos: This field is two bits indicating the class of service of the called end.

- Bit 0(calledRcos_minimumDelay) is the minimum delay bit. Delay RCOS is used if it is set; otherwise it uses the throughput RCOS.
- Bit 1(calledRcos_normalReliability) is the normal reliability bit. Normal reliability RCOS is used if set; otherwise high reliability RCOS is used.

Extended national address option

The *Extended national address* option contains the Extended national address digits for the calling, called, and original called addresses.

The Extended national address facility will be included in the accounting record only when Extended national addressing is enabled, and the given call uses an extended national address.

callingAddLength: The calling address extension.

calledAddLength: The called address extension.

orgCalledAdd: The extension called address.

Frame Relay options

The BDF for *frame relay* option is the base part of the traditional BDF, plus two frame relay-specific optional facilities, and one Original Called Address optional facility (when call redirection is supported).

All frame relay usage data is contained in two optional sections: the local (*calling*) section and the remote (*called*) section. The local section contains the last snapshot of frame relay data from the local (*calling*) end taken by the frame relay protocol. The remote section contains the last snapshot of frame relay data from the remote (*called*) end taken by the frame relay protocol.

The terms *calling* and *called* are used as part of some of the frame relay field names. The term *calling* refers to the calling end, which is that part of the switch interface that initiated the call request. The term *called* refers to the called end, which is that part of the switch interface that received the call request. For PVCs, the Master end initiated the call. For SVCs, the calling device initiated the call.

In normal conditions, the calling end will generate both the calling and the called records, and the called end will not generate any records. In unusual circumstances, such as when the trunk is down, the calling end will generate both calling and called records, and the called end will also generate both calling and called records. However, the remote section may be accurate only up to the last count exchange, which could take place up to 15 minutes prior.

For the byte counts described, the frame header, frame trailer, DLCI header, and CRC bytes are not included.

For further details on frame relay refer to the following document
241-1001-120 DPN-100 Frame Relay Service User Guide and Specification

Local (calling) Frame Relay field descriptions

The following are the field descriptions for the local (*calling*) portion of the frame relay option.

EP(callingClassOfService_emissionPri): the emission priority. The emission priority lets you choose between two distinct classes of services. This value is provisioned. If set to 1, high emission priority is selected for low delay. If set to 0, normal emission priority is selected.

RAP (callingRateAdaptation_provisioned): The rate adaptation is a congestion control mechanism. This value is provisioned. If set to 1, the rate adaptation was provisioned “on”. If set to 0, the rate adaptation was provisioned “off”. Rate adaptation can only be turned on if rate enforcement is also turned on.

RAO (callingRateAdaptation_occured): If set to 1, rate adaptation occurred on the interface. If set to 0, rate adaptation did not occur on the interface during this accounting report interval.

RE (callingRateAdaptation_enforced): the rate enforcement. Rate Enforcement prevents one user from using an inequitable share of the network’s resources by ensuring that user traffic follows the provisioned CIR and EIR rates. If set to 1, the rate enforcement was provisioned “on”. If set to 0, the rate enforcement was provisioned “off”.

UNI (flags_callingUni): If set to 1, the interface is UNI. If set to 0, the interface is NNI.

flags_callingFramesOrSegments (DPN): If set to 1, the accounting record contains frame counts. If set to 0, the accounting record contains segment counts.

AC (flags_callingAbit): If set to 1, the A-bit to or from the interface was changed on the interface during the current accounting period. If set to 0, the A-bit was not changed.

callingCir: The calling end provisioned CIR (Committed Information Rate). It is rounded to the next 800 bits per second.

callingEir: The calling end provisioned EIR (Excess Information Rate). It is provisioned as excessive burst size and is rounded to the next 800 bytes per second.

callingEgressBytes: The total number of bytes received at the calling end dlcI. This is the number of bytes sent by frameRelay dlcI to the application. With no congestion and no errors, this number will be equal

to the total number of ingress bytes (*calledIngressBytes*) at the called end of vc connection. The value for this field is metered at point E as shown in “Accounting meter reference points” (page 122).

callingIngressBytes: The total number of bytes sent at the calling end dlci. This is the number of bytes sent by the application to frameRelay dlci. With no congestion and no errors, this number will be equal to the total number of egress bytes (*calledEgressBytes*) at the called end of vc connection. Only good frames are passed to dlci from the application. The frames discarded due to CRC error are not included in this count. It does include the count *callingDiscardBytes*. The value for this field is metered at point A as shown in “Accounting meter reference points” (page 122).

callingEirEgressBytes: The total number of EIR bytes received from the network at the calling end with discard eligibility bit set. This total is included in the value of *callingEgressBytes* at the called end of the connection. Note that discard eligibility bit can be set by application or by frameRelay. The value for this field is metered at point D as shown in “Accounting meter reference points” (page 122).

callingEirIngressBytes: The total number of EIR bytes sent to the network at the calling end from application to frameRelay dlci. This total is included in *callingIngressBytes* value. It is the number of bytes or frames that have the discard eligibility bit set by the application. The value for this field is metered at point C as shown in “Accounting meter reference points” (page 122).

callingEgressFrames and **callingEgressSegments:** The total number of frames/segments sent at the calling end. This is the total number of frames/segments sent by frameRelay dlci to the application. Frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. A billing system uses

flags_callingFramesOrSegments to determine whether this count reports frames or segments. The value for this field is metered at point E as shown in “Accounting meter reference points” (page 122).

callingIngressFrames and **callingIngressSegments**: The total number of frames/segments sent at the calling end. This is the number of frames/segments sent by the application to frameRelay dlci. The frames discarded due to CRC error are not included in this count. Frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. The value for this field is metered at point A as shown in “Accounting meter reference points” (page 122).

callingEirEgressFrames or **callingEirEgressSegments**: The total number of EIR frames/segments received at the calling end. This is the total number of frames/segments received by frameRelay dlci from the network with discard eligibility bit set. This number is included in *callingEgressFrames/callingEgressSegments*. Note that discard eligibility bit can be set by application or by frameRelay. Frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. A billing system uses *flags_callingFramesOrSegments* to determine whether this count reports frames or segments. The value for this field is metered at point D as shown in “Accounting meter reference points” (page 122).

callingEirIngressFrames or **callingEirIngressSegments**: The total number of EIR frames/segments sent to the network at the calling end. This is the total number of frames/segments sent by frameRelay dlci to the network with the discard eligibility bit set. Note that discard eligibility bit can be set by application or by frameRelay. Frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. A billing system uses *flags_callingFramesOrSegments* to determine whether this count reports frames or segments. The value for this field is metered at point C as shown in “Accounting meter reference points” (page 122).

callingDiscardedFrames or **callingDiscardedSegments**: The number of Ingress frames/segments discarded at the calling end for the following reasons: rate enforcement, A-Bit is off, Ingress frame is too long, or Ingress frame is too short. The value for this field is metered at point B as shown in “Accounting meter reference points” (page 122).

callingDiscardedBytes: The number of Ingress bytes discarded at the calling end for the following reasons: A-Bit is off, Ingress frame is too long, Ingress frame is too short, or discarded due to rate enforcement. The value for this field is metered at point B as shown in “Accounting meter reference points” (page 122).

callingTransferPriority: The transfer priority is a preference specified by an application according to its time-sensitivity requirement. Frames with high transfer priority are served by the network before the frames with normal priority. The transfer priority standard is defined to have 16 levels from 0 to 15, with 0 denoting the lowest priority and 15 the highest.

Remote (called) Frame Relay field descriptions

The following are the field descriptions for the remote (*called*) portion of the frame relay option.

EP (calledClassOfService_emissionPri): the emission priority. The emission priority lets you choose between two distinct classes of services. This value is provisioned. If set to 1, high emission priority is selected for low delay. If set to 0, normal emission priority is selected.

RAP (calledRateAdaptation_provisioned): The rate adaptation is a congestion control mechanism. This value is provisioned. If set to 1, the rate adaptation was provisioned “on”. If set to 0, the rate adaptation was provisioned “off”. Rate adaptation can only be turned on if rate enforcement is also turned on.

RAO (calledRateAdaptation_occured): If set to 1, rate adaptation occurred on the interface. If set to 0, rate adaptation did not occur on the interface during this accounting report interval.

RE (calledRateAdaptation_enforced): the rate enforcement. Rate Enforcement prevents one user from using an inequitable share of the network's resources by ensuring that user traffic follows the provisioned CIR and EIR rates. If set to 1, the rate enforcement was provisioned "on". If set to 0, the rate enforcement was provisioned "off".

UNI (flags_calledUni): If set to 1, the interface is UNI. If set to 0, the interface is NNI.

flags_calledFramesOrSegments (DPN): If set to 1, the accounting record contains frame counts. If set to 0, the accounting record contains segment counts.

AC (flags_calledAbit): If set to 1, the A-bit to or from the interface was changed on the interface during the current accounting period. If set to 0, the A-bit was not changed.

calledCir: The called end provisioned CIR (Committed Information Rate). It is rounded to the next 800 bits per second.

calledEir: The called end provisioned EIR (Excess Information Rate). It is provisioned as excessive burst size and is rounded up to the next 800 bytes per second.

calledEgressBytes: The total number of bytes received at the called end dlci. This is the number of bytes sent by frameRelay dlci to the application. With no congestion and no errors, this number will be equal to the total number of ingress bytes (*callingIngressBytes*) at the calling end of vc connection. The value for this field is metered at point E as shown in "Accounting meter reference points" (page 122).

calledIngressBytes: The total number of bytes sent at the called end dlci. This is the number of bytes sent by the application to frameRelay dlci. With no congestion and no errors, this number will be equal to the total number of egress bytes (*callingEgressBytes*) at the calling end of vc connection. Only good frames are passed to dlci from the application. The frames discarded due to CRC error are not included in this count. It

does include the count *calledDiscardBytes*. The value for this field is metered at point A as shown in “Accounting meter reference points” (page 122).

calledEirEgressBytes: The total number of EIR bytes received from the network at the called end with discard eligibility bit set. This total is included in the value of *calledEgressBytes* at the called end of the connection. Note that discard eligibility bit can be set by application or by frameRelay. The value for this field is metered at point D as shown in “Accounting meter reference points” (page 122).

calledEirIngressBytes: The total number of EIR bytes sent to the network at the called end from application to frameRelay dlcI. This total is included in *calledIngressBytes* value. It is the number of bytes or frames that have the discard eligibility bit set by the application. The value for this field is metered at point C as shown in “Accounting meter reference points” (page 122).

calledEgressFrames and **calledEgressSegments:** The total number of frames/segments sent at the called end. This is the total number of frames/segments sent by frameRelay dlcI to the application. Frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. A billing system uses *flags_calledFramesOrSegments* to determine whether this count reports frames or segments. The value for this field is metered at point E as shown in “Accounting meter reference points” (page 122).

calledIngressFrames and **calledIngressSegments:** The total number of frames/segments sent at the called end. This is the number of frames/segments sent by the application to frameRelay dlcI. The frames discarded due to CRC error are not included in this count. Frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. The value for this field is metered at point A as shown in “Accounting meter reference points” (page 122).

calledEirEgressFrames or **calledEirEgressSegments:** The total number of EIR frames/segments received at the called end. This is the total number of frames/segments received by frameRelay dlcI from the

network with discard eligibility bit set. This number is included in *calledEgressFrames/calledEgressSegments*. Note that discard eligibility bit can be set by application or by frameRelay. Frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. A billing system uses *flags_calledFramesOrSegments* to determine whether this count reports frames or segments. The value for this field is metered at point D as shown in “Accounting meter reference points” (page 122).

calledEirIngressFrames or **calledEirIngressSegments**: The total number of EIR frames/segments sent to the network at the called end. This is the total number of frames/segments sent by frameRelay dlci to the network with discard eligibility bit set. Note that discard eligibility bit can be set by application or by frame relay. Frames or segments to be counted is provisioned as attribute *unitsCounted* in the *ModuleData VirtualCircuitSystem* component. A billing system uses *flags_calledFramesOrSegments* to determine whether this count reports frames or segments. The value for this field is metered at point C as shown in “Accounting meter reference points” (page 122).

calledDiscardedFrames or **calledDiscardedSegments**: The number of Ingress frames/segments discarded at the called end for the following reasons: rate enforcement, A-Bit is off, Ingress frame is too long, or Ingress frame is too short. The value for this field is metered at point B as shown in “Accounting meter reference points” (page 122).

calledDiscardedBytes: The number of Ingress bytes discarded at the called end for the following reasons: A-Bit is off, Ingress frame is too long, Ingress frame is too short, or discarded due to rate enforcement. The value for this field is metered at point B as shown in “Accounting meter reference points” (page 122).

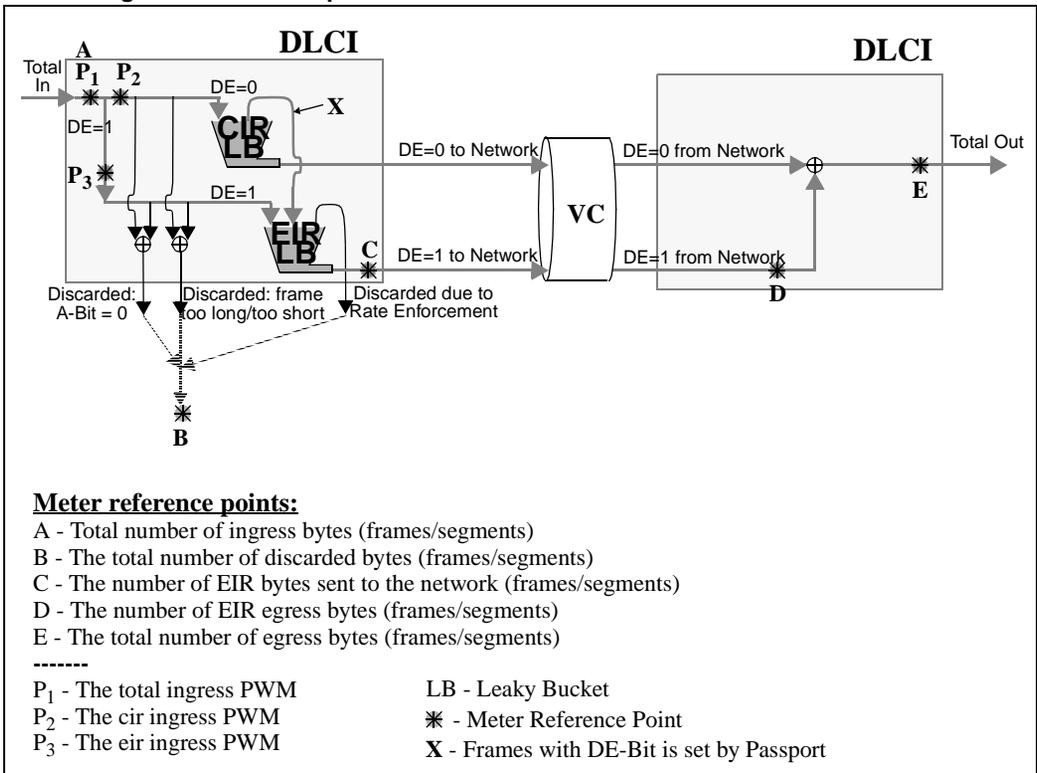
calledTransferPriority: The transfer priority is a preference specified by an application according to its time-sensitivity requirement. Frames with high transfer priority are served by the network before the frames with normal priority. The transfer priority standard is defined to have 16 levels from 0 to 15, with 0 denoting the lowest priority and 15 the highest.

Accounting meter reference points

The traffic at a DLCI or a VC is metered at reference points as shown in “Accounting meter reference points” (page 122). Although traffic is only shown in one direction in the diagram, the same metering of the DLCI-VC-DLCI set is done in both directions.

Note: “Accounting meter reference points” (page 122) only depicts traffic metering reference points at which accounting data is collected. In addition to values collected by the accounting meter, real-time statistics contains other data, such as the number of bytes/frames discarded when A-Bit is off, the number of frames discarded due to the frame being too short, the number of bytes/frames discarded due to the frame being too long, or the number of bytes/frames discarded due to rate enforcement.

Figure 8
Accounting meter reference points



Customized accounting information option

The customized accounting information option provides from 0 to 63 bytes of call information that usually comes from an off-network database. The data format is network specific. It can be in any format defined in the database providing the information. Currently, this information can be supplied by CSTIP databases that support customized accounting information. The output format of the customized accounting information is a hexadecimal string.

customAccInfo: is a variable length field with between 0 to 63 bytes of customized accounting information.

Chapter 3

DPN statistics records

Each DPN module in a network provides statistics information on network component availability, average and peak loads, and the number of packets received and transmitted. These DPN statistics enable coordinated network maintenance, provide an overview of network behavior, and provide a base for planning future improvements to subscriber service.

The following fields are common to DPN statistics records:

switchType: dpn

dataType: sta

subDataType: subtype #

dateTime: *yyyymmDDTHHMMSS.ss* yyyy=year; mm=month; DD=day; HH=hour; MM=minute; SS=seconds; ss=tenths of a second

customerId: Customer Network Management Identifier assigned to network components (PEs and POs) dedicated to a Virtual Private Network

dspId: Module DSP group ID

namsId: Integers which identify AM, RM, or PM

seqNum: Sequence number assigned by the network administration and management system (NAMS) device.

componentId: A component identifier describes a NAMS object. A component identifier is composed of a list of qualifiers; each with two elements—category (PE, PI, PO, CH) and a uniquely numbered name.

All other fields within BDF sta files are controlled by the 39file.

DPN statistics types quick reference

Table 24
DPN statistics types quick reference

Statistics group	Statistics name	Comment
AM/RM Office, PE, VC	"OfficeMaster (subtype 18)" (page 211) "PE286386 (subtype 30)" (page 214) "HighPerfPE (subtype 75)" (page 211) "VC (subtype 67)" (page 253) "GVCG34 (subtype 74)" (page 257)	
RM Servers	"CallRouterSrc (subtype 28)" (page 144) "CallRouterDst (subtype 29)" (page 142) "CallRedirSvrRSI (subtype 33)" (page 142) "NUIValidRemSvrRSI (subtype 34)" (page 210) "CallRouterGateDst (subtype 40)" (page 143) "CallRouterGateSrc (subtype 39)" (page 143)	
AM/RM Servers	"CallRedirSvr (subtype 32)" (page 141) "DialOutRouting (subtype 36)" (page 148) "NCSOpAgent (subtype 42)" (page 196) "NUIDatabaseInt (subtype 43)" (page 207)	
API Service	"APIDevice (subtype 19)" (page 134) "APILink (subtype 11)" (page 135)	
Asynchronous Services	"ITI (subtype 12)" (page 182) "ITIEnhanced (subtype 57)" (page 184)	G28 and earlier G29 and later
(Sheet 1 of 3)		

Table 24 (Continued)
DPN statistics types quick reference

Statistics group	Statistics name	Comment		
Frame Relay	"FrameRelay (subtype 59)" (page 149)	G2908, G3007, G31 and later		
	"ITI (subtype 12)" (page 182)			
ISDN, LAPD Service	"FrameRelayDLCIG33 (subtype 69)" (page 158)	G2908, G3007, G31 and later		
	"FrameRelayDLCIG34 (subtype 73)" (page 162)			
	"LAPDPhysical (subtype 23)" (page 187)			
	"LAPDSignalling (subtype 47)" (page 190)			
SNA, Token-Ring, 3270, SDLC Services	"ISDN DLCI (subtype 24)" (page 170)	G2908, G3007, G31 and later		
	"ISRB Phase 2 (subtype 68)" (page 172)			
	"3270TermPAD (subtype 20)" (page 131)			
	"3270HostPAD (subtype 22)" (page 128)			
	"SNALink (subtype 26)" (page 219)			
	"SNALogicalUnit (subtype 27)" (page 221)			
	"TokenRing (subtype 53)" (page 223)			
	"TokenRingDevice (subtype 52)" (page 226)			
	Trunk and Network Link		"NLLAPBEnhanced (subtype 58)" (page 197)	G29 and later
			"UTP (subtype 35)" (page 238)	RAUTP286 image
"UTPEnhanced (subtype 65)" (page 242)		UTP386 image only, G31 and later		
"NLOverFR (subtype 72)" (page 247)				
"TrunkRMUTP (subtype 25)" (page 229)		RAUTP286 image		
"TrunkRMUTPEnhanced (subtype 64)" (page 233)		UTP386 image only, G31 and later		
X25 Services		"X25MLAgent (subtype 61)" (page 274)	G30 and later	
(Sheet 2 of 3)				

Table 24 (Continued)
DPN statistics types quick reference

Statistics group	Statistics name	Comment
X75 Services	"X25MLSerPort (subtype 62)" (page 282)	G30 and later
	"X25Port (subtype 63)" (page 262)	G30 and later
	"X75PortG32 (subtype 70)" (page 300)	G32 and later
	"X75MLAgentG32 (subtype 71)" (page 293)	G32 and later
Custom Services	"GASTrtxtUnivGate (subtype 44)" (page 168)	
	"BOLDTermPAD (subtype 45)" (page 138)	
	"ACPPAD (subtype 76)" (page 140)	
	"SABRE (subtype 46)" (page 216)	
	"ClusterRemoteDU (subtype 51)" (page 145)	
(Sheet 3 of 3)		

3270HostPAD (subtype 22)

Table 25
3270 Host PAD statistics (subtype 22)

Field	Type
genPollCnt	integer
selectPollCnt	integer
selectCnt	integer
frmSnt	integer
frmRcv	integer
enqSnt	integer
enqRcv	integer
cmdAbort	integer
respUndlvrd	integer
frmRxmt	integer
(Sheet 1 of 2)	

Table 25 (Continued)
3270 Host PAD statistics (subtype 22)

Field	Type
crcVrcErr	integer
overruns	integer
underruns	integer
badFrmSize	integer
modemChanges	integer
protocolErr	integer
lrcErr	integer
noBuffer	integer
frmAbort	integer
maxCalls	integer
frmSsqSnt	integer
frmSsqRcv	integer
frmBytesSnt	integer
frmBytesRcv	integer
(Sheet 2 of 2)	

genPollCnt	The number of general polls which the HPAD has received.
selectPollCnt	The number of specific polls which the HPAD has received.
selectCnt	The number of selects which the HPAD has received.
frmSnt	The number of data frames that the HPAD has sent to the host.
frmRcv	The number of data frames which the HPAD has received from the host.
enqSnt	The number of BSC ENQ sequences sent to the host.

enqRcv	The number of BSC ENQ sequences received from the host.
cmdAbort	The number of times the host goes down or becomes inactive.
respUndlvrd	The number of device responses which failed to be delivered to the host.
frmRxmt	The number of frames which had to be retransmitted.
crcVrcErr	The number of vertical redundancy check errors and LRC errors received by the HPAD. Only applicable when the ASCII character set is used.
overruns	The number of times a received character was lost because the HPAD was busy processing the preceding character.
underruns	The number of times the HPAD was unable to provide a character to the link when it had completed output of the previous character.
badFrmSize	The number of frames received from the host which exceeded the maximum allowable frame.
modemChanges	The number of modem status changes detected.
protocolErr	The number of BSC.3 protocol errors detected from the host.
lrcErr	The number of frames which were available for transmission, but in which an LRC error prevented them from being transmitted.
noBuffer	The number of failures in obtaining a free block for incoming data.
frmAbort	The number of frames received with the last character being ENQ. This indicates a frame abort being sent from the host.
maxCalls	The maximum number of calls up in the 15 minute period.

frmSsqSnt	The sum square of the total number of transmitted bytes to the host.
frmSsqRcv	The sum square of the total number of received bytes from the host.
frmBytesSnt	The total number of transmitted data bytes.
frmBytesRcv	The total number of received data bytes.

3270TermPAD (subtype 20)

Table 26
3270 TPAD statistics (subtype 20)

Field	Type
genPollCnt	integer
selectPollCnt	integer
selectCnt	integer
frmSnt	integer
frmRcv	integer
enqSnt	integer
enqRcv	integer
respAbort	integer
cmdUndlvr	integer
frmRxmt	integer
crcVrcErr	integer
overruns	integer
underruns	integer
badFrmSize	integer
modemChanges	integer
protocolErr	integer
lrcErr	integer
(Sheet 1 of 2)	

Table 26 (Continued)
3270 TPAD statistics (subtype 20)

Field	Type
noBuffer	integer
frmAbort	integer
minbuffAbort	integer
frmSsqSnt	integer
frmSsqRcv	integer
frmBytesSnt	integer
frmBytesRcv	integer
(Sheet 2 of 2)	

genPollCnt	The number of times the TPAD has polled the controllers.
selectPollCnt	The number of times that the TPAD has requested status data from a controller.
selectCnt	The number of SELECTs performed by the TPAD.
frmSnt	The number of frames that the TPAD has sent to the controller.
frmRcv	The number of frames that the TPAD has received from the controller.
enqSnt	The number of BSC ENQ sequences sent to the controller.
enqRcv	The number of BSC ENQ sequences received from the controllers.
respAbort	The number of invalid message sequences which the TPAD received and responded with an abort sequence.
cmdUndlvrd	The number of commands sent by the TPAD which were not correctly acknowledged by the controllers.
frmRxmt	The number of frames which had to be retransmitted.

crcVrcErr	The number of cyclic redundancy check errors received by the TPAD. Only applicable when the EBCDIC character set is used.
overruns	The number of times a received character was lost because the TPAD was busy processing the preceding character.
underruns	The number of times the TPAD was unable to provide a character to the link when it had completed output of the previous character.
badFrmSize	The number of bad size frames detected. Frames were too long.
modemChanges	The number of modem status changes occurring in the last 15-minute period.
protocolErr	The number of BSC.3 protocol errors detected by the TPAD.
lrcErr	Total number of packets received with LRC errors.
noBuffer	The count of failures in obtaining a free block for incoming data.
frmAbort	The number of frames received with the last character being ENQ. This indicates a data check condition by the controller.
minbuffAbort	The number of minimum buffer aborts detected on the link.
frmSsqSnt	The sum square of the total number of transmitted bytes to the host.
frmSsqRcv	The sum square of the total number of received bytes from the host.
frmBytesSnt	The total number of transmitted data bytes.
frmBytesRcv	The total number of received data bytes.

APIDevice (subtype 19)

Table 27
API Device statistics (subtype 19)

Field	Type
nacksToSelects	integer
noRespToPoll	integer
nacksToInMsg	integer
nacksToOutMsg	integer
parityErr	integer
msgAbortRcv	integer
timeoutOnSelect	integer
enqSnt	integer
enqRcv	integer
timeoutRcv	integer
devSuspWindowClosure	integer
hostSntIntPkts	integer
padCallAtmpt	integer
dataMsgsRcv	integer
dataMsgsSnt	integer

nacksToSelects	The number of NAK responses to select.
noRespToPoll	The number of timeouts (No response to poll).
nacksToInMsg	The number of NAKs sent to input messages.
nacksToOutMsg	The number of NAK responses to output messages.
parityErr	The number of parity errors on control sequences.
msgAbortRcv	The number of message aborts (EOT received).
timeoutOnSelect	The number of timeouts on select.
enqSnt	The number of BSC ENQ sequences sent to the controllers.

enqRcv	The number of BSC ENQ sequences received from the controllers.
timeoutRcv	The number of timeouts received.
devSuspWindow Closure	The number of times the device was suspended due to window closure.
hostSntIntPkts	The number of times the device was suspended due to host sending Interrupt packet.
padCallAttmpt	The number of call attempts by the PAD, whether good or bad.
dataMsgsRcv	The number of incoming data messages from the device.
dataMsgsSnt	The number of outgoing data messages to the device.

APILink (subtype 11)

Table 28
API Link statistics (subtype 11)

Field	Type
frmSnt	integer
frmRcv	integer
crcErr	integer
lrcErr	integer
underruns	integer
overruns	integer
protocolErr	integer
zeroCnt	integer
breakErr	integer
frmErr	integer
longMsg	integer
incpltMsg	integer
(Sheet 1 of 2)	

Table 28 (Continued)
API Link statistics (subtype 11)

Field	Type
circuitReset	integer
errMsg	integer
frmBytesRcv	integer
frmBytesSnt	integer
frmSsqRcv	integer
frmSsqSnt	integer
histogramRcv	integer
histogramSnt	integer
(Sheet 2 of 2)	

frmSnt	The number of messages sent.
frmRcv	The number of messages received.
crcErr	The number of messages with cyclic redundancy check errors.
lrcErr	The number of messages with LRC errors.
underruns	The number of times the service was unable to provide a character to the link when it had completed output of previous character.
overruns	The number of times a received character was lost because the service was busy processing the preceding character.
protocolErr	The number of protocol violations detected by all logical links.
zeroCnt	Not used (always 0).
breakErr	Not used (always 0).
frmErr	The number of invalid frames received from device or sent to device.

longMsg	Host message received by the API is too long.
incpltMsg	Incomplete message received from device.
circuitReset	The number of circuit resets sent.
errMsg	The number of error messages sent and received.
frmBytesRcv	The total number of data characters received.
frmBytesSnt	The total number of data characters transmitted.
frmSsqRcv	The sum of the square of received message lengths.
frmSsqSnt	The sum of the square of transmitted message lengths.
histogramRcv	Histogram of data message length for received messages, in increments of 10 characters up to 250 characters and increments of 50 characters up to 500 characters. One additional field for messages greater than 500 characters.
histogramSnt	Histogram of data message length for sent messages, in increments of 10 characters up to 250 characters and increments of 50 characters up to 500 characters. One additional field for messages greater than 500 characters.

BOLDTermPAD (subtype 45)

Table 29
BOLD TermPAD link statistics (subtype 45)

Field	Type
frmSnt	integer
frmRcv	integer
enqSnt	integer
enqRcv	integer
frmRxmt	integer
crcErr	integer
overruns	integer
underruns	integer
badFrm	integer
modemChanges	integer
protocolErr	integer
noBuffer	integer
frmAbort	integer
frmSsqSnt	integer
frmSsqRcv	integer
frmBytesSnt	integer
frmBytesRcv	integer

frmSnt	The number of messages sent.
frmRcv	The number of messages received.
enqSnt	The number of BSC ENQ sequences sent to the host.
enqRcv	The number of BSC ENQ sequences received from the host.
frmRxmt	Frames recent to the DTE.
crcErr	CRC errors on frames from DTE.

overruns	The number of times a received character was lost because the service was busy processing the preceding character.
underruns	The number of times the service was unable to provide a character to the link when it had completed output of previous character.
badFrm	Frames from DTE that are either longer than allowed by service data, or are less than 4 octets long.
modemChanges	Modem status changes.
protocolErr	The total number of SWIFT BOLD PAD protocol violations for all PUs on this link during the past interval.
noBuffer	No message blocks are available for interrupt handle.
frmAbort	The number of frames received with the last character being ENQ. This indicates a frame abort being sent from the host.
frmSsqSnt	The sum of the square of the lengths of all RUs sent to the DTE (field length = 6 bytes).
frmSsqRcv	The sum of the square of the lengths of all RUs received from the DTE (field length = 6 bytes).
frmBytesSnt	The sum of the lengths of all RUs sent to the DTE (excludes CRC and flag) (field length = 4 bytes).
frmBytesRcv	The sum of the lengths of all RUs received from the DTE (excludes CRC and flag) (field length = 4 bytes).

ACPPAD (subtype 76)

Table 30
ACPPAD statistics (subtype 76)

Field	Type
lpcErrors	integer
overruns	integer
underruns	integer
badInputMessages	integer
modemChanges	integer
parityErrors	integer
lineUtl	integer
frmSnt	integer
frmRcv	integer
frmDisc	integer
xmitDataBytes	integer
rcvDataBytes	integer

lpcErrors	The number of received messages with LPC(C) error.
overruns	The number of times a received character was lost because the PAD was busy processing the preceding characters.
underruns	The number of times the PAD was unable to provide a character to the link when it had completed output of the previous character.
badInputMessages	The number of bad input messages.
modemChanges	The number of modem changes.
parityErrors	The number of received messages that contain a parity error.

lineUtl	This is a measure of the line utilization, and is in units of characters per second.
frmSnt	This is a measure of the total number of frames that have been transmitted.
frmRcv	This is a measure of the total number of frames that have been received.
frmDisc	This is a measure of the total number of frames that have been discarded.
xmitDataBytes	The total number of data bytes transmitted. This consists of the bytes contained between the STX and ETX character sequence of information messages.
rcvDataBytes	The total number of data bytes received. This consists of the bytes contained between the STX and ETX character sequence of information messages.

CallRedirSvr (subtype 32)

Table 31
Call redirection statistics (subtype 32)

Field	Type
redir	integer
redirFail	integer
dnaRedir	integer
dnicRedir	integer
numberAttempts	integer

redir	The number of calls received by the redirection server.
redirFail	The number of call redirection failures. Requests for DNA redirection not defined in the DNA redirection lists.
dnaRedir	The number of calls received for DNA redirection.

dnicRedir	The number of calls received for DNIC redirection.
numberAttempts	The number of call redirection attempts to redirection members in the list (member 1 to member 7).

CallRedirSvrRSI (subtype 33)

Table 32
Call redirection RSI statistics (subtype 33)

Field	Type
dnaReq	integer
ridReq	integer

dnaReq	The number of DNA requests handled.
ridReq	The number of requests handled by RID redirection.

CallRouterDst (subtype 29)

Table 33
Destination call router statistics (subtype 29)

Field	Type
callsProc	integer
callsFwd	integer
midCalls	integer
callsRedir	integer

callsProc	The number of calls processed.
callsFwd	The number of calls call-forwarded.
midCalls	The number of calls to MIDs.
callsRedir	The number of calls redirected.

CallRouterGateDst (subtype 40)

Table 34
Gateway destination call router statistics (subtype 40)

Field	Type
callsToGty	integer
callsToGscr	integer

callsToGty Number of calls routed to a gateway.

callsToGscr Number of calls sent to the GSCR for re-routing purposes.

CallRouterGateSrc (subtype 39)

Table 35
Gateway source call router statistics (subtype 39)

Field	Type
callsToRpoa	integer
barredCalls	integer
clgCldCalls	integer
dstAdrCalls	integer

callsToRpoa The number of calls routed by RPOA.

barredCalls The number of barred calls.

clgCldCalls The number of calls routed, based on the calling/
called prefix DNA pair.

dstAdrCalls The number of calls routed, based on the destination
network address.

CallRouterSrc (subtype 28)

Table 36
Source call router statistics (subtype 28)

Field	Type
callsProc	integer
ridCalls	integer
anidCalls	integer
nnidCalls	integer

callsProc The number of calls processed.

ridCalls The number of calls routed to a RID.

anidCalls The number of calls routed to an ANID.

nnidCalls The number of calls routed to an NNID.

ClusterRemoteDU (subtype 51)

One link level statistics record is maintained for the CRDU TPAD service. The statistics record is reported in response to a statistics probe. These statistics are reset after the service has responded to a statistics probe. The statistics are reported on a per TPAD basis.

Table 37
CRDU statistics (subtype 51)

Field	Type
nacks	integer
krpts	integer
parityErr	integer
msgTooLong	integer
msgExceed	integer
protocolErr	integer
modemChanges	integer
overruns	integer
underruns	integer
avgRespTime	integer
lineUtl	integer
bytesSnt	integer
bytesRcv	integer

nacks

The number of NACK messages that the TPAD relays from the CRDU to the AF Host.

krpts

The number of KRPT messages that the TPAD relays from the CRDU to the AF Host.

parityErr

The number of messages received by the TPAD from the CRDU which contain parity errors.

msgTooLong The number of messages received by the TPAD from the CRDU which exceed maximum message lengths defined in service data (either single or changed messages).

msgExceed A count of the number of occurrences of message response times exceeding the maximum normal response time defined in service data.

protocolErr A count of the occurrences of protocol errors detected by the TPAD.

Where protocol errors on messages received from the CRDU consist of:

- ELG field of the TC1 envelope is less than H.2B (i.e. TC1 envelope is too short).
- Message is not in a recognizable field specification format.
- Message from the CRDU is an invalid response to the message just sent to the CRDU.
- Message from a device on a CRDU configured as TC1, does not contain a TC1 envelope.

Alternatively, message from a device on a CRDU configured as TU, does contain a TC1 envelope.

Where protocol errors on messages received from the AF Host consist of:

- F0 (CONTROL MESSAGE) field of EAF has a value of 1.
- TERMINAL IRT field of the EAF does not match the configured IRT number of the TPAD that received this AF host message.
- TERMINAL TYPE field of the EAF does not match the configured type of CRDU for which this AF host message is intended.
- CRT# field of the EAF is not a configured device on the TPAD.
- LEN OF ENV field of the EAF is less than H.06.
- LGC field of the EAF is not equal to the length of the TEXT portion of the EAF message (including TC1, TEXT and ETX).
 - ELG field of the TC1 envelope is less than H.2B (i.e. TC1 envelope is too short).

modemChanges	The number of modem status changes occurring in the last 15-minute period.
overruns	The number of times a received character was lost because the TPAD was busy processing the preceding character.
underruns	The number of times the TPAD was unable to provide a character to the link when it had completed output of the previous character.
avgRespTime	Indicates the average response time of all response times gathered (from the CRDU) by the TPAD. This average is in units of tenths of a second.

lineUtl This is a measure of the line utilization, and is in units of characters per second. This line utilization is calculated as follows:

$$\text{Utilization (chars./sec.)} = (\text{TxBytes} + \text{RxBytes})/900 \text{ sec.}$$

where:

TxBytes: total number of transmitted data bytes

RxBytes: total number of received data bytes

900: represents the 15 minute measurement interval (in seconds).

bytesSnt The total number of data bytes transmitted to the CRDU. This consists of the bytes contained between the SOH and ETX character sequence of information messages.

bytesRcv The total number of data bytes received from CRDU. This consists of the bytes contained between the SOH and ETX/EOT character sequence of information messages (includes ACKs, NACKs and KRPTs).

DialOutRouting (subtype 36)

Table 38
Dial out routing statistics (subtype 36)

Field	Type
doCallReq	integer
doCallFail	integer
altDoAttmpt	integer
altDoFail	integer
prefixDoas	string

doCallReq	The number of dial-out call requests received.
doCallFail	The number of original dial-out call failures.
altDoAttmp	The number of alternate dial attempts.
altDoFail	The number of alternate dial-out failures.
prefixDoas	Variable length prefix/DOAS.

FrameRelay (subtype 59)

Table 39
Frame Relay statistics (subtype 59)

Field	Type
version	integer
service	integer
lclRid	integer
lclMid	integer
maxCalls	integer
linkSpeed	integer
currFrmQueued	integer
modType	integer
frmQueued	integer
overruns	integer
frmAbort	integer
noBuffer	integer
badFrmSize	integer
lrcErr	integer
crcErr	integer
cardErr	integer
underruns	integer
(Sheet 1 of 2)	

Table 39 (Continued)
Frame Relay statistics (subtype 59)

Field	Type
modemChanges	integer
protocolErr	integer
frmSnt	long integer
frmRcv	long integer
ifrmSnt	integer
ifrmRcv	integer
ifrmSsqSnt	integer
ifrmSsqRcv	integer
bytesSnt	integer
bytesRcv	integer
(Sheet 2 of 2)	

- version** This field indicates the version number of the statistics record, currently zero.
- service** This field indicates the access service with which this link is running. Values:
- 0 = Frame Relay Service
 - 1 = Network Link over Frame Relay.
- lclRid** This field specifies the routing id (RID) of the module on which this link is running. This field is 0 for an AM.
- lclMid** This field specifies the module id (MID) of the module on which this link is running. If 0, the module does not have a module id.
- maxCalls** This field indicates the maximum number of calls which were active over the frame relay service during the statistics interval.

linkSpeed	<p>This field specifies the speed of the link. The speed is denoted in kbit/s.</p> <p>For example, this field can contain a decimal value of 56, representing a link speed of 56,000 bit/s. A speed of 2,048,000 bit/s is represented by a decimal value of 2048.</p>
currFrmQueued	<p>This field specifies the number of frames queued for transmission to the link, when the statistics probe was processed by the link.</p>
modType	<p>This field specifies the type of module on which this link is running. If set to 3, then the module is an AM; if set to 2, then the module is an RM.</p>
frmQueued	<p>This field specifies the high-water-mark number of frames queued for transmission to the link when the statistics probe was processed by the link. If the number of frames queued is greater than 255, then this field has a value of 255, or H.FF. The true count of frames queued is in the statistical counter CURRENTFRAMESQD. This statistic usually indicates that the link is overloaded, and may have to be reengineered to handle the offered traffic.</p>

NOTE: This field is not as useful as the CURRENTFRAMESQD, but is used at the link for thresholding. The thresholds all have maximum values of 254, as there is no need in most cases to keep counting the number of link errors after hitting the upper bound. For example, the number of modem status changes in a statistics interval is counted until a value of 254 is reached, then the counting stops. In the case of frames queued, the true count of frames queued is available in CURRENTFRAMESQD.

overruns

This field specifies the number of overruns in this statistics interval on the link. The PE is unable to accept a frame before the next frame has been accepted by the receiver on the PI.

This statistic usually indicates that the PE is overloaded, and should be reengineered in order that the traffic capacity not be exceeded.

frmAbort

This field specifies the number of aborted frames in this statistics interval on the link.

This statistic usually indicates a problem at the remote end of the line (it is sending aborts), or on the facility (it is turning data on the line into aborts).

noBuffer

This field specifies the number of times in this statistics interval that the receive buffer used by the interrupt handler was insufficient to handle incoming traffic from the link.

This statistic usually indicates that the link is overloaded, and requires reengineering.

badFrmSize

This field specifies the number of frames received in this statistics interval from the link with a bad frame size (too long or too short).

This usually indicates a problem at the remote end of the line; it is sending frames of an incorrect size.

lrcErr

This field specifies the number of LRC errors in this statistics interval on the link.

This statistic usually indicates a data corruption problem somewhere in the module.

crcErr	<p>This field specifies the number of CRC errors in this statistics interval on the link.</p> <p>This statistic usually indicates a problem with the quality of the line.</p>
cardErr	<p>This field specifies the number of line errors in this statistics interval.</p> <p>This statistic usually indicates a problem on the SCC of the PI.</p>
underruns	<p>This field specifies the number of underruns in this statistics interval on the link. The interrupt handler is unable to give the next character to the SCC in time.</p>
modemChanges	<p>This field specifies the number of modem status changes in this statistic interval on the link.</p> <p>This statistic usually indicates a problem with the PI, the facility or the remote end. An unterminated cable plugged into the PI may also cause excessive modem status changes.</p>
protocolErr	<p>This field indicates the number of link protocol errors in this statistics interval on the link.</p> <p>This statistic, if coupled with other statistics errors, can indicate a problem on the hardware for the line. Otherwise, it indicates a problem with the remote end, where the protocol has not been implemented properly.</p>
frmSnt	<p>This field specifies the number of frames sent to the link. The count includes both the I-frames and the LMI frames.</p>
frmRcv	<p>This field specifies the number of frames received from the link. The count includes both the I-frames and the LMI frames.</p>

ifrmSnt	This field specifies the number of I-frames sent to the link. I-frames do not include LMI frames.
ifrmRcv	This field specifies the number of I-frames received from the link. I-frames do not include LMI frames.
ifrmSsqSnt	This field specifies the statistical sum of squares for the number of bytes in I-frames sent to the link in this statistics interval. I-frames do not include LMI frames.
ifrmSsqRcv	This field specifies the statistical sum of squares for the number of bytes in I-frames received from the link in this statistics interval. I-frames do not include LMI frames.
bytesSnt	This field specifies the total number of user bytes sent to the link in the statistics interval.
bytesRcv	This field specifies the total number of user bytes received from the link in the statistics interval.

FrameRelayDLCIEnhanced (subtype 66)

Table 40
Frame Relay DLCI Enhanced statistics (subtype 66)

Field	Type
service	integer
lclRid	integer
lclMid	integer
frmSnt	integer
frmRcv	integer
bytesSnt	integer
bytesRcv	integer
deFrmSnt	integer
deFrmRcv	integer
excessFrmRcv	integer
deBytesSnt	integer
deBytesRcv	integer
excessBytesRcv	integer
rateReductions	integer
reductionPeriod	integer
fecnSnt	integer
fecnRcv	integer
becnSnt	integer
becnRcv	integer
becnByPh	integer
discFrmEgressCong	integer
discFrmIngressCong	integer
discFrmRateEnforce	integer
(Sheet 1 of 2)	

Table 40 (Continued)
Frame Relay DLCI Enhanced statistics (subtype 66)

Field	Type
duplicatePkts	integer
pktsOutOfRange	integer
(Sheet 2 of 2)	

service	This field indicates the access service for this link. Values: 0 = Frame Relay Service.
lclRid	This field specifies the routing identifier (RID) of the module for this link. This field is 0 for an AM.
lclMid	This field specifies the module identifier (MID) of the module for this link. If zero, the module does not have a MID.
frmSnt	This field specifies the number of frames (DE = 0 and DE = 1) sent to the link.
frmRcv	This field specifies the number of frames (DE = 0 and DE = 1) accepted from the link.
bytesSnt	This field specifies the total number of user bytes in the frames (DE = 0 and DE = 1) sent to the link.
bytesRcv	This field specifies the total number of user bytes in the frames (DE = 0 and DE = 1) accepted from the link.
deFrmSnt	This field specifies the number of DE = 1 frames sent to the link.
deFrmRcv	This field specifies the number of (user-set) DE = 1 frames accepted from the link.
excessFrmRcv	This field specifies the number of DE = 0 frames accepted from the link, and subsequently set to DE = 1 by the PH.
deBytesSnt	This field specifies the number of bytes in the DE = 1 frames sent to the link.

deBytesRcv	This field specifies the number of bytes in the (user-set) DE = 1 frames accepted from the link.
excessBytesRcv	This field specifies the number of bytes in the DE = 0 frames accepted from the link, and subsequently set to DE = 1 by the PH.
rateReductions	This field indicates the number of times the allowable bandwidths have been reduced due to the rate adaptation. Once the counter reaches the high value, it will stay at 65535 until the next statistics probe.
reductionPeriod	This field indicates the accumulated time interval in 0.1-sec units during which the rate adaptation mechanism is in effect.
fecnSnt	This field specifies the number of FECN = 1 frames sent to the link. Once the counter reaches the high value, it will stay at 65535 until the next statistics probe.
fecnRcv	This field specifies the number of FECN = 1 frames accepted from the link. Once the counter reaches the high value, it will stay at 65535 until the next statistics probe.
becnSnt	This field specifies the number of BECN = 1 frames sent to the link. Once the counter reaches the high value, it will stay at 65535 until the next statistics probe.
becnRcv	This field specifies the number of BECN = 1 frames accepted from the link. Once the counter reaches the high value, it will stay at 65535 until the next statistics probe.
becnByPh	This field specifies the total number of BECN = 0 frames that the PH sets to BECN = 1 for all frames that have been sent to and receive from the link. Once the counter reaches the high value, it will stay at 65535 until the next statistics probe.

discFrmEgressCong	This field specifies the number of frames discarded at the egress access point due to congestion conditions. Once the counter reaches the high value, it will stay at 65535 until the next statistics probe.
discFrmIngress Cong	This field specifies the number of frames discarded at the ingress access point due to congestion conditions. Once the counter reaches the high value, it will stay at 65535 until the next statistics probe.
discFrmRateEnforce	This field specifies the total number of frames discarded at the ingress access point due to rate enforcement, or because the DLCI is inactive. Once the counter reaches the high value, it will stay at 65535 until the next statistics probe.
duplicatePkts	This field specifies the number of duplicated subnet packets received by the egress virtual circuit.
pktsOutOfRange	This field specifies the number of out-of-range subnet packets received by the egress virtual circuit.

FrameRelayDLCIG33 (subtype 69)

Table 41
Frame Relay DLCI G33 statistics (subtype 69)

Field	Type
version	integer
service	integer
lclRid	integer
lclMid	integer
frmSnt	integer
frmRcv	integer
bytesSnt	integer
bytesRcv	integer
deFrmSnt	integer
(Sheet 1 of 3)	

Table 41 (Continued)
Frame Relay DLCI G33 statistics (subtype 69)

Field	Type
deFrmRcv	integer
excessFrmRcv	integer
deBytesSnt	integer
deBytesRcv	integer
excessBytesRcv	integer
rateReductions	integer
reductionPeriod	integer
fecnSnt	integer
fecnRcv	integer
becnSnt	integer
becnRcv	integer
becnPhToSubnet	integer
becnPhToLink	integer
becnFromNetRcv	integer
fecnFromNetRcv	integer
frmToDteDisc	integer
deFrmToDteDisc	integer
bytesToDteDisc	integer
deBytesToDteDisc	integer
frmFromDteDisc	integer
deFrmFromDteDisc	integer
bytesFromDteDisc	integer
deBytesFromDteDisc	integer
discFrmRateEnforce	integer
duplicatePkts	integer
(Sheet 2 of 3)	

Table 41 (Continued)
Frame Relay DLCI G33 statistics (subtype 69)

Field	Type
pktsOutOfRange	integer
aBitChanges	integer
(Sheet 3 of 3)	

version	This field indicates the version number of the statistics record, currently zero.
service	The access service for this link. Values: 0 = Frame Relay Service.
lclRid	The routing identifier (RID) of the module for this link. This field is zero for an AM.
lclMid	The module identifier (MID) of the module for this link. If zero, the module does not have a MID.
frmSnt	The number of frames (DE = 0 and DE = 1) sent to the link.
frmRcv	The number of frames (DE = 0 and DE = 1) accepted from the link.
bytesSnt	The total number of user bytes in the frames (DE = 0 and DE = 1) sent to the link.
bytesRcv	The total number of user bytes in the frames (DE = 0 and DE = 1) accepted from the link.
deFrmSnt	The number of DE = 1 frames sent to the link.
deFrmRcv	The number of (user-set) DE = 1 frames accepted from the link.
excessFrmRcv	The number of DE = 0 frames accepted from the link, and subsequently set to DE = 1 by the PH.
deBytesSnt	The number of bytes in the DE = 1 frames sent to the link.
deBytesRcv	The number of bytes in the (user set) DE = 1 frames accepted from the link.

excessBytesRcv	The number of bytes in the DE = 0 frames accepted from the link, and subsequently set to DE = 1 by the PH.
rateReductions	The number of times the allowable bandwidths have been reduced due to rate adaptation. Once the counter reaches the high value, it stays at 65535 until the next statistics probe.
reductionPeriod	The accumulated time interval in 0.1-sec. units during which the rate adaptation mechanism is in effect.
fecnSnt	The number of FECN = 1 frames sent to the link. Once the counter reaches the high value, it will stay at 65535 until the next statistics probe.
fecnRcv	The number of FECN = 1 frames accepted from the link. Once the counter reaches the high value, it stays at 65535 until the next statistics probe.
becnSnt	The number of BECN = 1 frames sent to the link. Once the counter reaches the high value, it stays at 65535 until the next statistics probe.
becnRcv	The number of BECN = 1 frames accepted from the link. Once the counter reaches the high value, it stays at 65535 until the next statistics probe.
becnPhToSubnet	The number of BECNs sent to the subnet by the local Frame Relay service.
becnPhToLink	The number of BECNs sent to the link by the local Frame Relay service.
becnFromNetRcv	The number of BECNs received since the BCI bit was set by the network.
fecnFromNetRcv	The number of FECNs received since the FCI bit was set by the network.
frmToDteDisc	The number of frames discarded at egress due to local congestion.

deFrmToDteDisc	The number of frames with DE=1 discarded at egress due to local congestion.
bytesToDteDisc	The number of bytes discarded at egress due to local congestion.
deBytesToDteDisc	The number of bytes with DE=1 discarded at egress due to local congestion.
frmFromDteDisc	The number of frames discarded at ingress due to local congestion.
deFrmFromDteDisc	The number of frames with DE=1 discarded at ingress due to local congestion.
bytesFromDteDisc	The number of bytes discarded at ingress due to local congestion.
deBytesFromDteDisc	The number of bytes with DE=1 discarded at ingress due to local congestion.
discFrmRateEnforce	The total number of frames discarded at the ingress access point due to rate enforcement, or because the DLCI is inactive. Once the counter reaches the high value, it stays at 65535 until the next statistics probe.
duplicatePkts	The number of duplicated subnet packets received by the egress virtual circuit.
pktsOutOfRange	The number of out-of-range subnet packets received by the egress virtual circuit.
aBitChanges	The number of times the effective A-bit (for both directions) changes from active to inactive.

FrameRelayDLCIG34 (subtype 73)

Table 42
Frame Relay DLCI G34 statistics (subtype 73)

Field	Type
version	integer
service	integer
(Sheet 1 of 3)	

Table 42 (Continued)
Frame Relay DLCI G34 statistics (subtype 73)

Field	Type
lclRid	integer
lclMid	integer
frmSnt	integer
frmRcv	integer
bytesSnt	integer
bytesRcv	integer
deFrmSnt	integer
deFrmRcv	integer
excessFrmRcv	integer
deBytesSnt	integer
deBytesRcv	integer
excessBytesRcv	integer
rateReductions	integer
reductionPeriod	integer
fecnSnt	integer
fecnRcv	integer
becnSnt	integer
becnRcv	integer
becnPhToSubnet	integer
becnPhToLink	integer
becnFromNetRcv	integer
fecnFromNetRcv	integer
frmToDteDisc	integer
deFrmToDteDisc	integer
bytesToDteDisc	integer
(Sheet 2 of 3)	

Table 42 (Continued)
Frame Relay DLCI G34 statistics (subtype 73)

Field	Type
deBytesToDteDisc	integer
frmFromDteDisc	integer
deFrmFromDteDisc	integer
bytesFromDteDisc	integer
deBytesFromDteDisc	integer
discFrmRateEnforce	integer
discFrmTooShort	integer
discFrmTooLong	integer
discFrmInactive	integer
discBytesRateEnforce	integer
discBytesTooLong	integer
discBytesInactive	integer
duplicatePkts	integer
pktsOutOfRange	integer
aBitChanges	integer
(Sheet 3 of 3)	

version	This field indicates the version number of the statistics record, currently zero.
service	The access service for this link. Values: 0 = Frame Relay Service.
lclRid	The routing identifier (RID) of the module for this link. This field is zero for an AM.
lclMid	The module identifier (MID) of the module for this link. If zero, the module does not have a MID.
frmSnt	The number of frames (DE = 0 and DE = 1) sent to the link.

frmRcv	The number of frames (DE = 0 and DE = 1) accepted from the link.
bytesSnt	The total number of user bytes in the frames (DE = 0 and DE = 1) sent to the link.
bytesRcv	The total number of user bytes in the frames (DE = 0 and DE = 1) accepted from the link.
deFrmSnt	The number of DE = 1 frames sent to the link.
deFrmRcv	The number of (user-set) DE = 1 frames accepted from the link.
excessFrmRcv	The number of (user-set or PH set) DE = 1 frames accepted from the link.
deBytesSnt	The number of bytes in the DE = 1 frames sent to the link.
deBytesRcv	The number of bytes in the (user set) DE = 1 frames accepted from the link.
excessBytesRcv	The number of bytes in the (user-set or PH set) DE = 1 frames accepted from the link.
rateReductions	The number of times the allowable bandwidths have been reduced due to rate adaptation. Once the counter reaches the high value, it stays at 65535 until the next statistics probe.
reductionPeriod	The accumulated time interval in 0.1-sec. units during which the rate adaptation mechanism is in effect.
fecnSnt	The number of FECN = 1 frames sent to the link. Once the counter reaches the high value, it will stay at 65535 until the next statistics probe.
fecnRcv	The number of FECN = 1 frames accepted from the link. Once the counter reaches the high value, it stays at 65535 until the next statistics probe.
becnSnt	The number of BECN = 1 frames sent to the link. Once the counter reaches the high value, it stays at 65535 until the next statistics probe.

becnRcv	The number of BECN = 1 frames accepted from the link. Once the counter reaches the high value, it stays at 65535 until the next statistics probe.
becnPhToSubnet	The number of BECNs sent to the subnet by the local Frame Relay service.
becnPhToLink	The number of BECNs sent to the link by the local Frame Relay service.
becnFromNetRcv	The number of BECNs received since the BCI bit was set by the network.
fecnFromNetRcv	The number of FECNs received since the FCI bit was set by the network.
frmToDteDisc	The number of frames discarded at egress due to local congestion.
deFrmToDteDisc	The number of frames with DE=1 discarded at egress due to local congestion.
bytesToDteDisc	The number of bytes discarded at egress due to local PE congestion.
deBytesToDteDisc	The number of bytes with DE=1 discarded at egress due to local congestion.
frmFromDteDisc	The number of frames discarded at ingress due to local congestion.
deFrmFromDteDisc	The number of frames with DE=1 discarded at ingress due to local congestion.
bytesFromDteDisc	The number of bytes discarded at ingress due to local PE congestion.
deBytesFromDteDisc	The number of bytes with DE=1 discarded at ingress due to local congestion.
discFrmRateEnforce	The number of frames discarded due to rate enforcement.
discFrmTooShort	The number of frames discarded due to zero frame size.

discFrmTooLong	The number of frames discarded due to frame size being greater than the maximum frame size dN1.
discFrmInactive	The number of frames discarded due to the DLCI being inactive.
discBytesRate Enforce	The number of bytes discarded due to rate enforcement.
discBytesTooLong	The number of bytes discarded due to frame size being greater than the maximum frame size dN1.
discBytesInactive	The number of bytes discarded due to the DLCI being inactive.
duplicatePkts	The number of duplicated subnet packets received by the egress virtual circuit.
pktsOutOfRange	The number of out-of-range subnet packets received by the egress virtual circuit.
aBitChanges	The number of times the effective A-bit (for both directions) changes from active to inactive.

GASTrtxtUnivGate (subtype 44)

Table 43
GAS trtxt univ gate statistics (subtype 44)

Field	Type
newSmartSess	integer
maxSmartSess	integer
minSmartSess	integer
newDumbSess	integer
maxDumbSess	integer
minDumbSess	integer
newCsSess	integer
maxCsSess	integer
minCsSess	integer
newUavSess	integer
maxUavSess	integer
minUavSess	integer
newEumRedir	integer

- newSmartSess** This field indicates the total number of new smart sessions since the last probe.
- maxSmartSess** This field indicates the maximum number of smart sessions that occurred simultaneously since the last probe.
- minSmartSess** This field indicates the minimum number of smart sessions that occurred simultaneously since the last probe.
- newDumbSess** This field indicates the total number of new dumb sessions since the last probe.
- maxDumbSess** This field indicates the maximum number of dumb sessions that occurred simultaneously since the last probe.

minDumbSess	This field indicates the minimum number of dumb sessions that occurred simultaneously since the last probe.
newCsSess	This field indicates the total number of new CS sessions since the last probe.
maxCsSess	This field indicates the maximum number of new CS sessions that occurred simultaneously since the last probe.
minCsSess	This field indicates the minimum number of CS sessions that occurred simultaneously since the last probe.
newUavSess	This field indicates the total number of UAV queries completed since the last probe.
maxUavSess	This field indicates the maximum number of UAV queries that occurred simultaneously since the last probe.
minUavSess	This field indicates the minimum number of UAV queries that occurred simultaneously since the last probe.
newEumRedir	This field indicates the total number of EUM redirections completed since the last probe.

ISDN DLCI (subtype 24)

Table 44
ISDN DLCI statistics (subtype 24)

Field	Type
minCalls	integer
maxCalls	integer
maxNuiErr	integer
lclFlowCtrl	integer
rmtFlowCtrl	integer
frmSnt	integer
frmRcv	integer
badFrmSize	integer
protocolErr	integer
frmRxmt	integer
lrcErr	integer
frmQueued	integer
ifrmSnt	integer
ifrmRcv	integer
rrSnt	integer
rrRcv	integer
mrSnt	integer
mrRcv	integer
rejSnt	integer
rejRcv	integer
frmrSnt	integer
frmrRcv	integer
bytesSnt	integer
bytesRcv	integer

minCalls	Minimum number of calls up in a 15 minute period.
maxCalls	Maximum number of calls up in a 15 minute period.
maxNuiErr	Maximum number of NUI validation errors.
lclFlowCtrl	Total time interval which DPN access module has flow controlled on input (2 words).
rmtFlowCtrl	Total time interval which DPN access module has been flow controlled by the DTE (2 words).
frmSnt	Number of frames transmitted to the DTE, including information, supervisory, unnumbered and retransmitted frames, but not aborted frames (2 words).
frmRcv	Number of frames received from the DTE, including I, S and U frames, but not aborted frames, frames received with CRC errors or IBADFRMSIZE errors (2 words).
badFrmSize	Total number of frames received of the wrong size by level 2.
protocolErr	Total number of protocol violations detected.
frmRxmt	Total number of retransmits caused by an REJ or after a timeout on an information frame.
lrcErr	Total number of frames received with LRC errors.
frmQueued	Number of frames queued.
ifrmSnt	Number of I-frames transmitted.
ifrmRcv	Number of I-frames received.
rrSnt	Number of RRs transmitted.
rrRcv	Number of RRs received.
rnrSnt	Number of RNRs transmitted.
rnrRcv	Number of RNRs received.
rejSnt	Number of REJs transmitted.

rejRcv	Number of REJs received.
frmrSnt	Number of FRMRs transmitted.
frmrRcv	Number of FRMRs received.
bytesSnt	Number of bytes transmitted in I-frames information field (2 words).
bytesRcv	Number of bytes received in I-frames information field (2 words).

ISRB Phase 2 (subtype 68)

Table 45
ISRB Phase 2 statistics (subtype 68)

Field	Type
srbFrmFromLan	integer
srbFrmToSubnet	integer
srbFrmFromSubnet	integer
srbFrmToLan	integer
srbByteFromLan	integer
srbByteToSubnet	integer
srbByteFromSubnet	integer
srbByteToLan	integer
arbFrmFromLan	integer
arbFrmToSubnet	integer
arbFrmFromSubnet	integer
arbFrmToLan	integer
arbByteFromLan	integer
arbByteToSubnet	integer
arbByteFromSubnet	integer

(Sheet 1 of 4)

Table 45 (Continued)
ISRB Phase 2 statistics (subtype 68)

Field	Type
arbByteToLan	integer
srfFrmFromLan	integer
srfFrmToSubnet	integer
srfFrmFromSubnet	integer
srfFrmToLan	integer
srfByteFromLan	integer
srfByteToSubnet	integer
srfByteFromSubnet	integer
srfByteToLan	integer
frmDeBitOnIn	integer
frmDeBitOnOut	integer
frmLwvcMacCmprEligIn	integer
frmLwvcMacCmprIn	integer
frmLwvcMacCmprOut	integer
frmProxyUpdateIn	integer
frmProxyUpdateOut	integer
frmProxiedOut	integer
frmIbpduIn	integer
frmIbpduOut	integer
frmDiscSrbIn	integer
frmDiscSrbOut	integer
frmDiscRingFiltIn	integer
frmDiscRingFiltOut	integer
frmDiscDmacFiltIn	integer
frmDiscDmacFiltOut	integer
(Sheet 2 of 4)	

Table 45 (Continued)
ISRB Phase 2 statistics (subtype 68)

Field	Type
frmDiscSmacFiltIn	integer
frmDiscSmacFiltOut	integer
frmDiscDsapFiltIn	integer
frmDiscDsapFiltOut	integer
frmDiscSsapFiltIn	integer
frmDiscSsapFiltOut	integer
frmDiscInvFormatIn	integer
frmDiscInvFormatOut	integer
frmDiscInvIbpdu	integer
frmDiscREnforceIn	integer
frmDiscCongIn	integer
frmDiscCongOut	integer
frmDiscOtherIn	integer
frmDiscOtherOut	integer
frmDiscLwvcDown	integer
frmDiscExcessHopsIn	integer
frmDiscExcessHopsOut	integer
frmDiscOvrsizeLwvcIn	integer
frmDiscOvrsizeLwvcOut	integer
frmDiscOvrsizeBcstIn	integer
frmDiscOvrsizeBcstOut	integer
qToBcstOverflow	integer
frmDiscCompFiltIn	integer
frmDiscCompFiltOut	integer
(Sheet 3 of 4)	

Table 45 (Continued)
ISRB Phase 2 statistics (subtype 68)

Field	Type
frmDiscServREnforceln	integer
frmProxyDiscardln	integer
(Sheet 4 of 4)	

srbFrmFromLan	The number of single-route broadcast frames received from the physical Token-Ring LAN.
srbFrmToSubnet	The number of single-route broadcast frames sent to the DPN subnet.
srbFrmFromSubnet	The number of single-route broadcast frames received from the DPN subnet.
srbFrmToLan	The number of single-route broadcast frames sent to the Token-Ring LAN.
srbByteFromLan	The number of single-route broadcast frames (in bytes) received from the physical Token-Ring LAN.
srbByteToSubnet	The number of single-route broadcast frames (in bytes) sent to the DPN subnet.
srbByteFromSubnet	The number of single-route broadcast frames (in bytes) received from the DPN subnet.
srbByteToLan	The number of single-route broadcast frames (in bytes) sent to the Token-Ring LAN.
arbFrmFromLan	The number of all-routes broadcast frames received from the physical Token-Ring LAN.
arbFrmToSubnet	The number of all-routes broadcast frames sent to the DPN subnet.
arbFrmFromSubnet	The number of all-routes broadcast frames received from the DPN subnet.
arbFrmToLan	The number of all-routes broadcast frames sent to the Token-Ring LAN.

arbByteFromLan	The number of all-routes broadcast frames (in bytes) received from the physical Token-Ring LAN.
arbByteToSubnet	The number of all-routes broadcast frames (in bytes) sent to the DPN subnet.
arbByteFromSubnet	The number of all-routes broadcast frames (in bytes) received from the DPN subnet.
arbByteToLan	The number of all-routes broadcast frames (in bytes) sent to the Token-Ring LAN.
srfFrmFromLan	The number of specifically routed frames received from the physical Token-Ring LAN.
srfFrmToSubnet	The number of specifically routed frames sent to the DPN subnet.
srfFrmFromSubnet	The number of specifically routed frames received from the DPN subnet.
srfFrmToLan	The number of specifically routed frames sent to the Token-Ring LAN.
srfByteFromLan	The number of specifically routed frames (in bytes) received from the physical Token-Ring LAN.
srfByteToSubnet	The number of specifically broadcast frames (in bytes) sent to the DPN subnet.
srfByteFromSubnet	The number of specifically routed frames (in bytes) received from the DPN subnet.
srfByteToLan	The number of specifically routed frames (in bytes) sent to the Token-Ring LAN.
frmDeBitOnIn	The number of frames that came from the LAN and were sent to the DPN subnet with the discard eligible (DE) bit on. A frame is tagged with the DE bit when it passes through the service-level rate enforcement with credit from the Service Excess Burst Size.

frmDeBitOnOut	The number of frames that came from the DPN subnet for the LAN with the discard eligible (DE) bit on. A frame is tagged with the DE bit when it passes through the service level rate enforcement with credit from the Service Excess Burst Size at the remote ISRB.
frmLwvcMacCmprElign	The number of frames received from the LAN that are eligible for LWVC MAC header compression. The number includes both compressed and uncompressed frames. This field is updated only if LWVC MAC header compression is in use.
frmLwvcMacCmprIn	The number of frames received from the LAN that are subjected to LWVC MAC header compression before being sent through the LWVC. This field is updated only if LWVC MAC header compression is in use.
frmLwvcMacCmprOut	The number of frames received from the DPN subnet that have been MAC header compressed. This field is updated only if LWVC MAC header compression is in use.
frmProxyUpdateIn	The number of proxy update frames which have been sent to the Broadcast System. A proxy update frame may be a new proxied frame or a change to an existing proxied frame.
frmProxyUpdateOut	The number of proxy update frames which came from the Broadcast System. A proxy update frame may be a new proxied frame or a change to an existing proxied frame.
frmProxiedOut	The number of proxied frames sent to the LAN by this ISRB on behalf of the remote ISRBs.
frmIbpduln	The number of ISRB Bridge Protocol Data Units (IBPDU) received.

frmIbpdOut	The number of ISRB Bridge Protocol Data Units (IBPDU) sent to the physical Token-Ring network.
frmDiscSrbln	The number of single-route broadcast frames received from the LAN that have been discarded because the bridge is a not a single-route broadcast bridge.
frmDiscSrbOut	The number of single-route broadcast frames received from the DPN subnet that have been discarded because this ISRB is a not a single-route broadcast bridge.
frmDiscRingFiltIn	The number of frames received from the LAN that have been discarded because of ring filtering.
frmDiscRingFiltOut	The number of frames received from the DPN subnet that have been discarded because of ring filtering.
frmDiscDmacFiltIn	The number of frames received from the LAN that have been discarded because of destination MAC address filtering.
frmDiscDmacFiltOut	The number of frames received from the DPN subnet that have been discarded because of destination MAC address filtering.
frmDiscSmacFiltIn	The number of frames received from the LAN that have been discarded because of source MAC address filtering.
frmDiscSmacFiltOut	The number of frames received from the DPN subnet that have been discarded because of source MAC address filtering.
frmDiscDsapFiltIn	The number of frames received from the LAN that have been discarded because of destination SAP address filtering.

frmDiscDsapFiltOut	The number of frames received from the DPN subnet that have been discarded because of destination SAP address filtering.
frmDiscSsapFiltIn	The number of frames received from the LAN that have been discarded because of source SAP address filtering.
frmDiscSsapFiltOut	The number of frames received from the DPN subnet that have been discarded because of source SAP address filtering.
frmDiscInvFormatIn	<p>The number of frames received from the LAN that have been discarded because of the following conditions:</p> <ul style="list-style-type: none"> •Frame too short •Frame too long •Duplicate ring numbers in routing information •Invalid routing information field
frmDiscInvFormatOut	<p>The number of frames received from the DPN subnet that have been discarded because of the following conditions:</p> <ul style="list-style-type: none"> •Frame too short •Frame too long •Duplicate ring numbers in routing information •Invalid routing information field
frmDiscInvlb pdu	The number of invalid ISRB BPDUs that have been received and discarded.
frmDiscREnforceln	The number of frames received from the LAN that have been discarded because of rate enforcement.
frmDiscCongln	The number of frames received from the LAN that have been discarded because of congestion or rate adaptation.

frmDiscCongOut	The number of frames received from the DPN subnet that have been discarded because of congestion or rate adaptation.
frmDiscOtherIn	The number of frames received from the LAN that have been discarded because of uncatagorized reasons.
frmDiscOtherOut	The number of frames received from the DPN subnet that have been discarded because of uncatagorized reasons.
frmDiscLwvcDown	The number of frames received from the LAN that have been discarded because the Light Weight Virtual Circuit cannot be setup.
frmDiscExcessHopsIn	The number of frames received from the LAN that have been discarded because of their excessive hop counts.
frmDiscExcessHopsOut	The number of frames received from the DPN subnet that have been discarded because of their excessive hop counts.
frmDiscOvrsizeLwvcIn	The number of frames received from the LAN that have been discarded because the frames are too large for the LWVCs.
frmDiscOvrsizeLwvcOut	The number of frames received from the LWVCs that have been discarded because the frames are too large.
frmDiscOvrsizeBcstIn	The number of frames received from the LAN that have been discarded because the frames are too large for the immediate broadcast server.
frmDiscOvrsizeBcstOut	The number of frames received from the broadcast server that have been discarded because the frames are too large.
qToBcstOverflow	The number of messages that have been discarded because the send queue to the broadcast server has reached a threshold.

frmDiscCompFiltIn	The number of frames received from the LAN that have been discarded because of composite filtering.
frmDiscCompFiltOut	The number of frames received from the DPN subnet that have been discarded because of composite filtering.
frmDiscServREnforceIn	The number of frames received from the LAN that have been discarded because of service-level rate enforcement.
frmProxyDiscardIn	The number of frames received from the LAN that have been discarded because of proxying of certain types of broadcast frames. Other ISRBs in the broadcast group already have a copy of these frames and will send these frames to their attached LANs on behalf of this ISRB. This field is updated only if proxy broadcast is in use.

ITI (subtype 12)

Table 46
ITI statistics (subtype 12)

Field	Type
modemChanges	integer
pktsRcv	integer
pktsSnt	integer
bytesRcv	integer
bytesSnt	integer
pktsSsqRcv	integer
pktsSsqSnt	integer
nuiReq	integer
nuiValid	integer
nuiInvalid	integer
nuiCnv	integer
length	integer
parityErr	integer
frmErr	integer
overruns	integer
nuiErr	integer
lrcErr	integer
l1PktsErr	integer
params	integer
parityThresh	integer
overrunsThresh	integer
nuiDie	integer
nuiDegrades	integer

modemChanges	The number of modem status changes occurring in the last 15-minute period.
pktsRcv	The sum of all packets received from the DTE.
pktsSnt	The sum of all packets sent to the DTE.
bytesRcv	The sum of all bytes received from the DTE.
bytesSnt	The sum of all bytes sent to the DTE.
pktsSsqRcv	The sum of squares of received packet lengths.
pktsSsqSnt	The sum of squares of sent packet lengths.
nuiReq	The number of NUI/translation requests sent for validation to the database.
nuiValid	The number of valid NUI/translation replies received.
nuiInvalid	The number of invalid NUI/translation replies received.
nuiCnv	The number of cannot validate NUI/translation replies received.
length	The length in bytes of the remaining statistics in this record.
parityErr	The number of incoming character parity errors.
frmErr	The number of incoming character framing errors.
overruns	The number of times a received character was lost because the service was busy processing the preceding character.
nuiErr	The number of illegal NUIs received.
lrcErr	The number of messages with LRC errors.
l1PktsErr	The number of invalid L1-packets received.
params	Parameter_1 to Parameter_22, Parameter_121, Parameter_122, Parameter_125: The ITI parameters P1 to P22, P121, P122, P125.

parityThresh	An alarm is generated if the number of parity errors exceeds this threshold set by operator (default = 2).
overrunsThresh	An alarm is generated if the number of overruns exceeds this threshold set by operator (default = 20).
nuiDie	Threshold set by the operator (default = 12), if exceeded on a dedicated line then an alarm is sent out and the link is disabled, if exceeded on a dial-up port then the service is turned off and no alarm is sent.
nuiDegrades	Threshold set by the operator (default = 4), if exceeded on a dedicated line then input is blocked for one minute and no alarm is sent, if exceeded on a dial-up port the service is turned off and no alarm is sent.

ITIEnhanced (subtype 57)

Table 47
ITI Enhanced statistics (subtype 57)

Field	Type
modemChanges	integer
pktsRcv	integer
pktsSnt	integer
bytesRcv	integer
bytesSnt	integer
pktsSsqRcv	integer
pktsSsqSnt	integer
nuiReq	integer
nuiValid	integer
nuiInvalid	integer
nuiCnv	integer
(Sheet 1 of 2)	

Table 47 (Continued)
ITI Enhanced statistics (subtype 57)

Field	Type
length	integer
parityErr	integer
frmErr	integer
overruns	integer
nuiErr	integer
lrcErr	integer
l1PktsErr	integer
params	integer
parityThresh	integer
overrunsThresh	integer
nuiDie	integer
nuiDegrades	integer
callReqRcv	integer
callConnSnt	integer
accessNoCall	integer
(Sheet 2 of 2)	

modemChanges	The number of modem status changes occurring in the last 15-minute period.
pktsRcv	The sum of all packets received from the DTE.
pktsSnt	The sum of all packets sent to the DTE.
bytesRcv	The sum of all bytes received from the DTE.
bytesSnt	The sum of all bytes sent to the DTE.
pktsSsqRcv	The sum of squares of received packet lengths.
pktsSsqSnt	The sum of squares of sent packet lengths.

nuiReq	The number of NUI/translation requests sent for validation to the database.
nuiValid	The number of valid NUI/translation replies received.
nuiInvalid	The number of invalid NUI/translation replies received.
nuiCnv	The number of cannot validate NUI/translation replies received.
length	The length in bytes of the remaining statistics in this record.
parityErr	The number of incoming character parity errors.
frmErr	The number of incoming character framing errors.
overruns	The number of times a received character was lost because the service was busy processing the preceding character.
nuiErr	The number of illegal NUIs received.
lrcErr	The number of messages with LRC errors.
l1PktsErr	The number of invalid L1-packets received.
params	Parameter_1 to Parameter_22, Parameter_121, Parameter_122, Parameter_125: The ITI parameters P1 to P22, P121, P122, P125.
parityThresh	An alarm is generated if the number of parity errors exceeds this threshold set by operator (default = 2).
overrunsThresh	An alarm is generated if the number of overruns exceeds this threshold set by operator (default = 20).
nuiDie	Threshold set by the operator (default = 12), if exceeded on a dedicated line then an alarm is sent out and the link is disabled, if exceeded on a dial-up port then the service is turned off and no alarm is sent.

nuiDegrades	Threshold set by the operator (default = 4), if exceeded on a dedicated line then input is blocked for one minute and no alarm is sent, if exceeded on a dial-up port the service is turned of and no alarm is sent.
callReqRcv	The number of call request packets received from link.
callConnSnt	The number of incoming/outgoing service signals to the link.
accessNoCall	The number of times switched circuit call to the PAD port did not result in call setup.

LAPDPhysical (subtype 23)

Table 48
LAPD Physical statistics (subtype 23)

Field	Type
badFrmSize	integer
underruns	integer
modemChanges	integer
lineFail	integer
frmAbort	integer
noBuffer	integer
crcErr	integer
overruns	integer
frmSnt	integer
frmRcv	integer
bytesSnt	integer
bytesRcv	integer
rrSnt	integer
rnrSnt	integer
(Sheet 1 of 2)	

Table 48 (Continued)
LAPD Physical statistics (subtype 23)

Field	Type
rejSnt	integer
frmrSnt	integer
ifrmSnt	integer
rrRcv	integer
mnrRcv	integer
rejRcv	integer
frmrRcv	integer
ifrmRcv	integer
l2BadFrmSize	integer
protocolErr	integer
frmRxmt	integer
frmQueued	integer
lrcErr	integer
lclFlowCtrl	integer
rmtFlowCtrl	integer
(Sheet 2 of 2)	

badFrmSize	Total number of frames received of the wrong size by the interrupt handler.
underruns	Total number of times the PE has been unable to supply data fast enough to keep up continuous flow.
modemChanges	Total number of modem errors detected by the PI.
lineFail	Total number of line failures detected.
frmAbort	Count of frames aborted. Indicates a transmission fault detected by the PI.

noBuffer	Count of number of times that no buffers were available to the interrupt handler when data was coming in.
crcErr	Total number of frames received with CRC errors.
overruns	Number of overruns. PE cannot handle data at this rate.
frmSnt	Number of frames sent. This includes, I, S, U frames of all logical links of the port (2 words).
frmRcv	Number of frames received. This includes I, S, U frames of all logical links of the port (2 words).
bytesSnt	Number of bytes sent. This includes the I-frames information field of all logical links on the port (2 words).
bytesRcv	Number of bytes received. This includes the I-frames information field of all logical links on the port (2 words).
rrSnt	Number of RRs transmitted for all logical links.
rrrSnt	Number of RNRs transmitted for all logical links.
rejSnt	Number of REJs transmitted for all logical links.
frmrSnt	Number of FRMRs transmitted for all logical links.
ifrmSnt	Number of I-frames transmitted for all logical links.
rrRcv	Number of RRs received for all logical links.
rrrRcv	Number of RNRs received for all logical links.
rejRcv	Number of REJs received for all logical links.
frmrRcv	Number of FRMRs received for all logical links.
ifrmRcv	Number of I-frames received for all logical links.
I2BadFrmSize	Total number of frames received of the wrong size by all logical links.
protocolErr	Total number of protocol violations detected by all logical links.

frmRxmt	Total number of retransmits caused by an REJ or after a timeout on an information frame.
frmQueued	Number of frames queued for all logical links.
lrcErr	Total number of frames received with LRC errors for all logical links.
lclFlowCtrl	Total time interval which DPN access module has flow controlled on input (2 words) (for all logical links).
rmtFlowCtrl	Total time interval which DPN access module has been flow controlled by the DTE (2 words) (for all logical links).

LAPDSignalling (subtype 47)

Table 49
LAPD signalling statistics (subtype 47)

Field	Type
minCalls	integer
maxCalls	integer
callsTotal	integer
lclFlowCtrl	integer
rmtFlowCtrl	integer
frmSnt	integer
frmRcv	integer
badFrmSize	integer
protocolErr	integer
frmRxmt	integer
lrcErr	integer
frmQueued	integer
ifrmSnt	integer
(Sheet 1 of 2)	

Table 49 (Continued)
LAPD signalling statistics (subtype 47)

Field	Type
ifrmRcv	integer
rrSnt	integer
rrRcv	integer
rnrSnt	integer
rnrRcv	integer
rejSnt	integer
rejRcv	integer
frmrSnt	integer
frmrRcv	integer
bytesSnt	integer
bytesRcv	integer
(Sheet 2 of 2)	

minCalls	Minimum number of calls up in a 15-minute period.
maxCalls	Maximum number of calls up in a 15-minute period.
callsTotal	Total number of calls that reached the active state within the statistics interval.
lclFlowCtrl	Total time interval which DPN access module has flow controlled on input (2 words).
rmtFlowCtrl	Total time interval which DPN access module has been flow controlled by the DTE (2 words).
frmrSnt	Number of frames transmitted to the DTE, including information, supervisory, unnumbered and retransmitted frames, but not aborted frames (2 words).

frmRcv	Number of frames received from the DTE, including I, S and U frames, but not aborted frames, frames received with CRC errors or IBADFRMSIZE errors (2 words).
badFrmSize	Total number of frames received of the wrong size by level 2.
protocolErr	Total number of protocol violations detected.
frmRxmt	Total number of retransmits caused by an REJ or after a timeout on an information frame.
lrcErr	Total number of frames received with LRC errors.
frmQueued	Number of frames queued.
ifrmSnt	Number of I-frames transmitted.
ifrmRcv	Number of I-frames received.
rrSnt	Number of RRs transmitted.
rrRcv	Number of RRs received.
rnrsnt	Number of RNRs transmitted.
rnrrcv	Number of RNRs received.
rejSnt	Number of REJs transmitted.
rejRcv	Number of REJs received.
frmrSnt	Number of FRMRs transmitted.
frmrRcv	Number of FRMRs received.
bytesSnt	Number of bytes transmitted in I-frames information field (2 words).
bytesRcv	Number of bytes received in I-frames information field (2 words).

LAPDE_Physical (subtype 77)

Note: This statistic is not currently generated by the DPN-100.

Table 50
LAPDE Physical statistics (subtype 77)

Field	Type
badFrmSize	integer
underruns	integer
modemChanges	integer
lineFail	integer
frmAbort	integer
noBuffer	integer
crcErr	integer
overruns	integer
frmSnt	integer
frmRcv	integer
bytesSnt	integer
bytesRcv	integer
rrSnt	integer
rnrSnt	integer
rejSnt	integer
frmrSnt	integer
ifrmSnt	integer
rrRcv	integer
rnrRcv	integer
rejRcv	integer
frmrRcv	integer
ifrmRcv	integer
(Sheet 1 of 2)	

Table 50 (Continued)
LAPDE Physical statistics (subtype 77)

Field	Type
l2BadFrmSize	integer
protocolErr	integer
frmRxmt	integer
frmQueued	integer
lrcErr	integer
lclFlowCtrl	integer
rmtFlowCtrl	integer
resetSnt	integer
resetRcv	integer
(Sheet 2 of 2)	

badFrmSize	Total number of frames received of the wrong size by the interrupt handler.
underruns	Total number of times the PE has been unable to supply data fast enough to keep up continuous flow.
modemChanges	Total number of modem errors detected by the PI.
lineFail	Total number of line failures detected.
frmAbort	Count of frames aborted. Indicates a transmission fault detected by the PI.
noBuffer	Count of number of times that no buffers were available to the interrupt handler when data was coming in.
crcErr	Total number of frames received with CRC errors.
overruns	Number of overruns. PE cannot handle data at this rate.
frmSnt	Number of frames sent. This includes, I, S, U frames of all logical links of the port (2 words).

frmRcv	Number of frames received. This includes I, S, U frames of all logical links of the port (2 words).
bytesSnt	Number of bytes sent. This includes the I-frames information field of all logical links on the port (2 words).
bytesRcv	Number of bytes received. This includes the I-frames information field of all logical links on the port (2 words).
rrSnt	Number of RRs transmitted for all logical links.
rnrSnt	Number of RNRs transmitted for all logical links.
rejSnt	Number of REJs transmitted for all logical links.
frmrSnt	Number of FRMRs transmitted for all logical links.
ifrmSnt	Number of I-frames transmitted for all logical links.
rrRcv	Number of RRs received for all logical links.
rnrRcv	Number of RNRs received for all logical links.
rejRcv	Number of REJs received for all logical links.
frmrRcv	Number of FRMRs received for all logical links.
ifrmRcv	Number of I-frames received for all logical links.
l2BadFrmSize	Total number of frames received of the wrong size by all logical links.
protocolErr	Total number of protocol violations detected by all logical links.
frmRxmt	Total number of retransmits caused by an REJ or after a timeout on an information frame.
frmQueued	Number of frames queued for all logical links.
lrcErr	Total number of frames received with LRC errors for all logical links.
lclFlowCtrl	Total time interval which DPN access module has flow controlled on input (2 words) (for all logical links).

rmtFlowCtrl	Total time interval which DPN access module has been flow controlled by the DTE (2 words) (for all logical links).
resetSnt	Number of resets sent.
resetRcv	Number of resets received.

NCSOpAgent (subtype 42)

Table 51
NCS Op Agent statistics (subtype 42)

Field	Type
alarmRcv	integer
statsRcv	integer
logsRcv	integer
accntRcv	integer

alarmRcv	The number of alarm records received from outside the Operating Agent (OA) across a virtual circuit.
statsRcv	The number of statistics records received from outside the OA across a virtual circuit.
logsRcv	The number of log records received from outside the OA across a virtual circuit.
accntRcv	The number of accounting records received from outside the OA across a virtual circuit.

NLLAPBEnhanced (subtype 58)

Table 52
NL (extended) statistics (subtype 58)

Field	Type
lclType	integer
lclRid	integer
lclMid	integer
rmtType	integer
rmtRid	integer
rmtMid	integer
rmtNamsid	integer
rmtPortType	integer
rmtPortId	integer
lclPkts	integer
tdmPkts	integer
currFrmQueued	integer
lclFlowCtrl	integer
rmtFlowCtrl	integer
frmSnt	integer
frmRcv	integer
frmQueued	integer
overruns	integer
frmAbort	integer
noBuffer	integer
badFrmSize	integer
crcErr	integer
cardErr	integer
underruns	integer
(Sheet 1 of 3)	

Table 52 (Continued)
NL (extended) statistics (subtype 58)

Field	Type
lrcErr	integer
frmRxmt	integer
modemChanges	integer
protocolErr	integer
nrSnt	integer
nrRcv	integer
rejSnt	integer
rejRcv	integer
frmrSnt	integer
frmrRcv	integer
rrSnt	integer
rrRcv	integer
ifrmSnt	integer
ifrmRcv	integer
ifrmSsqSnt	integer
ifrmSsqRcv	integer
ifrmBytesSnt	integer
ifrmBytesRcv	integer
x21DiSuc	integer
x21DiInc	integer
x21DiErr	integer
x21DoSuc	integer
x21DoInc	integer
x21DoErr	integer
x21ClrErr	integer
(Sheet 2 of 3)	

Table 52 (Continued)
NL (extended) statistics (subtype 58)

Field	Type
x21ParityErr	integer
linkSpeed	integer
frmBytesSnt	integer
frmBytesRcv	integer
(Sheet 3 of 3)	

lclType This field specifies the module type of the module on which this Network Link is running. If set to 3, then the module is an AM or a DPNAC; if set to 2, then the module is an RM.

lclRid This field specifies the Routing ID (RID) of the module on which this link is running. If 0, then the module does not have a Routing ID. (On an AM, this field is 0; on an RM, this field is set).

lclMid This field specifies the Module ID (MID) of the module on which this link is running. If 0, then the module does not have a module ID.

rmtType This field specifies the module type of the module to which the RM UTP Network Link is connected. If set to 2, then the module is an RM; if set to 3, then the module is an AM.

rmtRid The number of the remote RID of this trunk.

rmtMid This field specifies the Module ID (MID) of the module to which this RM UTP Network Link is connected.

If 0, then the module on the far end of the RM UTP Network Link does not have a Module ID, or the RM UTP Network Link is not connected to a far RM UTP Network Link at the end of this statistics interval.

rmtNamsid	This field specifies the Unique NAMS ID of the module to which this RM UTP Network Link is connected. If 0, then the module on the far end of the RM UTP Network Link does not have access to the Unique NAMS ID, or the RM UTP Network Link is not connected to a far RM UTP Network Link at the end of this statistics interval.
rmtPortType	<p>The type of the port that the software generating this record runs on. Values:</p> <ul style="list-style-type: none"> •0 - Not running on port (reserved for Multilink procedure) •1 - V.24 •2 - V.35 •3 - X.21 <p>The Single Link Procedure runs always on port, so the only values the PORTTYPE may assume for this statistics record are 1, 2 or 3.</p>
rmtPortId	<p>This field identifies the port information of the far end of the Trunk.</p> <ul style="list-style-type: none"> •If the REMOTEPORTTYPE is set to a value of 1, then the first three bytes of this field specify: REMOTEPORTID [0] = Processor (decimal) REMOTEPORTID [1] = Scanner (decimal) REMOTEPORTID [2] = Line (decimal) •If the REMOTEPORTTYPE is set to a value of 2, then the first three bytes of this field specify: REMOTEPORTID [0] = PE (decimal) REMOTEPORTID [1] = PI (decimal) REMOTEPORTID [2] = Port (decimal) •If the REMOTEPORTTYPE is set to any other value, then this field is not useful.
lciPkts	Number of packets received that are destined for this node.
tdmPkts	Number of packets received that are destined for other nodes.

currFrmQueued	This field specifies the number of frames queued for transmission to the link at the point that the statistics probe was processed by the link.
lclFlowCtrl	Total time interval which DPN access module has flow controlled on input (2 words) (for all logical links).
rmtFlowCtrl	This field specifies the time, in seconds, that the Network Link has been flow controlled by the far end. This statistic usually indicates that the link is overloaded, and may have to be re-engineered to handle the offered traffic.
frmSnt	This field specifies the number of frames sent to the link. This count includes I-frames (containing internetwork packets, and link specific packets), and Level 2 control frames, such as RRs, RNRs, REJs, FREJs, SABM, UA, etc.
frmRcv	This field specifies the number of frames received from the link. This count includes I-frames (containing internetwork packets, and link specific packets), and Level 2 control frames, such as RRs, RNRs, REJs, FREJs, SABM, UA, etc.

frmQueued This field specifies the number of frames queued for transmission to the link at the point that the statistics probe was processed by the link. If the number of frames queued is greater than 255, then this field has a value of 255, or H.FF, and the true count of frames queued is in the statistical counter CURRENT FRAMES QD.

Note that this field is not quite as useful as the CURRENT FRAMES QD, it is used at the link for thresholding. The thresholds all have maximum values of 254, and there is no need in most cases, to keep counting the number of link errors after hitting the upper bound. For example, the number of modem status changes in a statistics interval is counted up to a value of 254, and then counting stops. In the case of frames queued, the true count of frames queued is available in the software, so is provided in CURRENT FRAMES QD.

This statistic usually indicates that the link is overloaded, and may have to be re-engineered to handle the offered traffic.

overruns This field specifies the number of overruns in this statistics interval on the link. The PE is unable to accept a character before the next character has been accepted by the SCC on the PI.

This statistic usually indicates that the PE is overloaded, and should be re-engineered in order that its traffic capacity is not exceeded.

frmAbort This field specifies the number of frame aborts in this statistics interval on the link.

This statistic usually indicates a problem at the remote end of the line (it is sending aborts), or on the facility (it is turning data on the line into aborts).

noBuffer	<p>This field specifies the number of times in this statistics interval that the receive buffer used by the interrupt handler was insufficient to handle incoming traffic from the link.</p> <p>This statistic usually indicates that the link is overloaded, and requires re-engineering.</p>
badFrmSize	<p>This field specifies the number of frames received in this statistics interval from the link with a bad frame size (too long or too short).</p> <p>This statistic usually indicates a problem at the remote end of the line, it is sending frames of an incorrect frame size.</p>
crcErr	<p>This field specifies the number of CRC errors in this statistics interval on the link.</p> <p>This statistic usually indicates a data corruption problem somewhere in the network.</p>
cardErr	<p>This field specifies the number of line errors in this statistics interval on the link.</p> <p>This statistic usually indicates a problem on the SCC on the PI.</p>
underruns	<p>This field specifies the number of underruns in this statistics interval on the link, (the interrupt handler is unable to give the next character to the SCC in time).</p> <p>This statistic usually indicates that the PE is overloaded, and should be re-engineered so that its traffic capacity is not exceeded.</p>
lrcErr	<p>This field specifies the number of LRC errors in this statistics interval on the link.</p> <p>This statistic usually indicates a data corruption problem somewhere in the network.</p>

frmRxmt	<p>This field specifies the number of times in this statistics interval on the link, that a frame had to be retransmitted due to frames being rejected or timeouts.</p> <p>This statistic usually indicates a problem with the PI, the facility or the remote end.</p>
modemChanges	<p>This field specifies the number of modem status changes in this statistics interval on the link.</p> <p>This statistic usually indicates a problem with the PI, the facility or the remote end; an unterminated EIA cable plugged into the PI may also cause excessive modem status changes.</p>
protocolErr	<p>This field specifies the number of link protocol errors in this statistics interval on the link.</p> <p>This statistic, if coupled with other statistics errors, can indicate a problem on the hardware for the line. cable plugged into the PI may also cause excessive modem status changes. Otherwise, it indicates a problem with the remote end, where the protocol has not been implemented properly.</p>
nrnSnt	<p>This field specifies the number of LAP-B Level 2 RNRs sent to the link in this statistics interval.</p>
nrnRcv	<p>This field specifies the number of LAP-B Level 2 RNRs received from the link in this statistics interval.</p>
rejSnt	<p>This field specifies the number of LAP-B Level 2 REJs sent to the link in this statistics interval.</p>
rejRcv	<p>This field specifies the number of LAP-B Level 2 REJs received from the link in this statistics interval.</p>
frmRnt	<p>This field specifies the number of LAP-B Level 2 FREJs sent to the link in this statistics interval.</p>
frmRcv	<p>This field specifies the number of LAP-B Level 2 FREJs received from the link in this statistics interval.</p>

rrSnt	This field specifies the number of LAP-B Level 2 RRs sent to the link in this statistics interval.
rrRcv	This field specifies the number of LAP-B Level 2 RRs received from the link in this statistics interval.
ifrmSnt	This field specifies the number of I-frames sent to the link. I-frames include all inter-network packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
ifrmRcv	This field specifies the number of I-frames received from the link. I-frames include all inter-network packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
ifrmSsqSnt	This field specifies the statistical sum of squares, for the number of bytes in I-frames sent to the link in this statistics interval. I-frames include all inter-network packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
ifrmSsqRcv	This field specifies the statistical sum of squares, for the number of bytes in I-frames received from the link in this statistics interval. I-frames include all inter-network packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
ifrmBytesSnt	This field specifies the number of bytes in I-frames sent to the link in this statistics interval. I-frames include all inter-network packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
ifrmBytesRcv	This field specifies the number of bytes in I-frames received from the link in this statistics interval. I-frames include all inter-network packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
x21DiSuc	Number of successful X.25/X.21 dial-in call attempts (level 1 established). Not meaningful for SLP.

x21DiInc	Number of incomplete X.25/X.21 dial-in call attempts (call cleared before level 1 established). Not meaningful for SLP.
x21DiErr	Number of successful X.25/X.21 call attempts (dial-in only) that encountered an X.21 protocol violation during disconnection. Not meaningful for SLP.
x21DoSuc	Number of successful X.25/X.21 dial-out call attempts (level 1 established). Not meaningful for SLP.
x21DoInc	Number of incomplete X.25/X.21 dial-out call attempts (call cleared before level 1 established). Not meaningful for SLP.
x21DoErr	Number of incomplete X.25/X.21 dial-out call attempts aborted by an X.21 protocol violation (level 1 not established). Not meaningful for SLP.
x21ClrErr	Number of successful X.25/X.21 call attempts (both dial-in and dial-out) that encountered an X.21 protocol violation during disconnection. Not meaningful for SLP.
x21ParityErr	Number of X21 parity errors. Applicable only if port is an X.21 one.
linkSpeed	This field indicates the speed of the link in bytes per second.
frmBytesSnt	The number of bytes including frame overhead sent to link in this statistics interval.
frmBytesRcv	The number of bytes including frame overhead received from link in this statistics interval.

NUIDatabaseInt (subtype 43)

Table 53
NDI statistics (subtype 43)

Field	Type
maxOutCalls	integer
maxInCalls	integer
noTransToDb	integer
noTransFromDb	integer
noTransPass	integer
noTransFail	integer
unableToValidate	integer
protocolErr	integer
pktSizeErr	integer
dbProbesFrDb	integer
dbProbesRepFromDb	integer
dbProbesToDb	integer

- maxOutCalls** This field indicates the maximum number of calls set up from the NDI to the off-network database in the statistical interval.
- maxInCalls** This field indicates the maximum number of calls set up to the NDI from NUI-RSIs and/or NSIs in the statistical interval.
- noTransToDb** This field indicates the number of transactions which were sent to the off-network database in the statistical interval.
- noTransFromDb** This field indicates the number of transactions which were received from the off-network database in the statistical interval.

noTransPass

This field indicates the number of database transactions received from the off-network database which were completely successful in the statistical interval.

If the off-network database simply validates NUIs (Network User Identifiers), then this count represents the number of valid NUIs within the statistical interval.

This is the case for both of the following types of off-network NUI databases:

- DPN Off-Network NUI
- LIDB Calling Card

If the off-network database performs further translations, then this count represents the number of transactions in which all translations were successful. This is the case for the following types of off-network databases:

- NSHS

noTransFail

This field indicates the number of database transactions received from the off-network database which were unsuccessful in the statistical interval.

If the off-network database simply validates NUIs (Network User Identifiers), then this count represents the number of invalid NUIs within the statistical interval. This is the case for the following types of off-network databases:

- DPN Off-Network NUI
- LIDB Calling Card

If the off-network database performs further translations, then this count represents the number of transactions in which any one translation was unsuccessful. This is the case for the following types of off-network databases:

- NSHS

unableToValidate

This field indicates the number of translation requests received from the network destined to the off-network database that could not be forwarded to the off-network database. The resultant translation is considered to be invalid.

Network translation requests are not sent to the off-network database for several reasons:

- All SVCs to the off-network database are down
- The NUI Database Interface (NDI) protocol does not support the specific translation request received from the network.
- The translation request received from the network is invalid.

protocolErr	This field indicates the number of packets received from the off-network database which were discarded by NDI due to a perceived protocol violation. This count is accumulated over the statistical interval.
pktSizeErr	This field indicates the number of translation requests received from the network destined to the off-network database which were discarded by NDI due to the fact that the packet size on the SVC between NDI and the off-network database could not accommodate the translation.
dbProbesFrDb	This field indicates the number of probes received from the off-network database. Currently, the only off-network database supported by NDI which handles probes is the DPN Off-Network NUI database. In all other cases, this statistical count is 0.
dbProbesRepFromDb	This field indicates the number of probes responses received from the off-network database. Currently, the only off-network database supported by NDI which handles probes is the DPN Off-Network NUI database. In all other cases, this statistical count is 0.
dbProbesToDb	This field indicates the number of probes sent to the off-network database. Currently, the only off-network database supported by NDI which handles probes is the DPN Off-Network NUI database. In all other cases, this statistical count is 0.

NUIValidRemSvrRSI (subtype 34)

Table 54
NUI validation RSI statistics (subtype 34)

Field	Type
srvReq	integer

srvReq The number of requests sent to NDIs.

OfficeMaster (subtype 18)

Table 55
OM statistics (subtype 18)

Field	Type
totalCommonBlks	integer
freeQueueSize	integer

totalCommonBlks For every sample taken, the number of memory blocks available at DPN-AM initialization is added to this field.

freeQueueSize The number of free blocks found by sampling.

HighPerfPE (subtype 75)

Table 56
High Performance PE statistics (subtype 75)

Field	Type
heapSize	integer
heapUse	integer
heapHwm	integer
pcbUse	integer
pcbHwm	integer
totalMsgBlks	integer
freeQueueSize	integer
freeQueueLwm	integer
intraSnt	integer
interSnt	integer

(Sheet 1 of 2)

Table 56 (Continued)
High Performance PE statistics (subtype 75)

Field	Type
msgRcv	integer
cpuUtl	integer
cpuUtlHwm	integer
procDisp	integer
portPkts	integer
ntwkPkts	integer
expdHeapUse	integer
expdHeapHwm	integer
pbusHeapUse	integer
pbusHeapHwm	integer
(Sheet 2 of 2)	

heapSize	Maximum number of K words of local secondary storage (LSS) allocated in PE. This quantity is a constant for a given PE subject to changes in software.
heapUse	Amount of K words of local secondary storage (LSS) that is being used. This value is subject to change as calls are made on existing lines or new lines/ service data is added to PE.
heapHwm	Maximum amount of K words of local secondary storage (LSS) that is used during each 15 minute interval.
pcbUse	Amount of blocks (64 bytes/block) of process control block (PCB) storage that is being used.
pcbHwm	Maximum number of blocks (64 bytes/block) of process control block that is used during each 15 minute interval.

totalMsgBlks	For every sample period, the number of free message blocks available at PE initialization is added to this field.
freeQueueSize	The number of local free message blocks found by sampling is kept in this field.
freeQueueLwm	The minimum number of free blocks found by sampling is kept in this field.
intraSnt	The number of messages originating from a local process.
interSnt	The number of messages originating from a local process and sent externally.
msgRcv	The number of messages originating from an external process.
cpuUtl	The processing element utilization calculated and put in this field.
cpuUtlHwm	PE CPU utilization High Water Mark (HWM) value.
procDisp	The number of processes that are run on PE over 1 minute interval.
portPkts	The total packets from port. (Include Call packets).
ntwkPkts	The total packets from network. (Include Call packets).
expdHeapUse	Amount of K words of local secondary storage (LSS) that is being used. This value is subject to change as calls are made on existing lines or new lines, or as service data is added to PE.
expdHeapHwm	Maximum amount of K words of local secondary storage (LSS) that is used during each 15 minute interval.

pbusHeapUse	Amount of peripheral bus static RAM heap storage that is currently in use. The value is in K words.
pbusHeapHwm	Maximum amount of K words of peripheral bus static RAM heap storage that is used during each 15 minute interval.

PE286386 (subtype 30)

Table 57
PE (80286) statistics (subtype 30)

Field	Type
heapSize	integer
heapUse	integer
heapHwm	integer
pcbUse	integer
pcbHwm	integer
totalMsgBlks	integer
freeQueueSize	integer
freeQueueLwm	integer
intraSnt	integer
interSnt	integer
msgRcv	integer
cpuUtl	integer
cpuUtilHwm *	integer
procDisp	integer
portPkts	integer
ntwkPkts	integer
expdHeapUse	integer
(Sheet 1 of 2)	

Table 57 (Continued)
PE (80286) statistics (subtype 30)

Field	Type
expdHeapHwm	integer
* The field pcbTotal was reused as cpuUtilHwm in G34.	
(Sheet 2 of 2)	

heapSize	Maximum number of K words of local secondary storage (LSS) allocated in PE. This quantity is a constant for a given PE subject to changes in software.
heapUse	Amount of K words of local secondary storage (LSS) that is being used. This value is subject to change as calls are made on existing lines or new lines/ service data is added to PE.
heapHwm	Maximum amount of K words of local secondary storage (LSS) that is used during each 15 minute interval.
pcbUse	Amount of blocks (64 bytes/block) of process control block (PCB) storage that is being used.
pcbHwm	Maximum number of blocks (64 bytes/block) of process control block that is used during each 15 minute interval.
totalMsgBlks	For every sample period, the number of free message blocks available at PE initialization is added to this field.
freeQueueSize	The number of local free message blocks found by sampling is kept in this field.
freeQueueLwm	The minimum number of free blocks found by sampling is kept in this field.
intraSnt	The number of messages originating from a local process.

interSnt	The number of messages originating from a local process and sent externally.
msgRcv	The number of messages originating from an external process.
cpuUtl	The processing element utilization calculate and put in this field.
cpuUtilHwm	PE CPU utilization High Water Mark (HWM) value.
procDisp	The number of processes that are run on PE over 1 minute interval.
portPkts	The total packets from port. (Include Call packets).
ntwkPkts	The total packets from network. (Include Call packets).
expdHeapUse	Amount of K words of local secondary storage (LSS) that is being used. This value is subject to change as calls are made on existing lines or new lines, or as service data is added to PE.
expdHeapHwm	Maximum amount of K words of local secondary storage (LSS) that is used during each 15 minute interval.

SABRE (subtype 46)

Table 58
Sabre statistics (subtype 46)

Field	Type
badOutSeg	integer
protocolErr	integer
cccErr	integer
overruns	integer
underruns	integer
badInSeg	integer
(Sheet 1 of 2)	

Table 58 (Continued)
Sabre statistics (subtype 46)

Field	Type
modemChanges	integer
badResp	integer
frmSnt	integer
frmRcv	integer
frmBytesSnt	integer
frmBytesRcv	integer
frmSsqSnt	integer
frmSsqRcv	integer
(Sheet 2 of 2)	

badOutSeg

This statistic will be updated for the following error conditions:

- inconsistent LN of the outbound segment
- no EOM character in the outbound segment
- outbound segment shorter than 5 bytes
- too long outbound segment

protocolErr

The number of link level protocol errors detected by the TPAD.

cccErr

The number of CCC errors received by the TPAD.

overruns

The number of times a received character was lost because the TPAD was busy processing the preceding character.

underruns

The number of times the TPAD was unable to provide a character to the link when it had completed output of the previous character.

badInSeg	<p>This statistic will be updated for the following error condition:</p> <ul style="list-style-type: none">•wrong EOM of an input segment•inconsistent IA TA in a multi-segment message•inconsistent Si in a multi-segment message•exceeding the allowable segment length•exceeding the allowable message length•inconsistent segment lengths in multi-segment message•timeout on long message assembly
modemChanges	<p>The number of modem status changes occurring in the last 15-minute period.</p>
badResp	<p>This statistic will be updated for the following error conditions:</p> <ul style="list-style-type: none">•S1 or Idle in the middle of poll response•Lacking 'Ga G Idle' at the end of poll response•Exceeding the allowable number of segments in a poll response
frmSnt	<p>The number of frames that the TPAD has on this link.</p>
frmRcv	<p>The number of frames that the TPAD has received on this link.</p>
frmBytesSnt	<p>The total number of transmitted data bytes.</p>
frmBytesRcv	<p>The total number of received data bytes.</p>
frmSsqSnt	<p>Sum of squares of bytes in transmitted frames.</p>
frmSsqRcv	<p>Sum of squares of bytes in received frames.</p>

SNALink (subtype 26)

Table 59
SNA link statistics (subtype 26)

Field	Type
modemChanges	integer
abortFrm	integer
underruns	integer
overwrite	integer
crcErr	integer
badFrmSize	integer
nonOctetFrm	integer
lrcErr	integer
frmRxmt	integer
rnRcv	integer
rnSnt	integer
freeQueueUse	integer
cardErr	integer
noMsgBlk	integer
protocolErr	integer
frmSnt	integer
frmRcv	integer
frmBytesSnt	integer
frmBytesRcv	integer
frmSsqSnt	integer
frmSsqRcv	long integer
flowCtrlDuration	long integer
duration	integer

modemChanges	Modem status changes.
abortFrm	Aborted frames.
underruns	Underruns.
overwrite	Number of overwrites.
crcErr	CRC errors on frames from DTE.
badFrmSize	Frames from DTE longer than allowed by service data, or less than 4 octets long.
nonOctetFrm	Non octet frames from DTE.
lrcErr	LRC errors on frames to DTE.
frmRxmt	Frames resent to DTE.
rnrRcv	RNRs received by the PAD.
rnrSnt	RNRs sent to the DTE.
freeQueueUse	Maximum of FREEQ (or high water mark) used by a port.
cardErr	Number of card errors.
noMsgBlk	No msg blks are available for interrupt handle.
protocolErr	Total number of SNA protocol violations for all PUs on this link during the past interval.
frmSnt	RUs sent to the DTE (field length = 4 bytes).
frmRcv	RUs received from the DTE (field length = 4 bytes).
frmBytesSnt	The sum of the lengths of all RUs sent to the DTE (excludes CRC and flag) (field length = 4 bytes).
frmBytesRcv	The sum of the lengths of all RUs received from the DTE (excludes CRC and flag) (field length = 4 bytes).
frmSsqSnt	The sum of the square of the lengths of all RUs sent to the DTE (field length = 6 bytes).
frmSsqRcv	The sum of the square of the lengths of all RUs received from the DTE (field length = 6 bytes).

flowCtrlDuration	Duration of the time interval covered by this statistics record during which the service was flow controlling boundary links by RNR polling. (expressed in milliseconds, field length = 4 bytes).
duration	Duration of the time interval covered by this statistics record in seconds. (field length = 2 bytes).

SNALogicalUnit (subtype 27)

Table 60
DPN-AM and DPN-RM SNA LU statistics (subtype 27)

Field	Type
frmSnt	integer
frmRcv	integer
frmBytesSnt	integer
frmBytesRcv	integer
frmSsqSnt	long integer
frmSsqRcv	long integer
duration	integer

frmSnt	RUs sent to the DTE (field length = 4 bytes).
frmRcv	RUs received from the DTE (field length = 4 bytes).
frmBytesSnt	The sum of the lengths of all RUs sent to the DTE (excludes CRC and flag) (field length = 4 bytes).
frmBytesRcv	The sum of the lengths of all RUs received from the DTE (excludes CRC and flag) (field length = 4 bytes).
frmSsqSnt	The sum of the square of the lengths of all RUs sent to the DTE (field length = 6 bytes).
frmBytesRcv	The sum of the lengths of all RUs received from the DTE (excludes CRC and flag) (field length = 4 bytes).
frmSsqSnt	The sum of the square of the lengths of all RUs sent to the DTE (field length = 6 bytes).

frmSsqRcv	The sum of the square of the lengths of all RUs received from the DTE (field length = 6 bytes).
duration	The period in seconds (coded in binary) over which the LU statistics have been collected. Normally indicates the 15 minute interval between statistics probes, but may be less if the collection of statistics was only active during part of that period. (field length = 2 bytes).

TokenRing (subtype 53)

Table 61
Token Ring statistics (subtype 53)

Field	Type
frmSnt	integer
frmRcv	integer
frmBytesSnt	integer
frmBytesRcv	integer
lineErr	integer
ariFciErr	integer
burstErr	integer
rcvCongest	integer
lostFrmErr	integer
frmCpdErr	integer
tokenErr	integer
dmaParityErr	integer
dmaBusErr	integer
signalLoss	integer
hardErr	integer
softErr	integer
sntBeacon	integer
lobeWireFlt	integer
autoRmErr	integer
rmRcv	integer
counterOvfl	integer
singleStation	integer
ringRecov	integer

frmSnt	The number of frames sent to the TokenRing.
frmRcv	The number of frames received from the Token-Ring.
frmBytesSnt	The number of bytes in frames sent to the Token-Ring.
frmBytesRcv	The number of bytes in frames received from the Token-Ring.
lineErr	<p>This count is incremented whenever:</p> <ol style="list-style-type: none">1) a frame is copied or repeated2) the error indicator bit is zero in the incoming frame, and3) if at least one of the following conditions exist: <ul style="list-style-type: none">•code violation exists between the starting delimiter and the ending delimiter of the frame•code violation in the token•frame check sequence error
ariFciErr	This count is incremented when the adapter detects that the upstream neighbor is unable to set the AIR/FCI bits in a frame that it has copied.
burstErr	The burst error count is incremented when the adapter detects the absence of transitions for five half-bit times between start and end delimiter or end and start delimiter.
rcvCongest	This count is incremented when the adapter recognizes a frame addressed to it and has no buffer space available to copy the frame.
lostFrmErr	This count is incremented when an adapter is in transmit mode and fails to receive the end of the frame it transmitted.
frmCpdErr	This count is incremented when the adapter recognizes a frame addressed to it but finds that the air bits are not set to 0.

tokenErr	<p>This count is active only if the adapter is the active monitor. It is incremented if the adapter detects at least one of the following conditions:</p> <ul style="list-style-type: none">•monitorCount bit of a token with non-zero priority equals 1•monitorCount bit of a frame equals 1•no token/frame received within a 10-ms window•the starting delimiter/token sequence has a code violation
dmaParityErr	Indicates the occurrences of DMA parity errors.
dmaBusErr	Indicates the occurrences of DMA bus errors.
signalLoss	This count is incremented when the adapter detects a signal loss on the ring.
hardErr	Indicates the occurrences of transmitting or receiving beacon frames to or from the ring.
softErr	This field is incremented when the adapter transmits a report error MAC frame.
sntBeacon	Indicates the occurrences of transmitting beacon frames to the ring.
lobeWireFlt	This field is incremented when the adapter is closed due to the detection of an open or short circuit in the cable to the wiring concentrator.
autoRmErr	This field is incremented when the adapter is closed due to failing the lobe wrap test from the beacon auto-removal process.
rmRcv	This field is incremented when the adapter is closed after receiving a remove ring station MAC frame request.
counterOvfl	This field is incremented when one of the adapter error counters has incremented from 254 to 255.

singleStation	This field is incremented when the adapter detects it is the only station on the ring.
ringRecov	This field is incremented when the adapter observes claim token MAC frames on the ring.

TokenRingDevice (subtype 52)

Table 62
Token Ring Device statistics (subtype 52)

Field	Type
devMacAddr	long integer
devSapAddr	integer
ifrmSnt	integer
ifrmBytesSnt	integer
ifrmRcv	integer
ifrmBytesRcv	integer
tiExp	integer
discRcv	integer
reallocate	integer
linkDown	integer
frmSnt	integer
frmRcv	integer
lclBusy	integer
rmtBusy	integer
ringProtocol	integer
qllcProtocol	integer
badFrm	integer
discTimeout	integer
xidTimeout	integer

devMacAddr	MAC address of the device.
devSapAddr	SAP address of the device.
ifrmSnt	This field specifies the number of information (I) frames and unnumbered information (UI) frames sent on all LLC connections to that device. Information frames do not include frames such as LLC control frames (for example, RR, RNR, and UA).
ifrmBytesSnt	This field specifies the number of bytes in information (I) frames and unnumbered information (UI) frames sent on all LLC connections to that device.
ifrmRcv	This field specifies the number of information (I) frames and unnumbered information (UI) frames received from all LLC connections to that device. Information frames do not include frames such as LLC control frames (for example, RR, RNR, and UA).
ifrmBytesRcv	This field specifies the number of bytes in information (I) frames and unnumbered information (UI) frames received from all LLC connections to that device.
tiExp	This field specifies the number of times that the inactivity timer has expired. The only occurrences that are counted are when the LLC is not in ABM.
discRcv	This field specifies the number of DM or DISC frames received.
reallocate	This field specifies the number of failed re-allocates where resources were not available for a link station.
linkDown	This field specifies the number of times that a link went down.
frmSnt	This field specifies the number of FRMR frames transmitted.

frmRcv	This field specifies the number of FRMR frames received.
lclBusy	This field specifies the number of times that local busy state was entered.
rmtBusy	This field specifies the number of times that the remote end has entered the local busy state.
ringProtocol	This field specifies the number of ring protocol errors encountered.
qllcProtocol	This field specifies the number of QLLC protocol errors encountered.
badFrm	This field specifies the number of bad frames received from the ring. This includes frames that have an incorrect length or invalid header.
discTimeout	This field specifies the number of connections that were cleared due to a disconnect timeout.
xidTimeout	This field specifies the number of connections that were cleared due to XID timeouts while XID contacting the device.

TrunkRMUTP (subtype 25)

Table 63
RM trunk statistics (subtype 25)

Field	Type
lclRid	integer
rmtRid	integer
lclMid	integer
rmtMid	integer
rmtNamsid	integer
rmtPortType	integer
rmtPortId	integer
frmRcv	integer
hiPrtySnt	integer
loPrtySnt	integer
lclPkts	integer
tdmPkts	integer
frmBytesRcv	integer
frmRxmt	integer
discPkts	integer
ayt	integer
windowClosure	integer
underruns	integer
lrcErr	integer
crcErr	integer
overruns	integer
modemChanges	integer
frmBytesSnt	integer
(Sheet 1 of 2)	

Table 63 (Continued)
RM trunk statistics (subtype 25)

Field	Type
linkSpeed	integer
bifurcation	integer
(Sheet 2 of 2)	

lclRid This field specifies the Routing ID (RID) of the module on which this Trunk is running.

rmtRid This field specifies the Routing ID (RID) of the module to which this Trunk is connected.

lclMid This field specifies the Module ID (MID) of the module on which this Trunk is running.

rmtMid This field specifies the Module ID (MID) of the module to which this Trunk is connected.

If 0, then the module on the far end of the Trunk does not have a Module ID, or the Trunk has not been connected to a far end in this statistics interval. (Where the far end of the Trunk is an NM, this field is 0; otherwise, this field is set).

rmtNamsid This field specifies the Unique NAMS ID of the module to which this Trunk is connected. If 0, then the module on the far end of the Trunk does not have access to the Unique NAMS ID, or the Trunk has not been connected to a far end in this statistics interval. (Where the far end of the Trunk is an NM, this field is 0; otherwise, this field is set).

rmtPortType	<p>If the Trunk has been up in this statistics interval, then this field indicates the type of port on the far end of the Trunk.</p> <p>If the Trunk has not been up in this statistics interval, then this field is 0.</p> <p>If this field is set to a value of 1, then the far end of the Trunk has a Port ID of the format (Processor, Scanner, Line).</p> <p>If this field is set to a value of 2, then the far end of the Trunk has a Port ID of the format (PE,PI, Port).</p> <p>If this field is set to a value other than 1, or 2, then the port type of the far end of the Trunk is not yet defined.</p>
rmtPortId	<p>This field identifies the port information of the far end of the Trunk.</p> <ul style="list-style-type: none"> •If the REMOTEPORTTYPE is set to a value of 1, then the first three bytes of this field specify: REMOTEPORTID [0] = Processor (decimal) REMOTEPORTID [1] = Scanner (decimal) REMOTEPORTID [2] = Line (decimal) •If the REMOTEPORTTYPE is set to a value of 2, then the first three bytes of this field specify: REMOTEPORTID [0] = PE (decimal) REMOTEPORTID [1] = PI (decimal) REMOTEPORTID [2] = Port (decimal) •If the REMOTEPORTTYPE is set to any other value, then this field is not useful.
frmRcv	Number of frames (data and control) received.
hiPrtySnt	Number of high priority frames transmitted.
loPrtySnt	Number of low priority frames transmitted.
lclPkts	Number of packets received that are destined for this node.

tdmPkts	Number of packets received that are destined for other nodes.
frmBytesRcv	Number of bytes received, including protocol overhead and control frames.
frmRxmt	Number of times retransmission occurs.
discPkts	Number of packets discarded due to local congestion (only those received from a trunk link) or transmit buffer overflow.
ayt	Number of times entered into AYT mode and recovered.
windowClosure	Number of times the transmit window closed.
underruns	Number of underruns detected.
lrcErr	Number of packets with LRC error on the transmit side. They are reported as they occur.
crcErr	Number of packets received with CRC error.
overruns	Number of overruns detected.
modemChanges	Number of transient modem status changed without causing a communication error.
frmBytesSnt	Number of bytes transmitted, including protocol overhead and control frames.
linkSpeed	This field gives the measured link speed reported in kilobits per second.
bifurcation	Number of tandem packets bifurcated when sending to the congested trunk.

TrunkRMUTPEnhanced (subtype 64)

Table 64
Frame trunk statistics (subtype 64)

Field	Type
lclRid	integer
rmtRid	integer
lclMid	integer
rmtMid	integer
rmtNamsid	integer
rmtPortType	integer
rmtPortId	integer
frmRcv	integer
hiPrtySnt	integer
loPrtySnt	integer
lclPkts	integer
tdmPkts	integer
frmBytesRcv	integer
frmRxmt	integer
discPkts	integer
ayt	integer
windowClosure	integer
underruns	integer
lrcErr	integer
crcErr	integer
overruns	integer
modemChanges	integer
frmBytesSnt	integer
(Sheet 1 of 2)	

Table 64 (Continued)
Frame trunk statistics (subtype 64)

Field	Type
linkSpeed	integer
bifurcation	integer
delayHiFrmSnt	integer
delayLoFrmSnt	integer
delayHiFrmBytesSnt	integer
thrptHiFrmBytesSnt	integer
delayLoFrmBytesSnt	integer
(Sheet 2 of 2)	

lclRid This field specifies the Routing ID (RID) of the module on which this Trunk is running.

rmtRid This field specifies the Routing ID (RID) of the module to which this Trunk is connected.

lclMid This field specifies the Module ID (MID) of the module on which this Trunk is running.

rmtMid This field specifies the Module ID (MID) of the module to which this Trunk is connected.

If 0, then the module on the far end of the Trunk does not have a Module ID, or the Trunk has not been connected to a far end in this statistics interval. (Where the far end of the Trunk is an NM, this field is 0; otherwise, this field is set).

rmtNamsid

This field specifies the Unique NAMS ID of the module to which this Trunk is connected. If 0, then the module on the far end of the Trunk does not have access to the Unique NAMS ID, or the Trunk has not been connected to a far end in this statistics interval. (Where the far end of the Trunk is an NM, this field is 0; otherwise, this field is set).

rmtPortType

If the Trunk has been up in this statistics interval, then this field indicates the type of port on the far end of the Trunk.

If the Trunk has not been up in this statistics interval, then this field is 0.

If this field is set to a value of 1, then the far end of the Trunk has a Port ID of the format (Processor, Scanner, Line).

If this field is set to a value of 2, then the far end of the Trunk has a Port ID of the format (PE,PI, Port).

If this field is set to a value other than 1, or 2, then the port type of the far end of the Trunk is not yet defined.

rmtPortId	<p>This field identifies the port information of the far end of the Trunk.</p> <ul style="list-style-type: none">•If the REMOTEPORTTYPE is set to a value of 1, then the first three bytes of this field specify: REMOTEPORTID [0] = Processor (decimal) REMOTEPORTID [1] = Scanner (decimal) REMOTEPORTID [2] = Line (decimal)•If the REMOTEPORTTYPE is set to a value of 2, then the first three bytes of this field specify: REMOTEPORTID [0] = PE (decimal) REMOTEPORTID [1] = PI (decimal) REMOTEPORTID [2] = Port (decimal)•If the REMOTEPORTTYPE is set to any other value, then this field is not useful.
frmRcv	Number of frames (data and control) received.
hiPrtySnt	Number of high priority frames transmitted.
loPrtySnt	Number of low priority frames transmitted.
lclPkts	Number of packets received that are destined for this node.
tdmPkts	Number of packets received that are destined for other nodes.
frmBytesRcv	Number of bytes received, including protocol overhead and control frames.
frmRxmt	Number of times retransmission occurs.
discPkts	Number of packets discarded due to local congestion (only those received from a trunk link) or transmit buffer overflow.
ayt	Number of times entered into AYT mode and recovered.
windowClosure	Number of times the transmit window closed.
underruns	Number of underruns detected.
lrcErr	Number of packets with LRC error on the transmit side. They are reported as they occur.

crcErr	Number of packets received with CRC error.
overruns	Number of overruns detected.
modemChanges	Number of transient modem status changed without causing a communication error.
frmBytesSnt	Number of bytes transmitted, including protocol overhead and control frames.
linkSpeed	This field gives the measured link speed reported in kilobits per second.
bifurcation	Number of tandem packets bifurcated when sending to the congested trunk.
delayHiFrmSnt	Contains the number of frames transmitted for delay and high priority class.
delayLoFrmSnt	Contains the number of frames transmitted for delay and low priority class.
delayHiFrmBytesSnt	Contains the number of bytes transmitted for delay and high priority class.
thrptHiFrmBytesSnt	Contains the number of bytes transmitted for throughput and high priority class.
delayLoFrmBytesSnt	Contains the number of bytes transmitted for delay and low priority class.

UTP (subtype 35)

Table 65
UTP statistics (subtype 35)

Field	Type
lclType	integer
lclRid	integer
lclMid	integer
rmtType	integer
rmtRid	integer
rmtMid	integer
rmtNamsid	integer
rmtPortType	integer
rmtPortId	integer
frmRcv	integer
hiPrtySnt	integer
loPrtySnt	integer
lclPkts	integer
tdmPkts	integer
frmBytesRcv	integer
frmRxmt	integer
discPkts	integer
ayt	integer
windowClosure	integer
underruns	integer
lrcErr	integer
crcErr	integer
overruns	integer
(Sheet 1 of 2)	

Table 65 (Continued)
UTP statistics (subtype 35)

Field	Type
modemChanges	integer
frmBytesSnt	integer
linkSpeed	integer
(Sheet 2 of 2)	

lclType This field specifies the module type of the module on which this Network Link is running. If set to 3, then the module is an AM or a DPNAC; if set to 2, then the module is an RM.

lclRid This field specifies the Routing ID (RID) of the module on which this link is running. If 0, then the module does not have a Routing ID. (On an AM, this field is 0; on an RM, this field is set).

lclMid This field specifies the Module ID (MID) of the module on which this link is running. If 0, then the module does not have a module ID.

rmtType This field specifies the module type of the module to which the RM UTP Network Link is connected. If set to 2, then the module is an RM; if set to 3, then the module is an AM.

rmtRid This field specifies the Routing ID (RID) of the module to which this RM UTP Network Link is connected.

If 0, the RM UTP Network Link is not connected to a far RM UTP Network Link at the end of this statistics interval, or the far end does not have RID.

rmtMid	<p>This field specifies the Module ID (MID) of the module to which this RM UTP Network Link is connected.</p> <p>If 0, then the module on the far end of the RM UTP Network Link does not have a Module ID, or the RM UTP Network Link is not connected to a far RM UTP Network Link at the end of this statistics interval.</p>
rmtNamsid	<p>This field specifies the Unique NAMS ID of the module to which this RM UTP Network Link is connected. If 0, then the module on the far end of the RM UTP Network Link does not have access to the Unique NAMS ID, or the RM UTP Network Link is not connected to a far RM UTP Network Link at the end of this statistics interval.</p>
rmtPortType	<p>The type of the port that the software generating this record runs on. Values:</p> <ul style="list-style-type: none">•0 - Not running on port (reserved for Multilink procedure)•1 - V.24•2 - V.35•3 - X.21 <p>The Single Link Procedure runs always on port, so the only values the PORTTYPE may assume for this statistics record are 1, 2 or 3.</p>

rmtPortId	<p>This field identifies the port information of the far end of the Trunk.</p> <ul style="list-style-type: none"> •If the REMOTEPORTTYPE is set to a value of 1, then the first three bytes of this field specify: REMOTEPORTID [0] = Processor (decimal) REMOTEPORTID [1] = Scanner (decimal) REMOTEPORTID [2] = Line (decimal) •If the REMOTEPORTTYPE is set to a value of 2, then the first three bytes of this field specify: REMOTEPORTID [0] = PE (decimal) REMOTEPORTID [1] = PI (decimal) REMOTEPORTID [2] = Port (decimal) •If the REMOTEPORTTYPE is set to any other value, then this field is not useful.
frmRcv	Number of frames (data and control) received.
frmBytesRcv	Number of bytes received, including protocol overhead and control frames.
frmRxmt	Number of times retransmission occurs.
discPkts	Number of packets discarded due to local congestion (only those received from an RM UTP Network link) or transmit buffer overflow.
ayt	Number of times entered into AYT mode and recovered.
windowClosure	Number of times the transmit window closed.
underruns	Number of underruns detected.
lrcErr	Number of packets with LRC error on the transmit side. They are reported as they occur.
crcErr	Number of packets received with CRC error.
overruns	Number of overruns detected.
modemChanges	Number of modem status changes.

frmBytesSnt	Number of bytes transmitted, including protocol overhead and control frames.
linkSpeed	This field specifies the speed of the link. The speed is in kilobits per second, and is a decimal value. For example, for a link speed of 128 kbit/s, this field contains a decimal value of 128.

UTPEnhanced (subtype 65)

Table 66
UTPEnhanced statistics (subtype 65)

Field	integer
lclType	integer
lclRid	integer
lclMid	integer
rmtType	integer
rmtRid	integer
rmtMid	integer
rmtNamsid	integer
rmtPortType	integer
rmtPortId (0)	integer
rmtPortId (1)	integer
rmtPortId (2)	integer
rmtPortId (3)	integer
rmtPortId (4)	integer
frmRcv	integer
hiPrtySnt	integer
loPrtySnt	integer
lclPkts	integer
tdmPkts	integer
(Sheet 1 of 2)	

Table 66 (Continued)
UTPEnhanced statistics (subtype 65)

Field	integer
frmBytesRcv	integer
frmRxmt	integer
discPkts	integer
ayt	integer
windowClosure	integer
underruns	integer
lrcErr	integer
crcErr	integer
overruns	integer
modemChanges	integer
frmBytesSnt	integer
bifurcation	integer
linkSpeed	integer
delayHiFrmSnt	integer
delayLoFrmSnt	integer
delayHiFrmBytesSnt	integer
thrptHiFrmBytesSnt	integer
delayLoFrmBytesSnt	integer
(Sheet 2 of 2)	

lclType

This field specifies the module type of the module on which this Network Link is running. If set to 3, then the module is an AM or a DPNAC; if set to 2, then the module is an RM.

lclRid

This field specifies the Routing ID (RID) of the module on which this link is running. If 0, then the module does not have a Routing ID. (On an AM, this field is 0; on an RM, this field is set).

lclMid	<p>This field specifies the Module ID (MID) of the module on which this link is running. If 0, then the module does not have a module ID.</p> <p>If the module is a stub-AM, the lower 11 bits of the field contains the MID of one of the stub-AM's connected AMs.</p>
rmtType	<p>This field specifies the module type of the module to which the RM UTP Network Link is connected. If set to 2, then the module is an RM; if set to 3, then the module is an AM.</p>
rmtRid	<p>This field specifies the Routing ID (RID) of the module to which this RM UTP Network Link is connected.</p> <p>If 0, the RM UTP Network Link is not connected to a far RM UTP Network Link at the end of this statistics interval, or the far end does not have RID.</p>
rmtMid	<p>This field specifies the Module ID (MID) of the module to which this RM UTP Network Link is connected.</p> <p>If 0, then the module on the far end of the RM UTP Network Link does not have a Module ID, or the RM UTP Network Link is not connected to a far RM UTP Network Link at the end of this statistics interval.</p> <p>If the module is an AM linked to an RM, the field contains the value $1910 + \text{RM RID value}$.</p> <p>If the module is an AM connected to a stub-AM, the field contains the value $6144 + \text{stub-id value}$.</p>

rmtNamsid	This field specifies the Unique NAMS ID of the module to which this RM UTP Network Link is connected. If 0, then the module on the far end of the RM UTP Network Link does not have access to the Unique NAMS ID, or the RM UTP Network Link is not connected to a far RM UTP Network Link at the end of this statistics interval.
rmtPortType	<p>If the Trunk has been up in this statistics interval, then this field indicates the type of port on the far end of the Trunk.</p> <p>If the Trunk has not been up in this statistics interval, then this field is 0.</p> <p>If this field is set to a value of 1, then the far end of the Trunk has a Port ID of the format (Processor, Scanner, Line).</p> <p>If this field is set to a value of 2, then the far end of the Trunk has a Port ID of the format (PE,PI, Port).</p> <p>If this field is set to a value other than 1, or 2, then the port type of the far end of the Trunk is not yet defined.</p>

rmtPortId	<p>This field identifies the port information of the far end of the Trunk.</p> <ul style="list-style-type: none">•If the REMOTEPORTTYPE is set to a value of 1, then the first three bytes of this field specify: REMOTEPORTID [0] = Processor (decimal) REMOTEPORTID [1] = Scanner (decimal) REMOTEPORTID [2] = Line (decimal)•If the REMOTEPORTTYPE is set to a value of 2, then the first three bytes of this field specify: REMOTEPORTID [0] = PE (decimal) REMOTEPORTID [1] = PI (decimal) REMOTEPORTID [2] = Port (decimal)•If the REMOTEPORTTYPE is set to any other value, then this field is not useful.
frmRcv	Number of frames (data and control) received.
hiPrtySnt	Number of high priority frames transmitted.
loPrtySnt	Number of low priority frames transmitted.
lclPkts	Number of packets received that are destined for this node.
tdmPkts	Number of packets received that are destined for other nodes.
frmBytesRcv	Number of bytes received, including protocol overhead and control frames.
frmRxmt	Number of times retransmission occurs.
discPkts	Number of packets discarded due to local congestion (only those received from an RM UTP Network link) or transmit buffer overflow.
ayt	Number of times entered into AYT mode and recovered.
windowClosure	Number of times the transmit window closed.
underruns	Number of underruns detected.
lrcErr	Number of packets with LRC error on the transmit side. They are reported as they occur.

crcErr	Number of packets received with CRC error.
overruns	Number of overruns detected.
modemChanges	Number of modem status changes.
frmBytesSnt	Number of bytes transmitted, including protocol overhead and control frames.
bifurcation	Number of tandem packets bifurcated when sending to the congested trunk.
linkSpeed	This field specifies the speed of the link. The speed is in kilobits per second, and is a decimal value. For example, for a link speed of 128 kbit/s, this field contains a decimal value of 128.
delayHiFrmSnt	Contains the number of frames transmitted for delay and high priority class.
delayLoFrmSnt	Contains the number of frames transmitted for delay and low priority class.
delayHiFrmBytesSnt	Contains the number of bytes transmitted for delay and high priority class.
thrptHiFrmBytesSnt	Contains the number of bytes transmitted for throughput and high priority class.
delayLoFrmBytesSnt	Contains the number of bytes transmitted for delay and low priority class.

NLOverFR (subtype 72)

Table 67
NLOverFR statistics (subtype 72)

Field	Type
lclType	integer
lclRid	integer
lclMid	integer
rmtType	integer
(Sheet 1 of 3)	

Table 67 (Continued)
NLOverFR statistics (subtype 72)

Field	Type
rmtRid	integer
rmtMid	integer
rmtNamsid	integer
rmtPortType	integer
rmtPortId (0)	integer
rmtPortId (1)	integer
rmtPortId (2)	integer
rmtPortId (3)	integer
rmtPortId (4)	integer
netLinkType	integer
frmRcv	integer
hiPrtySnt	integer
loPrtySnt	integer
lclPkts	integer
tdmPkts	integer
frmBytesRcv	integer
frmRxmt	integer
discPkts	integer
ayt	integer
windowClosure	integer
underruns	integer
lrcErr	integer
crcErr	integer
overruns	integer
modemChanges	integer
(Sheet 2 of 3)	

Table 67 (Continued)
NLOverFR statistics (subtype 72)

Field	Type
frmBytesSnt	integer
linkSpeed	integer
delayHiFrmSnt	integer
delayLoFrmSnt	integer
delayHiFrmBytesSnt	integer
thrptHiFrmBytesSnt	integer
delayLoFrmBytesSnt	integer
frFrmLost	integer
(Sheet 3 of 3)	

lclType

This field specifies the module type of the module on which this Network Link is running. If set to 3, then the module is an AM or a DPNAC; if set to 2, then the module is an RM.

lclRid

This field specifies the Routing ID (RID) of the module on which this link is running. If 0, then the module does not have a Routing ID. (On an AM, this field is 0; on an RM, this field is set).

lclMid

This field specifies the Module ID (MID) of the module on which this link is running. If 0, then the module does not have a module ID.

If the module is a stub-AM, the lower 11 bits of the field contains the MID of one of the stub-AM's connected AMs.

rmtType

This field specifies the module type of the module to which the RM UTP Network Link is connected. If set to 2, then the module is an RM; if set to 3, then the module is an AM.

rmtRid

This field specifies the Routing ID (RID) of the module to which this RM UTP Network Link is connected.

If 0, the RM UTP Network Link is not connected to a far RM UTP Network Link at the end of this statistics interval, or the far end does not have RID.

rmtMid

This field specifies the Module ID (MID) of the module to which this RM UTP Network Link is connected.

If 0, then the module on the far end of the RM UTP Network Link does not have a Module ID, or the RM UTP Network Link is not connected to a far RM UTP Network Link at the end of this statistics interval.

If the module is an AM linked to an RM, the field contains the value 1910 + RM RID value.

If the module is an AM connected to a stub-AM, the field contains the value 6144 + stub-id value.

rmtNamsid

This field specifies the Unique NAMS ID of the module to which this RM UTP Network Link is connected. If 0, then the module on the far end of the RM UTP Network Link does not have access to the Unique NAMS ID, or the RM UTP Network Link is not connected to a far RM UTP Network Link at the end of this statistics interval.

rmtPortType	<p>If the Trunk has been up in this statistics interval, then this field indicates the type of port on the far end of the Trunk.</p> <p>If the Trunk has not been up in this statistics interval, then this field is 0.</p> <p>If this field is set to a value of 1, then the far end of the Trunk has a Port ID of the format (Processor, Scanner, Line).</p> <p>If this field is set to a value of 2, then the far end of the Trunk has a Port ID of the format (PE,PI, Port).</p> <p>If this field is set to a value other than 1, or 2, then the port type of the far end of the Trunk is not yet defined.</p>
rmtPortId	<p>This field identifies the port information of the far end of the Trunk.</p> <ul style="list-style-type: none"> •If the REMOTEPORTTYPE is set to a value of 1, then the first three bytes of this field specify: REMOTEPORTID [0] = Processor (decimal) REMOTEPORTID [1] = Scanner (decimal) REMOTEPORTID [2] = Line (decimal) •If the REMOTEPORTTYPE is set to a value of 2, then the first three bytes of this field specify: REMOTEPORTID [0] = PE (decimal) REMOTEPORTID [1] = PI (decimal) REMOTEPORTID [2] = Port (decimal) •If the REMOTEPORTTYPE is set to any other value, then this field is not useful.
netLinkType	<p>Identifies this Network Link as either a UTP Network Link or a Network Link running on Frame Relay:</p> <ul style="list-style-type: none"> •0 = UTP Network Link •1 = Network Link over Frame Relay

frmRcv	Number of frames (data and control) received.
hiPrtySnt	Number of high priority frames transmitted.
loPrtySnt	Number of low priority frames transmitted.
lciPkts	Number of packets received that are destined for this node.
tdmPkts	Number of packets received that are destined for other nodes.
frmBytesRcv	Number of bytes received, including protocol overhead and control frames.
frmRxmt	Number of times retransmission occurs.
discPkts	Number of packets discarded due to local congestion (only those received from an RM UTP Network link) or transmit buffer overflow.
ayt	Number of times entered into AYT mode and recovered.
windowClosure	Number of times the transmit window closed.
underruns	Number of underruns detected.
lrcErr	Number of packets with LRC error on the transmit side. They are reported as they occur.
crcErr	Number of packets received with CRC error.
overruns	Number of overruns detected.
modemChanges	Number of modem status changes.
frmBytesSnt	Number of bytes transmitted.
linkSpeed	This field specifies the speed of the link. The speed is in kilobits per second, and is a decimal value. For example, for a link speed of 128kbit/s, this field contains a decimal value of 128.
delayHiFrmSnt	Contains the number of frames transmitted for delay and high priority class.
delayLoFrmSnt	Contains the number of frames transmitted for delay and low priority class.

delayHiFrmBytesSnt	Contains the number of bytes transmitted for delay and high priority class.
thrpHiFrmBytesSnt	Contains the number of bytes transmitted for throughput and high priority class.
delayLoFrmBytesSnt	Contains the number of bytes transmitted for delay and low priority class.
frFrmLost	Number of frames lost on the External/Public Frame Relay connection for Network Link running over Frame Relay. This is detected by the sequence number mechanism. The transmit Network Link attaches a sequence number to each sent frame and the receive Network Link checks the numbers received. Any gap in sequence numbers means a lost frame, registered by this statistic.

This statistic applies only to the receive direction of transmission. The count is always 0 for regular Network Links.

VC (subtype 67)

Table 68
VC statistics (subtype 67)

Field	Type
srcDna	string
srcLcn	integer
destDna	string
destLcn	integer
callRef	integer
dataPktsSnt	integer
dataPktsRcv	integer
(Sheet 1 of 2)	

Table 68 (Continued)
VC statistics (subtype 67)

Field	Type
pktsRxmt	integer
outOfSeqPkts	integer
vcRecoveries	integer
rcvQueueHwm	integer
duplicatedPkts	integer
pktsNotXmt	integer
ackStaTimer	integer
windAdptTrig	integer
timeWindAdptState	integer
windCycles	integer
lwmDynamWind	integer
currSzeDynamWind	integer
fciFromSbnet	integer
bciFromSbnet	integer
bciToSbnet	integer
(Sheet 2 of 2)	

srcDna

This fields contain full calling address (including DNIC for X.121 and Country Code for E.164); maximum 15 digits.

Note: This is not necessary address of the source of the record. Source of the record can be identified from NAMS ID, PE# and component id fields.

srcLcn

Logical Channel Number of the calling end of the call.

destDna	This fields contain full called address (including DNIC for X.121 and Country Code for E.164); maximum 15 digits.
destLcn	Logical channel number of the called end of the call.
callRef	Call reference number. It is a three byte random number generated by the switch used as one of the fields in matching duplicated records generated at each end of the call.
dataPktsSnt	This statistic contains the total number of packets forwarded to the subnet by VC. This counter contains only the number of packets which were forwarded by an application (AP) i.e. the retransmissions are not included. The total number of retransmissions is given in the <i>packets retransmitted</i> statistic.
dataPktsRcv	This statistic contains the total number of data packets received from the remote end. The duplicated packets are counted as one.
pktsRxmt	This is the total number of times packets were retransmitted to the subnet after retransmission timer expired. This includes all the retransmitted packets (data packets, idle probe, reset packets). This statistic does not tell how many times individual packets were retransmitted but it provides the total number of retransmissions which took place.

outOfSeqPkts	All data packets sent by the VC are equipped with the unique sequence number. The expected sequence number for the next receive packet is in the range which is determined by the last sequence number received and the receive credits available. All data packets which are received with out of sequence number but within the expected sequence range are enqueued into the receive queue in order to wait for all the preceding packets. This statistic contains the total number of packets which had to be enqueued into receive queue.
vcRecoveries	If the acknowledgment for the packet is not received after two retransmissions it is assumed that the call path is broken and the call recovery procedure is requested. This statistic contains the total number of times the recovery was attempted after path failure and the call was successfully recovered
rcvQueueHwm	This statistic contains the high water mark for the receive queue.
duplicatedPkts	Due to the retransmissions done by the source VC or by the trunks, some packets can be received as duplicates of the previously received packets at the destination VC. This statistic contains the total number of duplicated packets received.
pktsNotXmt	All packets received from AP are immediately forwarded to the subnet by VC. However in some cases packets could not be forwarded due to congestion of local resources (common memory congestion, UTP trunk/NL congestion or destination PE congestion if destination resides on the same module). This counter contains the total number of instances were packets could not be forwarded to the subnet due to this reason.

ackStaTimer	When the ack stacking feature is enabled the acknowledgment is held for the finite time period in the hope that it can be piggybacked on the data packet heading in the other direction. This statistic contains the total number of times the ack stacking timer has expired.
windAdptTrig	The number of times the dynamic window enters the adaptation state. Once the count is incremented it is not incremented again until the window leaves the adaptation state and enters it again.
timeWindAdptState	The total time in minutes during which the window was throttled because of all adaptation periods.
windCycles	The total number of window cycles during all window adaptation periods.
lwmDynamWind	The lowest value that the window reached since the last probe.
currSizeDynamWind	The current size of the window.
fciFromSbnet	The total number of frames (not only data) received since the last probe with the FCI bit set.
bciFromSbnet	The number of frames received since the last probe with the BCI bit set.
bciToSbnet	The number of frames sent to the subnet with the BCI bit set.

GVCG34 (subtype 74)

Table 69
GVCG34 statistics (subtype 74)

Field	Type
srcDna	string
srcLcn	integer
destDna	string
(Sheet 1 of 2)	

Table 69 (Continued)
GVCG34 statistics (subtype 74)

Field	Type
destLcn	integer
callRef	integer
dataPktsSnt	integer
dataPktsRcv	integer
pktsRxmt	integer
outOfSeqPkts	integer
vcRecoveries	integer
rcvQueueHwm	integer
duplicatedPkts	integer
pktsNotXmt	integer
ackStaTimer	integer
windAdptTrig	integer
timeWindAdptState	integer
windCycles	integer
lwmDynamWind	integer
currSzeDynamWind	integer
fciFromSbnet	integer
bciFromSbnet	integer
bciToSbnet	integer
(Sheet 2 of 2)	

srcDna	<p>This fields contain full calling address (including DNIC for X.121 and Country Code for E.164); maximum 15 digits.</p> <p>Note: This is not necessary address of the source of the record. Source of the record can be identified from NAMS ID, PE# and component id fields.</p>
srcLcn	<p>Logical Channel Number of the calling end of the call.</p>
destDna	<p>This fields contain full called address (including DNIC for X.121 and Country Code for E.164); maximum 15 digits.</p>
destLcn	<p>Logical channel number of the called end of the call.</p>
callRef	<p>Call reference number. It is a three byte random number generated by the switch used as one of the fields in matching duplicated records generated at each end of the call.</p>
dataPktsSnt	<p>This statistic contains the total number of packets forwarded to the subnet by VC. This counter contains only the number of packets which were forwarded by an application (AP) i.e. the retransmissions are not included. The total number of retransmissions is given in the <i>packets retransmitted</i> statistic.</p>
dataPktsRcv	<p>This statistic contains the total number of data packets received from the remote end. The duplicated packets are counted as one.</p>

pktsRxmt	This is the total number of times packets were retransmitted to the subnet after retransmission timer expired. This includes all the retransmitted packets (data packets, idle probe, reset packets). This statistic does not tell how many times individual packets were retransmitted but it provides the total number of retransmissions which took place.
outOfSeqPkts	All data packets sent by the VC are equipped with the unique sequence number. The expected sequence number for the next receive packet is in the range which is determined by the last sequence number received and the receive credits available. All data packets which are received with out of sequence number but within the expected sequence range are enqueued into the receive queue in order to wait for all the preceding packets. This statistic contains the total number of packets which had to be enqueued into receive queue.
vcRecoveries	If the acknowledgment for the packet is not received after two retransmissions it is assumed that the call path is broken and the call recovery procedure is requested. This statistic contains the total number of times the recovery was attempted after path failure and the call was successfully recovered
rcvQueueHwm	This statistic contains the high water mark for the receive queue.
duplicatedPkts	Due to the retransmissions done by the source VC or by the trunks, some packets can be received as duplicates of the previously received packets at the destination VC. This statistic contains the total number of duplicated packets received.

pktsNotXmt	All packets received from AP are immediately forwarded to the subnet by VC. However in some cases packets could not be forwarded due to congestion of local resources (common memory congestion, UTP trunk/NL congestion or destination PE congestion if destination resides on the same module). This counter contains the total number of instances were packets could not be forwarded to the subnet due to this reason.
ackStaTimer	When the ack stacking feature is enabled the acknowledgment is held for the finite time period in the hope that it can be piggybacked on the data packet heading in the other direction. This statistic contains the total number of times the ack stacking timer has expired.
windAdptTrig	The number of times the dynamic window enters the adaptation state. Once the count is incremented it is not incremented again until the window leaves the adaptation state and enters it again.
timeWindAdptState	The total time in minutes during which the window was throttled because of all adaptation periods.
windCycles	The total number of window cycles during all window adaptation periods.
lwmDynamWind	The lowest value that the window reached since the last probe.
currSzeDynamWind	The current size of the window.
fciFromSbnet	The total number of frames (not only data) received since the last probe with the FCI bit set.
bciFromSbnet	The number of frames received since the last probe with the BCI bit set.
bciToSbnet	The number of frames sent to the subnet with the BCI bit set.

X25Port (subtype 63)

Table 70
DPN-AM and DPN-RM X.25 statistics (subtype 63)

Field	Type
version	integer
service	integer
type	integer
portType	integer
gtyld	integer
modType	integer
lclRid	integer
lclMid	integer
anid	integer
minCalls	integer
maxCalls	integer
maxNuiErr	integer
currFrmQueued	integer
lclFlowCtrl	integer
rmtFlowCtrl	integer
frmSnt	integer
frmRcv	integer
frmQueued	integer
overruns	integer
frmAbort	integer
noBuffer	integer
badFrmSize	integer
crcErr	integer
lineFail	integer
(Sheet 1 of 3)	

Table 70 (Continued)
DPN-AM and DPN-RM X.25 statistics (subtype 63)

Field	Type
underruns	integer
lrcErr	integer
frmRxmt	integer
modemChanges	integer
protocolErr	integer
ifrmSnt	integer
ifrmRcv	integer
rrSnt	integer
rrRcv	integer
rnrSnt	integer
rnrRcv	integer
rejSnt	integer
rejRcv	integer
frmrSnt	integer
frmrRcv	integer
frmSsqSnt	long integer
frmSsqRcv	long integer
ifrmBytesSnt	long integer
ifrmBytesRcv	long integer
x21DiSuc	integer
x21DiInc	integer
x21DiErr	integer
x21DoSuc	integer
x21DoInc	integer
x21DoErr	integer
(Sheet 2 of 3)	

Table 70 (Continued)
DPN-AM and DPN-RM X.25 statistics (subtype 63)

Field	Type
x21ClrErr	integer
x21ParityErr	integer
callReqSnt	integer
callReqRcv	integer
callConnSnt	integer
callConnRcv	integer
dataPktsSnt	integer
dataPktsRcv	integer
dataBytesSnt	integer
dataBytesRcv	integer
lclValAttmpt	integer
lclValFail	integer
lclXlateAttmpt	integer
lclXlateFail	integer
rmtValAttmpt	integer
rmtValFail	integer
rmtXlateAttmpt	integer
rmtXlateFail	integer
frmBytesSnt	long integer
frmBytesRcv	long integer
linkSpeed	integer
(Sheet 3 of 3)	

version	This field indicates the version number of the statistics record, currently 0.
service	This field indicates the Access Service that this link is running. Values: <ul style="list-style-type: none"> •0 - X.75 (not applicable in this case) •1 - X.25 •2 - X.25 Gateway
type	This TYPE field in combination with the SERVICE field indicates what component of a service is generating this statistics record and consequently what format the statistics will take on. Values: <ul style="list-style-type: none"> •0 Single Link Service (record format given here) •1-Multilink Service •2-Single Link Within Multilink Group
portType	The type of the port that the software generating this record runs on. Values: <ul style="list-style-type: none"> •0- Not running on port (reserved for Multilink Procedure) •1- V.24 •2- V.35 •3- X.21
gtyld	For this statistics record the only possible value is 0. If this statistics record is generated by an X.25 Gateway Service this field gives the unique GID assigned to this gateway.
modType	This field specifies the module type of the module on which this link is running. If set to 3, then the module is an AM or a DPNAC; if set to 2, then the module is an RM.

lclRid	<p>This field specifies the link access procedure implemented at the link layer, as follows:</p> <ul style="list-style-type: none">•LAPB = 0 (LAP-B)•LAPBES = 1 (LAP-B Extended, Single Octet)•LAPBED = 2 (LAP-B Extended, Double Octet)
lclMid	<p>This field specifies the Module ID (MID) of the module on which this link is running. If 0, then the module does not have a Module ID.</p>
anid	<p>This field contains the Adjacent Network ID of the adjacent network accessed through this X.25 gateway link. Values: 128 to 2048. This field is only relevant for X.25 gateways.</p>
minCalls	<p>Minimum number of calls up.</p>
maxCalls	<p>Maximum number of calls up.</p>
maxNuiErr	<p>Total number of illegal passwords received by the service over all its logical channels.</p>
lclFlowCtrl	<p>This field specifies the time, in seconds, that the link has flow controlled the remote end. This statistic usually indicates that the link is overloaded, and may have to be re-engineered to handle the offered traffic.</p>
rmtFlowCtrl	<p>This field specifies the time, in seconds, that the link has been flow controlled by the remote end. This statistic usually indicates that the link is overloaded, and may have to be re-engineered to handle the offered traffic.</p>
frmSnt	<p>This field specifies the number of frames sent to the link. This count includes I-frames (containing internetwork packets, and link specific packets), and Level 2 control frames, such as RRs, RNRs, REJs, FREJs, SABM, UA, etc.</p>

frmRcv	This field specifies the number of frames received from the link. This count includes I-frames (containing internetwork packets, and link specific packets), and Level 2 control frames, such as RRs, RNRs, REJs, FREJs, SABM, UA, etc.
currFrmQueued	<p>This field specifies the number of frames queued for transmission to the link at the point that the statistics probe was processed by the link.</p> <p>This field specifies the number of frames queued for transmission to the link at the point that the statistics probe was processed by the link. If the number of frames queued is greater than 255, then this field has a value of 255, or H.FF, and the true count of frames queued is in the statistical counter CURRENT FRAMES QD.</p> <p>Note that this field is not quite as useful as the CURRENT FRAMES QD, it is used at the link for thresholding. The thresholds all have maximum values of 254, and there is no need in most cases, to keep counting the number of link errors after hitting the upper bound. For example, the number of modem status changes in a statistics interval is counted up to a value of 254, and then counting stops. In the case of frames queued, the true count of frames queued is available in the software, so is provided in CURRENT FRAMES QD.</p> <p>This statistic usually indicates that the link is overloaded, and may have to be re-engineered to handle the offered traffic.</p>

overruns	<p>This field specifies the number of overruns in this statistics interval on the link. The PE is unable to accept a character before the next character has been accepted by the SCC on the PI.</p> <p>This statistic usually indicates that the PE is overloaded, and should be re-engineered in order that its traffic capacity is not exceeded.</p>
frmAbort	<p>This field specifies the number of frame aborts in this statistics interval on the link.</p> <p>This statistic usually indicates a problem at the remote end of the line (it is sending aborts), or on the facility (it is turning data on the line into aborts).</p>
noBuffer	<p>This field specifies the number of times in this statistics interval that the receive buffer used by the interrupt handler was insufficient to handle incoming traffic from the link.</p> <p>This statistic usually indicates that the link is overloaded, and requires re-engineering.</p>
badFrmSize	<p>This field specifies the number of frames received in this statistics interval from the link with a bad frame size (too long or too short).</p> <p>This statistic usually indicates a problem at the remote end of the line, it is sending frames of an incorrect frame size.</p>
crcErr	<p>This field specifies the number of CRC errors in this statistics interval on the link.</p> <p>This statistic usually indicates a data corruption problem somewhere in the network.</p>

lineFail	<p>This field specifies the number of line errors in this statistics interval on the link.</p> <p>This statistic usually indicates a problem on the SCC on the PI.</p>
underruns	<p>This field specifies the number of underruns in this statistics interval on the link, (the interrupt handler is unable to give the next character to the SCC in time).</p> <p>This statistic usually indicates that the PE is overloaded, and should be re-engineered so that its traffic capacity is not exceeded.</p>
lrcErr	<p>This field specifies the number of LRC errors in this statistics interval on the link.</p> <p>This statistic usually indicates a data corruption problem somewhere in the network.</p>
frmRxmt	<p>This field specifies the number of times in this statistics interval on the link, that a frame had to be retransmitted due to frames being rejected or timeouts.</p> <p>This statistic usually indicates a problem with the PI, the facility or the remote end.</p>
modemChanges	<p>This field specifies the number of modem status changes in this statistics interval on the link.</p> <p>This statistic usually indicates a problem with the PI, the facility or the remote end; an unterminated EIA cable plugged into the PI may also cause excessive modem status changes.</p>

protocolErr	<p>This field specifies the number of link protocol errors in this statistics interval on the link.</p> <p>This statistic, if coupled with other statistics errors, can indicate a problem on the hardware for the line. cable plugged into the PI may also cause excessive modem status changes. Otherwise, it indicates a problem with the remote end, where the protocol has not been implemented properly</p>
ifrmSnt	<p>This field specifies the number of I-frames sent to the link. I-frames include all internetwork packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.</p>
ifrmRcv	<p>This field specifies the number of I-frames received from the link. I-frames include all internetwork packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.</p>
rrSnt	<p>This field specifies the number of LAP-B Level 2 RRs sent to the link in this statistics interval.</p>
rrRcv	<p>This field specifies the number of LAP-B Level 2 RRs received from the link in this statistics interval.</p>
rnrSnt	<p>This field specifies the number of LAP-B Level 2 RNRs sent to the link in this statistics interval.</p>
rnrRcv	<p>This field specifies the number of LAP-B Level 2 RNRs received from the link in this statistics interval.</p>
rejSnt	<p>This field specifies the number of LAP-B Level 2 REJs sent to the link in this statistics interval.</p>
rejRcv	<p>This field specifies the number of LAP-B Level 2 REJs received from the link in this statistics interval.</p>
frmrSnt	<p>This field specifies the number of LAP-B Level 2 FREJs sent to the link in this statistics interval.</p>

frmrRcv	This field specifies the number of LAP-B Level 2 FREJs received from the link in this statistics interval.
frmSsqSnt	This field specifies the statistical sum of squares, for the number of bytes in I-frames sent to the link in this statistics interval. I-frames include all internetwork packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
frmSsqRcv	This field specifies the statistical sum of squares, for the number of bytes in I-frames received from the link in this statistics interval. I-frames include all internetwork packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
ifrmBytesSnt	This field specifies the number of bytes in I-frames sent to the link in this statistics interval. I-frames include all internetwork packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
ifrmBytesRcv	This field specifies the number of bytes in I-frames received from the link in this statistics interval. I-frames include all internetwork packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
x21DiSuc	Number of successful X.25/X.21 dial-in call attempts (level 1 established). Not meaningful for SLP.
x21DiInc	Number of incomplete X.25/X.21 dial-in call attempts (call cleared before level 1 established). Not meaningful for SLP.
x21DiErr	Number of incomplete X.25/X.21 dial-in call attempts aborted by an X.21 protocol violation (level 1 not established). Not meaningful for SLP.
x21DoSuc	Number of successful X.25/X.21 dial-out call attempts (level 1 established). Not meaningful for SLP.

x21DoInc	Number of incomplete X.25/X.21 dial-out call attempts (call cleared before level 1 established). Not meaningful for SLP.
x21DoErr	Number of incomplete X.25/X.21 dial-out call attempts aborted by an X.21 protocol violation (level 1 not established). Not meaningful for SLP.
x21ClrErr	Number of successful X.25/X.21 call attempts (both dial-in and dial-out) that encountered an X.21 protocol violation during disconnection. Not meaningful for SLP.
x21ParityErr	Number of X21 parity errors. Applicable only if port is an X.21 one.
callReqSnt	The number of call request packets sent to the link.
callReqRcv	The number of call request packets received from the link.
callConnSnt	The number of call connect packets sent to the link.
callConnRcv	The number of call connect packets received from the link.
dataPktsSnt	The number of data packets sent to the link.
dataPktsRcv	The number of data packets received from the link.
dataBytesSnt	The number of data bytes in the data packets sent to the link.
dataBytesRcv	The number of data bytes in the data packets received from the link.
lclValAtmpt	The number of attempts made to validate a local address at this gateway. Only relevant for X.25 gateway.
lclValFail	The number of failed local validation attempts made at this gateway. Only relevant for X.25 gateway.
lclXlateAtmpt	The number of attempts made to translate a local address at this gateway. Only relevant for X.25 gateway.

lclXlateFail	The number of failed local translation attempts made at this gateway. Only relevant for X.25 gateway.
rmtValAttmpt	The number of attempts made to validate a remote address at this gateway. Only relevant for X.25 gateway.
rmtValFail	The number of failed remote validation attempts made at this gateway. Only relevant for X.25 gateway.
rmtXlateAttmpt	The number of attempts made to translate a remote address at this gateway. Only relevant for X.25 gateway.
rmtXlateFail	The number of failed remote translation attempts made at this gateway. Only relevant for X.25 gateway.
frmBytesSnt	The number of bytes including the frame overhead sent to the link in this statistics interval.
frmBytesRcv	The number of bytes including the frame overhead received from the link in this statistics interval.
linkSpeed	This field indicates the speed of the link in bytes per second.

X25MLAgent (subtype 61)

This statistics record is generated by an X.25 Multilink Service and is generated by the Multilink Agent which manages all of the single links within the Multilink Group as well as provides packet layer processing. Each of the Single Links within the Multilink Group also generate a statistics record to provide link layer information. The statistics record generated by each of the Single Links is the X.25 SLP statistics record. The set of statistics records generated by an X.25 Multilink Group can be correlated by the GROUPID, a number unique in module (when the X.25 Multilink Service is not a gateway) or across the network (when the X.25 multilink is an X.25 gateway).

Table 71
DPN-AM and DPN-RM X.25 multilink agent statistics (subtype 61)

Field	Type
version	integer
service	integer
type	integer
portType	integer
gtyld	integer
modType	integer
lclRid	integer
lclMid	integer
anid	integer
minCalls	integer
maxCalls	integer
maxNuiErr	integer
maxOutPktsQueued	integer
maxInPktsQueued	integer
lostPktsOut	integer
lostPktsIn	integer
(Sheet 1 of 2)	

Table 71 (Continued)
DPN-AM and DPN-RM X.25 multilink agent statistics (subtype 61)

Field	Type
duplicPkts	integer
discPkts	integer
mlReset	integer
slpRxmtPkts	integer
pktsSnt	integer
pktsRcv	integer
badPktSize	integer
mlProtocolErr	integer
callReqSnt	integer
callReqRcv	integer
callConnSnt	integer
callConnRcv	integer
dataPktsSnt	integer
dataPktsRcv	integer
dataBytesSnt	integer
dataBytesRcv	integer
lclValAttmpt	integer
lclValFail	integer
lclXlateAttmpt	integer
lclXlateFail	integer
rmtValAttmpt	integer
rmtValFail	integer
rmtXlateAttmpt	integer
rmtXlateFail	integer
linkId	string
(Sheet 2 of 2)	

version	This field indicates the version number of the statistics record, currently 0.
service	<p>This field indicates the Access Service that this link is running. Values:</p> <ul style="list-style-type: none">•0 - X.75 (not applicable for the format described here)•1 - X.25•2 - X.25 Gateway <p>For this statistics record (Multilink Agent), the only possible values are 1 or 2.</p>
type	<p>This TYPE field in combination with the SERVICE field indicates what component of a service is generating this statistics record and consequently what format the statistics will take on. Values:</p> <ul style="list-style-type: none">•0 - Single Link Service•1 - Multilink Service (record described here)•2 - Single Link Within Multilink Group (see above)•For this statistics record (Multilink Agent), the only possible value is 1.
portType	<p>The type of the port that the software generating this record runs on. Values:</p> <ul style="list-style-type: none">•0- Not running on port (reserved for Multilink Procedure)•1- V.24•2- V.35•3- X.21•For this statistics record the only possible value is 0.

gtyld	This field identifies the Multilink Group handled by this Multilink Agent.
	If the X.25 Service is not a Gateway, then this is a MutlinkGroupID (1..64), a number unique in the module. If the X.25 Multilink Service is an X.25 gateway, then this number is a GatewayID, a number unique throughout the network.
	In an X.25 Multilink Service, several statistics records are generated. Each Single Link within the Multilink Group generates an X.25 SLP statistics record, as well as does the Multilink Agent that oversees the group and its single links and provides the packet layer processing. This number enables correlation of this different statistic records.
modType	This field specifies the module type of the module on which this link is running. If set to 3, then the module is an AM or a DPNAC; if set to 2, then the module is an RM.
lclRid	This field specifies the Routing ID (RID) of the module on which this link is running. If 0, then the module does not have a Routing ID. (On an AM, this field is 0; on an RM, this field is set).
lclMid	This field specifies the Module ID (MID) of the module on which this link is running. If 0, then the module does not have a Module ID.
anid	This field contains the Adjacent Network ID of the adjacent network accessed through this X.25 gateway link.
	This field only applies for X.25 Gateway statistics records.
	Values: 128 to 2048
minCalls	Minimum number of calls up.

maxCalls	Maximum number of calls up.
maxNuiErr	Total number of illegal passwords received by the service over all its logical channels.
maxOutPktsQueued	<p>This field specifies the maximum number of multilink frames which were queued (at any given time) for output to the remote end by the Multilink Procedure.</p> <p>Multilink frames include all X.25 information frames (or Level 3 packets) which traverse a single link in the multilink configuration. This includes all packets on a virtual call, and permanent virtual circuit. It also includes all control packets such as the Level 3 restart request packet, and restart confirmation packet, and the multilink reset request packet, and multilink reset confirmation packet.</p> <p>This field does not include supervisory and unnumbered Link Layer frames.</p>
maxInPktsQueued	<p>This field specifies the maximum number of multilink frames which were queued (at any given time) on input from the remote end by the X.25 Multilink Procedure.</p> <p>Multilink frames include all X.25 information frames (or Level 3 packets) which traverse a single link in the multilink configuration. This includes all packets on a virtual call, and permanent virtual circuit. It also includes all control packets such as the Level 3 restart request packet, and restart confirmation packet, and the multilink reset request packet, and multilink reset confirmation packet.</p> <p>This field does not include supervisory and unnumbered Link Layer frames.</p>

lostPktsOut	This field specifies the total number of X.25 information frames (or Level 3 packets) destined to the X.25 Multilink which were discarded or perceived to be lost since the last statistics probe.
lostPktsIn	This field specifies the total number of X.25 information frames (or Level 3 packets) received from the X.25 Multilink which were perceived to be lost since the last statistics probe.
duplicPkts	This field indicates the number of packets that have been received from the remote which are duplicates of packets already received by the Multilink Procedure since the last statistics probe.
discPkts	<p>This field indicates the number of packets that have been received from the remote which are outside of the Multilink Received Window since the last statistics probe.</p> <p>This statistic may be indicative of a problem in the Multilink Procedure implementation at the remote end. It may also indicate that the Multilink Window Sizes on the local and remote end are not identical.</p>
mlReset	This field specifies the total number of times that the Multilink Resetting Procedure has been invoked since the last statistics probe.
slpRxmtPkts	<p>This field indicates the number of packets that have been returned from a single link to the Multilink Procedure for retransmission over the same or a different single link since the last statistics probe.</p> <p>This statistic is indicative of a transmission problem on at least one of the X.25 Single Links within the Multilink Group. Examine the statistics record for the Single Links within the same Multilink Group to determine which link is problematic and diagnose the problem.</p>

pktsSnt	<p>This field indicates the number of packets sent to the X.25 Multilink. Packets include all information packets, and level 3 control packets, such as Idle Probes and Restarts.</p> <p>This field will closely match the sum of the IFRAMESENT fields from all of the X.25 SLP statistics records from each of the X.25 Single Links.</p>
pktsRcv	<p>This field indicates the number of packets received from the X.25 Multilink. Packets include all information packets, and level 3 control packets, such as Idle Probes and Restarts.</p> <p>This field will closely match the sum of the IFRAMESRCVD fields from all of the X.25 SLP statistics records from each of the X.25 Single Links.</p>
badPktSize	<p>This field indicates the number of packets received from the X.25 Multilink which do not contain a full Multilink Header.</p> <p>This statistic indicates that the remote side of the link is generating packets incorrectly.</p>
mlProtocolErr	<p>This field indicates that the Multilink Control Field in the Multilink Header of the packet received from the X.25 Multilink is formatted incorrectly.</p> <p>This statistic indicates that the remote side of the link is generating packets incorrectly.</p>
callReqSnt	<p>The number of call request packets sent to the link.</p>
callReqRcv	<p>The number of call requests packets received from the link.</p>
callConnSnt	<p>The number of call connect packets sent to the link.</p>
callConnRcv	<p>The number of call connect packets received from the link.</p>
dataPktsSnt	<p>The number of data packets sent to the link.</p>

dataPktsRcv	The number of data packets received from the link.
dataBytesSnt	The number of data bytes in the data packets sent to the link.
dataBytesRcv	The number of data bytes in the data packets received from the link.
lclValAttmpt	The number of attempts made to validate a local address at this gateway. Only valid when this multilink group is a gateway.
lclValFail	The number of failed local validation attempts made at this gateway. Only valid when this multilink group is a gateway.
lclXlateAttmpt	The number of attempts made to translate a local address at this gateway. Only valid when this multilink group is a gateway.
lclXlateFail	The number of failed local translation attempts made at this gateway. Only valid when this multilink group is a gateway.
rmtValAttmpt	The number of attempts made to validate a remote address at this gateway. Only valid when this multilink group is a gateway.
rmtValFail	The number of failed remote validation attempts made at this gateway. Only valid when this multilink group is a gateway.
rmtXlateAttmpt	The number of attempts made to translate a remote address at this gateway. Only valid when this multilink group is a gateway.
rmtXlateFail	The number of failed remote translation attempts made at this gateway. Only valid when this multilink group is a gateway.
linkId	Link identifier.

X25MLSerPort (subtype 62)

This statistics record is generated by each Single Link within an X.25 Multilink Group.

Table 72
DPN-AM and DPN-RM X.25 SLP statistics (subtype 62)

Field	Type
version	integer
service	integer
type	integer
portType	integer
gtyld	integer
modType	integer
linkProc	integer
lclRid	integer
lclMid	integer
anid	integer
currFrmQueued	integer
lclFlowCtrl	integer
rmtFlowCtrl	integer
frmSnt	integer
frmRcv	integer
frmQueued	integer
overruns	integer
frmAbort	integer
noBuffer	integer
badFrmSize	integer
crcErr	integer
lineFail	integer
(Sheet 1 of 3)	

Table 72 (Continued)
DPN-AM and DPN-RM X.25 SLP statistics (subtype 62)

Field	Type
underruns	integer
lrcErr	integer
frmRxmt	integer
modemChanges	integer
protocolErr	integer
ifrmSnt	integer
ifrmRcv	integer
rrSnt	integer
rrRcv	integer
rnrSnt	integer
rnrRcv	integer
rejSnt	integer
rejRcv	integer
frmrSnt	integer
frmrRcv	integer
frmSsqSnt	long integer
frmSsqRcv	long integer
ifrmBytesSnt	long integer
ifrmBytesRcv	long integer
x21DiSuc	integer
x21DiInc	integer
x21DiErr	integer
x21DoSuc	integer
x21DoInc	integer
x21DoErr	integer
(Sheet 2 of 3)	

Table 72 (Continued)
DPN-AM and DPN-RM X.25 SLP statistics (subtype 62)

Field	Type
x21ClrErr	integer
x21ParityErr	integer
frmBytesSnt	long integer
frmBytesRcv	long integer
linkSpeed	integer
(Sheet 3 of 3)	

- version** This field indicates the version number of the statistics record, currently 0.
- service** This field indicates the access service that this link is running. Values:
- 0 - X.75
 - 1 - X.25
 - 2 - X.25 Gateway
 - For this statistics record (Single Link in Multilink Group) this field can assume only values 1 or 2.
- type** This field in combination with the service field indicates what component of a service is generating this statistics record and consequently what format the statistics will take on. Values:
- 0 - Single Link Service
 - 1 - Multilink Service
 - 2 - Single Link Within Multilink Group
 - For this statistics record (Single Link in Multilink Group) this field can only assume a value of 2.

portType	<p>The type of the port that the software generating this record runs on. Values:</p> <ul style="list-style-type: none">•0 - Not running on port (reserved for Multilink procedure)•1 - V.24•2 - V.35•3 - X.21 <p>•The Single Link Procedure runs always on port, so the only values the PORTTYPE may assume for this statistics record are 1, 2 or 3.</p>
gtyld	<p>This field identifies the Multilink Group to which the SLP that generates this record belongs. If the Multilink service is an X.25 service (not a gateway), then this is a MultilinkGroupID (1...64). When the Multilink service is an X.25 gateway, then this field is a GatewayID (1...9999). This field can be used to correlate the various statistics records from the different components of the Multilink Group.</p>
modType	<p>This field specifies the module type of the module on which this link is running. If set to 3, then the module is an AM or a DPNAC; if set to 2, then the module is an RM.</p>
linkProc	<p>This field specifies the link access procedure implemented at the link layer, as:</p> <ul style="list-style-type: none">•LAPB = 0 (LAP-B)•LAPBES = 1 (LAP-B Extended, Single Octet)•LAPBED= 2 (LAP-B Extended, Double Octet)
lclRid	<p>This field specifies the Routing ID (RID) of the module on which this link is running. If 0, then the module does not have a Routing ID. (On an AM, this field is 0; on an RM, this field is set).</p>

lclMid	This field specifies the Module ID (MID) of the module on which this link is running. If 0, then the module does not have a module ID.
anid	This field contains the Adjacent Network ID of the adjacent network accessed through this X.25 gateway link. This field only applies for X.25 gateway statistics records. Values: 128 to 2048
lclFlowCtrl	This field specifies the time, in seconds, that the link has flow controlled the remote end. This statistic usually indicates that the link is overloaded, and may have to be re-engineered to handle the offered traffic.
rmtFlowCtrl	This field specifies the time, in seconds, that the link has been flow controlled by the remote end. This statistic usually indicates that the link is overloaded, and may have to be re-engineered to handle the offered traffic.
frmSnt	This field specifies the number of frames sent to the link. This count includes I-frames (containing internetwork packets, and link specific packets), and Level 2 control frames, such as RRs, RNRs, REJs, FREJs, SABM, UA, etc.
frmRcv	This field specifies the number of frames received from the link. This count includes I-frames (containing internetwork packets, and link specific packets), and Level 2 control frames, such as RRs, RNRs, REJs, FREJs, SABM, UA, etc.

currFrmQueued

This field specifies the number of frames queued for transmission to the link at the moment that the statistics probe was processed by the link.

This field specifies the number of frames queued for transmission to the link at the moment that the statistics probe was processed by the link. If the number of frames queued is greater than 255, then this field has a value of 255, or H.ff, and the true count of frames queued is in the statistical counter CURRENT FRAMES QD.

Note that this field is not quite as useful as the CURRENT FRAMES QD, but it is used at the link for thresholding. The thresholds all have maximum values of 254, and there is no need in most cases, to keep counting the number of link errors after hitting the upper bound. For example, the number of modem status changes in a statistics interval is counted up to a value of 254, and then counting stops. In the case of frames queued, the true count of frames queued is available in the software, so is provided in CURRENT FRAMES QD.

This statistic usually indicates that the link is overloaded, and may have to be re-engineered to hand the offered traffic.

overruns

This field specifies the number of overruns in this statistics interval on the link. The PE is unable to accept a character before the next character has been accepted by the SCC on the PI.

This statistic usually indicates that the PE is overloaded, and should be re-engineered in order that its traffic capacity is not exceeded.

frmAbort	<p>This field specifies the number of frame aborts in this statistics interval on the link.</p> <p>This statistic usually indicates a problem at the remote end of the line (it is sending aborts), or on the facility (it is turning data on the line into aborts).</p>
noBuffer	<p>This field specifies the number of times in this statistics interval that the receive buffer used by the interrupt handler was insufficient to handle incoming traffic from the link.</p> <p>This statistic usually indicates that the link is overloaded and requires re-engineering.</p>
badFrmSize	<p>This field specifies the number of frames received in this statistics interval from the link with a bad frame size (too long or too short).</p> <p>This statistic usually indicates a problem at the remote end of the line, it is sending frames of an incorrect frame size.</p>
crcErr	<p>This field specifies the number of CRC errors in this statistics interval on the link.</p> <p>This statistic usually indicates a data corruption problem somewhere in the network.</p>
lineFail	<p>This field specifies the number of line errors in this statistics interval on the link.</p> <p>This statistic usually indicates a problem on the SCC on the PI.</p>

underruns	<p>This field specifies the number of underruns in this statistics interval on the link, (the interrupt handler is unable to give the next character to the SCC in time).</p> <p>This statistic usually indicates that the PE is overloaded, and should be re-engineered so that its traffic capacity is not exceeded.</p>
lrcErr	<p>This field specifies the number of LRC errors in this statistics interval on the link.</p> <p>This statistic usually indicates a data corruption problem somewhere in the network.</p>
frmRxmt	<p>This field specifies the number of times in this statistics interval on the link, that a frame had to be retransmitted due to frames being rejected or timeouts.</p> <p>This statistic usually indicates a problem with the PI, the facility or the remote end.</p>
modemChanges	<p>This field specifies the number of modem status changes in this statistics interval on the link.</p> <p>This statistic usually indicates a problem with the PI, the facility or the remote end; an unterminated EIA cable plugged into the PI may also cause excessive modem status changes.</p>
protocolErr	<p>This field specifies the number of link protocol errors in this statistics interval on the link.</p> <p>This statistic, if coupled with other statistics errors, can indicate a problem on the hardware for the link. Otherwise, it indicates a problem with the remote end, where the protocol has not been implemented properly.</p>

ifrmSnt	This field specifies the number of I-frames send to the link. I-frames include all inter-network packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
ifrmRcv	This field specifies the number of I-frames received from the link. I-frames include all internetwork packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
rrSnt	This field specifies the number of LAP-B Level 2 RRs sent to the link in this statistics interval.
rrRcv	This field specifies the number of LAP-B Level 2 RRs received from the link in this statistics interval.
rnrSnt	This field specifies the number of LAP-B Level 2 RNRs sent to the link in this statistics interval.
rnrRcv	This field specifies the number of LAP-B Level 2 RNRs received from the link in this statistics interval.
rejSnt	This field specifies the number of LAP-B Level 2 REJs sent to the link in this statistics interval.
rejRcv	This field specifies the number of LAP-B Level 2 REJs received from the link in this statistics interval.
frmRstSnt	This field specifies the number of LAP-B Level 2 FREJs sent to the link in this statistics interval.
frmRstRcv	This field specifies the number of LAP-B Level 2 FREJs received from the link in this statistics interval.
frmSsqSnt	This field specifies the statistical sum of squares, for the number of bytes in I-frames sent to the link in this statistics interval. I-frames include all internetwork packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.

frmSsqRcv	This field specifies the statistical sum of squares, for the number of bytes in I-frames received from the link in this statistics interval. I-frames include all internetwork packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
ifrmBytesSnt	This field specifies the number of bytes in I-frames sent to the link in this statistics interval. I-frames include all internetwork packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
ifrmBytesRcv	This field specifies the number of bytes in I-frames received from the link in this statistics interval. I-frames include all inter-network packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
x21DiSuc	Number of successful X.25/X.21 dial-in call attempts (level 1 established). Not meaningful for SLP.
x21DiInc	Number of incomplete X.25/X.21 dial-in call attempts (call cleared before level 1 established). Not meaningful for SLP.
x21DiErr	Number of incomplete X.25/X.21 dial-in call attempts aborted by an X.21 protocol violation (level 1 not established). Not meaningful for SLP.
x21DoSuc	Number of successful X.25/X.21 dial-out call attempts (level 1 established). Not meaningful for SLP.
x21DoInc	Number of incomplete X.25/X.21 dial-out call attempts (call cleared before level 1 established). Not meaningful for SLP.
x21DoErr	Number of incomplete X.25/X.21 dial-out call attempts aborted by an X.21 protocol violation (level 1 not established). Not meaningful for SLP.

x21ClrErr	Number of successful X.25/X.21 call attempts (both dial-in and dial-out) that encountered an X.21 protocol violation during disconnection. Not meaningful for SLP.
x21ParityErr	Number of X21 parity errors. Applicable only if port is an X.21 one.
frmBytesSnt	The number of bytes including the frame overhead sent to the link in this statistics interval.
frmBytesRcv	The number of bytes including the frame overhead received from the link in this statistics interval.
linkSpeed	This field specifies the speed of the link. For speeds below 32 kbytes/sec (=256 kbits/sec) linkspeed is specified in bytes per second, is a decimal value, and has bit 16 set to 0. For speeds 32 kbytes/sec, or more, linkspeed is specified in kbytes per second, is a decimal value, and has bit 16 set to 1.

X75MLAgentG32 (subtype 71)

Table 73
X.75 ML (G32) statistics (subtype 71)

Field	Type
version	integer
service	integer
type	integer
gtyld	integer
modType	integer
lclRid	integer
lclMid	integer
anid	integer
maxCalls	integer
locValAttempts	integer
locValFailures	integer
locXlatAttempts	integer
locXlatFailures	integer
locRcValAttempts	integer
locRcValFailures	integer
remValAttempts	integer
remValFailures	integer
remXlatAttempts	integer
remXlatFailures	integer
remRcValAttempts	integer
remRcValFailures	integer
maxOutPktsQueued	integer
maxInPktsQueued	integer
lostPktsOut	integer
(Sheet 1 of 2)	

Table 73 (Continued)
X.75 ML (G32) statistics (subtype 71)

Field	Type
lostPktsIn	integer
duplicPkts	integer
discPkts	integer
mlReset	integer
slpRxmtPkts	integer
pktsSnt	integer
pktsRcv	integer
badPktSize	integer
mlProtocolErr	integer
callReqSnt	integer
callReqRcv	integer
callConnSnt	integer
callConnRcv	integer
dataPktsSnt	integer
dataPktsRcv	integer
dataBytesSnt	integer
dataBytesRcv	integer
linkId	string
(Sheet 2 of 2)	

Note: If this statistics record is generated from an X.75 single link within an X.75 gateway offering multilink service, (that is, the TYPE field is set to 2), then these fields contain a value of 0, and are not relevant. The multilink agent, which also produces its own statistics record, provides the information for these fields as a whole.

version	This field indicates the version number of the statistics record, currently 0.
service	This field indicates the access service that this link is running. Values: 0 - X.75
type	This statistics record is generated by the X.75 multilink agent in an X.75 gateway offering multilink service, as identified by this field. Values: 1 - X.75 gateway multilink service
gtyld	<p>This field indicates the unique (network wide) gateway ID that identifies the X.75 gateway.</p> <p>An X.75 gateway offering multilink service, generates several statistics records. Each single link within the multilink group generates an X.75 port statistics record. The multilink agent which oversees the X.75 gateway and its single links, provides the packet layer processing which generates these statistic records, also using the unique gateway ID. The X.75 gateway offering multilink service, uses this gateway ID to correlate the various statistics records from the different components of the multilink gateway.</p>
modType	This field specifies the module type of the module on which the X.75 gateway is running. If set to 3, then the module is an AM or a DPNAC; if set to 2, then the module is an RM.
lclRid	This field specifies the routing ID, (RID), of the module on which this link is running. If 0, then the module does not have a routing ID. (On an AM, this field is 0; on an RM, the field is set).
lclMid	This field specifies the module ID, (MID), of the module on which this X.75 link is running. If 0, then the module does not have a module ID.

anid	This field contains the adjacent network ID of the adjacent network accessed through the X.75 link. Values: 128 to 2048
maxCalls	This field indicates the maximum number of calls which were up over the X.75 gateway, during the statistics interval. Values: 0 to 1000
locValAttempts	The number of attempts made to validate a local address at this gateway.
locValFailures	The number of failed local validations made at this gateway.
locXlatAttempts	The number of attempts made to translate a local address at this gateway.
locXlatFailures	The number of failed local translation attempts made at this gateway.
locRcValAttempts	The number of attempts made to validate the presence of the reverse charge facility against the local address specified, at this gateway.
locRcValFailures	The number of failed local reverse charging validations made at this gateway.
remValAttempts	The number of attempts made to validate a remote address at this gateway.
remValFailures	The number of failed remote validations made at this gateway.
remXlatAttempts	The number of attempts made to translate a remote address at this gateway.
remXlatFailures	The number of failed remote translation attempts made at this gateway.
remRcValAttempts	The number of attempts made to validate the presence of the reverse charge facility against the remote address specified, at this gateway.
remRcValFailures	The number of failed remote reverse charging validations made at this gateway.

maxOutPktsQueued This field specifies the maximum number of multilink frames which were queued (at any given time) for output to the remote STE, due to the X.75 gateway implementing the multilink procedure.

Multilink frames include all X.75 information frames (or level 3 packets) which traverse a single link in the multilink configuration. This includes all packets on a virtual call, and on a permanent virtual circuit. It also includes all control packets such as the level 3 restart request packet, the restart confirmation packet, the multilink reset request packet, and the multilink reset confirmation packet. This field does not include supervisory and unnumbered link layer frames.

maxInPktsQueued This field specifies the maximum number of multilink frames which were queued (at any given time) on input from the remote STE, due to the X.75 gateway implementing the multilink procedure.

Multilink frames include all X.75 information frames (or level 3 packets) which traverse a single link in the multilink configuration. This includes all packets on a virtual call, and on a permanent virtual circuit. It also includes all control packets such as the level 3 restart request packet, the restart confirmation packet, the multilink reset request packet, and the multilink reset confirmation packet. This field does not include supervisory and unnumbered link layer frames.

lostPktsOut This field specifies the total number of X.75 information frames (or Level 3 packets) destined to the X.75 gateway, which were discarded or perceived to be lost since the last statistics probe.

lostPktsIn	This field specifies the total number of X.75 information frames (or Level 3 packets) received from the X.75 gateway, which were perceived to be lost since the last statistics probe.
duplicPkts	This field indicates the number of packets received from the remote end, which are duplicates of packets already received by the multilink procedure, since the last statistics probe.
discPkts	This field indicates the number of packets received from the remote STE, which are outside of the multilink received window, since the last statistics probe. This statistic may be indicative of a problem in the multilink procedure implementation at the remote STE. It may also indicate that the multilink window sizes on the local and remote STEs are not identical.
mlReset	This field specifies the total number of times that the multilink resetting procedure has been invoked since the last statistics probe.
slpRxmtPkts	This field indicates the number of packets that have been returned from a single link to the multilink procedure for retransmission over the same or a different single link since the last statistics probe. This statistic is indicative of a transmission problem on at least one of the X.75 single links within the multilink group. Examine the statistics record for the single links within the same multilink group to determine which link is problematic and diagnose the problem.

pktsSnt	<p>This field indicates the number of packets sent to the X.75 gateway. Packets include all internetwork packets, and include X.75 link specific packets such as level 3 idle probes and restarts.</p> <p>This field will closely match the sum of the IFRAMESENT fields from all of the X.75 port statistics records from each of the X.75 single links.</p>
pktsRcv	<p>This field indicates the number of packets received from the X.75 gateway. Packets include all internetwork packets, and include X.75 link specific packets such as level 3 idle probes and restarts.</p> <p>This field will closely match the sum of the IFRAMESRCVD fields from all of the X.75 port statistics records from each of the X.75 single links.</p>
badPktSize	<p>This field indicates the number of packets received from the X.75 gateway which do not contain a full multilink header.</p> <p>This statistic indicates that the remote end of the link is generating packets incorrectly.</p>
mlProtocolErr	<p>This field indicates that the multilink control field in the multilink header of the packet, received from the X.75 gateway, is formatted incorrectly.</p> <p>This statistic indicates that the remote end of the link is generating packets incorrectly.</p>
callReqSnt	The number of call request packets sent to link.
callReqRcv	The number of call request packets received from link.
callConnSnt	The number of call connect packets sent to link.
callConnRcv	The number of call connect packets received from link.
dataPktsSnt	The number of data packets sent to link.

dataPktsRcv	The number of data packets received from link.
dataBytesSnt	The number of data bytes in the data packets sent to link.
dataBytesRcv	The number of data bytes in data packets received from link.
linkId	Link identifier.

X75PortG32 (subtype 70)

Table 74
X.75 port (G32) statistics (subtype 70)

Field	Type
version	integer
service	integer
type	integer
gtyld	string
modType	integer
linkProc	integer
lclRid	integer
lclMid	integer
anid	integer
maxCalls	integer
locValAttempts	integer
locValFailures	integer
locXlatAttempts	integer
locXlatFailures	integer
locRcValAttempts	integer
locRcValFailures	integer
remValAttempts	integer

(Sheet 1 of 3)

Table 74 (Continued)
X.75 port (G32) statistics (subtype 70)

Field	Type
remValFailures	integer
remXlatAttempts	integer
remXlatFailures	integer
remRcValAttempts	integer
remRcValFailures	integer
currFrmQueued	integer
lclFlowCtrl	integer
rmtFlowCtrl	integer
frmSnt	integer
frmRcv	integer
frmQueued	integer
overruns	integer
frmAbort	integer
noBuffer	integer
badFrmSize	integer
crcErr	integer
cardErr	integer
underruns	integer
lrcErr	integer
frmRxmt	integer
modemChanges	integer
protocolErr	integer
ifrmSnt	integer
ifrmRcv	integer
rrSnt	integer
(Sheet 2 of 3)	

Table 74 (Continued)
X.75 port (G32) statistics (subtype 70)

Field	Type
rrRcv	integer
rnrSnt	integer
rnrRcv	integer
rejSnt	integer
rejRcv	integer
frmSnt	integer
frmRcv	integer
ifrmSsqSnt	long integer
ifrmSsqRcv	long integer
ifrmBytesSnt	long integer
ifrmBytesRcv	long integer
linkSpeed	integer
callReqSnt	integer
callReqRcv	integer
callConnSnt	integer
callConnRcv	integer
dataPktsSnt	integer
dataPktsRcv	integer
dataBytesSnt	integer
dataBytesRcv	integer
frmBytesSnt	long integer
frmBytesRcv	long integer
(Sheet 3 of 3)	

Note: If this statistics record is generated from an X.75 single link within an X.75 gateway offering multilink service, (that is, the TYPE field is set to 2), then these fields contain a value of 0, and are not relevant. The multilink agent, which also produces its own statistics record, provides the information for these fields as a whole.

version	This field indicates the version number of the statistics record, currently 0.
service	This field indicates the access service which this link is running. Values: 0 - X.75
type	<p>As this statistics record is generated both by the X.75 gateway offering single link service (directly running a LAP-B port), and by each X.75 single link within an X.75 multilink group, (each also directly running a LAP-B port), this field distinguishes between the two types of services.</p> <p>Values:</p> <ul style="list-style-type: none">•0 - X.75 gateway single link service•2 - X.75 single link within multilink group

gtyld	<p>If this statistics record is generated by an X.75 gateway offering single link service, (that is, the TYPE field is set to 0), then this field indicates the unique (network wide), gateway ID that identifies the X.75 gateway.</p> <p>If this statistics record is generated from an X.75 single link within an X.75 gateway offering multilink service, (that is, the TYPE field is set to 2), then this field also indicates the unique gateway ID identifying the X.75 multilink gateway. However, in the case of the multilink, several statistics records are generated. Each single link within the multilink group generates this statistics record. The multilink agent which oversees the X.75 gateway and its single links, provides the packet layer processing and generates a different statistics record also with this unique gateway ID. For the X.75 gateway offering multilink service, this gateway ID can be used to correlate the various statistics records from the different components of the multilink gateway.</p>
modType	<p>This field specifies the module type of the module on which this link is running. If set to 3, then the module is an AM or a DPNAC; if set to 2, then the module is an RM.</p>
linkProc	<p>This field specifies the link access procedure implemented at the link layer, as:</p> <ul style="list-style-type: none">•LAPB = 0 (LAP-B)•LAPBES = 1 (LAP-B extended, Single octet)•LAPBED = 2 (LAP-B extended, double octet)
lclRid	<p>This field specifies the routing ID, (RID), of the module on which this link is running. If 0, then the module does not have a routing ID. (On an AM, this field is 0; on an RM, this field is set.)</p>

lclMid	This field specifies the module ID, (MID), of the module on which this link is running. If 0, then the module does not have a module ID.
anid	This field contains the adjacent network ID of the adjacent network accessed through this X.75 link. Values: 128 to 2048
maxCalls	<p>If this statistics record is generated from an X.75 gateway offering single link service, (that is, the TYPE field is set to 0), then this field indicates the maximum number of calls which were up over the X.75 gateway during the statistics interval.</p> <p>If this statistics record is generated from an X.75 single link within an X.75 gateway offering multilink service, (that is, the TYPE field is set to 2), then this field contains a value of 0, and is not relevant. The multilink agent, which also produces its own statistics record, provides the maximum number of calls up over the X.75 gateway as a whole.</p> <p>Values: 0 to 1000</p>
locValAttempts	The number of attempts made to validate a local address at this gateway.
locValFailures	The number of failed local validations made at this gateway.
locXlatAttempts	The number of attempts made to translate a local address at this gateway.
locXlatFailures	The number of failed local translation attempts made at this gateway.
locRcValAttempts	The number of attempts made to validate the presence of the reverse charge facility against the local address specified, at this gateway.
locRcValFailures	The number of failed local reverse charging validation made at this gateway.

remValAttempts	The number of attempts made to validate a remote address at this gateway.
remValFailures	The number of failed remote validation made at this gateway.
remXlatAttempts	The number of attempts made to translate a remote address at this gateway.
remXlatFailures	The number of failed remote translation attempts made at this gateway.
remRcValAttempts	The number of attempts made to validate the presence of the reverse charge facility against the remote address specified, at this gateway.
remRcValFailures	The number of failed remote reverse charging validations made at this gateway.
currFrmQueued	This field specifies the number of frames queued for transmission to the link, when the statistics probe was processed by the link.
lclFlowCtrl	<p>This field specifies the time, in seconds, that the link has flow controlled the remote end.</p> <p>This statistic usually indicates that the link is overloaded, and may have to be re-engineered to handle the offered traffic.</p>
rmtFlowCtrl	This field specifies the time, in seconds, that the link has been flow controlled by the remote end. This statistic usually indicates that the link is overloaded, and may have to be re-engineered to handle the offered traffic.
frmSnt	This field specifies the number of frames sent to the link. This count includes I-frames (containing internetwork packets, and link specific packets), and Level 2 control frames, such as RRs, RNRs, REJs, FREJs, SABM, UA, etc.

frmRcv	This field specifies the number of frames received from the link. This count includes I-frames (containing internetwork packets, and link specific packets), and Level 2 control frames, such as RRs, RNRs, REJs, FREJs, SABM, UA, etc.
frmQueued	<p>This field specifies the high water mark of the number of frames queued for transmission to the link when the statistics probe was processed by the link. If the number of frames queued is greater than 255, then this field has a value of 255, or H.FF. The true count of frames queued is in the statistical counter CURRENT FRAMES QD. This statistic usually indicates that the link is overloaded, and may have to be re-engineered to handle the offered traffic.</p> <p>This field is not quite as useful as the CURRENT FRAMES QD, but it is used at the link for thresholding. The thresholds all have maximum values of 254, as there is no need in most cases, to keep counting the number of link errors after hitting the upper bound. For example, the number of modem status changes in a statistics interval is counted until a value of 254 is reached, then counting stops. In the case of frames queued, the true count of frames queued is available in the software, (provided in CURRENT FRAMES QD).</p>
overruns	This field specifies the number of overruns in this statistics interval on the link. The PE is unable to accept a character before the next character has been accepted by the SCC on the PI. This statistic usually indicates that the PE is overloaded, and should be re-engineered in order that its traffic capacity not be exceeded.

frmAbort	<p>This field specifies the number of frame aborts in this statistics interval on the link.</p> <p>This statistic usually indicates a problem at the remote end of the line (it is sending aborts), or on the facility (it is turning data on the line into aborts).</p>
noBuffer	<p>This field specifies the number of times in this statistics interval that the receive buffer, used by the interrupt handler, was insufficient to handle incoming traffic from the link.</p> <p>This statistic usually indicates that the link is overloaded, and requires re-engineering.</p>
badFrmSize	<p>This field specifies the number of frames, received in this statistics interval from the link, with a bad frame size (too long or too short). This statistic usually indicates a problem at the remote end of the line, it is sending frames of an incorrect frame size.</p>
crcErr	<p>This field specifies the number of CRC errors in this statistics interval on the link. This statistic usually indicates a data corruption problem somewhere in the network.</p>
cardErr	<p>This field specifies the number of line errors in this statistics interval. This statistic usually indicates a problem on the SCC on the PI.</p>
underruns	<p>This field specifies the number of underruns in this statistics interval on the link, (the interrupt handler is unable to give the next character to the SCC in time). This statistic usually indicates that the PE is overloaded, and should be re-engineered so that its traffic capacity is not exceeded.</p>
lrcErr	<p>This field specifies the number of LRC errors in this statistics interval on the link. This statistic usually indicates a data corruption problem somewhere in the module.</p>

frmRxmt	This field specifies the number of times in this statistics interval on the link, that a frame had to be retransmitted due to frames being rejected or timeouts. This statistic usually indicates a problem with the PI, the facility or the remote end.
modemChanges	This field specifies the number of modem status changes in this statistics interval on the link. This statistic usually indicates a problem with the PI, the facility or the remote end; an unterminated cable plugged into the PI may also cause excessive modem status changes.
protocolErr	This field specifies the number of link protocol errors in this statistics interval on the link. This statistic, if coupled with other statistics errors, can indicate a problem on the hardware for the line. A cable plugged into the PI may also cause excessive modem status changes. Otherwise, it indicates a problem with the remote end, where the protocol has not been implemented properly.
ifrmSnt	This field specifies the number of I-frames sent to the link. I-frames include all internetwork packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
ifrmRcv	This field specifies the number of I-frames received from the link. I-frames include all internetwork packets, and include link specific packets such as Level 3 Idle Probes, and Restarts.
rrSnt	This field specifies the number of LAP-B level 2 RRs sent to the link in this statistics interval.
rrRcv	This field specifies the number of LAP-B level 2 RRs received from the link in this statistics interval.
rnrSnt	This field specifies the number of LAP-B level 2 RNRs sent to the link in this statistics interval.

rnRcv	This field specifies the number of LAP-B level 2 RNRs received from the link in this statistics interval.
rejSnt	This field specifies the number of LAP-B level 2 REJs sent to the link in this statistics interval.
rejRcv	This field specifies the number of LAP-B level 2 REJs received from the link in this statistics interval.
frmrSnt	This field specifies the number of LAP-B level 2 FREJs sent to the link in this statistics interval.
frmrRcv	This field specifies the number of LAP-B level 2 FREJs received from the link in this statistics interval.
ifrmSsqSnt	This field specifies the statistical sum of squares, for the number of bytes in I-frames sent to the link in this statistics interval. I-frames include all internetwork packets, and include link specific packets such as level 3 idle probes, and restarts.
ifrmSsqRcv	This field specifies the statistical sum of squares, for the number of bytes in I-frames received from the link in this statistics interval. I-frames include all internetwork packets, and include link specific packets such as level 3 idle probes, and restarts.
ifrmBytesSnt	This field specifies the number of user bytes in I-frames sent to the link in this statistics interval. I-frames include all internetwork packets, and include link specific packets such as level 3 idle probes, and restarts.
ifrmBytesRcv	The number of bytes including frame overhead received from the link in this statistics interval.
linkSpeed	This field indicates the speed of the link in bytes per second.
callReqSnt	The number of call request packets sent to link.
callReqRcv	The number of call request packets received from link.

callConnSnt	The number of call connect packets sent to link.
callConnRcv	The number of call connect packets received from link.
dataPktsSnt	The number of data packets sent to link.
dataPktsRcv	The number of data packets received from link.
dataBytesSnt	The number of data bytes in the data packets sent to link.
dataBytesRcv	The number of data bytes in data packets received from link.
frmBytesSnt	The number of bytes including frame overhead sent to the link in this statistics interval.
frmBytesRcv	This field specifies the number of user bytes in I-frames received from the link in this statistics interval. I-frames include all internetwork packets, and include link specific packets such as level 3 idle probes, and restarts.

Chapter 4

DPN alarm records

DPN alarms are generated when hardware or software malfunctions are detected or cleared. Alarms provide information on fault conditions which is sent to monitoring devices such as local consoles or Preside Multiservice Data Manager (MDM) workstations. This provides a network-wide view of occurring problems, which in turn helps to pinpoint malfunctions or abnormalities at any site in the network.

The following fields are common to all BDF DPN alarm records:

switchType: dpn

dataType: ala

dateTime: `yyyymmddTHHMMSS.ss`, where `yyyy`=year; `mm`=month; `dd`=day; `HH`=hour; `MM`=minute; `SS`=seconds; `ss`=tenths of a second

customerId: Customer Network Management Identifier assigned to network components (PEs and POs) dedicated to a Virtual Private Network

dspId: Module DSP group ID

namsId: Integers which identify AM, RM, or PM

seqNum: Sequence number assigned by the network administration and management system (NAMS) device

componentId: A component identifier describes a NAMS object. A component identifier is composed of a list of qualifiers; each with two elements—category (PE, PI, PO, CH) and a uniquely numbered name.

All other fields within BDF alarm files are controlled by the `dpn_ala.rdf` file.

Alarm output format

The alarm record fields are briefly described below. For information on interpreting alarms, see 241-1001-506 *DPN-100 Alarm Console Indications*.

Table 75
ASCII alarm output format

Field	Type
severity	enumerated
alarmType	enumerated
sourceId	integer and hexadecimal
faultCode	integer and hexadecimal
condition	string
action	enumerated
operatorData	formatted hexadecimal
expertData	formatted hexadecimal
comment	string

severity	The severity level of the alarm is: process degrade overload minor major wildcard
alarmType	<p>This field specifies how this alarm interacts with any active alarm list that this alarm might be directed to. Values are:</p> <p>set clear message</p> <p>A clear removes a set alarm in an active alarm list. The alarm is removed if the following fields match the clear alarm: Severity, Condition number, Action Code, NCS-NAMS Component Identifier. A generalized clear operation is achieved by a “wild card” value in the fields of an alarm clear. This excludes this field from comparison when performing an alarm clear.</p> <p>The NCS-NAMS Component Identifier if not completely specified in the clear alarm removes alarms that match the component as far as specified. Any alarms in the active alarm list with components that match the clear to this point but have been further specified are cleared.</p>
sourceId	Use as index into 241-1001-506 <i>DPN-100 Alarm Console Indications</i> .
faultCode	Use as index into 241-1001-506 <i>DPN-100 Alarm Console Indications</i> .
condition	The condition of the alarm in ASCII.

action	This code indicates what category of action this alarm requires. Values are: serviceData hardwareFault softwareFault securityViolation protocolViolation debugInfo network engineering operation wildcard
operatorData	Extra data that would be useful to the operator.
expertData	Extra data that would be useful to an expert at an STC. Trap data could go here.
comment	This is the ASCII text that is generated by the module for the module's local operator.

Note: Alarm comment and condition field use the BDF escape sequence.

Chapter 5

DPN operator command logs

Each operator command (local or remote) that is issued to a node or module in the network is logged and transferred to a Management Data Provider (MDP). The log records are also displayed on the selected NCS consoles or on the Preside MDM log display tool.

The following fields are common to all BDF DPN log records:

switchType: dpn

dataType: log

dateTime: *yyyymmDDTHHMMSS.ss*, where *yyyy*=year; *mm*=month; *DD*=day; *HH*=hour; *MM*=minute; *SS*=seconds; *ss*=tenths of a second

customerId: Customer Network Management Identifier assigned to network components (PEs and POs) dedicated to a Virtual Private Network

dspId: Module DSP group ID

namsId: Integers which identify AM, RM, or PM

seqNum: Sequence number assigned by the network administration and management system (NAMS) device

All other fields within BDF log files are controlled by the `dpn_log.rdf` file.

Log record format

“Log records” (page 318) describes the log format used for operator command logs.

Table 76
Log records

Field	Type
significantData	string
otherData	string

significantData This field holds a string that indicates why this data is being logged. An example would be the transaction termination code of an operator command.

otherData This field holds the data that is to be logged. An example would be an operator command.

Note: Both *significantData* and *otherData* use the BDF escape sequence.

Chapter 6

Outage records

This chapter describes MDP Outage Calculator outage records.

DPN outage records

Outage records can be generated for DPN-100 nodes. The records use a Bulk Data Format (BDF) and are generated from alarm records processed by an MDP outage calculator. The content and sequence of all fields within BDF outage files are controlled by the `dpn_otg.rdf` RDF file.

For DPN BDF outage record details, see “DPN BDF outage record fields” (page 319).

Table 77
DPN BDF outage record fields

Field name	Description or value
switchType	dpn
dataType	otg
customerId	Customer Network Management Identifier assigned to network components (PEs and POs) dedicated to a Virtual Private Network.
dspld	Module DSP group ID.
namsId	Integers which identify AM, RM, or PM.
seqNum	Sequence number assigned by the network administration and management system (NAMS) device.
(Sheet 1 of 2)	

Table 77 (Continued)
DPN BDF outage record fields

Field name	Description or value
componentId	A component identifier describes a NAMS object. A component identifier is composed of a list of qualifiers; each with two elements—category (PE, PI, PO, CH) and a uniquely numbered name.
outageType	Can have a value of: Component, Possible, Combined, or Indeterminate.
componentType	Can have a value of: Processor, LinkTrunk, Module, or Port
startDateTime	The date and time of the first alarm record processed. The format is: yyyyymmDDTHHMMSS.t, where yyyy=year; mm=month; DD=day; HH=hour; MM=minute; SS=seconds; and t=tenths of a second.
endDateTime	The date and time of the last alarm record processed. The format is: yyyyymmddTHHMMSS.t, where yyyy=year; mm=month; DD=day; HH=hour; MM=minute; SS=seconds; and t=tenths of a second.
duration	Outage duration in seconds.
sourceID	Use as index into 241-1001-506 <i>DPN-100 Alarm Console Indications</i> (DPN)
faultCode	Use as index into 241-1001-506 <i>DPN-100 Alarm Console Indications</i> (DPN)
(Sheet 2 of 2)	

Note: Comment and condition fields use the BDF escape sequence.

Chapter 7

Availability report records

Availability report records use a Bulk Data Format (BDF) and are generated from outage file records created by an MDP Outage Calculator. The content and sequence of all fields within BDF availability records is controlled by the gen_ava.rdf RDF file. See “BDF availability record fields” (page 321).

Table 78
BDF availability record fields

Field name	Description or value
switchType	dpn for DPN-100
dataType	ava
dateTime	Start date and time of the availability sample period. The format is: YYYYMMDDThmmss.s, where YYYY=year; MM=month; DD=day; hh=hour; mm=minute; ss=seconds; s=tenths of a second.
switchID	For DPN-100, switchID identifies the AM, RM, or PM.
componentId	The component identifier formatted in pairs of: category/value.
componentType	For DPN-100: Module, Processor, LinkTrunk, Port.
sampleTime	Duration of the sample period in seconds
outageTime	Sum of individual outage durations in seconds
serviceTime	Total time, in seconds, of sample period less any service windows defined in the exceptions file
nofOutages	Number of individual outages
mtbf	Mean time between failure in seconds
(Sheet 1 of 2)	

Table 78 (Continued)
BDF availability record fields

Field name	Description or value
mtrr	Mean time to repair in seconds
availPcnt	Percentage availability
accsampleTime	Duration of the accumulated sample period in seconds
accOutageTime	Accumulated outage time in seconds
accServiceTime	Accumulated service time in seconds
accNofOutages	Accumulated number of outages
accMtbf	Accumulated mean time between failures in seconds
accMtrr	Accumulated mean time to repair in seconds
accAvailPcnt	Accumulated availability percentage
(Sheet 2 of 2)	

Note: Comment and condition fields use the BDF escape sequence.

Chapter 8

File processing audit records

Header records that identify each field in the file processing audit records are optionally provided at the top of each detail or summary report page. Header records are indicated by the letter H in field 1.

For a description of the file processing audit detail records, see “File processing audit detail records” (page 323).

For a description of the file processing audit summary records, see “File processing audit summary records” (page 325).

Table 79
File processing audit detail records

Field name	Field type	Default field position	Field value	Field description
recordType	str	001	always “D”	This field identifies this audit record as a detail record.
fileName	str	002		The filename of the converted raw data file.
dataType	str	003	one of: acc, ala, log, scn, or sta	This field identifies the datatype of the converted raw data file.
nodeID	str	004		The nodename or nodeID of the originating switch.
(Sheet 1 of 3)				

Table 79 (Continued)
File processing audit detail records

Field name	Field type	Default field position	Field value	Field description
sizeBytes	str	005	7 digits, 0 filled	The size of the spooled data file in bytes.
recordsCount	str	006	7 digits, 0 filled	The number of successfully converted records.
bytesCorrupt	str	007	7 digits, 0 filled	The number of corrupted bytes in the spooled file.
conversionResult	str	008	one of: CONVERTED PARTIAL ERROR OTHER_ERR ZERO_LEN NONE UNKNOWN	This field indicates the status of the file conversion. CONVERTED indicates successful BDF conversion. PARTIAL indicates some records were converted and a BDF file was generated. ERROR indicates that no records were converted. OTHER_ERR indicates that other processing errors occurred while processing this file. ZERO_LEN indicates that this file was zero length and removed from the spool directory. NONE indicates that no conversion occurred because the file was missing, was waiting in the spool directory, or is not to be converted (for example, an SRS file). UNKNOWN indicates the file status is unknown.
(Sheet 2 of 3)				

Table 79 (Continued)
File processing audit detail records

Field name	Field type	Default field position	Field value	Field description
deliverResult	str	009	one of: MISSING <seq> INSPOOL DELIVERED NONE	This optional field indicates the status of the file delivery. MISSING <seq> indicates that a spool file with sequence number <seq> was not received. INSPOOL indicates that a data file is received and waiting in the spool directory. DELIVERED indicates that the file was delivered to a customer host. NONE indicates that the file was not delivered to a customer host.
receivingHost/server	str	010	maximum 3 values, comma delimited	This optional field identifies the customer host(s) that received the converted BDF file. This field can contain hostnames or IP addresses.
(Sheet 3 of 3)				

Table 80
File processing audit summary records

Field name	Field type	Default field position	Field value	Field description
recordType	str	001	always "S"	This field identifies this audit record as a summary record.
date	str	002	YYYYMMDD	The date of the record.
dataType	str	003	one of: acc, ala, log, scn, or sta	This field identifies the datatype of the converted raw data file.
(Sheet 1 of 2)				

Table 80 (Continued)
File processing audit summary records

Field name	Field type	Default field position	Field value	Field description
nodeID	str	004		The nodename or nodeID of the originating switch.
detailCount	str	005	5 digits, 0 filled	The number of records per node.
totalBytes/node	str	006	10 digits, 0 filled	The number of bytes processed per node.
totalRecs/node	str	007	10 digits, 0 filled	The number of successfully converted records per node.
corruptedBytes/node	str	008	10 digits, 0 filled	The number of corrupted bytes per node.
zeroCount	str	009	5 digits, 0 filled	The number of zero length files per node.
errorCount	str	010	5 digits, 0 filled	The number error files per node.
missgCount	str	011	5 digits, 0 filled	The number of missing spool files per node.
delivCount	str	012	5 digits, 0 filled	The number of BDF files successfully delivered to customer hosts.
(Sheet 2 of 2)				

Index

3270HostPAD 128

3270TermPAD 131

A

accounting BDF records

common fields 83

Extended national address option 113

Frame Relay option 114

GAS option 108

Gateway option 112

International CUG option 110

Inter-network option 106

National CUG option 109

NUI option 106

optional fields 105

Routing class of service option 113

Sensor identifier option 112

Videotex option 112

X.75 interface identifier option 112

accounting Published Format records

Extended national address option 65

Frame Relay option 66

GAS option 55

Gateway option 61

International CUG option 57

Inter-network option 50

National CUG option 55

NUI option 47

optional fields 45

Original called address option 54

Routing class of service option 64

Sensor identifier option 60

Videotex option 62

X.75 interface identifier option 61

accounting records

DPN Published Format 20

Passport Published Format 20

ACPPAD 140

API_Link 135

APIDevice 134

audit records 323

Availability records 321, 323

availability report records 321

B

BDF

short file names 83

BOLDTermPAD 138

Bulk Data Format (BDF)

common fields 83

customized accounting information
option 123

DPN alarm records 313, 319, 321, 323

Extended national address option 113

frame relay option 114

Free-format NUI option 106

GAS option 108

Gateway option 112

International CUG option 110

Inter-network option 106

National CUG option 109
NUI option 106
optional fields 105
Routing class of service option 113
Sensor identifier option 112
Videotex option 112
X.75 interface identifier option 112

C

CallRedirSvr 141
CallRedirSvrRSI 142
CallRouterDst 142
CallRouterGateDst 143
CallRouterGateSrc 143
CallRouterSrc 144
ClusterRemoteDU 145
common fields
 BDF 83
 Published format 23

D

DialOutRouting 148
DPN 319
 Peak water mark fields 100
DPN accounting records 81
DPN alarm record common fields 313
DPN alarm records 313, 319
DPN availability 321, 323
DPN log record common fields 317
DPN log records 317
DPN statistics records 125

E

Extended national address option 65, 113

F

formats
 DPN alarm output format 314
 DPN log record format 318
Frame Relay option 66, 114
FrameRelay 149

FrameRelayDLCIEnhanced 155
FrameRelayDLCIG33 158
FrameRelayDLCIG34 162

G

GAS option 55, 108
GASTrtxtUnivGate 168
Gateway option 61, 112
GVC 257

H

HighPerfPE 211

I

International CUG option 57, 110
Inter-network option 50
ISDNDLCI 170
ISRBPhase2 172
ITI 182
ITIEnhanced 184

L

LAPDPhysical 187, 193
log record format 318

N

National CUG option 55, 109
NLLAPBEnhanced 197
NLOverFR 247
NUI
 DPN-100 and Calling Card 106
 free-format option 106
NUI BDF option 106
NUI option 47
NUIDatabaseInt 207
NUIValidRemSvrRSI 210

O

OfficeMaster 211
operator command logs 317
optional fields

BDF 105
Published format 45
Original called address option 54
outage 319
outage records 319

P

Passport
Peak water mark fields 100
Passport accounting records 19
Passport availability 321, 323
PE286386 214
Peak water mark fields
DPN and Passport 100
Published Format
accounting record fields 22
common fields 23
DPN accounting records 20
Extended national address option 65
frame relay option 66
GAS option 55
Gateway option 61
International CUG option 57
Inter-network option 50
National CUG option 55
NUI option 47
optional fields 45
Original called address option 54
Passport accounting records 20
Routing class of service option 64
Sensor identifier option 60
Videotex option 62
X.75 interface identifier option 61

R

RDFs
file naming 13
records
audit 323
availability report 321
DPN accounting 81

DPN alarms 313, 319
DPN logs 317
DPN statistics 125
DPN statistics common fields 125
Passport accounting 19
Routing class of service option 64, 113

S

SABRE 216
Sensor identifier option 60, 112
SNALink 219
SNALogicalUnit 221
statistics
3270HostPAD 128
3270TermPAD 131
ACPPAD 140
APIDevice 134
APILink 135
BOLDTermPAD 138
CallRedirSvr 141
CallRedirSvrRSI 142
CallRouterDst 142
CallRouterGateDst 143
CallRouterGateSrc 143
CallRouterSrc 144
ClusterRemoteDU 145
DialOutRouting 148
FrameRelay 149
FrameRelayDLCIEnhanced 155
FrameRelayDLCIG33 158
FrameRelayDLCIG34 162
GASTrtxtUnivGate 168
GVC 257
HighPerfPE 211
ISDNDLCI 170
ISRBPhase2 172
ITI 182
ITIEnhanced 184
LAPDPhysical 187, 193
NLLAPBEnhanced 197
NLOverFR 247

- NUIDatabaseInt 207
- NUIValidRemSvrRSI 210
- OfficeMaster 211
- overview 125, 313, 317
- PE286386 214
- quick reference 126
- records 125, 313, 317, 319, 321
- SABRE 216
- SNALink 219
- SNALogicalUnit 221
- subtype 11 135
- subtype 12 182
- subtype 18 211
- subtype 19 134
- subtype 20 131
- subtype 22 128
- subtype 23 187
- subtype 24 170
- subtype 25 229
- subtype 26 219
- subtype 27 221
- subtype 28 144
- subtype 29 142
- subtype 30 214
- subtype 32 141
- subtype 33 142
- subtype 34 210
- subtype 35 238
- subtype 36 148
- subtype 39 143
- subtype 40 143
- subtype 43 207
- subtype 44 168
- subtype 45 138
- subtype 46 216
- subtype 51 145
- subtype 52 226
- subtype 53 223
- subtype 57 184
- subtype 58 197
- subtype 59 149
- subtype 61 274
- subtype 62 282
- subtype 63 262
- subtype 64 233
- subtype 65 242
- subtype 66 155
- subtype 67 253
- subtype 68 172
- subtype 69 158
- subtype 70 300
- subtype 71 293
- subtype 72 247
- subtype 73 162
- subtype 74 257
- subtype 75 211
- subtype 76 140
- subtype 77 193
- TokenRing 223
- TokenRingDevice 226
- TrunkRMUTP 229
- TrunkRMUTPEnhanced 233
- UTP 238
- UTPEnhanced 242
- VC 253
- X25MLAgent 274
- X25MLSerPort 282
- X25Port 262
- X75MLAgentG32 293
- X75PortG32 300

T

- TokenRing 223
- TokenRingDevice 226
- TrunkRMUTP 229
- TrunkRMUTPEnhanced 233

U

- UTP 238
- UTPEnhanced 242

V

VC 253

Videotex option 62, 112

X

X.75 interface identifier option 61, 112

X25MLAgent 274

X25MLSerPort 282

X25Port 262

X75MLAgentG32 293

X75PortG32 300

Preside Multiservice Data Manager MDP Data Formats for DPN

Reference

Release: R15.1

Copyright © 2004 Nortel Networks.
All Rights Reserved.

NORTEL, NORTEL NETWORKS, the globemark design, the NORTEL NETWORKS corporate logo, PRESIDE, DPN, and PASSPORT are trademarks of Nortel Networks. UNIX is a trademark licensed exclusively through X/Open Company Ltd. SUN is a trademark of Sun Microsystems Inc. IBM is a trademark of International Business Machines Corporation. VAX is a trademark of Digital Equipment Corporation.

Publication: 241-6001-806
Document status: Standard
Document version: 15.1RSUP
Document date: August 2004
Printed in Canada

