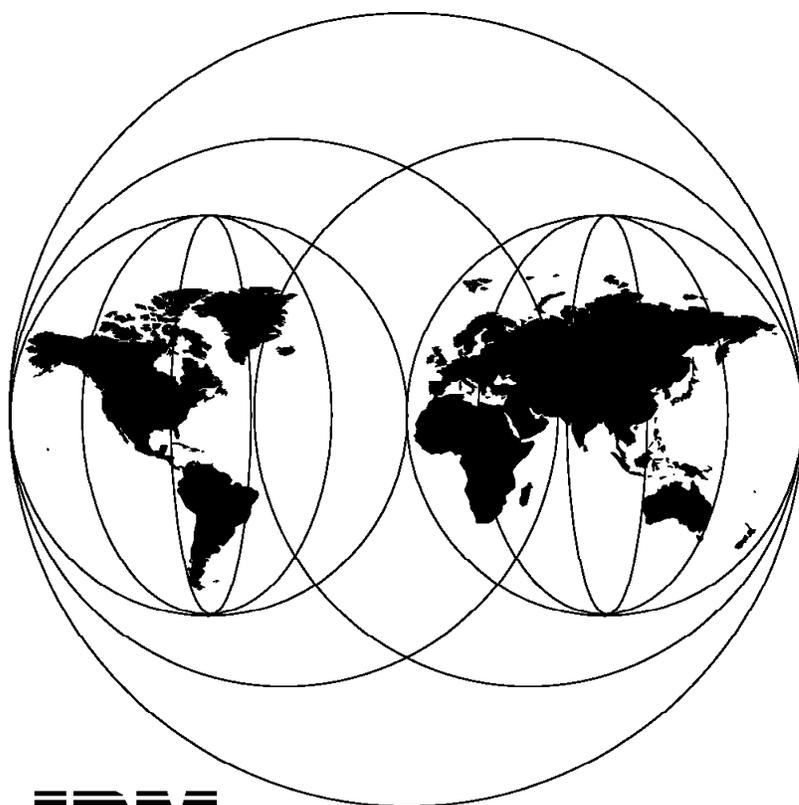


RS/6000 AIXLink/X.25 Cookbook

November 1996



**International Technical Support Organization
Austin Center**



International Technical Support Organization

SG24-4475-01

RS/6000 AIXLink/X.25 Cookbook

November 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix I, "Special Notices" on page 297.

Second Edition (November 1996)

This edition applies to Version 1.1.3 of AIXLink/X.25, 5696-926 for use with Version 4 of the AIX/6000 Operating System.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JN9B Building 045 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1996. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	xi
Tables	xiii
Preface	xv
How This Redbook Is Organized	xv
The Team That Wrote This Redbook	xvi
Comments Welcome	xvii
Chapter 1. An Introduction to the X.25 Protocol	1
1.1 Network Communications with X.25	1
1.1.1 Description	1
1.1.2 Standardization	2
1.2 Advantages and Disadvantages of X.25	3
1.2.1 Advantages	3
1.2.2 Disadvantages	4
1.3 Components of a PSDN: DSEs, DTEs and DCEs	5
1.4 The Three Levels of X.25	6
1.4.1 Physical Level	7
1.4.2 Link Level	8
1.4.3 Packet Level	9
1.5 Network User Address (NUA)	12
1.6 Logical Channels and Virtual Circuits	13
1.6.1 Definition	13
1.6.2 Logical Channel	14
1.6.3 Virtual Circuit	14
1.6.4 Multiplexing	14
1.6.5 Types of Virtual Circuits	15
1.7 Facilities	15
1.7.1 Definition	15
1.7.2 Use of Facilities	16
1.7.3 Types of Facilities	17
1.8 X.25 Network Subscription	19
1.9 Summary of IBM Support Experiences	20
1.9.1 Planning Considerations	20
1.9.2 Application Selection	21
Chapter 2. RISC System/6000 X.25 Support	23
2.1 RISC System/6000 Functions Summary	23
2.1.1 Number of Virtual Circuits Supported	25
2.2 RISC System/6000 X.25 Components	25
2.2.1 IBM X.25 Interface Co-Processor and Co-Processor/2	25
2.2.2 IBM ARTIC Portmaster Adapter/A	26
2.2.3 IBM ISA Multiport Model II Adapter	28
2.2.4 IBM ARTIC960 Adapter	30
2.2.5 X.25 Device Driver and X.25 Ports	33
2.2.6 X.25 Commands	34
2.2.7 PAD Support	35
2.2.8 NPI Programming Interface	35
2.2.9 DLPI Programming Interface	35
2.2.10 COMIO Emulation	35

2.2.11	SNMP	35
2.2.12	SMP machines and X.25	36
2.2.13	X.32	36
2.3	X.25 Power Management	37
2.3.1	General Power Management and X.25	37
2.3.2	Impact to Network Provider	37
2.3.3	Impact to Local Applications	38
2.3.4	Power Management Limitation Warnings	38
2.4	Hardware Requirements	39
2.5	Differences Between X.25 Support on AIX Versions	40
2.5.1	Adapter Speeds	41
 Chapter 3. X.25 LPP Installation and Setup		 43
3.1	Overview and Fast Path	43
3.1.1	Scenario 1: Two RISC System/6000s Connected to a PSDN	43
3.1.2	Scenario 2: Two RISC System/6000s Connected by a Modem Eliminator	44
3.2	Hardware Installation	45
3.2.1	Installing the Adapter	45
3.2.2	Physical Connections to the Network	47
3.2.3	X.25 Connection Test Scenarios	49
3.2.4	Modem Considerations	52
3.3	Software Customization	52
3.3.1	Installing the X.25 LPP Software	52
3.3.2	Configuring the X.25 Device Driver	55
3.3.3	Configuring an X.25 LPP Port	56
3.3.4	Updating the Number of Virtual Channels	58
3.3.5	Modifying Other Parameters	60
3.3.6	Customization for the Back-to-Back Scenario	62
3.4	Returning to the Default Configuration Parameters	64
3.5	Configuring V.25bis Dialing	64
3.6	Configuring ISA X.25 Adapters	65
3.6.1	Configuring ISA X.25 Adapters at the Hardware Level	65
3.6.2	Configuring ISA X.25 Adapters at the Software Level	67
3.7	X.25 Facilities	68
3.7.1	Facilities Requested by the DTE	68
3.7.2	More on Configuring Facilities	71
 Chapter 4. Tools		 73
4.1	The xtalk Tool	73
4.1.1	Description	73
4.1.2	Overview and Fast Path	73
4.1.3	The xtalk Command	74
4.1.4	Testing Communication with xtalk	75
4.1.5	Requesting the Use of a Facility with xtalk	78
4.2	The x25mon Tool	79
4.2.1	Description	79
4.2.2	Using the x25mon Tool	79
4.3	The xroute Tool	82
4.3.1	The Routing Table	82
4.3.2	Updating the X.25 Routing Table with the xroute Tool	83
4.4	Other X.25 Commands	86
4.5	Configuration Tools	89
4.5.1	The backupx25 Script	89
4.5.2	The restorex25 Script	89

4.5.3 The removex25 Script	89
Chapter 5. Packet Assembler Disassembler (PAD)	91
5.1 Using a PAD	91
5.2 Function of the PAD	92
5.2.1 X.3, X.28 and X.29 Standards	93
5.2.2 Outgoing PADs	94
5.2.3 Incoming PADs	95
5.3 CCITT PAD Interfaces	95
5.3.1 PAD Parameters	95
5.3.2 PAD Commands	97
5.3.3 PAD Profiles	100
5.4 Configuring the PAD	102
5.4.1 PAD for AIXLink/X.25 1.1.2 Running on AIX V.4	103
5.4.2 PAD for AIXLink/X.25 1.1.3 (and later) on AIX V.4	104
5.4.3 Default Initial Application	105
5.4.4 Selectable Profile	107
5.4.5 Configurable Profile	108
5.4.6 Security Features	108
5.5 PAD printing	109
5.5.1 Setting up the PAD Printer Queue	110
5.5.2 PAD Printing Process	110
5.5.3 Configuration for Remotely Initiated Printing	110
5.5.4 Configuration of Locally Initiated Printing	111
5.6 PAD Usage	112
5.7 Incoming PAD Setup	112
5.7.1 Listing the Configured Ports	112
5.7.2 Stopping the X.29 Daemon	113
5.7.3 Starting the X.29 Daemon	113
5.8 Setup of an Integrated PAD	114
5.8.1 Starting the PAD Program	114
5.8.2 Exiting	115
5.9 X.25 PAD and C-Kermit	115
5.9.1 Requirements	115
5.9.2 Documentation	115
5.9.3 AIX X.25 PAD as a C-Kermit Client	115
5.9.4 AIX X.25 PAD as a C-Kermit Server	116
5.9.5 Session Statistics	116
Chapter 6. APIs: COMIO, NPI and DLPI	117
6.1 Implementation of X.25 in the STREAMS Environment	117
6.2 STREAMS	118
6.2.1 STREAMS Definition	118
6.2.2 STREAMS Components	118
6.2.3 Benefits of STREAMS	120
6.2.4 How to use STREAMS	122
6.3 NPI API	123
6.3.1 NPI Implementation on the RS/6000	123
6.3.2 NPI Primitives Available on the RS/6000	125
6.3.3 Use of the Primitives	125
6.3.4 Description of the NPI Samples Available with the LPP	126
6.4 DLPI API	128
6.4.1 DLPI Implementation on the RS/6000	129
6.4.2 Establishment of a Stream to DLPI	129
6.4.3 DLPI Primitives Available on the RS/6000	130

6.4.4 Description of the DLPI Samples Available with the X.25 LPP	130
6.5 COMIO API	132
6.5.1 COMIO Definition	132
6.5.2 Using COMIO	133
6.5.3 X.25 Example Programs	134
Chapter 7. TCP/IP Setup	137
7.1 Introduction	137
7.2 TCP/IP X.25 Connection Setup and Test	138
7.2.1 Creating Entries in /etc/hosts	139
7.2.2 Initializing and Starting the IP/X.25 Interface	140
7.2.3 Adding Routes for the Remote Systems	141
7.2.4 Mapping IP Addresses to X.25 NUAs	142
7.2.5 Testing the Connection	143
7.2.6 Suppressing the X.25 LPP IP Interface	144
7.3 Requesting the Use of a Facility with TCP/IP	145
7.4 Changing TCP/IP CUD Values	146
7.5 Configuring SNMP	147
Chapter 8. Accessing an SNA Network with X.25	149
8.1 QLLC with Reference to RISC System/6000 X.25 Support	150
8.2 Introduction to SNA Server/6000	150
8.2.1 SNA Application Programming Interfaces (APIs)	152
8.2.2 SNA Configuration Profiles	153
8.2.3 SNA Configuration Database and Commands	154
8.2.4 SNA Server Components	156
8.3 AIX SNA Server/6000 Setup	158
8.3.1 Installation of the AIX SNA Server/6000 LPP	158
8.3.2 Installation of the Data Link Control	159
8.3.3 Getting Host Definitions	160
8.3.4 SNA Profiles for an X.25 LU 2	161
8.3.5 Performing Initial Node Setup	161
8.3.6 Setting Up the SNA profiles for LU 2	163
8.3.7 Verifying SNA Profiles	172
8.3.8 Starting SNA Server/6000 Server	174
8.3.9 Starting SNA Server/6000 Link Station	175
8.3.10 Listing Current Status of SNA Server/6000	175
8.4 3270 Host Connection Program/6000	176
8.4.1 Overview	176
8.4.2 Setup and Use of 3270 Host Connection Program/6000	177
8.4.3 SNA Problems	178
8.4.4 HCON Problems	179
Chapter 9. X.25 Problem Determination	181
9.1 X.25 Problem Diagnosis	181
9.2 Diagnosing Problems with Connecting to the X.25 Network	181
9.3 Diagnosing Problems with Making an Outgoing X.25 Call	182
9.4 Diagnosing Problems with Receiving an Incoming X.25 Call	183
9.5 Diagnosing X.25 Packet Problems	183
9.6 Diagnosing X.25 Command Problems	184
9.7 Diagnosing xtalk Problems	185
9.7.1 Device Driver Problems	185
9.7.2 X.25 Protocol Problems	186
9.8 Diagnosing TCP/IP Problems	191
9.8.1 Preliminary Checks	191

9.8.2 Testing with ping	192
9.8.3 Function Keys not Working Properly	194
9.9 Diagnosing SNA Server Problems on AIX V4	194
9.10 Basic Information Required	196
9.10.1 Problem Definition	196
9.10.2 Profiles	197
9.10.3 SNA Server/6000 Traces	198
9.10.4 LU 0 Information	201
9.10.5 System Error Log	202
9.10.6 HCON Trace	202
9.10.7 SNA_ABEND	203
9.10.8 Trace Hooks for X.25	203
9.11 System Errors	203
9.11.1 Flashing 888	203
9.11.2 System Dump	205
9.12 Collecting Information for Resolution of X.25 Problems	206
9.12.1 List of Elements to Gather	206
9.13 How to Resolve an X.25 'Device Busy' Condition	207
9.13.1 Problem Identification	207
9.13.2 Problem Solution	207
9.13.3 If All Else Fails	208
9.14 Improving FTP Performance over X.25	208
9.15 ARTIC960 Software Problems	208
9.16 Examples of X.25 PAD Problem Determination	209
9.16.1 Providing Initial Statement of Fault	209
9.16.2 X.25 Hardware Connection	210
9.16.3 Device Driver	211
9.16.4 Connection State	211
9.16.5 Total Line Termination or Failure to Start	212
9.16.6 Connected to X.25 Network No Traffic Flow	214
9.16.7 Transient Dropouts	214
9.16.8 Application	216
9.16.9 Printing	216
Chapter 10. Performance and Tuning	219
10.1 Line Speed	219
10.2 Transit Delay Versus Data Throughput	219
10.3 Factors That May Limit Throughput	220
10.3.1 X.25 Adapter	220
10.3.2 DTE-to-DCE Line Speed and Throughput Class	220
10.3.3 Influence of the Application Design	221
10.4 Network Configuration Parameters	221
10.5 Applications	221
10.5.1 X.25 LPP PAD Versus TCP/IP	221
10.5.2 TCP/IP	222
10.5.3 SNA	224
10.6 Memory Buffers	224
Appendix A. Summary of the SMIT Main Menus Used for Configuration	227
A.1 X.25 Adapter Configuration	227
A.2 TCP/IP Configuration	231
A.3 SNA Configuration	234
Appendix B. Differences Between X.25 LPP and AIX V3 Base X.25 Support	239
B.1 Hardware Differences	239

B.2	Functionality Differences	239
B.2.1	Differences between (CCITT) 1988 and 1984 X.25	241
B.3	Configuration and Setup Differences	243
B.4	Command Differences	244
B.5	ODM/SMITAttributes Differences	244
B.6	Default Values of Important Parameters	246
Appendix C. X.25 Cables and Connectors		247
C.1	IBM X.25 Co-Processor and IBM X.25 Co-Processor/2	247
C.1.1	X.25 Co-Processor 37-Pin Connector Pin Assignment	248
C.1.2	X.25 Co-Processor Modem Attachment Pin Assignment	249
C.1.3	X.25 Co-Processor Interconnection Cables	253
C.1.4	X.25 Co-Processor Adapter and Cable Diagnostics	255
C.2	IBM Portmaster Adapter/2 and Multiport Model 2	257
C.2.1	IBM ARTIC Portmaster Adapter/A Pin-Out Information	257
C.2.2	Modem Attachment Pin Assignment	260
C.2.3	X.25 Portmaster and Multiport Interconnection Cables	261
C.2.4	X.25 Portmaster, Multiport and ARTIC960 Diagnostic Wrap Plugs	262
Appendix D. CCITT Causes and Diagnostics		269
D.1	CCITT Clear and Reset Causes	269
D.2	CCITT Diagnostic Codes	269
D.3	Product-Specific Diagnostic Codes	271
D.4	ISO 8208 Diagnostic Codes	271
D.5	SNA Diagnostic Codes	271
D.5.1	List of Diagnostic Codes Used by xtalk	274
D.5.2	Logical Channel States	274
Appendix E. Facilities		275
E.1	Supported Facilities for X.25 Communications	275
E.1.1	Defining Facilities	275
E.1.2	Facilities Format	275
E.1.3	Index of the Facilities	276
E.2	X.25 Facilities	277
E.2.1	Packet Size Selection	277
E.2.2	Window Size Selection	277
E.2.3	Throughput Class	278
E.2.4	Closed User Group Selection	278
E.2.5	CUG with Outgoing Access	279
E.2.6	Bilateral Closed User Group Selection	279
E.2.7	Reverse Charging and Fast Select	279
E.2.8	Network User Identification	280
E.2.9	Charging Information Request	280
E.2.10	Charging (Monetary Unit)	281
E.2.11	Charging (Segment Count)	281
E.2.12	Charging (Call Duration)	281
E.2.13	RPOA Selection	282
E.2.14	Called Line Address Modified Notification	282
E.2.15	Call Redirection Notification	283
E.2.16	Transit Delay Selection and Indication	284
E.3	CCITT Specified Facilities to Support the OSI Network	284
E.3.1	Calling Address Extension	284
E.3.2	Called Address Extension	285
E.3.3	Minimum Throughput Class	285
E.3.4	End-to-End Transmit Delay Facility	286

E.3.5 Expedited Data Negotiation	286
Appendix F. PAD Parameters and Commands	287
F.1 PAD Parameters	287
F.2 Detailed Description of PAD Parameters	289
F.3 Commands Entered from the PAD Prompt	290
F.4 Messages from the PAD to the Terminal	292
Appendix G. CIO and X.25 Device Driver Error Codes	293
Appendix H. Country Networks Default Parameters	295
Appendix I. Special Notices	297
Appendix J. Related Publications	299
J.1 International Technical Support Organization Publications	299
J.2 Redbooks on CD-ROMs	299
J.3 Other Publications	299
J.3.1 Articles	300
J.4 Non-IBM References	300
How To Get ITSO Redbooks	303
How IBM Employees Can Get ITSO Redbooks	303
How Customers Can Get ITSO Redbooks	304
IBM Redbook Order Form	305
Abbreviations	307
Index	309

Figures

1.	Advantages and Disadvantages of X.25 Compared to Dial-up Transmission	3
2.	An X.25 Network	6
3.	OSI, X.25 Layers	7
4.	HDLC Frame Structures	9
5.	Establishing a Switched Virtual Circuit	11
6.	Clearing a Switched Virtual Circuit	11
7.	DTE Initiated Reset	11
8.	DCE Initiated Reset	11
9.	Packet Format	12
10.	Network User Address (NUA) Structure	13
11.	Logical Channels and Virtual Circuits	14
12.	Facilities and Call Packets	16
13.	Call Packet Formats	17
14.	IBM X.25 Interface Co-Processors	26
15.	IBM ARTIC Portmaster Adapter/A Configurations	28
16.	IBM ARTIC Multiport Adapter Model II Configurations	30
17.	IBM ARTIC960 Adapter Configurations	32
18.	Adapter, Device Driver, Port, Interface	34
19.	RISC System/6000s Connected to the TYMNET PSDN	43
20.	RISC System/6000s Connected Back-to-Back	44
21.	Two RISC System/6000s Connected to an X.25 Network	49
22.	One RISC System/6000 Connected to an X.25 Network	50
23.	Two RISC System/6000s Connected to an X.25 Network Simulator	50
24.	Two RISC System/6000s Connected Back-to-Back	51
25.	One RISC System/6000 with Two X.25 Co-Processors	51
26.	Two RISC System/6000s Connected Back-to-Back	62
27.	xtalk Test	73
28.	The Various Types of PADs	92
29.	X.3, X.28 and X.29 Standards	93
30.	Public PAD vs. Integrated Private PAD	94
31.	Private PADs Used to Carry Asynchronous Connections over a PSDN	95
32.	X.25 In the STREAMS Environment	117
33.	Components of a Stream	118
34.	The Portability of the STREAMS Architecture	120
35.	Easy Migration with the STREAMS Architecture	121
36.	Reusability of STREAMS Modules	121
37.	Multiplexing Facility of the STREAMS	122
38.	NPI Components	124
39.	The Building of a NPI Stream	124
40.	Structure of the npiserver.c Program	127
41.	Structure of the npiclient.c Program	128
42.	DLPI Components	129
43.	Structure of the dlpiserver.c Program	131
44.	Structure of the dlpiclient.c Program	132
45.	COMIO Emulation	133
46.	Test Configuration for TCP/IP	139
47.	Scenario for an SNA Connection Using X.25	149
48.	Generic AIX SNA Server/6000 Profiles for HCON	153
49.	AIX SNA Server/6000 Configuration Process	155
50.	X.25 AIX SNA Server/6000 Profiles for HCON	161

51.	Establishing a Switched Virtual Circuit	187
52.	Call Packet Structure	188
53.	Call Cleared by the DCE	190
54.	Call Cleared by the Remote DTE	190
55.	Call Cleared by the Calling DTE	190
56.	AIX SNA Services/6000 Main Profiles	234
57.	Co-Processor Adapters	247
58.	X.21 Interface Cable	253
59.	X.21bis/V.24 Interface Cable	254
60.	X.21bis/V.35 Interface Cable	255
61.	D-37 Wrap Plug	255
62.	D-15 Wrap Plug	256
63.	D-25 Wrap Plug	256
64.	M/34 Wrap Plug	256
65.	Portmaster and Multiport	257
66.	X.21 Interface Cable	261
67.	X.21bis/V.24 Interface Cable	261
68.	X.21bis/V.35 Interface Cable	262
69.	D-15 Wrap Plug	262
70.	D-25 Wrap Plug	263
71.	M/34 Wrap Plug	263
72.	D-25 Wrap Plug	263
73.	D-25 Wrap Plug	264
74.	D-37 Wrap Plug	264
75.	D-78 X.21 Wrap Plug	265
76.	D-100 Wrap Plug	266
77.	D-100 V.24 Wrap Plug	267
78.	D-100 V.36/V.35 Wrap Plug	268
79.	Facilities Format	275
80.	Facility Code Format	276
81.	Packet Size Selection	277
82.	Window Size Selection	277
83.	Throughput Class	278
84.	Closed User Group Selection (Basic Format)	278
85.	Closed User Group Selection (Extended Format)	278
86.	CUG with Outgoing Access (Basic Format)	279
87.	CUG with Outgoing Access (Extended Format)	279
88.	Bilateral Closed User Group Selection	279
89.	Reverse Charging and Fast Select	279
90.	Network User Identification	280
91.	Charging Information Request	280
92.	Charging (Monetary Unit)	281
93.	Charging (Segment Count)	281
94.	Charging (Call Duration)	281
95.	RPOA Selection (Basic Format)	282
96.	RPOA Selection (Extended Format)	282
97.	Called Line Address Modified Notification	282
98.	Called Line Address Modified Notification (CLAMN)	283
99.	Call Redirection Notification	283
100.	Transit Delay Selection and Indication	284
101.	Calling Address Extension	284
102.	Called Address Extension	285
103.	Minimum Throughput Class	285
104.	End-to-End Transmit Delay Facility	286
105.	Expedited Data Negotiation	286

Tables

1.	X.25 Optional Facilities	17
2.	Throughput of the X.25 Adapters	40
3.	Differences Between AIX Version 3.2.5, AIXLink Ver. 1.1 and AIXLink Ver. 1.1.3	40
4.	Standard Value of Important Parameters	60
5.	NUAs Used for Back-to-Back Scenario	62
6.	Interrupt Level Switch Positions	65
7.	X.25 Adapter: Switch Block 1	66
8.	X.28 Facility Codes	99
9.	Clearing Codes	100
10.	Parameter Settings for Predefined PAD Profiles	101
11.	Parameter Settings for Default PAD Profiles	102
12.	SNA Host Definitions	160
13.	SNA Problem Checklist	195
14.	Information to Gather for Specific Problem Categories	197
15.	Trace Reports and Associated Hook IDs	201
16.	X.25 Problem Checklist	210
17.	SMIT Menus for the Base X.25 Configuration	227
18.	SMIT Menus for the TCP/IP Configuration	231
19.	SMIT Menus for the SNA Configuration	235
20.	X.25 Supported Adapters	239
21.	Functional Differences	240
22.	SMIT Fastpaths Differences	243
23.	Differences Between xmonitor and x25mon	244
24.	SMIT Attribute Names	244
25.	Default Values of Important Parameters	246
26.	X.25 Co-Processor 37-Pin Connector Pin Assignment	248
27.	Pin Assignment of V.11 Type of Interface Circuits to 15-Pin Connectors	249
28.	V.24/X.21bis Pin Assignment to 25-Pin Connector (Speeds up to 20 Kbps)	251
29.	V.35/X.21bis Pin Assignment to 34-Pin Connector (Speeds above 20 kbps)	252
30.	V.36 Pin Assignment to 37-Pin Connector	253
31.	Portmaster and Multiport: V.35	257
32.	Portmaster and Multiport: V.24	259
33.	Portmaster and Multiport: X.21	260
34.	List of CCITT Clear Causes	269
35.	List of CCITT Reset Causes	269
36.	List of CCITT Diagnostic Codes	269
37.	List of Product-Specific Diagnostic Codes	271
38.	List of ISO 8208 Diagnostic Codes	271
39.	List of SNA Diagnostic Codes	271
40.	List of xtalk Diagnostic Codes	274
41.	Index of Facilities	276
42.	User-Customizable PAD Parameters Defined in X.3	287
43.	Detailed Descriptions of PAD Parameters Defined in X.3	289
44.	Commands that can be Issued from the PAD Prompt	290
45.	Service Signals Sent from the X.3 PAD to the X.28 Terminal	292
46.	Default Parameters for the Various Networks	295

Preface

This document is intended for users of AIXLink/X.25 V1.1.3 Licensed Program Product.

The *AIX/V3 X.25 Communication Cookbook*, GG24-3692-01, is still available and is intended for users of the AIX X.25 product shipped with the AIX 3.2 base operating system.

This document describes the set up and testing of X.25 communications on the RISC System/6000. It explains, step-by-step, how to install hardware and software, customize the X.25 support, test the connection to the network, and establish a virtual call between two systems. The use of the APIs available and the setup of the included PAD of TCP/IP and SNA 3270 emulator in an X.25 environment are also described. Descriptions of the X.25 concepts, components, and protocols are also included.

This document is intended for customers and system engineers who plan to install X.25 Licensed Program Product (LPP) communications on the RISC System/6000. A knowledge of AIX/6000 is assumed.

How This Redbook Is Organized

This redbook contains 313 pages. It is organized as follows:

- Chapter 1, "An Introduction to the X.25 Protocol"
This chapter is an overview of the X.25 protocol and terminology.
- Chapter 2, "RISC System/6000 X.25 Support"
This chapter briefly describes all the features included in the X.25 LPP on the RISC System/6000.
- Chapter 3, "X.25 LPP Installation and Setup"
This chapter describes the hardware and software installation of the X.25 LPP on the RISC System/6000, the modem setup and the minimal customization of the X.25 LPP for switched virtual circuits.
- Chapter 4, "Tools"
This chapter describes the use of the three main tools available in the X.25 LPP and describes briefly each command specific to the X.25 LPP.
- Chapter 5, "Packet Assembler Disassembler (PAD)"
This chapter reviews the PAD protocols (CCITT X.3, X.28, X.29), also called Triple X, and describes the customization and use of the PAD included in the X.25 LPP.
- Chapter 6, "APIs: COMIO, NPI and DLPI"
This chapter presents the APIs available in the X.25 LPP. Before the presentation of the APIs, a review of the STREAMS mechanism is included, since these APIs are based on the Portable STREAMS Environment
- Chapter 7, "TCP/IP Setup"

This chapter shows how to set up and test TCP/IP communications over an X.25 network. It also explains the setup of SNMP to use the X.25 MIB available with the X.25 LPP.

- Chapter 8, “Accessing an SNA Network with X.25”

This chapter describes how to set up and test SNA communications over an X.25 network. The end of the chapter describes the use of 3270 Host Connection Program/6000, which allows the RISC System/6000 to communicate with System/390 computer systems.

- Chapter 9, “X.25 Problem Determination”

This chapter provides information that is intended to be helpful in problem determination.

- Chapter 10, “Performance and Tuning”

This chapter discusses performance considerations on the RISC System/6000 X.25 adapters and on the main applications (PAD, TCP/IP, SNA) over an X.25 link.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

Al Mitchell is a Senior SE Support Representative at the International Technical Support Organization, Austin Center. He writes extensively and teaches IBM classes worldwide on all areas of AIX communications. Before joining the ITSO two years ago, he worked in USM&S as a Senior AIX Marketing Support Specialist.

Sue Foulds is an AIX Systems Support Specialist in the UK. She has five years of experience in the AIX Networking field. Her areas of expertise include X.25 and SNA. She has written extensively on these subjects.

Praben Primasaputra is an AIX Country Support Systems Engineer in Jakarta, Indonesia. He has been with IBM for six years, and has five years of experience in AIX, with the last two years specializing in Networking. He holds a degree in Telecommunication Engineering from Sepuluh November Institute of Technology, Surabaya, Indonesia. His areas of expertise include, HACMP/6000, SNA Server, X.25, AIX Performance Tuning, and TME 10 (Tivoli). He has written extensively on AIX System Management for IBM Indonesia operation.

Eduard Spiess is a AIX Support Professional in Germany. He has 10 years of experience in the AIX communication field and has worked at IBM for 27 years. His areas of expertise include SNA, X.25, OSI/6000 and AS/400 communications. He has written extensively on AIXLink/X.25.

Paul Gunther is a Software Service Specialist in Australia. He has 11 years of experience in the UNIX/AIX field and has worked at IBM for eight years. He holds a degree in Computer Science from the Queensland University of Technology. His areas of expertise include kernel, graphics, and internetworking. He has written extensively on problem-determination techniques and Internet exploitation via the World Wide Web.

Thanks to the following people for their invaluable contributions to this project:

Mark Grosch
AIX WAN Software Development

Marissa Borrego
AIX WAN Software Development

Jackie Wilson
AIX WAN Software Development

Will Faveash
AIX WAN Software Development

Cindy Young
AIX System Center

Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Chapter 1. An Introduction to the X.25 Protocol

This chapter is an introduction to the X.25 network protocol. It is intended to provide an understanding of X.25 for users of the AIX X.25 products. Along with descriptions of the components and layers of X.25, sections are included to explain the advantages and limitations of the protocol.

The following subjects are discussed in this chapter:

- Network communications with X.25
- Advantages and disadvantages of X.25
- DSEs, DTEs and DCEs
- The three levels of X.25
- Network user address (NUA)
- Logical channels and virtual circuits
- X.25 packet types
- X.25 network subscription
- A summary of IBM support experiences

1.1 Network Communications with X.25

X.25 networks provide wide area communications capabilities in heterogeneous environments.

1.1.1 Description

X.25 is a set of recommendations from the International Telegraph and Telephone Consultative Committee (CCITT) that defines a standard network access protocol for attaching diverse types of computer equipment to a Packet-Switched Data Network (PSDN). The CCITT consists of representatives from the Post Telegraph and Telephone authorities from nations around the world, nearly all of which offer X.25 networking services.

Although some corporations have created private networks, most companies subscribe to a public PSDN. A PSDN is an interconnecting set of intelligent switching nodes that enables subscribers to exchange data using a standard protocol and packet-switching technology. Such a network carries messages, divided into parts called packets, over circuits that are shared by many network users. A single physical line into an office can handle many concurrent connections, each called a virtual circuit. A packet consists of a sequence of data and control elements in a special format that is always transmitted as a whole. The network packet size is commonly 128 bytes (octets in X.25 terminology); however, this value can vary from 16 to 4096 bytes.

You can use X.25 communications to provide a network service for higher level protocols, such as SNA or TCP/IP. You can also use an X.25 network directly, either by using the `xtalk` command or by developing applications using one of the provided X.25 Application Program Interfaces (APIs).

Because several users simultaneously share the same circuits, a protocol is necessary to ensure that the network correctly routes data to its destination.

1.1.2 Standardization

In 1976, a protocol for attaching user equipment to a PSDN was defined by CCITT in *CCITT Recommendation X.25* (commonly called the “Orange Book”). The CCITT had full sessions every four years, with proceedings published after each.

Note: CCITT is now ITU-T, International Telegraph and Telephone Union - Telecommunications Sector, and it will publish recommendations as they are adopted, not at 4-year intervals.

Between 1976 and 1980, there were radical changes to the X.25 recommendations. The 1980 session produced the Yellow Book. This gave firm specifications for many aspects that were previously open to different interpretation. Subsequently, the 1984 Red Book and 1988 Blue Book contained comparatively minor enhancements.

The main changes from 1976 to the present in the basic standard and associated facilities are:

1976

- First X.25 Recommendation issued
- Call barring: Incoming and Outgoing
- Closed User Groups
- Default packet and window size
- LAP (Link Access Procedure) only at Data Link Level
- Reverse charging

(1978)

- X.75 first defined

1980

- Closed User Group facilities enhanced
- D-bit
- Datagrams
- Fast Select
- Interrupt
- LAPB at Data Link Level introduced
- PVCs made essential
- Reset
- Throughput Class negotiation

1984

- Call Redirection
- Datagrams dropped
- Expedited data negotiation
- Facility field length extended to 109 octets
- Hunt group
- Multilink procedure

- NUI
- Online facilities registration
- OSI Address Extension
- OSI Quality of Service Facilities
- X.3 Parameters: four added
- X.32 Recommendation

1988

- Call Deflection facility added
- Long address format for Call Request and Call Clearing packets

The International Organization for Standardization (ISO) also published the X.25 recommendations as *ISO 8208* and *ISO 7776*.

1.2 Advantages and Disadvantages of X.25

Figure 1 shows the advantages and disadvantages of X.25 networking.

Advantages	Disadvantages
Basis of OSI	Unpredictable performance
Worldwide network	
Vendor independence	
Access to public databases	
Multiple logical connections over a single physical link	
Network options for security	
Cost effectiveness	Expensive if traffic is very high or very low
High data integrity	Overhead with higher-level protocols
Network is responsibility of network vendor	User has no control over network

Figure 1. Advantages and Disadvantages of X.25 Compared to Dial-up Transmission

1.2.1 Advantages

X.25 can be a cost-effective means of networking systems in a wide geographical area, compared to traditional dial-up (circuit switched) connections or remote-bridged Local Area Networks (LANs) connected by leased lines up to 1.33 Mbps (nearly T1). It provides worldwide interconnection for international corporations. Typically, PSDN suppliers charge a one-time connection fee, a

monthly subscription charge, and a usage fee based on the number of packets transmitted across the network during that month.

Organizations use X.25 for many reasons:

- **International Standards:** The X.25 recommendations are open standards for Wide Area Networking (WAN) that form the basis of the Open Systems Interconnection (OSI) protocols, standards set by the International Organization for Standardization (ISO).
- **Worldwide Network:** Users can communicate with other systems around the world.
- **Vendor Independence:** Every major computer manufacturer supports X.25.
- **High Level of Data Integrity:** X.25 provides built-in error detection and data correction features.
- **Security:** X.25 offers built-in security features. For example, Closed User Groups allow a limited set of addresses that can communicate with each other to be defined, while refusing access to outside parties. Also, system administrators can block outgoing calls to stop people from using the network and running up network costs.
- **Network Responsibility:** For smaller organizations, this is perhaps the most important feature. The network supplier, whether a Post Telegraph and Telephone (PTT) authority or an independent company, takes full responsibility for running and managing the network.
- **Access to Information:** X.25 is a popular access method for public databases.
- **Multiple Logical Connections over a Single Physical Link:** With one physical connection, users can multiplex several connections simultaneously.

1.2.2 Disadvantages

Although most of these features are benefits, some users may find them to be drawbacks. While vendor responsibility for network operations is an advantage for small to medium sized companies, very large organizations might consider vendor reliance a serious disadvantage. If something goes wrong, users have no control unless they are accessing their own private network.

Also, organizations using higher-level protocols, such as Systems Network Architecture (SNA) or Transmission Control Protocol/Internet Protocol (TCP/IP), will find that running these higher level protocols over X.25 networks results in higher overhead costs. X.25 is a fairly high-level protocol with built-in facilities like flow-control mechanisms. Since network costs reflect per-packet usage, other protocols that also provide error detection/correction (TCP/IP and SNA) produce added overhead with extra packets.

Perhaps the biggest disadvantage to AIX users accustomed to the speed of LANs is the performance of X.25 networks. X.25 usually operates on transmission media that are slower than LANs. In addition, the response time is not predictable. The inconsistency of response time is most evident with interactive applications such as the TCP/IP telnet command. For more information on performance see Chapter 10, "Performance and Tuning" on page 219.

1.3 Components of a PSDN: DSEs, DTEs and DCEs

The terms DTE, DCE and DSE are used here as X.25 functional concepts. Unfortunately, the modem world also uses this terminology with a slightly different meaning, which sometimes causes confusion. We will be using the CCITT-defined X.25 terminology.

The CCITT has defined the following terminology:

- A switching node in a packet-switched data network is called **Data-Switching Equipment (DSE)**
- A computer that uses the network for communications is called **Data-Terminal Equipment (DTE)**
- A device at the point of access to the network is called **Data Circuit-terminating Equipment (DCE)**

Every *DTE* must have an associated *DCE*.

Note: As DTE and DCE are functional definitions, they need not correspond to specific items of equipment. For example, a single device may be a DSE and may also provide multiple DCE interfaces.

While X.25 is a standard network-access protocol, it is not a complete end-to-end protocol. The *CCITT Recommendation X.25* defines a standard protocol for information exchange in packet mode between a DTE and a DCE (that is, between an individual user's equipment, such as a RISC System/6000, and the network provider's equipment). X.25 does not define the network; the network is often drawn as a *cloud* because the exact configuration and internetworking vary from network to network. They are implementation dependent.

The network is composed of DCEs and DSEs that route the packets of data through the network to the intended destination. The path that a user's data will take through the cloud might vary with every packet. All the user knows is that the data goes from their DTE into the DCE, and that it arrives at the other end in the correct order.

Operation and maintenance of DCEs and DSEs is the responsibility of the network provider. If a link between two DSEs goes down, the provider must reroute traffic. X.25 does not define the route through the network or the protocols employed within it.

Figure 2 on page 6 shows the elements of a packet-switched data network.

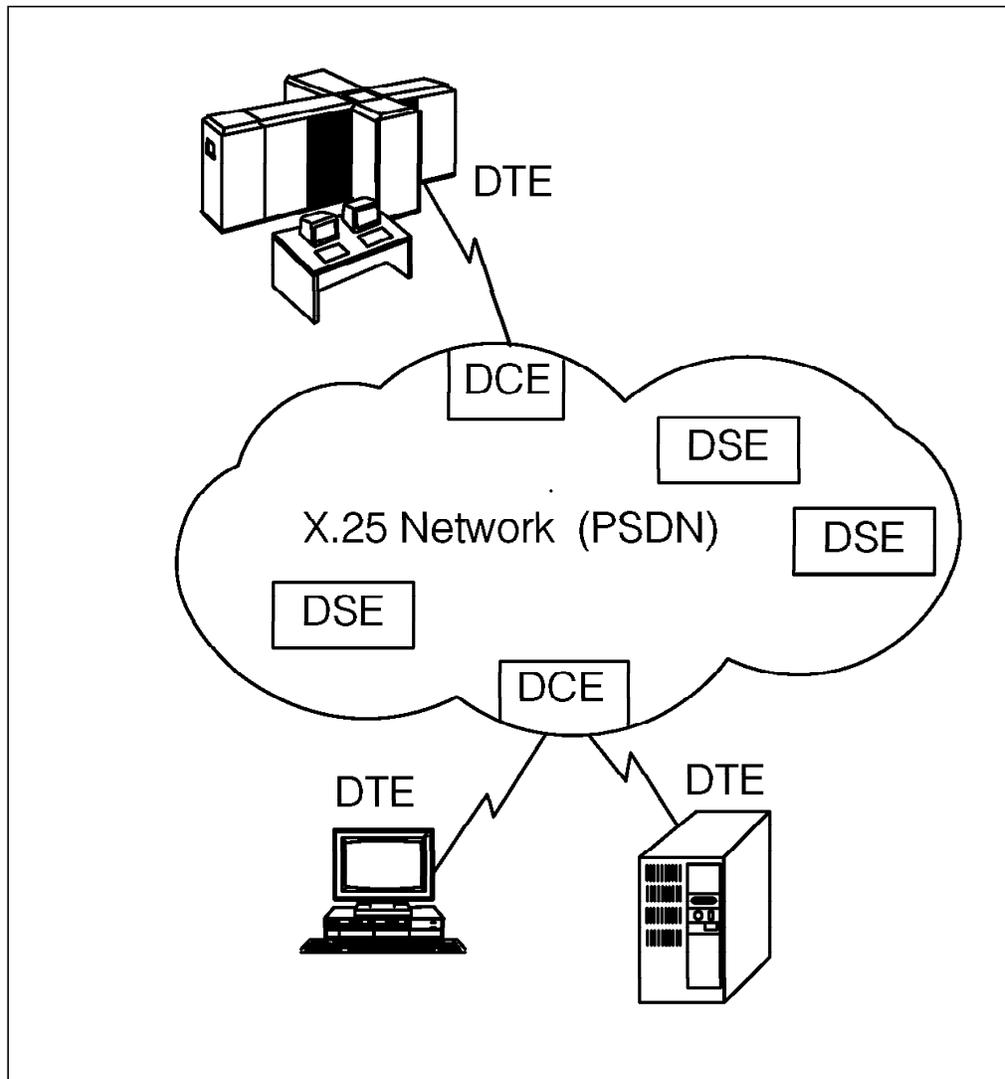


Figure 2. An X.25 Network

1.4 The Three Levels of X.25

The Open System Interconnection (OSI) reference model of the International Standards Organization (ISO) defines a 7-layer model to specify how networks work.

Levels one to three are network specific and differ depending on the physical network used. Levels four to seven are network independent. These are the higher-level functions. The X.25 protocol corresponds to the three network-specific layers.

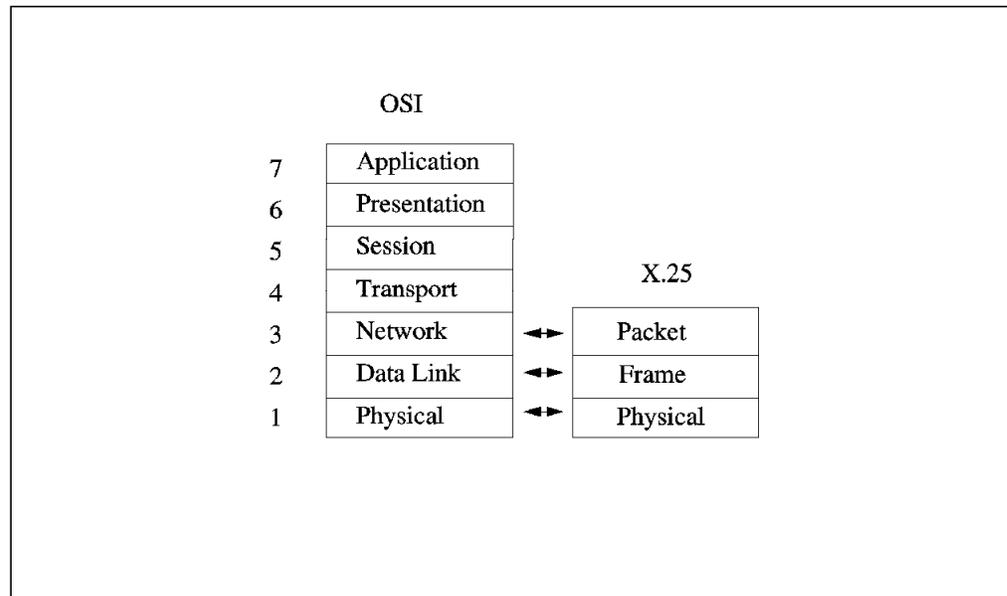


Figure 3. OSI, X.25 Layers

1.4.1 Physical Level

The physical layer is responsible for the transmission of raw bits of data across some physical medium. In the X.25 protocol, the physical level activates, maintains, and deactivates the physical circuit between a DTE and a DCE. The AIX X.25 LPP implements the physical level as a STREAMS driver; the physical layer performs the following functions:

- Maintains line characteristics of the selectable interface
- Indicates faulty incoming HDLC frames, such as frames with the wrong length
- Allows configuration of auto call units (ACU) for systems with dial-up X.25 connections

The physical level is defined in *CCITT Recommendation X.21* and in *CCITT Recommendations X.21 and X.21bis*. *CCITT Recommendation X.21bis* defines the V.24 or V.35 interface between a DTE and a DCE.

The physical characteristics are divided into four distinct categories: physical, electrical, functional, and procedural. The terms used for each kind of interface in the LPP are V.24, V.35 and X.21:

V.24 stands for V.24 + V.28 + ISO 2110

V.35 stands for V.24 + V.35 + ISO 2593

X.21 stands for X.21 + V.11 + ISO 4903

V.24 (X.21bis) and X.21 are functional recommendations which define the use of each pin. For instance, in the V.24 recommendation, the Clear To Send (CTS) is the number 106, pin 5.

V.28, V.35 and V.11 are electrical recommendations which specify the electrical levels. For instance, in V.28 recommendations, the logical state of each signal is represented by a voltage transition in either the positive or negative direction: +3V < binary 0 < +15V and -3V > binary 1 > -15V.

ISO 2110, ISO 2593 and ISO 4903 are the physical recommendations which describe the mechanical aspects of the connectors. ISO 2110 is a DB-25 connector, ISO 2593 is a DB-34 connector and ISO 4903 is a DB-15 connector.

1.4.2 Link Level

The X.25 frame level is equivalent to the OSI Data Link Level.

1.4.2.1 Purpose of this Layer

The packet layer produces X.25 packets to establish calls and transfer data. All these packets are then passed to the frame layer for transmission to the local DCE.

The X.25 recommendation for link level or frame layer describes the procedures for data interchange between a DTE and a DCE. The procedures are regrouped into three categories: link initialization/disconnection, error control and flow control. The aim is to ensure an orderly and reliable exchange of information under any of the variety of conditions that can affect a link.

The link level uses a link access procedure to ensure that data and control information are accurately exchanged over the physical circuit between the DTE and DCE. It provides recovery procedures and is based on a subset of the high-level data-link control (HDLC) called LAP_B. This procedure is synchronous and full-duplex. Once a link is started, either station can transfer information on its own initiative without waiting for permission from the other.

In HDLC, all commands, responses and data are transmitted in *frames*. Each frame has a header containing address and control information, as well as a trailer containing a frame-check sequence.

1.4.2.2 Types of Frames

There are three types of frames:

- I (information) frames transfer user data. I_frames are numbered sequentially. All X.25 packets are transferred within I_frames.
- S (supervisory) frames supervise the link performing such functions as:
 - Acknowledging I_frames
 - Requesting retransmission of I_frames
 - Requesting temporary suspension of transmission of I_framesS frames are numbered sequentially.
- U (unnumbered) frames describe the mode of operation, for example, Set Asynchronous Balanced Mode (SABM).

1.4.2.3 More about the Frames

Frames are made up of several fields:

Frame address: X.25 uses HDLC only in point-to-point mode, so the address field is used only to separate link commands from responses. The field address is set to 01 to identify frames containing commands from a DTE to a DCE and responses to these commands from a DCE to a DTE. The field address is set to 03 to identify frames containing commands from a DCE to a DTE and responses from a DTE to a DCE.

Poll/Final bit (P/F bit): The Poll bit (P) is used by the sender to insist on an immediate response. This same bit becomes the receiver's Final bit (F). The receiver always turns the Final bit on in its response to a command from the sender with the Poll bit set. The P/F bit may be used when either end becomes unsure about proper frame sequencing, perhaps from a missing acknowledgement, and wants to re-establish a point of reference.

Send and Receive Counters: I_frames are assigned independent pairs of sequence numbers that operate in both send and receive directions to ensure that no frames are lost or interpreted out of order. The N(S) counter is incremented each time a frame is sent, and the N(R) counter is incremented each time a frame is received. These counters are ranged from 0 to 7 (modulo 8) or 0 to 127 (modulo 128).

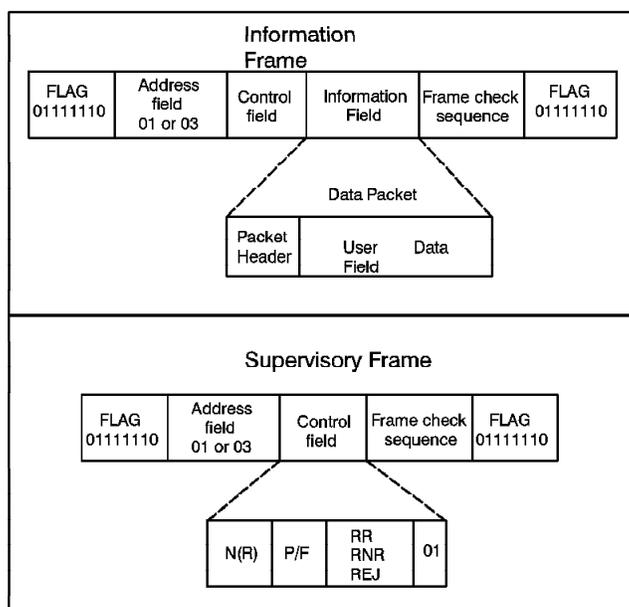


Figure 4. HDLC Frame Structures

For more information on frame control fields, see 4.2, "The x25mon Tool" on page 79.

1.4.3 Packet Level

The X.25 packet layer is equivalent to the ISO network level.

1.4.3.1 Role of the Packet Layer

The *packet layer* is the most complex layer of the three layers which compose the X.25 protocol. This complexity is a result of the designed-in flexibility and the reliable nature of the packet layer protocol. As the X.25 protocol is connection-oriented, the primary function of this layer is to give users access to the network by establishment of a connection, in the form of virtual circuit.

The *packet level protocol* specifies how virtual circuits between DTEs are established, maintained and cleared. This level defines how a single physical channel (the access link) can be treated as a set of multiple logical channels, each providing a virtual circuit. It also defines the structure of data packets and

control packets used to establish and manage a virtual circuit between two DTEs in a PSDN.

The recommendations for the packet level are not as specific as those for the physical and link levels, and they allow network providers some freedom in implementing the packet functions. For example, some networks do not support the Diagnostic Code field in the Reset and Clear Indication packets.

The X.25 packet level includes the following functions:

Multiplexing

Supports multiple concurrent connections

Data Transfer

Sends and receives data

Interrupt Transfer

Sends and receives a small amount of information independent of the data stream

Error Control

Detects packet level errors

Reset and Restart

Reinitializes communication paths if packet level errors occur

1.4.3.2 X.25 Packet Types

Different types of packets are used for such purposes as making a call, accepting a call, transferring data and terminating a call. The X.25 communications software does most of the work involved in creating the packets. You do not have to know the detailed content of each packet; you only have to supply the information that is needed to create the packet.

In some circumstances, the contents of the packet when it reaches the *called* DTE are different from when it left the *calling* DTE. This is because some information is different for each DTE (for example, logical channel number), or only applicable to one DTE, or it is information inserted by the network.

Packets are grouped into the following categories:

- Call establishment and clearing
 - CALL_REQUEST
 - INCOMING_CALL
 - CALL_ACCEPTED
 - CALL_CONNECTED
 - CLEAR_INDICATION
 - CLEAR_REQUEST
 - CLEAR_CONFIRMATION
- Data and interrupt
 - DATA
 - INTERRUPT
 - INTERRUPT_CONFIRMATION
- Flow control and reset
 - RESET_REQUEST
 - RESET_INDICATION
 - RESET_CONFIRMATION

RECEIVE_READY
 RECEIVE_NOT_READY

- Restart

RESTART_REQUEST
 RESTART_INDICATION
 RESTART_CONFIRMATION
 DIAGNOSTICS

Some of the most common control packet exchanges are:

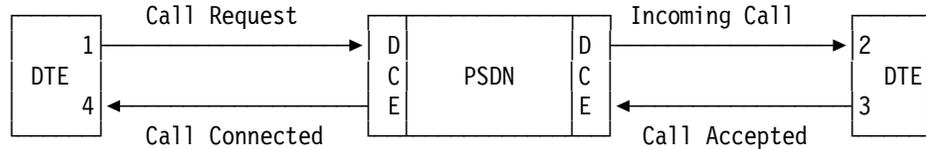


Figure 5. Establishing a Switched Virtual Circuit

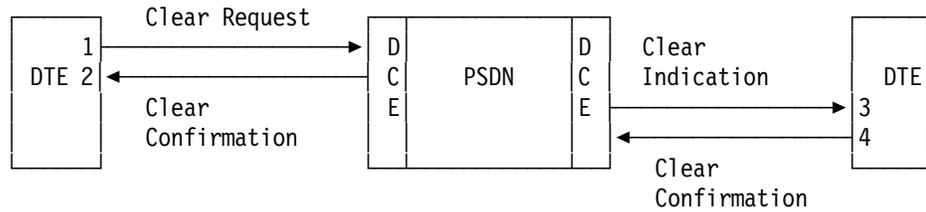


Figure 6. Clearing a Switched Virtual Circuit

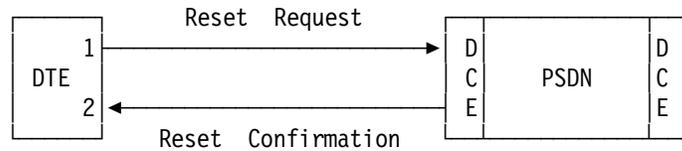


Figure 7. DTE Initiated Reset

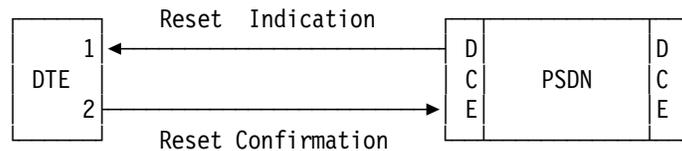


Figure 8. DCE Initiated Reset

1.4.3.3 More Details on the Packets

Packets are made up of several control fields and user data.

More-bit (M bit): Data sent during a call is divided into units. The size of these units is the *packet size*. Packet size applies to the size of the data packets, not all packets. The default packet size is 128 bytes. When one unit of data that is greater than the packet size needs to be sent, a number of packets is sent. The M bit is set to 1 to inform the destination DTE that more data will follow in the next packet. Thus, packets can be logically chained together to convey a large block of related information. Packets with the M bit set are always full. The last packet in the chain has the M bit set to 0, indicating that the series is completed.

The M bit can also be used to match unequal packet sizes at the DTE/DCE interface on either side of the network.

Delivery-bit (D bit): The D bit is normally set to 0 to indicate that the packet layer acknowledgments have only local significance between one DTE and DCE. When set to 1 in a call setup or data packet, the D bit indicates that acknowledgments have end-to-end significance between source and destination DTE. The D bit may be turned on and off selectively on a per-packet basis.

The use of the D bit increases overall network traffic.

Qualifier bit (Q bit): The data qualifier bit indicates the kind of data carried by the packet. A Q bit set to 0 means that the packet carries conventional user data. A Q bit set to 1 means that the packet carries control information and the packet should be interpreted differently by the called DTE. The use of this bit is only standardized with X.29 control packets used by the asynchronous PADs.

User Data (UD): The final field of a packet is the user data field. The use and length of this field is determined by the Packet Type Identifier.

Counters: The packet layer has its own send and receive counters, P(S) and P(R), that operate independently of the Data Link Level counters, N(R) and N(R). These counters are used for level 3 window control.

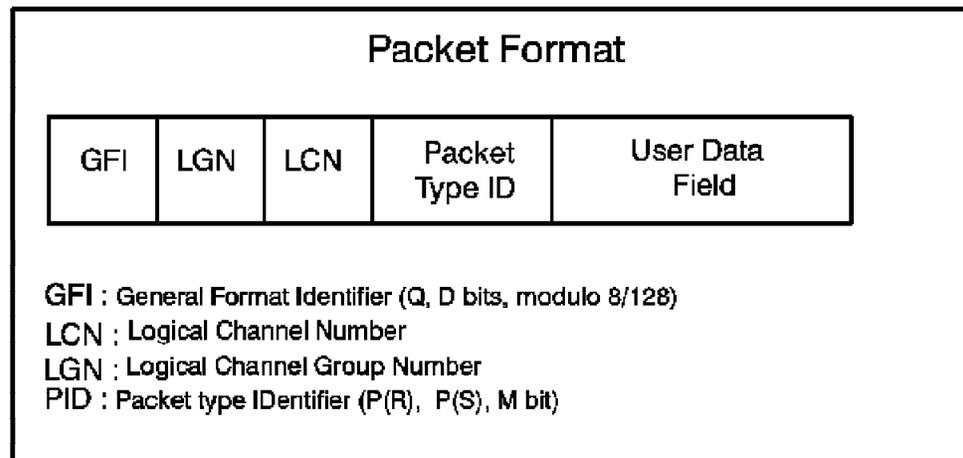


Figure 9. Packet Format

For more information on packet format, see 4.2, "The x25mon Tool" on page 79.

1.5 Network User Address (NUA)

On an X.25 network, each system (DTE) is identified by an address supplied by the network provider. This address is called *network user address* (NUA). To ensure unique DTE addressing worldwide, X.121 defines an international numbering scheme. Most public networks use the X.121 addressing standard to create NUAs.

Under the X.121 addressing standard, the network user address is built by:

- Data Network Identification Code (DNIC), which is four digits long:

- Data Country Codes (DCC)
 - First digit to identify a world geographic zone
 - Second and third digits to identify a specific country
- Fourth digit for specific Public Data Network

The number of PDNs within the U.S. exceeded the limit of 10, so the CCITT has granted the U.S. the use of the last two digits of the DNIC as the PDN identifier.

- National Terminal Number (NTN): Following the DNIC are 10 digits assigned by the PDN. When a communication is made within a given network, the NTN is often used as the NUA. The DNIC is usually used when the remote system is on a different network from the calling system. There is no rule of how NTN's should be built. However, most PDNs leave the last 2 digits as an optional subaddress to allow the X.25 subscriber to manipulate these two digits for their own applications. This subaddress is not processed by the PDN and can be used to identify a finer granularity of address on the remote system.

The following figure shows the structure of the network user address (NUA), leaving two digits for the subaddress.

Data Network Identification	National Terminal Number	Optional Subaddress
1234	56789012	34

Figure 10. Network User Address (NUA) Structure

The X.121 addressing standard also defines a one-digit optional prefix for use on international calls:

- If a call is beyond PDN boundaries, a 0 or 1 is added at the beginning of the NUA.
- If a call is within PDN boundaries, no prefix is added to it.

The maximum length of an NUA is 15 digits.

1.6 Logical Channels and Virtual Circuits

Communications in X.25 networks are via logical channels and virtual connections.

1.6.1 Definition

An AIX X.25 adapter provides one or more physical connections to the network. Each physical connection supports multiple, concurrent logical connections (*virtual circuits*) to other DTEs on the network. A virtual circuit is a data circuit between the local and remote system, which may have its route switched within the network. *Logical channels* are the conversation paths between a DTE and its DCE.

1.6.2 Logical Channel

The number of concurrent connections possible depends on the number of logical channels the network provider defines and the capabilities of the DTE hardware and software. To have 15 simultaneous connections across a network, the network supplier must provide 15 logical channels. Valid logical channel numbers range from 1 to 4095 (logical channel number 0 typically is reserved for diagnostics). The network provider assigns the specific logical channel numbers, and each number must match between the DTE and its DCE; a DTE using logical channels 51-58, for example, could not communicate with a DCE using logical channels 3002-3009. However, when a DTE talks with another DTE across a PSDN network, the DTE logical channel numbers do not have to match. The logical channel is not end-to-end in the network; it must only match between each DTE/DCE pair, as shown in Figure 11.

1.6.3 Virtual Circuit

When a user application initiates a conversation with another DTE on the network, a virtual circuit is established from one DTE to the other via the DCEs on the network. This virtual circuit only exists for the duration of the call. When the conversation is over, the virtual circuit is closed. The logical channel is then free to be allocated to another application. Each application running between two hosts requires a virtual circuit.

1.6.4 Multiplexing

With AIX, once the virtual circuit has been established between two machines for TCP/IP, all TCP/IP traffic between those two machines will flow on this single virtual circuit. The same is true for SNA on the RISC System/6000. This multiplexing of traffic helps lower the number of logical channels users need, reducing networking costs.

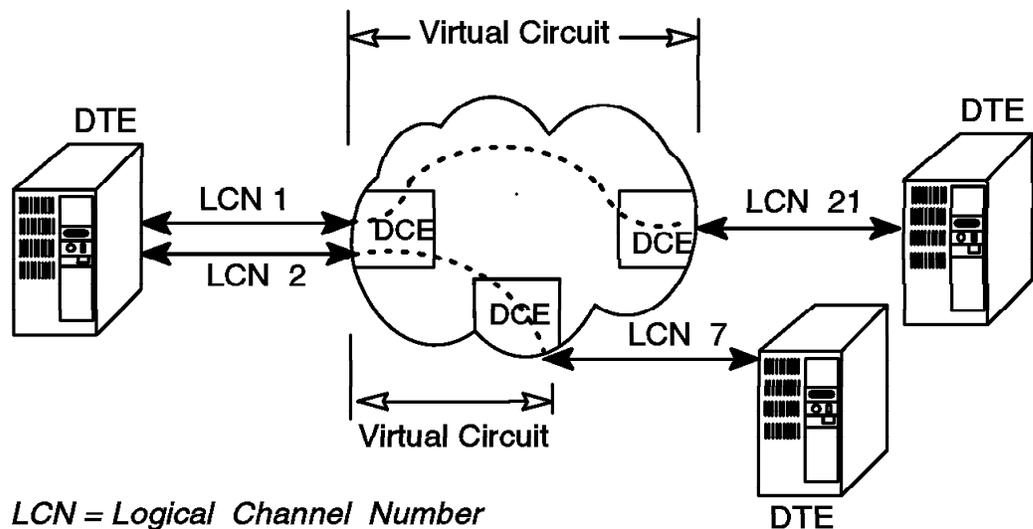


Figure 11. Logical Channels and Virtual Circuits

1.6.5 Types of Virtual Circuits

X.25 defines two basic types of virtual circuits: *Permanent Virtual Circuits* (PVCs) and *Switched Virtual Circuits* (SVCs). PSDN networks allow users to establish a PVC between two DTE addresses to save time in establishing calls. It is like having a leased line, a permanent telephone call connected, just in case you want to say something. PVCs are much less flexible than SVCs because they tie up a logical channel permanently, cost more and cannot (according to the CCITT) span different PSDN networks. This means, for example, users could not establish a PVC outside their own country. For these reasons, SVCs are much more common.

SVCs need three distinct phases for the connection establishment: call setup, data transfer and call clearing; while PVCs require only the data transfer phase.

SVCs are virtual circuits that exist only for the duration of the call, acting like a connection over the normal telephone network. For SVCs, there are three types of logical channels:

- **Incoming:** The DTE can only receive calls on this channel
- **Outgoing:** The DTE can only initiate calls on this channel
- **Two-Way:** The DTE can both receive and make calls on this channel

These channel types are used only for call initiation. Once a virtual circuit has been established, it is always for two-way communication. Typically, one or two types of SVCs are used, depending on security requirements. However, if more than one type is used, in order to minimize call collisions, the CCITT states that the logical channel numbers must be assigned within the following hierarchy, from the lowest logical channel number to the highest:

- PVC
- Incoming SVCs
- Two-way SVCs
- Outgoing SVCs

Two-way SVCs are the most common types of virtual circuits, as they offer the greatest flexibility and

least administration by allowing both incoming and outgoing calls. One-way incoming SVCs may be used for security reasons.

1.7 Facilities

Facilities allow the subscriber greater control of the X.25 environment. Their availability is optional and determined by the network provider.

1.7.1 Definition

In order to offer flexibility, X.25 recommendations define a rich set of optional user facilities. These facilities allow a network planner to fine-tune the network's handling of such major areas as security, accounting, routing and performance. The X.2 recommendations define more than 30 International User Services and Facilities in Public Data Networks. The X.25 recommendations define only the facilities related to packet-switched Public Data Networks. So at each call setup time, users may request special network services beyond normal default offerings by using the optional user facilities.

Some facilities are standard; other, optional facilities must be subscribed to by the users as a part of their contract with the network provider. The ability to negotiate the packet size, for example, is a standard facility on almost all the networks, while *reverse charging acceptance* is optional and must be allowed in the network subscription.

Some optional facilities, when included in the user subscription, are valid for all virtual calls. The reverse charging acceptance is an example of such a facility and is in effect for all incoming calls. Other facilities, such as *reverse charging*, must be specifically requested for the duration of a call.

The facilities and their coding are defined in the *CCITT Recommendation X.25* sections 6 and 7. The ones you can actually use are defined in your X.25 subscription.

1.7.2 Use of Facilities

Call time facilities: To use a facility valid for the duration of a virtual call, a *facility request* specifying the nature of the facility and the corresponding parameters must be inserted in the call packet. The X.25 application on the RISC System/6000 (DTE) can insert a facility request either in the call request (calling DTE) or call accepted (called DTE) packet. Coding of the facilities is the responsibility of the application developer and not the device driver or X.25 microcode.

The network (DCE) can also notify the DTE of the use and parameters of a facility. In this case, the DCE inserts a *facility indication* either in the incoming call or call connected packet. Figure 12 shows the name of the different call packets.

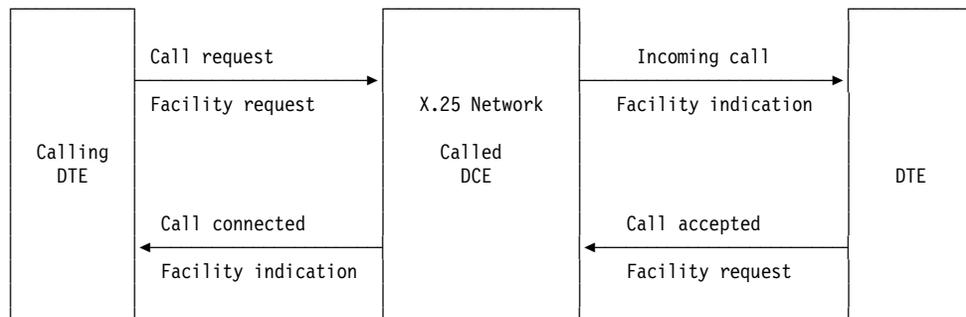


Figure 12. Facilities and Call Packets

Coding and decoding facilities: Figure 13 on page 17 shows the structure of a call packet. Call request, incoming call, call accepted and call connected packets all have the same structure.

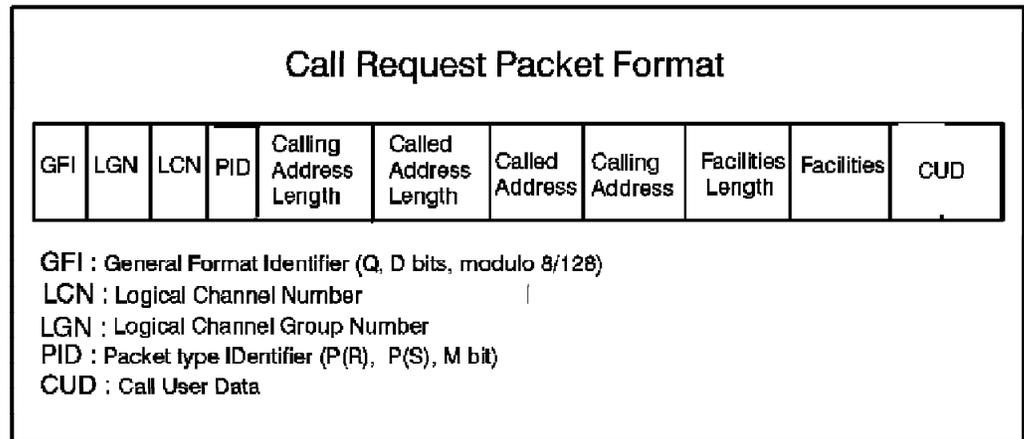


Figure 13. Call Packet Formats

The facility requests or facility indications are inserted between the address block and the CUD (Customer User Data) and are prefixed by an octet containing the total length of the facilities. Appendix E, "Facilities" on page 275 explains how to code or decode facilities.

1.7.3 Types of Facilities

The following table lists the main facilities that may be requested for the duration of a call either by the DTE or DCE.

See Appendix E, "Facilities" on page 275 for an exhaustive list of the facilities available with X.25.

Table 1. X.25 Optional Facilities

Optional facilities	1980	Req	Ind
Throughput class selection	y	y	y
Flow-control parameters selection	y	y	y
CUG selection	*	y	
Reverse charging	y	y	y
Fast select	y	y	y
Recognized Private Operating Agency (RPOA) selection	y	y	
Network user identification (NUI)			y
Call redirection notification			y
Charging requesting service		y	
Note: * Basic CUG only in CCITT 1980.			

Throughput class negotiation The throughput class negotiation, when it is subscribed to, allows you to change your default throughput class (measuring the transmission speed within the network) to a lower value. This does not affect the DTE-to-DCE speed, only the speed at which a packet traverses the switching nodes in the network. If both the calling DTE and the called DTE have subscribed to this facility, the

mechanism to select the throughput class that will be used during a virtual call is the following:

1. The calling DTE requests a throughput class for both directions of traffic with a throughput class facility request in the call request packet, or it implicitly chooses the default values. *The requested value must be lower than the default.*
2. The DCE will indicate to the called DTE in the incoming call packet the throughput class which will be the lowest value among:
 - a. The default throughput class, subscribed to by the calling DTE
 - b. The throughput class requested by the calling DTE
 - c. The default throughput class subscribed to by the called DTE
3. The called DTE may accept the proposed value if it is lower than its default or may request, with a facility in the call accepted packet, a throughput class that is lower than the value from the incoming call.
4. Finally, the value chosen for the throughput class is indicated to the calling DTE in the call connected packet.

Flow-control parameter negotiation The flow-control parameter negotiation is the determination of the packet size and packet window size that will be agreed upon by the DTE and the DCE for a given call. When two DTEs are communicating through an X.25 network, the flow-control parameters on the side of the calling DTE and on the side of the called DTE are completely independent. For example a DTE A can use a packet size of 256 and a window size of 4 and communicate successfully with a DTE B operating with a packet size of 1024 and a window size of 2.

When the network supports flow-control parameter negotiation, the incoming call packet will contain the packet size and the window size from which the negotiation will start.

By default (that is, if you do not change the default parameters) the packet size used for the call will be determined as follows:

- If the size proposed in the incoming call is within the limits of the maximum and minimum packet size, it is accepted.
- If the size proposed is larger than the maximum size defined in the DTE, there will be a negotiation, and the DTE will send a facility request in the call accepted packet accepting a packet size equal to the maximum.

The window size determination is done the same way as the packet size. If the value proposed in the incoming call packet is acceptable to the DTE, it is used. If it is not, the DTE negotiates an acceptable value by using a window negotiation facility request in the call accepted packet.

Closed User Group (CUG) A closed user group (CUG) is a facility that allows a user to collect a number of DTEs into a single logical group. Access to the group may be restricted to receiving incoming calls from and/or making outgoing calls to the unrestricted open portion of the network. So a member of one subgroup may be allowed to communicate only with other members of the same subgroup. A public network supports up to 99 CUGs for a single DTE; an extended

CUG format allows up to 10,000 CUGs. The CUG selection facility allows the DTE to specify what CUG it will be working with. It appears in the incoming call packet to identify to the called DTE which CUG has been selected.

Reverse charging option Reverse charging allows a DTE to request that the cost of a call it makes be charged to the called DTE.

Fast select option Fast select means the user has 128 bytes of call user data instead of the ordinary 16. Also, 128 bytes may be sent in the call accepted packet. Fast select must be enabled at subscription time for that particular DTE. With it, you can make applications that depend entirely on the virtual call protocol, clearing virtual circuits as soon as calls come in.

RPOA selection With the RPOA selection facility, you can select one or several specific transit networks to carry your virtual circuit. RPOA stands for Recognized Private Operating Agency, the CCITT's term for a gateway carrier between PDNs. When making internetwork or international calls, RPOA Selection in the call request packet allows the DTE to specify a particular transit network (gateway), when more than one exists. No attempt at alternate routing is made if the call cannot be successfully routed through the transit-network gateway. An extended version of this facility permits the selection of more than one RPOA transit network.

Network user identification Network User Identification (NUI) allows the requesting DTE to provide billing, security or management information on a per-call basis to the DCE.

Call redirection notification The call redirection notification informs the caller that the call has been redirected to another DTE.

Charging requesting service Charging requesting service specifies that charging information (segment count data, monetary unit data or call duration data) is required.

1.8 X.25 Network Subscription

The network supplier must give some information about the connection before the users can connect to the network. The network provider assigns the DTE address (NUA), the logical channel numbers and the types of virtual circuits. There are many more attributes that must match between the DTE and the network's DCE.

The network supplier provides the X.25 attachment attributes that must be used for DTE configuration in the network subscription. Suppliers create subscriptions based on their network conventions, DCE hardware and customers' requirements (performance, number of concurrent connections, security needs and so on). The network attachment speed and the DCE hardware will determine the modem cable interface, so customers should check with the network provider before ordering cables.

To configure an AIX DTE, the network provider should be told that the X.25 device driver expects the following attachment characteristics:

- Full-duplex, synchronous transmission
- Leased-line attachment or dial-up capability through X.32

- Network or modem-provided clocking

AIXlink/X.25 Version 1.1.3 supports CCITT up to 1988.

One RISC System/6000 may have several X.25 physical connections or ARTIC Portmaster/A, and each connection requires a network subscription. The network supplier may provide automatic call rerouting (when a port is busy) as part of the subscription.

1.9 Summary of IBM Support Experiences

Users with experience in X.25 networking find that configuring the X.25 network attachment on AIX hosts is familiar and easy. The following comments will be helpful to users who have no previous experience with X.25.

The IBM country support centers have been involved in hundreds of X.25 installations around the world since the general availability of AIX Version 3.1 in July 1990. While each situation has been unique, some similarities have been observed.

Compared to the traditional modem and LAN connectivity methods, X.25 problems often take weeks rather than days to resolve. Lack of knowledge and inexperience with X.25 has been the cause of problems in the majority of cases, thus creating a misconception of the project. AIX systems administrators tend to assume X.25 is like a token-ring or Ethernet LAN, when it is more analogous to SNA, VTAM or NCP. This misconception is understandable. Most AIX systems administrators work with LANs every day and have developed great skill in software, hardware and troubleshooting.

X.25 is more complex than LAN. On the RISC System/6000, there are roughly 100 parameters that users or systems administrators can configure (or misconfigure) with the system management interface tool (SMIT). Sometimes, a single attribute can make the difference between success and failure.

1.9.1 Planning Considerations

Organizations considering using X.25 can take the following steps to ensure as smooth an installation as possible. Understanding the scope of the task, planning appropriately and setting the proper expectations can make all the difference.

- **Learn about X.25** Ask questions and, if possible, get some hands-on self-education.
- **Plan** Define applications and select them carefully. Make sure everybody understands the interdependencies.
- **Set realistic installation time expectations** Allow plenty of time to correct errors.
- **Subscribe to a network** Find a contact at the network provider and get them involved in the planning process.
- **Take the installation in stages** Get each stage working before moving to the next. When finished, make a backup of the parameters that work.

Subscribing to a real PSDN network is key to a successful X.25 installation. While the back-to-back method (described in Chapter 3, "X.25 LPP Installation

and Setup” on page 43) is useful for early testing, it fails to provide an ongoing production solution. The link startup procedure would be cumbersome, and X.25 packets would add overhead, thereby reducing data throughput.

It is unrealistic for the X.25 novice to expect to set up any X.25 connection in one day. The X.25 attributes set in the SMIT menus are critical to success, and working out mismatches with the network can take some time. Good planning and communication between all parties can reduce installation time dramatically.

It is helpful to establish a contact who can speak for the DCE configuration early in the planning cycle. Only the network provider can advise users on the parameters that must be set in the SMIT menus.

The installation should be done in stages. Using the `x25mon` command, users can isolate frame- and packet-level problems. However, the display rate must keep up with the rate of data received. For an ASCII terminal, users should redirect `x25mon` output to a file.

Once successfully connected to the network, users should save a copy of the X.25 SMIT configuration. To assure complete recovery from a failure, users must also note any changes to `xroute` and the applications they are using.

1.9.2 Application Selection

Many AIX systems administrators want to view X.25 WANs as giant LANs. It is unrealistic to expect LAN performance from X.25 networks. Yet performance is usually sufficient for all but interactive or high-throughput applications. X.25 networking is best for batch applications. Mail exchange and order inquiry are good examples. X.3, X.28 and X.29 (collectively called *Triple X*) also provide very practical solutions. For example, corporations with retail stores and branch offices use Triple X to make casual inquiries to the central site. Triple X allows them to put low-cost terminals and printers, or PCs and workstations running an asynchronous terminal emulation program, at outlying locations. To the users performing casual order entry, queries or mail, the attachment appears to be permanent. In reality, the network is accessed only as needed. At the central site host, the data is processed as if the terminal were attached to a local serial port.

X.25 networking does not suit highly interactive or time-critical applications, or those that involve extremely high volumes of data. Examples of these applications include transfer of large graphics files, network installation, Network File System (NFS), X-Windows and extensive use of `telnet`. With X.25, users are charged for the number of packets sent. The cost of transmitting the AIX operating system across the network in chunks of 128 bytes would be prohibitive, and the transfer would be slow. With `telnet`, packets often contain a single character, so the cost of editing a file across the network would be considerable. It would be better to use `ftp` to transfer the file to the local system, edit it and then use `ftp` to return the new copy. For these types of applications, users should bridge the LAN with a product like IBM’s Token-Ring Remote Bridge. Leased lines can connect sites at speeds up to 1.33 Mbps, giving AIX users the needed bandwidth.

Finally, it is important to consider the number of concurrent users needing support over X.25 (as this will be the number of logical channels needed in the network subscription) and what they will be doing once they log in. Would the machine you intend to use support them if they were locally attached?

Interactive applications use CPU resources heavily. Please refer to 10.5, “Applications” on page 221.

Chapter 2. RISC System/6000 X.25 Support

In this chapter, we will describe the functions provided by and AIXLink/X.25 Version 1.1.3 licensed program products.

2.1 RISC System/6000 Functions Summary

AIXLink/X.25 Version 1.1.3 is supported on AIX Version 4.1.4 and Version 4.2.

Other X.25 products are available from IBM for earlier versions of AIX.

Summary of the X.25 LPP support on the RS/6000

- Complies with the CCITT 1988 X.25 Recommendations
- Supports up to 8 X.25 adapters per system
- Supports Micro Channel and ISA bus adapters
- Provides X.3, X.28 and X.29 Packet Assembler/Disassembler (PAD) support
- Supports PAD printing
- Provides SNMP support for data items from the MIBs for the packet and frame layers
- Provides the NPI packet layer programming interface
- Provides the DLPI frame layer programming interface
- Provides a compatible API for applications written to the X.25 support included in AIX Version 3
- Provides dedicated or switched (X.32, V.25 bis) network access
- Allows automatic or user-defined point-to-point DTE/DCE configuration
- Provides applications for messaging, file transfer, link status and line monitoring
- Supports TCP/IP
- Supports SNA LU 0, 1, 2, 3 and 6.2
- Supports up to 512 logical channels per line
- Physical interface can be V.24, V.35, V.36 or X.21
- Supports speeds up to 2 Mbps, depending on the physical link
- Supports several IBM communications adapters
- Priced by number of virtual circuits, not processor group

Supported Adapters

- ARTIC Portmaster Adapter/A (FC 7006/7008)
 - Micro Channel
 - 1MB - FC 7006
 - 2MB - FC 7008 (supported, but no longer available)
 - 8 Ports - V.24 interface
 - 6 Ports - V.35 or X.21 Interface
- ARTIC Multitport Model II (PS/2 FC 6590/5306*)
Note: *PS/2 FC 5306 is used when ordering in EMEA.
 - ISA
 - 8 Ports - V.24 interface
 - 6 Ports - V.35 or X.21 Interface
- X.25 Interface Co-Processor/2 - FC 2960
 - Single Port V.24/V.35/X.21
 - Micro Channel
- X.25 Interface Co-Processor - FC 6753
 - *Supported by AIXLink X.25 for AIX Version 4.x ONLY*
 - Single Port V.24/V.35/X.21
 - ISA
- ARTIC960 Adapter - FC 2929/2935/2938
 - Supported by AIXLink X.25 V1.1.3 or higher
 - Micro Channel
 - 8 port - V/24 (128 Kbps)
 - 6 port - V.36 (2 Mbps)
 - 6 port - V.35 (64Kbps)
 - 8 port - X.21 (2Mbps)

Note: The IBM X.25 Interface Co-Processor and X.25 Interface Co-Processor/2 provide equivalent function on RISC/6000 models with different I/O bus architectures. These adapters have identical pin assignments and attach with identical cables.

AIXLink/X.25 Version 1.1.3 is available and priced by number of virtual circuits:

Virtual Circuit Ordering Option

- AIXLink/X.25 Entry Level Capacity (Up to 4 Virtual Circuits)
- AIXLink/X.25 Basic Capacity (Up to 16 Virtual Circuits)
- AIXLink/X.25 Extended Capacity (17 to 64 Virtual Circuits)
- AIXLink/X.25 Advanced Capacity (65 to 256 Virtual Circuits)
- AIXLink/X.25 Unlimited Capacity (257 Virtual Circuits and above)

On a system running AIX V 4.x, the X.25 LPP requires the virtual circuit license information (see 3.3, "Software Customization" on page 52).

2.1.1 Number of Virtual Circuits Supported

The maximum number of virtual circuits supported is 512 per port. The realistic number of virtual circuits will be determined by several factors:

- Number of adapters per system

Note: Maximum number of adapters is determined by supported hardware configurations.

- Number of ports per adapter, 1 - 8
- Licensing option, AIXLink/X.25 for AIX 4.x
- Packets per second per adapter, 200 128byte packets/sec/adapter
- Acceptable performance, see Chapter 10, "Performance and Tuning" on page 219
- Recommended Number of Virtual Circuits is 1024 per adapter

2.2 RISC System/6000 X.25 Components

To participate in an X.25 network, the RISC System/6000 requires a number of components.

2.2.1 IBM X.25 Interface Co-Processor and Co-Processor/2

As stated before, the two adapters are functionally equivalent with different I/O bus interfaces.

The X.25 adapter design conforms to the ISO 8208 standard which is compatible with the CCITT 1980, CCITT 1984 and CCITT 1988 recommendations.

The IBM X.25 Interface Co-Processor(/2) provides support for attaching a RISC System/6000 unit to an X.25 network. The adapter provides a single port that will accommodate three selectable interfaces: X.21, EIA-232D/V.24 and V.35. The port has a 37-pin female D-shell connector.

The X.25 Interface Co-Processor(/2) features the following:

- Full-duplex, synchronous protocol
- Support for the following interfaces:
 - X.21 interface at up to 64 Kbps
 - EIA-232D/V.24 interface at up to 19.2 Kbps
 - V.35 interface at up to 56 Kbps
- Wrap plug for 37-pin connector supplied for testing

Feature codes for the X.25 Co-Processors are:

- For Micro Channel: Feature Code #2960
- For ISA Bus: Feature Code #6753

X.25 Co-Processor FC #6753

The Co-Processor for ISA bus has switches to set the adapter interrupt level. Refer to the adapter and system unit reference manuals to determine a valid level for your installation.

Realtime Control Microcode (RCM)

The Realtime Control Microcode is now delivered as part of the X.25 LPP and is no longer a separate installation. The RCM diskette is no longer needed.

The following figure illustrates the different configurations of the IBM X.25 Interface Co-Processors with each kind of cable:

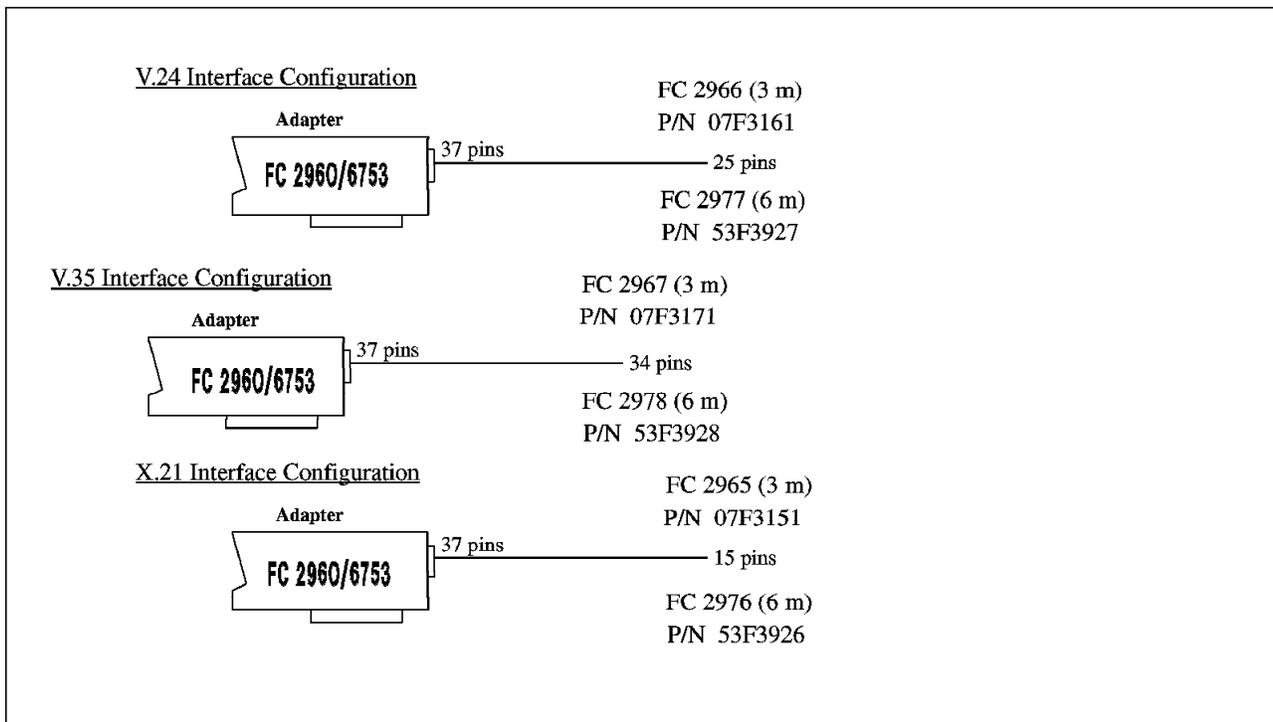


Figure 14. IBM X.25 Interface Co-Processors

2.2.2 IBM ARTIC Portmaster Adapter/A

The IBM Realtime Interface Co-Processor Portmaster Adapter/A, with 1MB (#7006) or 2MB (#7008) of memory can be used to connect a RISC System/6000 unit to an X.25 network. The ARTIC Portmaster Adapter/A must be used with an Electronic Interface Board (EIB) with the appropriate device interface cable (fanout box).

Note: Feature code 7008 is supported but no longer available.

The IBM ARTIC Portmaster Adapter/A has the following characteristics:

- Micro Channel.
- 12.5 MHz Intel 80186 microprocessor.

- Adapter-to-adapter and adapter-to-system Micro Channel bus master support.
- Up to eight serial I/O ports (asynchronous or synchronous) available through a family of interface boards and cables.
- Performance up to 64 Kbps full duplex for each of eight ports running concurrently.
- Performance of up to 2.048 Mbps full duplex for single port. This throughput is a maximum that the card is capable of achieving and is not supported by AIXLink/X/25/.
- Dynamically managed adapter storage.

Depending on the interface you are using to attach your system to the X.25 network, select one of the following interface boards and cables:

- V.24 Interface:
 - 8-Port RS-232-D/CCITT V.24 Interface Board/A (#7042)
 - 8-Port Cable (#7108)
 - FC 2939 can be used instead of the two above
- V.35 Interface:
 - 6-Port V.35 Interface Board/A (#7046)
 - 6-Port V.35 Cable (#7106)
- X.21 Interface:
 - 6-Port X.21 Interface Board/A (#7048)
 - 6-Port X.21 Cable (#7110)

The following figure illustrates the different configurations of the IBM ARTIC Portmaster Adapter/A with each kind of EIB, device interface and cable:

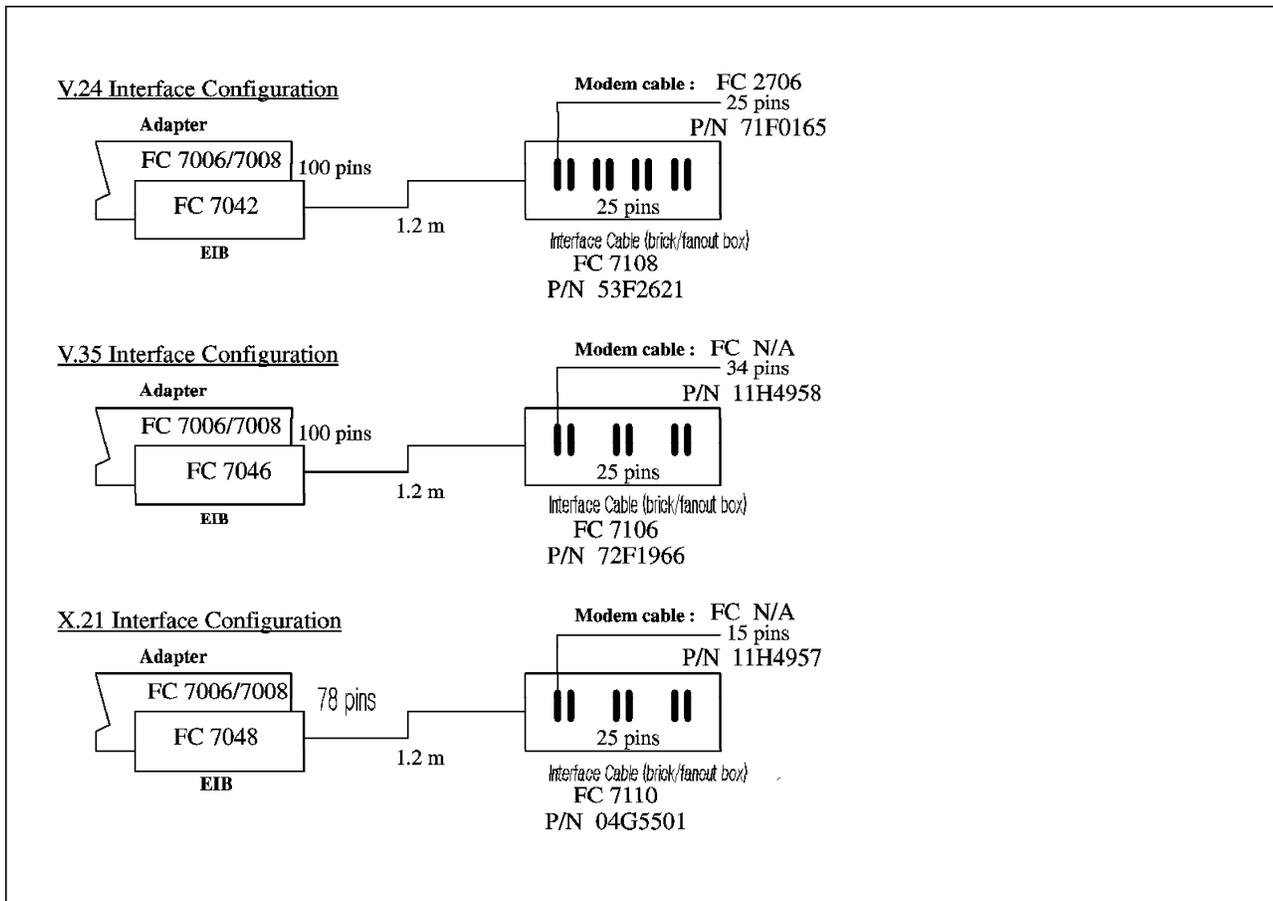


Figure 15. IBM ARTIC Portmaster Adapter/A Configurations

Note: The 4-port Multiprotocol Adapter is NOT supported for X.25 communications.

The Portmaster and Multiprotocol adapters are functionally equivalent, and the EIBs and cables are interchangeable.

2.2.3 IBM ISA Multiprotocol Model II Adapter

This base adapter provides an ISA bus equivalent to the Portmaster/A Micro Channel adapter solution supported in the original AIX X.25 LPP. In both cases, the adapter consists of a base adapter with at least 1MB of memory and one of the three daughter cards (Electronic Interface Boards or EIBs). The daughter cards and corresponding interface cables are interchangeable between the Micro Channel and ISA base adapters.

- 8 port V.24 (EIA 232)
- 6 port X.21
- 6 port V.35 connection

For V.24, the AIXLink V1.1.3 software supports speeds of up to 19.2 Kbps. For X.21 and V.35, the connections are supported up to 64 Kbps. The Multiprotocol Model II (ISA) base adapter is NOT available using a RS/6000 order number. Therefore, it can NOT be ordered through the normal RS/6000 channels and can NOT be specified via the RS/6000 configurator. This base adapter MUST be ordered through normal PC channels using the PS/2 order number, feature

#6590 for USA, Asia Pacific and Latin America, feature #5306 for EMEA. The daughter cards and corresponding interface cables (fanouts) are interchangeable between the Micro Channel and ISA base adapters. Daughter cards and fanouts may be ordered under the RS/6000 or PS/2 feature codes. Modem cables, which attach between the fanout and the modem or DSU/CSU, are also available through either channel.

The IBM Realtime Interface Co-Processor Multiport Adapter Model II can be used to connect a RISC System/6000 unit to an X.25 network. The ARTIC Multiport Adapter Model II must be used with an Electronic Interface Board (EIB) with the appropriate device interface cable (fanout box).

The adapter has the following characteristics:

- ISA bus.
- 12.5 MHz Intel 80186 microprocessor.
- Adapter-to-adapter and adapter-to-system Micro Channel bus master support.
- Up to eight serial I/O ports (asynchronous or synchronous) available through a family of interface boards and cables.
- Performance up to 64 Kbps full duplex for each of eight ports running concurrently.
- Performance of up to 2.048 Mbps full duplex for single port. This throughput is a maximum that the card is capable of achieving and is not supported by AIXLink/X.25.
- Dynamically managed adapter storage.

Depending on the interface you are using to attach your system to the X.25 network, select one of the following interface boards and cables:

- V.24 Interface:
 - 8-Port RS-232-D/CCITT V.24 Interface Board/A (PS/2 #6362 or RS/6000 #7042)
 - 8-Port Cable (PS/2 #6366 or RS/6000 #7108)
- V.35 Interface:
 - 6-Port V.35 Interface Board/A (PS/2 #1210 or RS/6000 #7046)
 - 6-Port V.35 Cable (PS/2 # 1210 or RS/6000 #7106)
- X.21 Interface:
 - 6-Port X.21 Interface Board/A (PS/2 #1170 or RS/6000 #7048)
 - 6-Port X.21 Cable (PS/2 #2028 or RS/6000 #7110)

Note: This adapter has switches to set the adapter interrupt level. Refer to the adapter and system unit reference manuals to determine a valid interrupt level for your installation.

The following figure illustrates the different configurations of the Multiport Model II Adapter with each kind of EIB, device interface and cable:

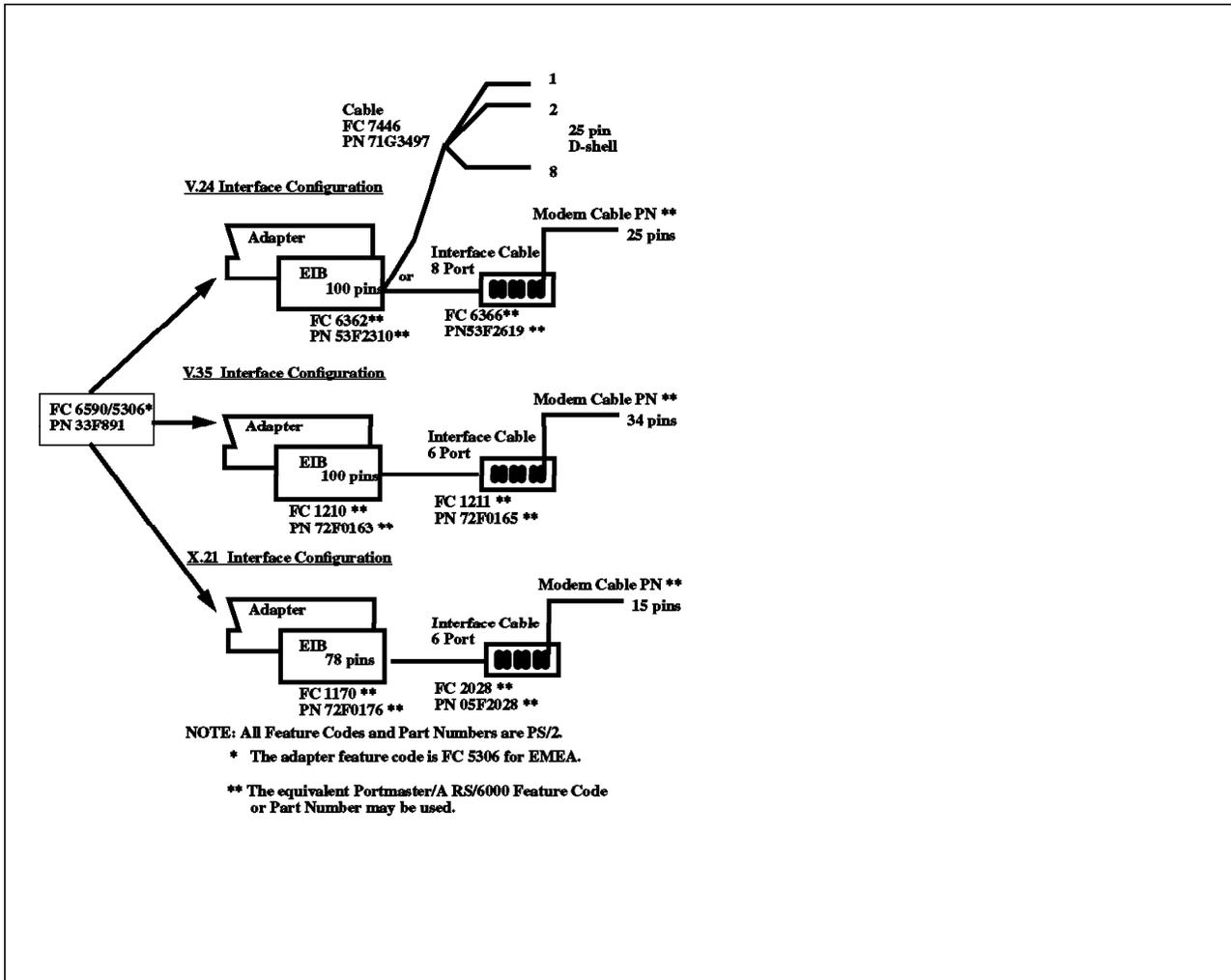


Figure 16. IBM ARTIC Multiport Adapter Model II Configurations

Note: The electronic interface boards, interface cables and modem cables used with the IBM ARTIC Portmaster Adapter/A, Figure 15 on page 28, may be substituted for the PS/2 components.

2.2.4 IBM ARTIC960 Adapter

The IBM Realtime Interface Co-Processor ARTIC960 Adapter can be used to connect a RISC System/6000 unit to an X.25 network.

Each Interface Co-Processor ARTIC960 Adapter is equipped with 4MB of memory and an electronic interface board. Feature codes 2929, 2935 and 2938 will order the adapter, 4M memory and V.24, V.36 or X.21 EIB, respectively. The ARTIC960 Adapter must be used with the appropriate device interface cable (octopus or fanout box).

The IBM ARTIC960 Adapter has the following characteristics:

- 25 MHz Intel 80C960 microprocessor.
- Adapter-to-adapter and adapter-to-system Micro Channel bus master support.
- Up to eight serial I/O ports (asynchronous or synchronous) available through a family of interface boards and cables.

- Performance up to 64 Kbps full duplex for each of eight ports running concurrently on FC 2929.
- Performance of up to 2.048 Mbps full duplex for all ports with FC 2935 and FC 2938. This throughput is a maximum that the card is capable of achieving and maybe reduced by the restrictions of the X.25 network.
- Dynamically managed adapter storage.

Depending on the interface you are using to attach your system to the X.25 network, select one of the following interface boards and cables:

- V.24 Interface:
 - 8-Port RS-232-E/CCITT V.24 (#2929)
 - 8-Port Cable (#7108 or 2939)
- V.35 Interface:
 - 6-Port V.36 (#2935)
 - 6-Port V.35 Cable (#7106)
- V.36 Interface:
 - 6-Port V.36 (#2935)
 - 6-Port V.36 Cable (#2941)
- X.21 Interface:
 - 8-Port X.21 (#2938)
 - 8-Port X.21 Cable (#2942)

The following figure illustrates the different configurations of the IBM ARTIC960 Adapter with each kind of EIB, device interface and cable:

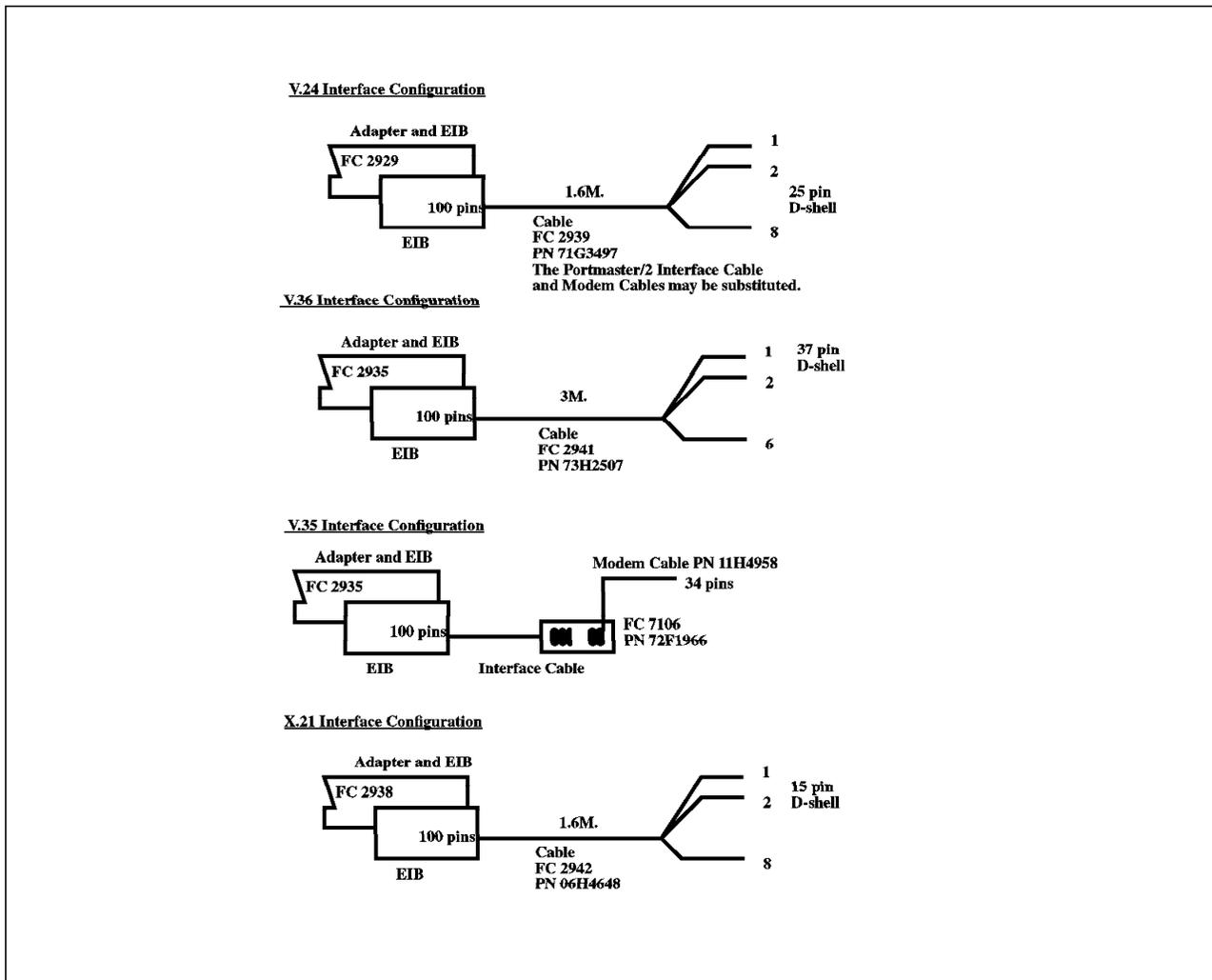


Figure 17. IBM ARTIC960 Adapter Configurations

This adapter provides an increased number of WAN ports at T1/E1 speeds and allows offloading of WAN protocols or applications onto the adapter. A powerful processor and 4MB of memory provide a platform capable of offloading WAN protocols or applications onto the adapter, freeing the host from these computer-intensive tasks. This product is four to six times faster than previously released ARTIC Portmaster family adapters and supports WAN link speeds of 2 Mbps. The adapter consists of a base adapter and one of the three protocol-specific daughter cards (Application Interface Boards or AIBs). The daughter cards offer:

- 8 port EIA 232 - (feature #2929, cable #2939)
- 6 port V.36 - (feature #2935, cable #2941)
- 8 port X.21 - (feature #2938, cable #2942)

With the new generation of adapters comes a new design in interface cables. Whereas the Portmaster fanout box requires separate modem cables for modem connection, the new "octopus" or fanout cable design combines the single adapter connector with six or eight modem cables.

V.36 offers, essentially, a new generation of V.35 connection. One visible difference is that V.36 uses a 37-pin D-shell connector while V.35 defines a

rectangular 34-pin connector. V.35 modem (or DSU/CSU) users may utilize the new ARTIC960 V.36 adapter and the Portmaster's V.35 fanout box and modem cables. This V.35 configuration is supported for port speeds up to 64 Kbps. The ARTIC960 EIA 232E/V.24 adapter users can also utilize the fanout box and cables that were used in the 8-port Portmaster/A configuration.

ARTIC960 users must install the IBM ARTIC960 AIX Support Program for RS/6000. This software is shipped with the adapter on diskette. This software must be at V1.1.3 or later. **Please ensure that the software on the diskette is saved.** To check that the software has been installed and is at the correct level use the `lslpp -l` command. The output should be:

```
devices.artic960.rte      1.1.3.0
```

If the software is installed in the wrong order, you will need to follow the recovery procedure included in the Hardware Installation section of the AIXLink/X.25 1.1 for AIX Guide and Reference.

2.2.5 X.25 Device Driver and X.25 Ports

As the new X.25 LPP supports multiport adapters, more than one X.25 port can be associated with a given adapter. The X.25 device driver handles the communication between the ports and the communication adapter. Only one device driver can be associated with each adapter. The X.25 ports are associated with the individual communication ports supported by the adapter, which can vary from one to eight.

Figure 18 on page 34 illustrates the relationship between driver instances:

- apm0-n** An X.25 Portmaster device.
- ampx0-n** An X.25 Co-Processor(/2) device.
- ricio-n** An ARTIC960 device.
- ddricio-n** IBM ARTIC960 Device Driver.
- twd0-n** The device driver associated with the adapter. Each adapter supports only ONE device driver.
- sx25a0-n** A port associated with the device driver. Only one port is used for the X.25 interface Co-Processor(/2) adapter and up to eight ports for the Portmaster Adapter/A.
- x25s0-n** An X.25 interface for COMIO emulation.
- xs2** The X.25 TCP/IP interface configured on the X.25 port sx25a3.

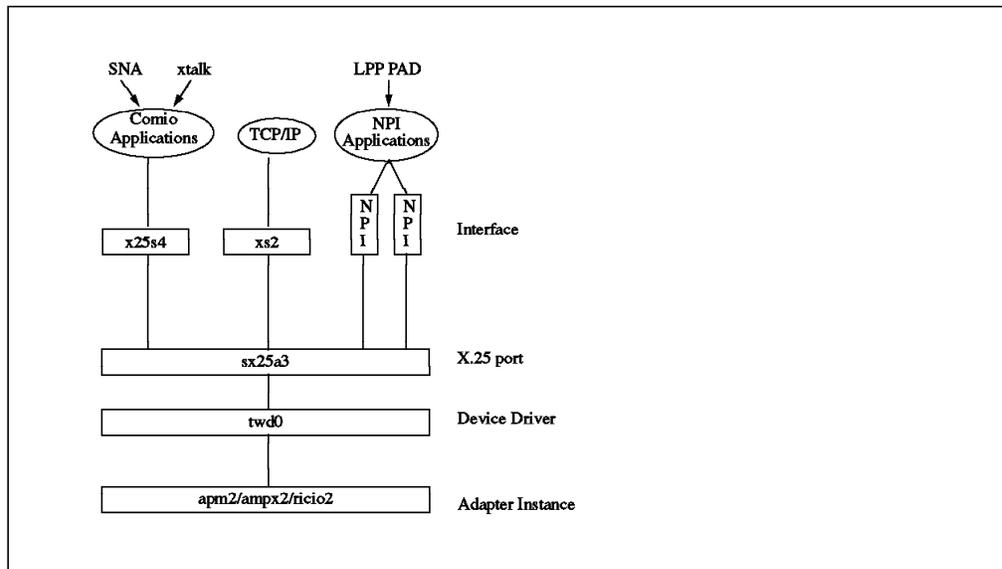


Figure 18. Adapter, Device Driver, Port, Interface

2.2.6 X.25 Commands

The X.25 commands enable you to make use of the X.25 network without doing any application programming yourself.

- The X.25 commands are:

xroute Manages COMIO X.25 routing table

xtalk Communicates with other people, manages address lists for outgoing calls

x25mon Monitors the activity on an X.25 port at the frame or packet level

x25status Returns the state of X.25 ports

For more details on these tools see Chapter 4, "Tools" on page 73.

- There are some AIX commands related to the setup and management of X.25 communications:

chsx25 Re-initializes the attributes of an X.25 port

lspvc Lists the non-default PVC attributes for an X.25 port

lsx25 Lists the configuration of the X.25 support on the system

mkpvc Creates or modifies a non-default PVC on an X.25 port

mksx25 Adds an X.25 port

rmsx25 Removes an X.25 port

chdev Allows modification of the X.25 adapter attributes

lsattr Shows the X.25 adapter and port attributes

- Other X.25 commands related to high-level networking like TCP/IP, SNA, PAD support include:

x25ip Manages a translation table from IP addresses into X.25 network user addresses (NUAs)

mksnaobj, chsnaobj, rmsnaobj, qrysnaobj	Creates, changes, removes and retrieves an SNA profile
xspad	Starts a terminal PAD session
x29d	Starts the x29 daemon

2.2.7 PAD Support

The new X.25 LPP supports X.3, X.28 and X.29 protocols. When the PAD support is enabled, the X.29 daemon, x29d, supports all the configured X.25 ports and services all remote terminals using the PAD. When the system is being used as a terminal PAD, each terminal has its own X.3/X.28 session. To start an X.3/X.28 session, run the xspad command. The use of PAD is described in 5.1, “Using a PAD” on page 91.

2.2.8 NPI Programming Interface

The Network Provider Interface (NPI) enables application access to the packet layer of the protocol stack. It provides a connection-oriented programming interface based on the UNIX International Standard NPI V2.0. A subset of the standard has been implemented with enhancements to better suit it to an X.25 environment.

The NPI consists of a set of primitives defined as STREAMS messages that provides access to the network layer services. For more details, see 6.3, “NPI API” on page 123 and *AIXLink/X.25 Version 1.1 for AIX: Guide and Reference manual*.

2.2.9 DLPI Programming Interface

The Data Link Provider Interface (DLPI) enables application access to the frame or LAP-B layer of the protocol stack. To access the frame layer directly on a port, the port’s configuration must be set to allow it. For more details, see 6.4, “DLPI API” on page 128 and *AIXLink/X.25 Version 1.1 for AIX: Guide and Reference manual*.

2.2.10 COMIO Emulation

COMIO Emulation provides application compatibility with applications developed to the interface supported with AIX Version 3 X.25 support, (SNA, xtalk, etc.). Applications developed for X.25 API provided with AIX Version 3 can use this emulation without recompilation. For more details on the use of the COMIO API, see 6.5, “COMIO API” on page 132 and *AIXLink/X.25 Version 1.1 for AIX: Guide and Reference manual*.

2.2.11 SNMP

The AIX X.25 LPPs now support Simple Network Management Protocol.

2.2.11.1 What is SNMP?

The Simple Network Management Protocol (SNMP) is a protocol used by network hosts to exchange information used in the management of networks. SNMP is defined in several TCP/IP Requests for Comments (RFCs).

SNMP network management is based on the familiar client-server model that is widely used in TCP/IP-based network applications. Each host that is to be managed runs an agent, which is a server process that maintains the MIB

database for the host. Hosts involved in network management decisions may run a process called a monitor. A monitor is a client application that generates requests for MIB information and processes responses. In addition, a monitor may send requests to agent servers to modify MIB information.

The Management Information Base (MIB) is a database containing the information pertinent to network management. The database is conceptually organized as a tree. The internal nodes of the tree represent subdivision by organization or function. MIB variable values are stored in the leaves of this tree.

2.2.11.2 X.25 and SNMP

The X.25 LPP SNMP support allows statistical data related to the X.25 interfaces to be gathered and transferred to an SNMP agent for analysis. The items of data gathered form a subset of the Management Information Base for X.25 specified in RFCs 1381 and 1382:

RFC 1381 Defines MIB variables for X.25 frame layer

RFC 1382 Defines the MIB variables for X.25 packet layer

For the list of supported objects, see *AIXLink/X.25 Version 1.1 for AIX: Guide and Reference*. For the use of the X.25 SNMP see 7.5, "Configuring SNMP" on page 147.

2.2.12 SMP machines and X.25

AIXLink/X.25 V1.1.2 and higher will run on the SMP RS/6000 machines. The AIXLink/X.25 LPP, however, is NOT SMP enabled. Nevertheless, the device drivers for the X.25 cards run in what is termed as FUNNELED mode. This means that the device drivers will run solely on the master processor and do not take advantage of the other processors.

2.2.13 X.32

The X.32 protocol defines the way you can access a Packet Switched Public Data Network (PSPDN) through a Public Switched Telephone Network (PSTN) or an Integrated Services Digital Network (ISDN) or a Circuit Switched Public Data Network (CSPDN).

The main advantages of using the X.32 protocol over dial-up lines are:

- Better link efficiency due to less protocol overhead
- The ability to run multiple sessions over a single dial-up link
- Error control using LAPB

The X.32 recommendation provides four distinct methods for DTE identification. These methods are:

- Identification provided by the public switched network
- Identification by means of a link layer Exchange Identification (XID) procedure
- Identification by means of a packet layer registration procedure
- Identification by means of the NUI selection facility in call setup packets

The X.25 LPP provides X.32 support, allowing you the usage of XID procedure.

2.3 X.25 Power Management

2.3.1 General Power Management and X.25

The AIXLink/X.25 1.1 for AIX drivers have been updated to provide Power Management Support. Power Management is a technique that enables hardware and software to minimize system power consumption. This technique tends to be emphasized only on low-end models and systems operating on battery. When Power Management is enabled, the system enters a power-saving mode under a number of conditions. Power Management state transitions include:

- enable
- standby
- suspend
- hibernation
- shutdown

Each transitions implies further decreasing the power supplied to various system components. Only suspend, hibernation and shutdown states have an impact on X.25 connections. Drivers with NO Power Management awareness can prevent Power Management capable systems from going into power-saving transitions.

The enhancement to AIXLink/X.25 V1.1.3 allows customers connected via X.25 to packet switched data networks who want to utilize the suspend or hibernate states to do so. This support is currently available to X.25 users running either the Multiport 2 or X.25 Interface Co-Processor adapters in Power Management-enabled ISA bus systems. When Power Management is configured and enabled, the entire system goes into hibernate mode after a user-configured period of inactivity from the keyboard. From an X.25 perspective, it is as though the X.25 adapter cable has been unplugged from the NTU (modem or DSU/CSU). All ports remain "Available" and applications continue to run although the physical connection is broken.

Keyboard activity keeps systems in full-on state. X.25 link activity will neither keep a system in full-on mode, nor reactivate it during hibernations. Therefore, remote users will not be able to access AIX systems that are in either suspend or hibernation mode. For this reason, Power Management is not recommended for X.25 attached hosts running as unattended data servers.

2.3.2 Impact to Network Provider

As Power Management from a X.25 perspective is a loss of power to the adapters, all network connections are lost. Since all signals are dropped at the physical layer, there is no opportunity for the local DTE packet layer to send out clear requests to the DCE. Also, the frame layer does not go through the usual DISC/UA sequence prior to the link going down. Once power is restored, all X.25 connections which were physically connected will be restored. Two exceptions are dial-up connections and those ports controlled directly by DLPI applications.

Restoring an X.25 connection refers to bringing up the physical, frame and packet layers between the local DTE and DCE. Switched Virtual Circuits that were active before the power loss will have been cleared. Permanent Virtual Circuits will be reset. It is therefore up to the corresponding applications to

re-establish Switched Virtual Circuits and to properly resynchronize the Permanent Virtual Circuits.

2.3.3 Impact to Local Applications

System shutdown impacts X.25 applications differently than suspend or hibernation. Shutdown causes ALL user applications to be terminated. When system shutdown is issued, no attempt is made to preserve the current user state before power off and all X.25 user applications need to be restarted after power-on. When suspend or hibernation states are entered, there is an attempt to preserve as much of the current system state as possible. This is to make the off-on cycle as transparent as possible to user applications. All user applications remain active. In the case of X.25 applications there is a direct impact as the network physical connections are lost. This makes the power-off and power-on cycles not completely transparent.

The following briefly highlights how each of the interfaces behave during the suspend or hibernation power-off/on cycles.

- DLPI** It is up to the DLPI application to handle the indication when a **DL_DISCONNECT** is received. The port is not active until a new **DL_CONNECT** is received by the application.
- TCP/IP** This interface is highly robust and transparent. In most cases, the application does not even know that the underlying circuit was reset or cleared.
- NPI** When the suspend or hibernation states are entered a **N_DISCON_INDs** is received by all SVCs and a **N_RESET_INDs** on all configured and bound PVCs. The application needs to be able to handle these calls. All **N_BIND_REQs** are maintained as long as the application does not issue an unbind. This implies that all PVCs remain active, but SVCs need to be re-established.
- COMIO Emulation** In the case of a base X.25 API application running through COMIO, all active SVCs are cleared and all PVCs are reset. Before any of the sessions can be re-established, the application must ensure that the sessions are properly aborted.
- PAD** All active PAD connections are cleared when the suspend or hibernation state is entered.

2.3.4 Power Management Limitation Warnings

The following is a list of things that it is NOT recommended to do while the system is in a Power Management state.

- **Change the configuration during suspend/hibernation**
Changing the configuration of devices or ports while the system is in suspend/hibernation states can cause unpredictable results. It can cause loss of data, filesystem corruption, a system crash or failure to resume from the transition state.
- **Non-PM-aware device drivers**
If a device driver is installed that is not Power Management aware, then the suspend and hibernation state must NEVER be used as the results are unpredictable.
- **Booting from CD-ROM or other media after hibernation**

Accessing the rootvg from maintenance mode, such as a CD-ROM boot when a valid hibernation image exists, can result in loss of data and filesystem corruption.

- **Maintenance modes after hibernation**

To avoid filesystem corruption and loss of data, only use maintenance modes after a NORMAL system shutdown.

- **Network connections during suspend/hibernation**

As network connections are disconnected during suspend/hibernation states, it is recommended that these states are NOT used for machines running following network interfaces.

X.25

TCP/IP

NFS

AFS

DCE

SNA

OSI

NetBIOS

- **Power Button Behavior**

When Power Management is enabled, the power button is software controlled. If there is a system problem, it should always be possible to turn off the power by pressing the power button three times quickly. This overrides whatever state transition was selected for for the power switch and requires a full boot.

2.4 Hardware Requirements

AIXLink/X.25 Version 1.1 can be used on all systems that support AIX Version 4.x with X.25 Interface Co-Processor/2 (FC 2960) or/and ARTIC Portmaster Adapter/A (FC 7006/7008) or/and ARTIC960 Adapter (FC 2929, 2935, 2938) or X.25 Interface Co-Processor (FC 6753). The X.25 Interface Co-Processor is available on models supporting ISA I/O bus architecture only and requires AIX Version 4.x.

AIX X.25 Version 1.1 is supported on all systems that support AIX Version 3.2.5 for RISC System/6000 and the X.25 Interface Co-Processor/2 or/and ARTIC Portmaster Adapter/A.

AIXLink/X25 Version 1.1 and AIX X.25 Version 1.1 require a minimum of 1 MB of memory and 8 MB of free space in the /usr file system.

The maximum number of adapters per system is determined by the system model.

Adapter	Software Support	Packets per second	Number Logical Channels
MicroChannel bus:			
X.25 Co-Processor/2	AIX V3 BOS	35	64 per adapter
X.25 Co-Processor/2	AIXLink LPP	100	512 per adapter
Portmaster/A	AIXLink LPP	200	512/port, 1024/adapter
ARTIC960	AIXLink V1.1.3	1,000	512/port, 1024/adapter
ISA bus:			
X.25 Co-Processor	AIXLink LPP	100	512 per adapter
Multiport Model II	AIXLink V1.1.3	200	512/port, 1024/adapter

The above assumes 128-byte packets, measured at the packet layer API.

AIX V3 BOS assumes basic X.25 support which was included in the AIX V3 Base Operating System.

2.5 Differences Between X.25 Support on AIX Versions

The following table shows the main characteristics and differences between X.25 support on AIX Version 3.2.x, AIXLink/X.25 versions 1.1 and 1.1.3. AIXL/X.25 Version 1.1.3 is supported on AIX Versions 4.1.4 and 4.2. For more information about the differences between the X.25 support on AIX V3 and the X.25 LPP see Appendix B, "Differences Between X.25 LPP and AIX V3 Base X.25 Support" on page 239.

Description	AIX Version 3.2.x	AIXLink/X.25 Version 1.1	AIXLink/X.25 Version 1.1.3
Applications:			
TCP/IP	x	x	
AIX Version 3 X.25 API	x	COMIO emulation	COMIO emulation
NPI and DLPI	-	x	x
SNA	x	x	x
PAD support	3rd Party	comes with the LPP	comes with the LPP
PAD printing	3rd Party	3rd party	comes with the LPP
SNMP	-	x	x
IBM-Written commands:			
xcomms	x	-	-
xroute	x	COMIO emulation	COMIO emulation
xmanage	x	-	-
x25status	-	-	x
xmonitor	x	-	-
x25mon	-	x	x
xtalk	x	COMIO emulation	COMIO emulation
Software Configuration:			
Microcode Installation separately	Install separately	Installed with LPP	Installed with LPP except ARTIC960

<i>Table 3 (Page 2 of 2). Differences Between AIX Version 3.2.5, AIXLink Ver. 1.1 and AIXLink Ver. 1.1.3</i>			
Description	AIX Version 3.2.x	AIXLink/X.25 Version 1.1	AIXLink/X.25 Version 1.1.3
Hardware:			
Adapter	IBM X.25 Co-Processor or Co-Processor/2		
Max. Adapter per system	4 / 8●●	-●	-●
Max. Virtual circuits per Adapter	64●	512	512
Adapter	Portmaster Adapter		
Max. Adapter per system	-●●	-●	-●
Max. Virtual circuits per Adapter	-●	1024●	1024●
Adapter	Multiport Model 2		
Max. Adapter per system	-	-	-●
Max. Virtual circuits per Adapter	-	-	1024●
Adapter	ARTIC960 Adapter		
Max. Adapter per system	-●	-●	-●
Max. Virtual circuits per Adapter	-●	-	1024●
X.25 Co-Processor, Co-Processor/2, Portmaster/A and Multiport Mod 2 - MODEM Cable / Maximum speeds:			
X.21 (15-pin)	64 Kbps	64 Kbps	64 Kbps
V.24 (25-pin)	19.2 Kbps	19.2 Kbps	19.2 Kbps
V.35 (34-pin)	64 Kbps●	64 Kbps●	64 Kbps●
X.25 ARTIC960 - MODEM Cable / Maximum speeds:			
X.21 (15-pin)	-	-	2 Mbps
V.24 (25-pin)	-	-	19.2 Kbps
V.35 (34-pin)	-	-	64 Kbps●
V.36 (37-pin)	-	-	2Mbps.
Note:			
<ul style="list-style-type: none"> ●Up to eight adapters are supported on systems with dual Micro Channel. Each Micro Channel supports 4 adapters. ●The number of adapters is limited to the number supported in each RISC System/6000. ●The ARTIC Portmaster adapter was not supported for X.25 communications on AIX V3.x without the X.25 LPP. ●1024 is the recommended maximum number of VCs. 512 VCs are supported for each port. ●The V.35 specification actually defines 56 Kbps as the maximum line speed. In the case of the RISC System/6000, we exceed the specification. <p>The X.25 Co-Processor is supported on systems with ISA bus only.</p>			

2.5.1 Adapter Speeds

Although the IBM adapters support line speeds up to 2Mbps, it is unlikely running at these speeds will be successful unless connected to superior networks. A high-quality line and external clocking are required.

Chapter 3. X.25 LPP Installation and Setup

In this chapter you will find a step-by-step description of the installation process for supported adapters, the modem setup and the minimal customization of the X.25 support for switched virtual circuits.

Two scenarios will be described: the first is a real-life configuration with two RISC System/6000s connected through a public packet switching data network, and the second is a test configuration with two systems connected back-to-back through a synchronous modem eliminator.

3.1 Overview and Fast Path

This section overviews the hardware and software configuration of an X.25 connection. This will be an overview of two connection scenarios. Detailed steps will be covered in following sections of this chapter.

3.1.1 Scenario 1: Two RISC System/6000s Connected to a PSDN

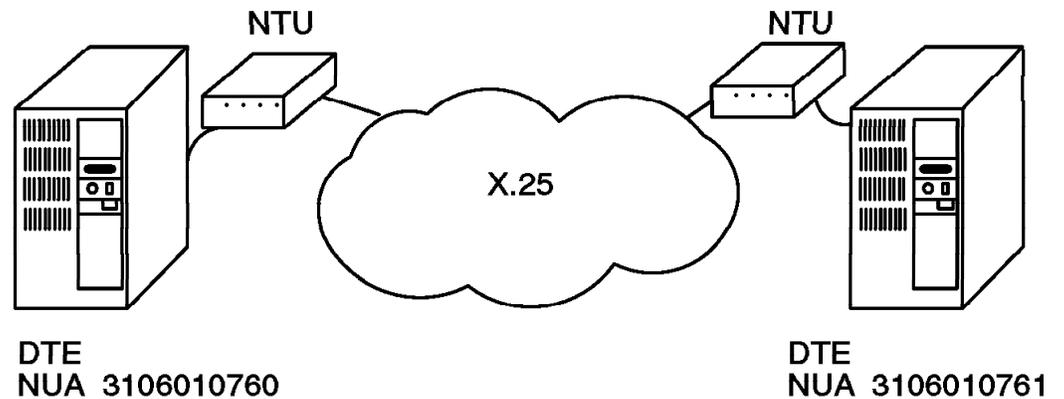


Figure 19. RISC System/6000s Connected to the TYMNET PSDN

Setup for the Connection to a PSDN

- Hardware preparation:
 1. Install the X.25 adapter.
 2. For ARTIC960 only, install the ARTIC960 AIX Support Program for RISC/6000 for AIX Version 4.
 3. Connect the system to the NTU (modem or DSU/CSU).
 - Software configuration:
 1. Install the X.25 LPP.
 2. Configure the X.25 device driver.
 3. Configure the X.25 port.
 4. With `smit x25str_mp_csp_g_sel`, configure the NUA, the number of logical channels and the virtual circuit types to match your PSDN network subscription.
- Virtual circuits can now be defined during X.25 port configuration.

Note: ARTIC960 is supported only by AIXLink/X.25 Ver. 1.1.3 on AIX Ver. 4.

If you install the ARTIC960 adapter after AIXLinkX.25, you may encounter a method error. If this occurs, run the `/usr/lpp/sx25/inst_root/sx25.ret.config` script.

3.1.2 Scenario 2: Two RISC System/6000s Connected by a Modem Eliminator

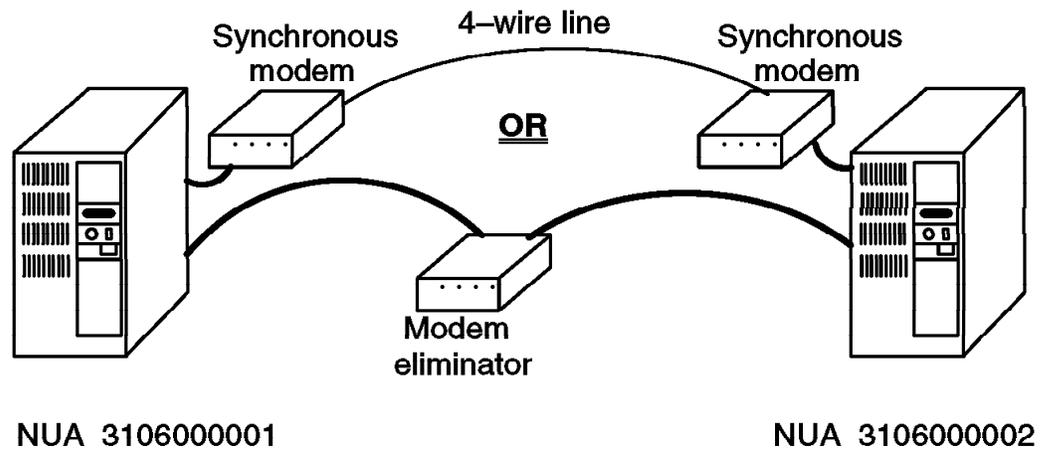


Figure 20. RISC System/6000s Connected Back-to-Back

Setup for Two RISC System/6000s Back-to-Back

- Hardware preparation:
 1. Install the X.25 adapter in any appropriate slot.
 2. Connect the system to the synchronous modem eliminator.
Note: Synchronous modems with 4-wire transmit/receive connections can be used if a modem eliminator is not available.
- Software configuration:
 1. Install X.25 LPP.
 2. Configure X.25 device driver.
 3. Configure X.25 port.
 4. Configure one system as *DTE* and the other as *DCE* or set the Allow automatic DTE/DCE detection parameter to yes-automatic. This is the default setting.
 5. With `smit x25str_mp_csp_g_sel`, configure the number of logical channels and virtual circuit types to match your PSDN network subscription.

Virtual circuits can now be defined during X.25 port configuration.

3.2 Hardware Installation

This section provides descriptions of the installation and configuration of the supported IBM X.25 adapters and related cables.

3.2.1 Installing the Adapter

The X.25 Co-Processor/2, ARTIC Portmaster Adapter/A or ARTIC960 adapter may be installed in any available Micro Channel slot. The X.25 Co-Processor or Multiport Model 2 adapter may be installed in any available ISA slot.

Note: It is good practice to execute the `diag -a` command before installing any adapters.

3.2.1.1 Installation of the X.25 Co-Processor(/2)

The X.25 Interface Co-Processor/2 may be installed in any of the available Micro Channel slots in the RISC System/6000.

The X.25 Interface Co-Processor may be installed in any of the available ISA slots in the RISC System/6000.

See 3.6, "Configuring ISA X.25 Adapters" on page 65 for information on ISA adapter interrupt switches.

Use SMIT to configure the adapter.

Once installed, check the adapter with:

```
lsdev -C -H -t articmpx
```

If you get an output like:

```
name status location description
ampx0 Available 00-05 X.25 Co-Processor/2 Adapter
```

Then the adapter is installed correctly.

But if you get an output like:

```
name status location description
ampx1 Defined 00-03 X.25 Co-Processor/2 or Multiport/2 Adapter
```

then the adapter was once installed in the machine, but now it is removed or must be configured using SMIT.

Note: The X.25 Co-Processor for ISA bus interrupt level is set using switches. Refer to the adapter and system unit documentation to determine a proper interrupt level.

3.2.1.2 Installation of the Portmaster or Multiport 2 Adapter

The portmaster Adapter may be installed in any of the available Micro Channel slots in the RISC System/6000.

The Multiport Model 2 adapter may be installed in any ISA slot.

Use SMIT to configure the adapter. Once installed, check the adapter with:

```
# lsdev -C -H -t portmaster
```

If you get an output like:

```
name status location description
apm0 Available 00-01 6-Port Portmaster Adapter/A V.35
apm1 Available 00-02 6-Port Multiport Model 2 V.35
```

then the adapter is installed correctly.

But if you get an output like:

```
name status location description
apm0 Defined 00-01 6-Port Portmaster Adapter/A V.35
```

then the adapter was once installed in the machine, but now it is removed or must be configured using SMIT.

3.2.1.3 Installation of the ARTIC960 Adapter

You must install the ARTIC960 AIX Support Program for RISC/6000 software for AIX Version 4 to use the ARTIC960 Adapter. The support program is shipped on a diskette packaged with the adapter. The support program should be installed before installing the adapter. If you install the support program after the adapter is installed, run the `/usr/lpp/sx25/inst_root/sx25.rte.config` script.

The ARTIC960 Adapter may be installed in any of the available Micro Channel slots in the RISC System/6000.

Use SMIT to configure the adapter. Once installed, check the adapter with:

```
# lsdev -C | grep 960
```

If you get an output like:

name	status	location	description
ricio0	Available	00-05	IBM ARTIC960 Adapter
ddricio0	Available	00-05-00	IBM ARTIC960 Device Driver

then the adapter is installed correctly.

But if you get an output like:

name	status	location	description
ricio0	Defined	00-05-00	IBM ARTIC960 Adapter

then the adapter was once installed in the machine, but now it is removed or must be configured using SMIT.

3.2.2 Physical Connections to the Network

The IBM X.25 adapters support three different line driver circuits to accommodate three different types of connections: X.21, X.21bis/V.24 and X.21bis/V.35. These three circuits are all connected to the X.25 Co-processor's 37-pin connector or to the Portmaster Adapter's Electrical Interface Board (fanout connector).

3.2.2.1 X.25 Co-Processor(/2) V.24, V.35 and X.21 Cables

Both IBM single port X.25 Co-Processors use the same cables.

Table 26 on page 248 shows which pins are attributed to which circuit. Using two cable identification pins, ID0 and ID1, the adapter can determine which cable is attached and activate the appropriate line driver.

X.21 Connection: The CCITT-defined X.21 attachment as the digital interface originally planned for an X.25 public network. In practice, very few countries are offering this capability yet. The X.21 cable is terminated by a 15-pin D-shell connector and is described in C.1.3.1, "X.21 Interface Cable" on page 253. The IBM references for the X.21 cable are:

#2965 PN 07F3151 for the 3-meter X.21 cable

#2976 PN 53F3926 for the 6-meter X.21 cable

When necessary, use the description given in C.1.3.1, "X.21 Interface Cable" on page 253 to make an X.21 cable.

X.21bis Connections: V.24 and V.35: As an interim measure before the generalization of X.21, the CCITT has issued the X.21bis recommendation defining the use of the popular V-series modems for X.25 networks. Normal synchronous V.24 modems can be used for speeds up to 19200 bps or V.35 modems for greater speeds. Modem pin assignments and cable descriptions are given in C.1.2.2, "V.24/X.21bis Pin Assignment" on page 251 and in C.1.2.3, "V.35/X.21bis Pin Assignment" on page 252.

The IBM feature and part numbers for the V.24 and V.35 cables are:

#2966 PN 07F3161 for the 3-meter V.24 cable

#2977 PN 53F3927 for the 6-meter V.24 cable

#2978 PN 53F3928 for the 6-meter V.35 cable

#2967 PN 07F3171 for the 3-meter V.35 cable

Note: In France, a V.35 pin adapter, Feature Code #2703, is required with the V.35 cables

If necessary, use the description given in C.1.3.2, "X.21bis/V.24 Interface Cable" on page 254 and C.1.3.3, "X.21bis/V.35 Interface Cable" on page 255 to make a cable.

3.2.2.2 Portmaster Adapter V.24, V.35 and X.21 Cables

C.2, "IBM Portmaster Adapter/2 and Multiport Model 2" on page 257 shows which pins are assigned to which circuit.

As we saw in the previous chapter, the Interface cables for the three interfaces are:

V.24 interface: FC 7108 or FC 2939 (See Figure 17 on page 32.)

V.35 interface: FC 7106

X.21 interface: FC 7110

IBM cables available for NTU(modem) connection to these are:

V.24 interface: FC 2706 (P/N 71F0165)

V.35 interface: FC 7107 (P/N 11H4958)

X.21 interface: FC 7111 (P/N 11H4957)

If necessary, use the description given in Appendix C, "X.25 Cables and Connectors" on page 247 to make the cables yourself.

3.2.2.3 Multiport Model 2 V.24, V.35 and X.21 Cables

C.2, "IBM Portmaster Adapter/2 and Multiport Model 2" on page 257 shows which pins are assigned to which circuit.

As we saw in the previous chapter, the Interface cables for the three interfaces are:

V.24 interface: PS/2 FC 6366

V.35 interface: PS/2 FC 1211

X.21 interface: PS/2 FC 2028

IBM cables available for NTU(modem) connection to these are:

V.24 interface: FC 2706 (P/N 71F0165)

V.35 interface: FC 7107 (P/N 11H4958)

X.21 interface: FC 7111 (P/N 11H4957)

If necessary, use the description given in Appendix C, "X.25 Cables and Connectors" on page 247 to make the cables yourself.

3.2.2.4 ARTIC960 V.24, V.35, V.36 and X.21 Cables

C.2, "IBM Portmaster Adapter/2 and Multiport Model 2" on page 257 shows which pins are assigned to which circuit.

As we saw in the previous chapter, the Interface cables for the four interfaces are:

V.24 interface: FC 2939

V.35 interface: FC 7106

V.36 interface: FC 2941

X.21 interface: FC 2942

IBM cables available for NTU(modem) connection to these are:

V.35 interface: FC not available yet (P/N 11H4958)

If necessary, use the description given in Appendix C, "X.25 Cables and Connectors" on page 247 to make the cables yourself.

3.2.3 X.25 Connection Test Scenarios

To test the X.25 connection between two RISC System/6000s, the best solution is to have two RISC System/6000s connected to a public X.25 packet-switched data network. If you don't have these resources, you can still perform some tests with an X.25 network simulator or with two RISC System/6000s connected back-to-back. It is even possible, in some cases, to use only one RISC System/6000. In the following paragraphs we will describe the feasible test configurations.

3.2.3.1 Two RISC System/6000s Connected to a Real X.25 Network

Each system must have a subscription to a PSDN. Each RISC System/6000 is connected to the network through a modem, DSU/CSU, provided by the network supplier called an NTU (Network Terminating Unit) whose settings you are not allowed to alter. Each RISC System/6000 is configured as a DTE and has its own NUA (Network User Address) assigned by the network provider. Figure 21 shows such a configuration.

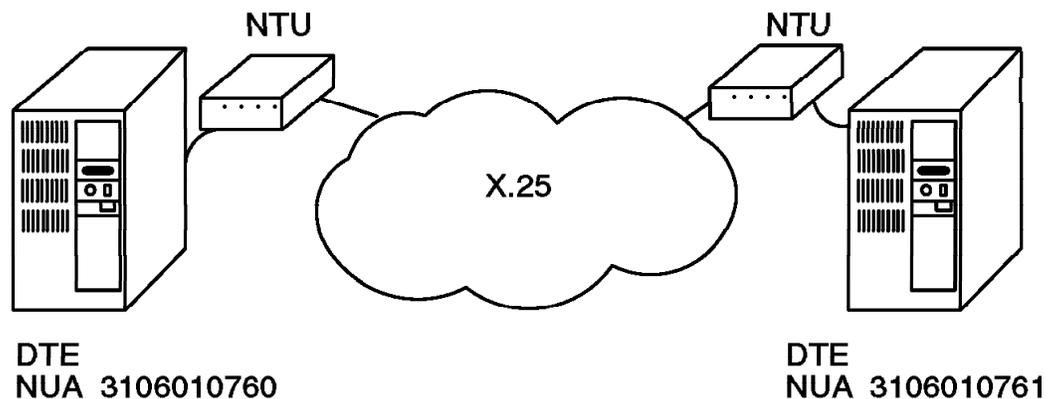


Figure 21. Two RISC System/6000s Connected to an X.25 Network

3.2.3.2 Single RISC System/6000 Connected to a Real X.25 Network

Most applications will allow you to call yourself through the X.25 network. We have tested this scenario with `xtalk` (see 4.1, "The `xtalk` Tool" on page 73). You cannot use this configuration to test TCP/IP or SNA over an X.25 network because those protocols determine that the host you are calling is the local host and that it doesn't need to go through the X.25 network to reach it.

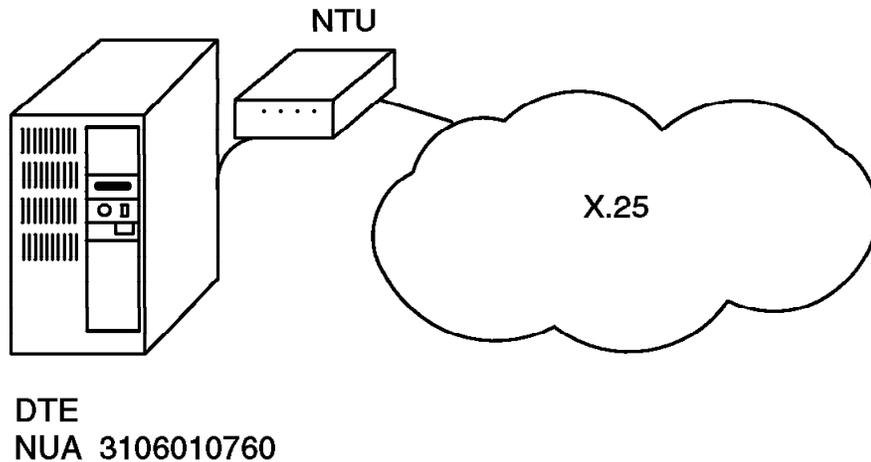


Figure 22. One RISC System/6000 Connected to an X.25 Network

Warning: Some X.25 networks may not allow this call loopback capability.

3.2.3.3 Two RISC System/6000s Connected to an X.25 Network Simulator

An X.25 network simulator is a commercially available device simulating an X.25 network to which X.25 DTEs can be attached, either remotely through a pair of modems or locally with a modem eliminator. The X.25 functions that can be tested depend on the capabilities of the simulator.

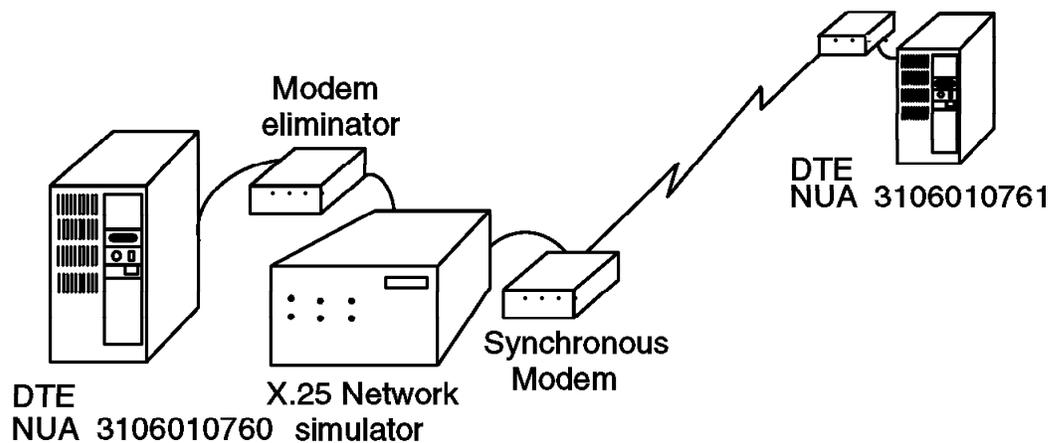


Figure 23. Two RISC System/6000s Connected to an X.25 Network Simulator

3.2.3.4 Two RISC System/6000s Connected Back-to-Back

In this point-to-point configuration, the two systems may be side-by-side, connected by a *synchronous* modem eliminator, or one may be remotely connected by a four-wire line through a pair of synchronous modems.

This configuration allows you to test communications with PVCs or SVCs but doesn't have the same customizing problems you will experience on a real

network. We don't encourage using this configuration in production environments; this is just for test purposes.

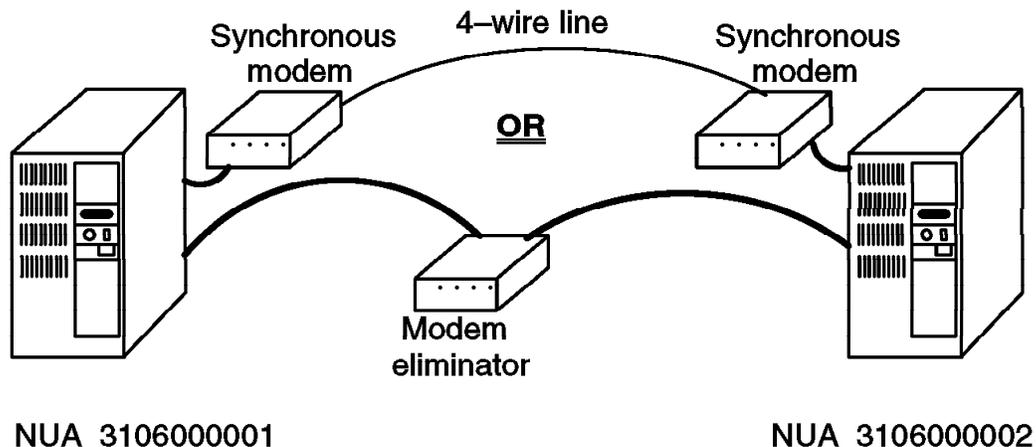


Figure 24. Two RISC System/6000s Connected Back-to-Back

3.2.3.5 One RISC System/6000 and No X.25 Network

Yes, it works. ... You can use two ports belonging to the same Portmaster Adapter, or you can use two adapters. (Each adapter can be a Portmaster Adapter or an X.25 Co-Processor Adapter.) One port must be configured as a DTE and the other as a DCE, or you must define the Allow automatic DTE/DCE detection parameter as yes-automatic. This configuration has the same limitations as the previous one. In addition, it is not suitable to test TCP/IP or SNA for the reasons given in the case of one machine connected to a network.

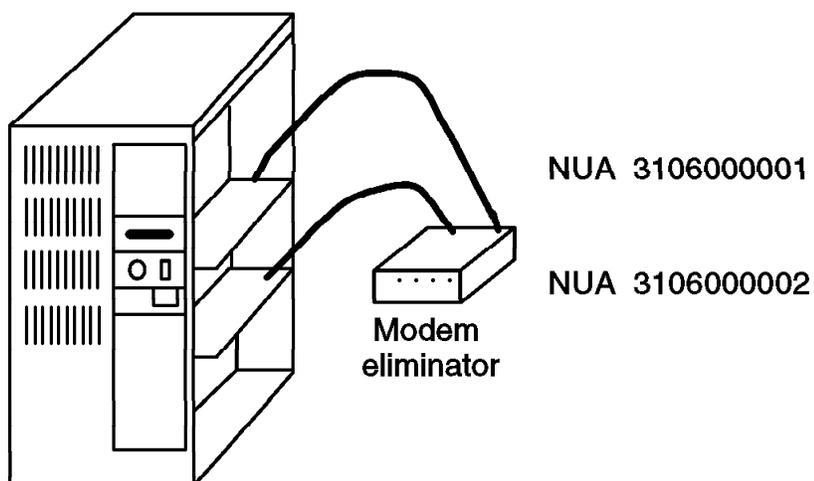


Figure 25. One RISC System/6000 with Two X.25 Co-Processors

3.2.4 Modem Considerations

If you are connected to a public X.25 network, you will probably have no problems with the modems because they are provided and configured by the network supplier. You are not allowed to change the configuration.

In other cases, you will have to select and set up the modems (or modem eliminator). You may, for example, choose a V.24 synchronous full-duplex modem for leased line, a limited distance synchronous modem or a V.24 synchronous modem eliminator. You can choose any speed between 1200 and 19200 bps, the faster, the better. There will be no need to specify this speed in the RISC System/6000 customization because the X.25 adapters will use the clock signal generated by the modem.

If your modem can be customized, verify the following settings:

1. The transmit (114) and receive (115) clock signals must be provided by the modem
2. The DSR signal may be either forced on (108-1) or controlled by the DTR (108-2)
3. The CTS signal (106) may be either forced on or controlled by the RTS
4. The RTS to CTS delay can be anything

The diagnosis of modem-related problems will be easier if you can get a V.24 breakout box to monitor the status of each V.24 line.

3.3 Software Customization

This section is a step-by-step description of the installation and customization of the X.25 LPP for both AIX 4.1.4 and AIX 4.2.

3.3.1 Installing the X.25 LPP Software

The first step on the software customization process is the X.25 LPP installation. To check if the X.25 is already installed, try:

```
# ls1pp -h "sx25*"
```

If your system is running AIX Version 3.2.5, you get:

```
Name
-----
  Fix Id  Release   Status   Action   Date       Time       User Nam
-----
Path: /usr/lib/objrepos
sx25.x25.obj
      01.01.0000.0000  COMPLETE  APPLY    09/20/94   21:03:35   root
```

Or if your system is running AIX Version 4.x, you get the following output from `ls1pp -l sx25*`

```
Fileset                               Level  State   Description
-----
Path: /usr/lib/objrepos
sx25.adt.comio                        1.1.3.0  COMMITTED  AIXlink/X.25 Application
                                         Development Toolkit - COMIO
                                         Support
```

sx25.adt.npi	1.1.3.0	COMMITTED	AIXlink/X.25 Application Development Toolkit - NPI/DLPI
sx25.adt.rte	1.1.3.0	COMMITTED	AIXlink/X.25 Application Development Toolkit - Runtime Env.
sx25.comio	1.1.3.0	COMMITTED	AIXlink/X.25 COMIO Compatibility Support and Applications
sx25.info.en_US.usr_gd	4.2.0.0	COMMITTED	X.25 User Guides - U.S. English
sx25.msg.De_DE.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - German IBM-850
sx25.msg.En_US.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - U.S. English IBM-850
sx25.msg.Es_ES.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - Spanish IBM-850
sx25.msg.Fr_FR.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - French IBM-850
sx25.msg.It_IT.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - Italian IBM-850
sx25.msg.Ja_JP.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - Japanese
sx25.msg.Nl_BE.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - Belgian Dutch IBM-850
sx25.msg.de_DE.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - German
sx25.msg.en_US.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - U.S. English
sx25.msg.es_ES.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - Spanish
sx25.msg.fr_FR.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - French
sx25.msg.it_IT.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - Italian
sx25.msg.ja_JP.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - Japanese IBM-eucJP
sx25.msg.nl_BE.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - Belgian Dutch
sx25.msg.zh_TW.rte	1.1.1.0	COMMITTED	AIXlink/X.25 Messages - Traditional Chinese
sx25.npi	1.1.3.0	COMMITTED	AIXlink/X.25 NPI and DLPI Support
sx25.pad	1.1.3.0	COMMITTED	AIXlink/X.25 Triple-X Packet Assembler/Disassembler (PAD)
sx25.rte	1.1.3.0	COMMITTED	AIXlink/X.25 Runtime Environment
sx25.server	1.1.3.0	COMMITTED	AIXlink/X.25 Server Support
sx25.tcpip	1.1.3.0	COMMITTED	AIXlink/X.25 TCP/IP Support
Path: /etc/objrepos			
sx25.comio	1.1.3.0	COMMITTED	AIXlink/X.25 COMIO Compatibility Support and Applications
sx25.rte	1.1.3.0	COMMITTED	AIXlink/X.25 Runtime Environment
sx25.tcpip	1.1.3.0	COMMITTED	AIXlink/X.25 TCP/IP Support

The X.25 LPP has already been installed in your system.

3.3.1.1 AIX V4.x X.25 LPP Installation

To retain a copy of the SMIT menus and commands used during X.25 customization, delete or rename `smit.log` and capture the image of each SMIT menu. This can be retained for future reference.

Note: After successfully customizing, save your X.25 environment using the `/usr/lpp/sx25/bin/backupx25` command.

Use the following SMIT sequence to install the X.25 LPP on AIX Version 4.1:

```
Software Installation and Maintenance
  -Install and Update Software
    -Install/Update Selectable Software (Custom Install)
      -Install Software Products at Latest Level
        -Install New Software Products at Latest Level
```

Or use the fastpath:

```
smit install_latest
```

Select your input device, and then you'll get:

```

      Install New Software Products at Latest Level

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
* INPUT device / directory for software      /dev/rmt0.1
* SOFTWARE to install                       [1.1.3.0 sx25      > +
PREVIEW only? (install operation will NOT occur)  no          +
COMMIT software updates?                     yes         +
SAVE replaced files?                         no          +
ALTERNATE save directory                     []
AUTOMATICALLY install requisite software?      yes         +
EXTEND file systems if space needed?          yes         +
OVERWRITE same or newer versions?            no          +
VERIFY install and check file sizes?         no          +
Include corresponding LANGUAGE filesets?      yes         +
DETAILED output?                             no          +
```

Use **F4** or key in the SOFTWARE to install.

The software is:

```
1.1.3.0 sx25 ALL sx25 1.1.3.0.all
```

Or you can press **F4** and then select this option. Use the following SMIT sequence to install the X.25 LPP on AIX Version 4.

```
Software Installation and Maintenance
  -Install and Update Software
    -Install and Update from LATEST Available Software
```

Select your input device, and then you will get:

```

Install and Update from LATEST Available Software

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
* INPUT device / directory for software      /dev/rmt0.1
* SOFTWARE to install                        []           +
PREVIEW only? (install operation will NOT occur)  no           +
COMMIT software updates?                    yes          +
SAVE replaced files?                        no           +
AUTOMATICALLY install requisite software?      yes          +
EXTEND file systems if space needed?          yes          +
OVERWRITE same or newer versions?            no           +
VERIFY install and check file sizes?         no           +
Include corresponding LANGUAGE filesets?      yes          +
DETAILED output?                            no           +

```

Choose SOFTWARE to install by pressing **F4**.

3.3.1.2 X.25 Virtual License Information

If the AIXLink/X.25 product is installed on a system running AIX Version 4.1, the X.25 LPP requires the virtual circuit license information. This information does not need to be supplied if the AIXLink/X.25 product is installed on a system running AIX Version 3.2.5 or if the AIXLink/X.25 entry level capacity product is installed.

To enter the X.25 license level, use the SMIT sequence:

```

Software License Management
  -Manage X.25 Server License Database

```

From the Change/Show Number of X.25 Virtual Circuits menu, select one of the following license levels depending on the X.25 virtual circuit license ordered:

- Entry (>5)
- Basic (<17)
- Extended (<65)
- Advanced (<257)
- Unrestricted

3.3.2 Configuring the X.25 Device Driver

After installing the X.25 LPP, you must configure the X.25 LPP device driver. The following steps are the same for AIX 4.1.4 and 4.2. From the SMIT main menu, select the following:

```

Devices
  -Communication
    -{choose your adapter type)
      -Adapter
        -Manage Device Drivers for (adapter)
          -Manage X.25 LPP Device Driver
            -Add a Device Driver

```

Or you can use the appropriate fast path as follows:

Adapter Type	Fastpath
X.25 Co-Processor/2 or Multiport/2 Adapter	cx25str_dd
Portmaster Adapter/A	x25str_dd
IBM ARTIC960 Adapter	ax25str_dd
X.25 Co-Processor/1 Adapter	cx25str_dd
Multiport Model 2 Adapter	x25str_isa_dd

Select the X.25 adapter:

```

Parent Adapter

Move cursor to desired item and press Enter.

  ampx0 Available 00-05 X.25 Co-Processor/2 Adapter
  apm0  Available 00-01 8-Port Portmaster Adapter/A RS-232
  apm0  Available 00-01 6-Port Portmaster Adapter/A X.21
  apm0  Available 00-01 6-Port Portmaster Adapter/A V.35
  ricio0 Available 00-07 IBM ARTIC960 Adapter
  ampx0 Available 01-02 X.25 Co-Processor/1 Adapter
  apm0  Available 01-02 6-Port Multiport Model 2 Adapter (V.35)
  apm0  Available 01-02 8-Port Multiport Model 2 Adapter (RS-232)
  apm0  Available 01-02 6-Port Multiport Model 2 Adapter/A (X.21)

F1=Help          F2=Refresh      F3=Cancel
F8=Image         F10=Exit       Enter=Do
/=Find          n=Find Next

```

3.3.3 Configuring an X.25 LPP Port

Once the device driver is available, you must configure an X.25 LPP port. You must know the Network User Address and Network Identifier of your network subscription. To add a port, select the following starting from the SMIT main menu:

Devices

- Communication
- (select one of the following adapter)

- X.25 Co-Processor/2 or Multiport/2 Adapter
- Portmaster Adapter/A
- IBM ARTIC960 Adapter
- X.25 Co-Processor/1 Adapter
- Multiport Model 2 Adapter

 -Adapter

- Manage Device Drivers for (your adapter type)
- Manage X.25 LPP Device Driver
- Manage X.25 Ports

Or use the SMIT fastpath:

Adapter Type	Fastpath
X.25 Co-Processor/2 or Multiport/2 Adapter	cx25str_mp
Portmaster Adapter/A	px25str_mp
IBM ARTIC960 Adapter	ax25str_mp
X.25 Co-Processor/1 Adapter	cx25str_mp
Multiport Model 2 Adapter	px25str_mp

Now, select the next menu:

→Add Port

Select the Adapter Driver that contains the port you want to configure, and fill in your network subscription data:

```

                                Add an X.25 Port

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
Parent adapter driver                twd0
* PORT number                        [0]                +
* Local network user address (NUA)   [3106010761]       #
* Network identifier                 [other public]      #
Country prefix                       []                #
PVC - lowest logical channel number [0]                +
PVC - number of logical channels     [0]                +
Incoming SVC - lowest logical channel number [0]            +
Incoming SVC - number of logical channels [0]            +
Two-way SVC - lowest logical channel number [1]            +
Two-way SVC - number of logical channels [1]            +
Outgoing SVC - lowest logical channel number [0]           +
Outgoing SVC - number of logical channels [0]           +

```

Note: The above menu is the same for each of the adapters you choose.

The AIX-generated command is:

```
mksx25 -c port -s star -t stx25 -p 'twd0' -w '0' -a local_nua='3106010761' -a
network_id='5' -a by_vc_start='1' -a bi_vc_num='1'
```

The contents of these fields are as follows:

- PORT number:
 - If you are using any of the following adapters, the port number must be zero.
 - X.25 Co-Processor/2
 - Co-Processor/1 Adapter
 - If you are using any of the following adapters, select the number of the port you are configuring.
 - Portmaster Adapter/A
 - IBM ARTIC960 Adapter
 - Multiport Model 2 Adapter
- Network User Address (NUA):
 - If you are connected to a public network, your subscription information will contain your network user address, (NUA). Use the international format with the country code in the first three positions.
 - If you are connected to a private network, your network administrator will provide you with an address.
- Network Identifier:

This is a multiple-choice question. Press **F4** and the following list will be displayed:

Datex-P
Datapac
Telenet
DDN
other public
other private
Extended PSS-1
Transpac

- If your network name is listed above, select it.
- If it is not listed and you are connected to a public network that allocates NUAs in accordance with the CCITT X.121 standard, select **other public**. In this case, the country code will be sufficient to identify the network and select the appropriate parameters.
- If you are connected to a test environment, or to any other private network where the NUAs do not have a country code, use **other private**.
- Local Network User Address: Must match you network subscription.
- Country prefix: Used to set default country code in NUA.

3.3.4 Updating the Number of Virtual Channels

This step modifies the definitions done as part of port configuration.

In your X.25 subscription, look for the number of logical channels you have subscribed to and for the number of the first channel. Then use SMIT to modify these values on the RISC System/6000:

```
Devices
-Communication
-(your adapter type)
-Adapter
-Manage Device Drivers for (your adapter type)
-Manage X.25 LPP Device Driver
-Manage X.25 Ports
-Change / Show Characteristics of Port
-Change / Show General Parameters
```

Or use the fastpath: `smit x25str_mp_csp_g_sel`

Since our network subscription was for 64 bidirectional SVCs starting at logical channel 1, the setting is:

```

Change / Show X.25 General Parameters

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                     [Entry Fields]
Port name                                sx25a0
Local network user address (NUA)         [3106010760]      #
Calling address in call request/accept packet [allow]          +
Enable DLPI interface ONLY               [no]             +
PVC - lowest logical channel number       [0]              +#
PVC - Number of logical channels          [0]              +#
Incoming SVC - lowest logical channel number [0]              +#
Incoming SVC - number of logical channels [0]              +#
Two-way SVC - lowest logical channel number [1]              +#
Two-way SVC - number of logical channels [64]             +#
Outgoing SVC - lowest logical channel number [0]              +#
Outgoing SVC - number of logical channels [0]              +#

X.32 Configuration
[MORE...14]

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command      F7=Edit        F8=Image
F9=Shell     F10=Exit        Enter=Do

```

SMIT checks for VC hierarchy and overlapping SVC and PVC ranges, see 1.6.5, "Types of Virtual Circuits" on page 15. For instance, if you try to configure two PVCs with lowest logical channel number one and 62 SVCs with lowest logical channel 2, you'll get this error:

```

COMMAND STATUS

Command: failed      stdout: no      stderr: yes

Before command completion, additional instructions may appear below.

Method error (/etc/methods/chgsx25):
    0514-018 The values specified for the following attributes
        are not valid:
chgsx25: Invalid two-way SVC lowest channel number
    bi_vc_start      Lowest logical channel number for a two-way SVC

F1=Help      F2=Refresh      F3=Cancel      F6=Command
F8=Image     F9=Shell        F10=Exit

```

Notes:

- A maximum of 512 virtual circuits is supported per line.
- The information concerning the number of virtual circuits must be correct; any larger or smaller number will cause problems.
- We assume that you have subscribed to bidirectional switched virtual circuits, but other possibilities are incoming SVCs, outgoing SVCs and PVCs. An incoming SVC can only receive calls, not make them. An outgoing SVC can only be used to make calls, not to receive them. Once the sessions are established, communications are two-way on the virtual circuit.

3.3.5 Modifying Other Parameters

Considering the number of parameters involved, the customization of a RISC System/6000 connected to an X.25 network may seem difficult. Actually, only a few parameters are essential, and they have CCITT-defined default values. In addition, the configuration process uses the country code of the NUA and the network type information entered at the beginning of the configuration to set default values that are unique to a country network.

The following parameter default values are CCITT standards that can be changed only as a subscription option. Check your subscription to see if any of the following parameters is mentioned with non-standard values:

Parameter	Default	SMIT Fast Path
Frame window size	7	x25str_mp_csp_f_sel
Frame modulo	8	x25str_mp_csp_f_sel
Packet modulo	8	x25str_mp_csp_p_sel
CCITT support	1988	x25str_mp_csp_p_sel
Default receive packet size	128	x25str_mp_csp_p_sel
Default transmit packet size	128	x25str_mp_csp_p_sel
Default receive packet window	3	x25str_mp_csp_p_sel
Default transmit packet window	3	x25str_mp_csp_p_sel
Default receive throughput class	64000	x25str_mp_csp_p_sel
Default transmit throughput class	64000	x25str_mp_csp_p_sel

Notes:

- None of these parameters can be defined arbitrarily by the user. You must use actual values defined on your network. Using the packet level negotiation facilities, you will be able to modify these values for each virtual call; you cannot do this by changing the defaults.
- If a default parameter, the packet size for example, is associated with parameters giving its maximum and/or minimum value, you must take care to modify them so that the default value falls between the maximum and the minimum.

In our case, we requested a non-standard default value of 1024 octets for the packet size (rather unusual). From the SMIT menu, we selected:

Devices

- Communications
 - X.25 Co-Processor/2 or Multiport/2 Adapter
 - Adapter
 - Manage Device Drivers for X.25 Co-Processor/2 or Multiport/2
 - Manage X.25 LPP Device Driver
 - Manage X.25 Ports
 - Change / Show Characteristics of Port
 - Change / Show Packet Parameters

Or use the fastpath: `smit x25str_mp_csp_p_sel`

```
Change / Show X.25 Packet Parameters

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                     [Entry Fields]
Port name                                sx25a0
CCITT support                             [1984]          +#
Packet modulo                             [8]            +#
Type of line                              [DTE]          +
Packet layer DTE/DCE line alteration policy [per OSI 8208 procedure] +
Disconnect on inactivity                  [no]           +
Registration                              [no]           +
A-bit                                     [off]          +
Network intermediate data unit (nidu) size [200]         +#

Default Attributes for SVCs
*****
Default receive packet size               [1024]         +#
Default transmit packet size              [1024]         +#
[MORE...37]

F1=Help          F2=Refresh      F3=Cancel      F4=List
F5=Reset         F6=Command      F7=Edit        F8=Image
F9=Shell         F10=Exit        Enter=Do
```

The maximum packet size would also have to be increased from the default of 256 to avoid errors in this example.

You have now set up all necessary X.25 parameters for a test environment. To have a convenient list of the X.25 parameters on your system that will help you if any problem occurs during the connection tests, type:

```
# lsattr -E -H -l sx25a0 | grep -v pvc
```

The output includes:

```
⋮
network_id      5          Network identifier
local_nua       3106010760 Network User Address
in_svc         0          Lowest logical channel number for an incoming SVC
num_in_svcs    0          Number of logical channels for incoming SVCs
in_out_svc     1          Lowest logical channel number for an two-way SVC
num_in_out_svc 20         Number of logical channels for two-way SVCs
.
```

:

The grep filter is used to eliminate the PVC parameters from the list by suppressing all the lines containing the string "pvc."

3.3.6 Customization for the Back-to-Back Scenario

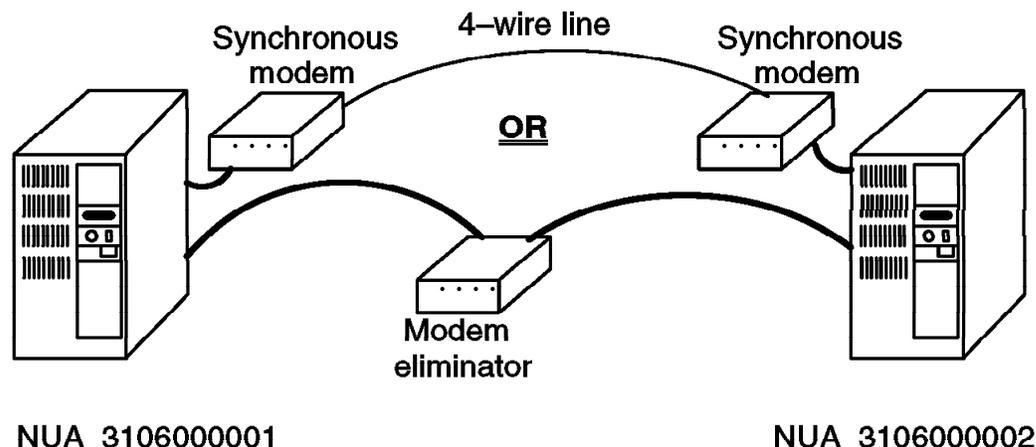


Figure 26. Two RISC System/6000s Connected Back-to-Back

If you are connecting two systems back-to-back, you can benefit from the automatic DTE/DCE detection feature. If this feature has been disabled, you must define one system as DTE and the other as DCE.

Because we will use switched virtual circuits (SVCs), we will have to assign X.25 addresses (NUAs) to our two hosts, fili and kili. NUAs should be assigned by a network provider.

In this instance, you are providing your network.

Note: Since this is a TEST environment, NUAs may be arbitrarily selected. In any production environment, NUAs must be obtained from the network provider.

The DNIC (first four digits of the NUA) of TYMNET being 3106, we have assigned the following NUA values for fili and kili:

<i>Table 5. NUAs Used for Back-to-Back Scenario</i>	
Machine Name	NUA
kili	3106000001
fili	3106000002

3.3.6.1 X.25 Adapter Initialization

Once the hardware has been prepared, we can continue by defining the device driver and then assigning a NUA to each machine. From the first menu of the SMIT interface, select:

Devices

- Communications
 - X.25 Co-Processor/2 or Multiport/2 Adapter
 - Adapter
 - Manage Device Drivers for X.25 Co-Processor/2 or Multiport/2 A da
 - Manage X.25 LPP Device Driver
 - Manage X.25 Ports
 - Add a Port

Or use the SMIT fastpath `cx25str_mp` (with X.25 Co-Processor/2) or `px25str_mp` (with a Portmaster Adapter).

kili

```

                                Add an X.25 Port
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
Parent adapter driver              twd0
* PORT number                      [0]
* Local network user address (NUA) [3106000001]
* Network identifier                [other public]
Country prefix                     []
```

fili

```

                                Add an X.25 Port
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
Parent adapter driver              twd0
* PORT number                      [0]
* Local network user address (NUA) [3106000002]
* Network identifier                [other public]
Country prefix                     []
```

The AIX-generated commands are:

```
#mksx25 -c port -s star -t stx25 -p 'twd0' -w '0'
-a local_nua='3106000001' -a network_id='5'
```

```
#mksx25 -c port -s star -t stx25 -p 'twd0' -w '0'
-a local_nua='3106000002' -a network_id='5'
```

Notes:

- Network User Address (NUA): This is required to initialize the X.25 adapter. Don't start the NUA with three zeroes.
- Field Network Identifier: Use **other public** if your NUA adheres to CCITT X.212 standards or **other private** if your NUA is purely arbitrary.
- Again, please be sure to configure the primary machine as DTE (the calling one) and the secondary machine as DTE (the listening one).

You have now set up all necessary X.25 parameters for a test environment. To have a convenient list of the X.25 parameters of your system that will help you if any problem occurs during the connection tests, type:

```
# lsattr -E -H -l sx25a0 | grep -v pvc
```

The output includes:

```

:
network_id      5          Network identifier
local_nua      3106000001  Network User Address
in_svc         0          Lowest logical channel number for an incoming SVC
num_in_svcs    0          Number of logical channels for incoming SVCs
in_out_svc     1          Lowest logical channel number for an two-way SVC
num_in_out_svc 20         Number of logical channels for two-way SVCs
:

```

The grep filter is used to eliminate the PVC parameters from the list by suppressing all the lines containing the string "pvc."

3.4 Returning to the Default Configuration Parameters

If it is necessary to start again, delete the X.25 port definition from AIX using the following commands:

If you are using COMIO, then first delete it using:

```
rmdev -l 'x25s0'
```

Then you may remove the X.25 port using the following command:

```
rmdev -d -l sx25a0
```

If you use COMIO and try to delete the sx25a0 directly, you will get the following error message:

```
Method error (/usr/lib/methods/ucfgsx25):
0514-029 Cannot perform the requested function because a
child device of the specified device is not in a correct state
```

3.5 Configuring V.25bis Dialing

If you are going to use V.25bis, you must:

1. Configure the Connection Type as V25bis.
2. Configure the Call Establishment Method as Addressed or Direct.
When addressed is specified on outgoing calls, the number must be supplied. On incoming calls, V.25bis commands are used to notify the DTE.
When direct is specified on outgoing calls, the number is stored in the modem. On incoming calls, the calling indicator is used to notify the DTE.
3. If you are using the Addressed V25bis Call Establishment Method, then you must fill the Phone Number or Address to Call attribute.
4. Set the Maximum Connection Delay attribute to be long enough for modems to perform handshaking/training operations.
5. Set the Enable/Disable DSR polling attribute to Enable.

For example, if you want to configure V.25bis with the addressed call establishment method and phone number 2291992, select from SMIT for the Co-Processor/2 adapter:

Devices

```
Communications
X.25 Co-Processor/2 or Multiport/2 Adapter
Aadapter
Manage Device Drivers for X.25 Co-Processor/2 or Multiport/2
Manage X.25 LPP Device Driver
Manage X.25 Ports
Change / Show Characteristics of Port
Change / Show General Parameters
```

```

Change / Show General Parameters

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...14]
*****
Use X.32 XID Exchange           [no]          +
X.32 XID Identity               []
X.32 XID signature              []

Dial-Up Configuration
*****
Connection Type                 [V25bis]      +
V25bis Call Establishment Method [Addressed]   +
Phone Number or Address to Call [2291992]
Maximum Connection Delay        [30]          #
Enable/Disable DSR Polling      [enable]     +
DSR polling timeout             [30]          #
[BOTTOM]

F1=Help      F2=Refresh    F3=Cancel    F4=List
F5=Reset     F6=Command    F7=Edit     F8=Image
F9=Shell     F10=Exit     Enter=Do

```

3.6 Configuring ISA X.25 Adapters

In this section, we will provide some information on how to set up an X.25 ISA adapter. In our test, we use RISC Systems/6000 Model 40P with AIX 4.1.4. For more information, please refer to *Managing AIX V4 on PCI-Based RISC System/6000 Workstations (40P/43P) SG24-2581-00*. We used IBM Realtime Interface Co-Processor Multiport Adapter Model 2. Detail information about this card could be found in the *Guide to Operations* which comes with it.

The IBM X.25 Co-Processor Multiport Adapter/2 is equipped with switches to set the operating system parameters. The procedure that is described in this section shows you how to set the operating system parameters at hardware level and at AIX level.

3.6.1 Configuring ISA X.25 Adapters at the Hardware Level

Before you install the ISA X.25 Co-Processor/2 adapter in your computer, first you need to configure the switches on the card. Please keep in mind that some parameters are set at the hardware and AIX level and they must match. Record every setting that you make.

The first three switches are used to configure interrupt level. For more detailed information on the other switches, please refer to the adapter manual. It also depends on the positions of the adapter, whether it is the first, second or third ISA adapter in your computer. The following explanation is only a guide.

Table 6 (Page 1 of 2). Interrupt Level Switch Positions			
Dip Switch 1	Dip Switch 2	Dip Switch 3	Interrupt Level
down	down	down	3
up	down	down	4

<i>Table 6 (Page 2 of 2). Interrupt Level Switch Positions</i>			
Dip Switch 1	Dip Switch 2	Dip Switch 3	Interrupt Level
down	up	down	7
up	up	down	2 or 9
down	down	up	10
up	down	up	11
down	up	up	12
up	up	up	15

<i>Table 7. X.25 Adapter: Switch Block 1</i>			
Dip Switch #	First Adapter	Second Adapter	Third Adapter
4	down	down	down
5	down	up	down
6	down	down	up
7	down	down	down
8	down	down	down
9	down	down	down
10	down	down	down

Notes:

- We use **up** and **down** instead of **off** and **on** respectively.
 - Dip switch 4 is factory set to indicate the size of RAM installed on the adapter. Verify that the position of this switch is **down** to reflect a RAM size of 512 KB.
 - Dip switch 9 indicates whether a one-edge (62-pin) or a two-edge (62-pin and 36-pin) connector is used to hold your adapter.
 - Dip switch 10 sets the bus width and should generally be set to 8-bit.
down=8-bit / up=16-bit
1. Install your ISA X.25 adapter in the computer.
 2. Insert the SMS (System Management Services) diskette into the diskette drive and turn the computer on.
 3. As soon as the first screen appears, press the **F4** key.
 4. Use the Up and Down arrow keys to select **Test the Computer** and press **Enter**. A list similar to the following will appear:

```

 Test All Sybsystems
 Test System Board
 Test Memory
 Test Keyboard
 Test Mouse
 Test SCSI Adapter
 Test Diskette
 Test Serial Port
 Test Parallel Port
 Test Performance Video
 Test Token-Ring Adapter 1
 Test Ethernet Adapter I/O 0x240
 Test Ethernet Adapter I/O 0x300
 Test X.25 Adapter I/O Addr 0x2A0
 Test X.25 Adapter I/O Addr 0x6A0

```

5. Use the spacebar to select or deselect the device.
6. Press **F6** to run the test.

If the test is successful, continue with Configuring ISA X.25 Adapters at AIX Level.

3.6.2 Configuring ISA X.25 Adapters at the Software Level

After successfully configuring the ISA X.25 Adapter at the hardware level, you will need to configure those parameters to the AIX.

1. Log in as root.
2. Use SMIT to add your adapter:

```

smit clxmpxa
  -Add an X.25 Adapter

```

Select the appropriate bus and fill in the following SMIT panel:

```

                                Add an X.25 CoProcessor/1 Adapter

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]

Description
Parent Adapter                    bus1
* ISA Virtual Connection           [0]
ISA Interrupt Level                [11]
Interrupt Priority                  2
BUS IO Address                     [0x2A0]
BUS Memory Address                 [0xE0000]
Window Size                        0x2000

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do

```

3. Verify whether the X.25 adapter is available:

```
lsdev -Cadapter -s isa
```

If the adapter has been successfully configured, the system will display device information similar to the following:

```
ampx0 Available 00-1X    N/A
```

4. Verify that the attributes are customized for the adapter. For example, for the device `ampx0` enter:

```
lsattr -l ampx0 -E
```

A list similar to the following will appear:

```
bus_intr_lvl  11      Bus Interrupt Level   True
intr_priority 2       Interrupt Priority    False
bus_io_addr   0x2A0   Bus IO address        True
bus_mem_addr  0xE0000 Bus memory address    True
window_size   0x20000 Bus memory window size False
```

5. Now, you have to configure the X.25 device driver. Use the following SMIT sequence:

```
smit cx25str_dd
  -Add a Device Driver
```

Select the parent adapter, and press **Enter**.

Enter the following command to verify the availability of the driver:

```
lsdev -Cl twd0
```

A line similar to the following should appear:

```
twd0 Available 00-1X-00 ARTIC Adapter Driver
```

6. After the driver has become available, you can configure the X.25 LPP port. For this task, you will need to obtain the details about your X.25 network subscription. Contact your X.25 network provider or your X.25 network administrator.

To start the X.25 port configuration, enter:

```
smit cx25str_mp
  -Add Port
```

Select the parent adapter driver, and provide the required information in the SMIT panel.

7. Enter the following command to verify whether the X.25 port has become available:

```
lsdev -C -t stx25 -H
```

The system will display information similar to the following:

```
name      status  location  description
sx25a0    Available 00-1X-00-00 X.25 Port
```

3.7 X.25 Facilities

In this section, we will provide some details about the function and use of the main X.25 facilities. For more information on facilities, see: 1.7.1, "Definition" on page 15 and Appendix E, "Facilities" on page 275.

3.7.1 Facilities Requested by the DTE

Only X.25 facilities included in the network subscription may be used. Facilities are coded in the layers above the packet layer. The IBM-provided applications `xtalk`, TCP/IP, and SNA Server allow the user to optionally define a facility request that will be inserted in the call packet.

X.25 facilities are configured using SMIT.

3.7.1.1 Configuring Fast Select Mode

Fast Select is an optional service of X.25 that, if configured on both DTEs, allows the transfer of up to 128 octets of user data within the call request. Configuration options for the Fast Select facility are:

- Allow
- Allow only incoming calls
- Allow only outgoing calls
- Allow only if not billed
- Forbid

The default is allow, but depending on your network subscription, you can change it. For example, if you want to change Fast Select mode to forbid, select from SMIT for the Co-Processor/2 adapter:

Devices

- Communications
 - X.25 Co-Processor/2 or Multiport/2 Adapter
 - Adapter
 - Manage Device Drivers for X.25 Co-processor/2 or Multiport/2
 - Manage X.25 LPP Device Driver
 - Manage X.25 Ports
 - Change / Show Characteristics of Port
 - Change / Show Packet Parameters

Or use the fastpath: `smit x25str_mp_csp_p_sel`

```
Change / Show Packet Parameters

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...37]                                     [Entry Fields]
R20 retransmission count restart timer          [3]                + #
R22 retransmission count reset timer            [3]                + #
R23 retransmission count clear timer            [3]                + #
R25 retransmission count window rotation timer  [0]                + #
R27 retransmission count reject timer           [0]                + #
R28 retransmission count registration timer     [1]                + #

Packet-Level Features
*****
Fast select mode                                [forbid]           +
D-bit                                            [allow]            +
Reverse charging                                [allow]            +
Maximum facility field length                   [109]              + #
Interrupt data size                             [32]               + #
[BOTTOM]

F1=Help          F2=Refresh          F3=Cancel          F4=List
F5=Reset         F6=Command           F7=Edit           F8=Image
F9=Shell         F10=Exit             Enter=Do
```

The AIX-generated command is:

```
chdev -l 'sx25a0' -a fs_mode='0'
```

3.7.1.2 Configuring Reverse Charging

Coming or outgoing calls can have a facility requesting that the receiving party pays for the call. Configuration options for the Reverse Charging facility are:

- Allow
- Allow only if not billed
- Forbid

The default is allow, but depending on your network subscription you can change it. For example, if we want to change Reverse Charging to forbid, we would select from SMIT main menu for a Co-Processor/2 adapter:

Devices

```
Gommunications
X.25 Co-Processor/2 or Multiport/2 Adapter
Adapter
Manage Device Drivers for X.25 Co-Processor/2 or Multiport/2
Manage X.25 LPP Device Driver
Manage X.25 Ports
Change / Show Characteristics of Port
Change / Show Packet Parameters
```

Or use the fastpath: `smit x25str_mp_csp_p_sel`

```
Change / Show Packet Parameters

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...37]                                     [Entry Fields]
R20 retransmission count restart timer          [3]                +#
R22 retransmission count reset timer            [3]                +#
R23 retransmission count clear timer            [3]                +#
R25 retransmission count window rotation timer [0]                +#
R27 retransmission count reject timer           [0]                +#
R28 retransmission count registration timer     [1]                +#

Packet-Level Features
*****
Fast select mode                               [allow]            +
D-bit                                           [allow]            +
Reverse charging                               [forbid]           +
Maximum facility field length                  [109]              +#
Interrupt data size                            [32]               +#
[BOTTOM]

F1=Help          F2=Refresh      F3=Cancel        F4=List
F5=Reset         F6=Command      F7=Edit          F8=Image
F9=Shell         F10=Exit        Enter=Do
```

The AIX generated command is:

```
chdev -l 'sx25a0' -a rev_charge='0'
```

3.7.1.3 Configuring Maximum Facility Field Length

You can configure the maximum facility field size in call and clear registration packets. For compliance with CCITT 1984 and later, the facility field length should be set to 109. Use a value of 63 for networks of prior to CCITT 1984. For example, if we want to change the maximum facility size, we select from SMIT for a coprocessor adapter:

Devices

- ~~C~~ommunications
 - ~~X~~.25 Co-Processor/2 or Multiport/2 Adapter
 - ~~A~~dapter
 - ~~M~~anage Device Drivers for X.25 Co-Processor/2 or Multiport/2
 - ~~M~~anage X.25 LPP Device Driver
 - ~~M~~anage X.25 Ports
 - ~~C~~hange / Show Characteristics of Port
 - ~~C~~hange / Show Packet Parameters

Or use the fastpath: `smit x25str_mp_csp_p_sel`

```
Change / Show Packet Parameters

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...37]                                [Entry Fields]
R20 retransmission count restart timer      [3]                + #
R22 retransmission count reset timer        [3]                + #
R23 retransmission count clear timer        [3]                + #
R25 retransmission count window rotation timer [0]                + #
R27 retransmission count reject timer       [0]                + #
R28 retransmission count registration timer  [1]                + #

Packet-Level Features
*****
Fast select mode                            [allow]            +
D-bit                                        [allow]            +
Reverse charging                            [allow]            +
Maximum facility field length                [63]               + #
Interrupt data size                         [32]               + #

[BOTTOM]

F1=Help          F2=Refresh          F3=Cancel          F4=List
F5=Reset         F6=Command           F7=Edit           F8=Image
F9=Shell         F10=Exit             Enter=Do
```

The AIX generated command is:

```
chdev -l 'sx25a0' -a fac_length='63'
```

3.7.2 More on Configuring Facilities

There are examples of defining facilities for TCP/IP and SNA in 7.3, "Requesting the Use of a Facility with TCP/IP" on page 145 and 8.3.6.4, "Defining X.25 Optional Facilities Profile" on page 168. Facilities formats can be found in Appendix E, "Facilities" on page 275.

Chapter 4. Tools

In this chapter, we will present the tools:

- xtalk
- x25mon
- x25status
- xroute

More details are available in *AIXLink/X.25 Version 1.1 for AIX: Guide and Reference* manual.

4.1 The xtalk Tool

Xtalk can be used as a tool or as a simple application.

4.1.1 Description

The xtalk tool enables you to initiate or receive calls over SVCs and communicate with remote hosts by exchanging messages and files. Xtalk is a tool that can be used as a sample application to verify a connection across an X.25 network. The xtalk tool can also be used to test an X.25 network before application programs such as TCP/IP, SNA/X.25, a terminal emulator or a PAD are configured.

Note: The xtalk tool is implemented over COMIO emulation and so can be used only over ports that have COMIO emulation configured.

4.1.2 Overview and Fast Path

In our test, we will use xtalk to:

- Start xtalk
- Define a remote host
- Establish a connection over an SVC
- Exchange messages

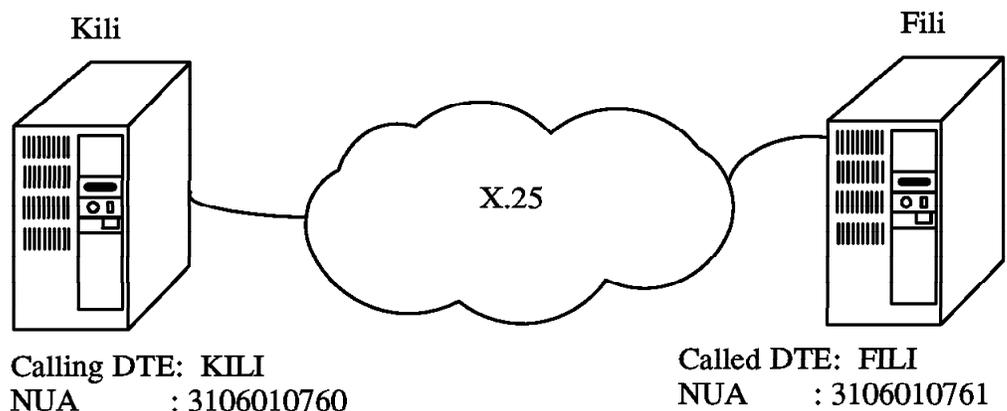


Figure 27. xtalk Test

Testing an SVC Communication with xtalk

1. On fili, load xtalk to listen to the IBMXTALK entry in the routing table:

```
# xtalk -s -l IBMXTALK
```
2. On kili:
 - Load xtalk:

```
# xtalk -s -l IBMXTALK
```
 - Use the ADD function to create an entry in the address list for fili; enter its NUA.
 - Use the TALK function to call fili.
3. On fili, accept the incoming call, and then press **F2**.
4. On kili, press **F2** and enter a message. It is transmitted on the network and then displayed on fili.

Notes:

- If you have only one system, open two windows (or two sessions). In the first, start one copy of xtalk that will correspond to the called machine and will listen for incoming calls whose characteristics are defined in the routing table entry IBMXTALK:

```
# xtalk -s -l IBMXTALK
```

In the other window, start xtalk without a routing parameter:

```
# xtalk -s
```

Ignore the warning message: You cannot receive incoming calls... by pressing **Esc**. You can now proceed from this window the same way as explained for the system kili.

- Xtalk cannot be used to test PVCs. Use the sample programs instead. (See Chapter 6, "APIs: COMIO, NPI and DLPI" on page 117.)

4.1.3 The xtalk Command

The xtalk command provides a panel-driven environment where you can make or receive a virtual call and then either talk to another person by typing messages on a panel, or send and receive files. You can store the details of the systems you want to communicate with under a symbolic name in an address list. The xtalk command allows you to view, change, add or delete entries of this list.

Syntax:

```
xtalk [ -n ] [ -l EntryName &] [ -q | -s ]
```

Flags:

- n Runs the xtalk process in the background to listen for incoming calls.
- l Listens for calls for the routing list entry specified by *EntryName*.
- q Displays the title panel for two seconds.
- s Does not display the title panel.

Related files:

/etc/xtalk.names	Contains the systemwide xtalk address list, which can be used by all users on the system to route outgoing calls.
\$HOME/xtalk.names	Contains the individual user's xtalk address list used to route outgoing calls.
/etc/xrt.names	Contains the X.25 routing list, used to route incoming calls with the COMIO interface.
/OtherDTENAME.log	Names the message logging file. OtherDTENAME specifies the address list name of the DTE with whom messages were exchanged, or the last eight digits of its network user address.

Introducing the xtalk Command

The xtalk command has several features that are analogous to using a telephone. To have a conversation or transfer files, one party must first make a call and the other party must receive and accept the call. Either party can end the call at any time. With xtalk you can:

- Listen for calls
- Make a call
- Receive a call
- Have a conversation
- Transfer files
- End a call

You can run the xtalk command in the background to listen for calls and notify you when they arrive. When a call arrives, you can start xtalk again in the foreground and then choose whether to accept or reject the call.

To have a conversation, each party types messages; both your messages and the other party's messages appear on your display. You can use a log file to record the messages you exchange during a conversation.

4.1.4 Testing Communication with xtalk

You can use the xtalk command to test X.25 communications; this test only applies for communicating RISC System/6000s.

4.1.4.1 Start xtalk on fili

First, start xtalk on fili to listen to the incoming call from kili.

```
# xtalk -s -l IBMXTALK
```

IBMXTALK is the identification of a record in the routing table giving the characteristics of the incoming calls that must be routed to the xtalk application. This name and the corresponding routing table entry are already defined in the system, and you do not have to worry about them. The function of the routing table and the way to customize it are defined in this chapter when we describe the use of the xroute tool.

After a while, this is what is displayed:

X.25 Communications XTALK					
TALK	ADD	BROWSE	CHANGE	DELETE	QUIT
You can make and receive calls					
Name	Port	NUA	Extension		

4.1.4.2 Start xtalk on kili

Start xtalk on kili the same way as on fili:

```
# xtalk -s -l IBMXTALK
```

Create an Entry for fili in kili's address list:: Each user you can talk to over an X.25 network has a network user address (similar to a telephone number). The xtalk command maintains a system address list that is available to all users, so users can make calls without knowing another user's network address. Each user can also choose to keep a local address list containing modifications and additions to the system address list. An entry for a user in your local address list overrides the entry for the same user in the system address list.

The following screen dialogs show how to add an entry for fili in kili's system address list, which you will use later to call it. This is like maintaining a phone directory for valid NUAs that a machine can dial to or can be dialed from.

The xtalk command will display a dialog screen, from which you can choose the **ADD** option to create a new entry in the address list:

X.25 Communications XTALK					
TALK	ADD	BROWSE	CHANGE	DELETE	QUIT
You can make and receive calls					
Name	Port	NUA	Extension		

X.25 Communications XTALK					ADD
Name	==>	<u>FILI</u>			
Port	==>	<u>x25s0</u>			
Address	==>	<u>3106010761</u>			
Extended Address	==>				
Facilities	==>				
Press Enter when finished					Esc = Cancel

In the same manner, you will add the kili entry to fili's address list so that the NUA in incoming calls from kili will be translated into the name kili and listed on the screen when a call occurs.

Calling fili:: You are now ready to make a call. By moving the highlighted zone with the vertical cursor movement keys, select the system you want to call.

From the menu screen, select **TALK**:

```

X.25 Communications XTALK
  TALK  ADD  BROWSE  CHANGE  DELETE  QUIT
-----
You can make and receive calls
-----
Name          Port      NUA          Extension
-----
Your address list:

System Address List:
  FILI      x25s0    3106010761
-----
                                F1 = Help          F3 = Shell
  
```

Then press **Enter** to select fili:

```

X.25 Communications XTALK                                     TALK
-----
Name/NUA          ==>FILI
Port              ==>x25s0
-----
Press Enter when finished                                     Esc = Cancel
  
```

Then press **Enter** again to make the call.

4.1.4.3 Exchanging Messages between kili and fili

On kili, the next screen shows that a call is in progress while on fili, a message shows that there is an incoming call.

kili

```

X.25 Communications XTALK                                     WAIT
-----
Making call to FILI ( 3106010761 on x25s0)
Press Esc to cancel the call
  
```

fili

```

X.25 Communications XTALK                                     INFORMATION
-----
Incoming call from KILI (3106010760 on x25s0)
  ACCEPT  REJECT
  
```

Choose the **ACCEPT** option from the menu on fili to start a conversation between the two machines. Once the call has been accepted, a menu screen appears that allows you to send and receive messages and files.

kili

```

X.25 Communications XTALK                               Message logging is OFF
-----
Connected to FILI (3106010761 on x25s0)
-----

TRANSFER FILE
BEGIN LOGGING
END LOGGING
CHANGE LOG FILENAME
QUIT CALL

F1 = Help      F2 = Messages      F3 = Shell

```

fili

```

X.25 Communications XTALK                               Message logging is OFF
-----
Connected to KILI (3106010760 on x25s0)
-----

TRANSFER FILE
BEGIN LOGGING
END LOGGING
CHANGE LOG FILENAME
QUIT CALL

F1 = Help      F2 = Messages      F3 = Shell

```

The F2 key is a toggle key between a dialog screen to enter and receive messages, and the menu shown above. We press F2 on kili to send the *Hello world* message to fili. After having been transported on the X.25 network, the message is displayed on fili:

kili

```

X.25 Communications XTALK                               Message logging is OFF
-----
Connected to FILI (3106010761 on x25s0)
-----
Hello World!

```

fili

```

X.25 Communications XTALK                               Message logging is OFF
-----
Connected to KILI (3106010760 on x25s0)
-----
Hello World!

```

If you have not successfully completed this test, go to Chapter 9, “X.25 Problem Determination” on page 181.

Transferring Files: To transfer a file with xtalk, one user sends the file, and the other can choose to accept or reject it. If a file of the same name already exists on the recipient’s system, the recipient can choose to append or overwrite the existing file, or save the transferred file under a new name.

4.1.5 Requesting the Use of a Facility with xtalk

In the following example, we will explain how to reverse the charges during a virtual call made by using xtalk. (We assume that the called number has subscribed to the reverse charging acceptance facility.)

First, we look at E.1, “Supported Facilities for X.25 Communications” on page 275, which explains how to code a facility request. The code for reversing the charges is:

Value	Description
0x01	Reverse charging or fast select
0x01	Reverse charging requested

Then, after having started xtalk, use the CHANGE panel to add this facility request in the definition of the X.25 host fili:

X.25 Communications XTALK		CHANGE
Name	==>FILI	
Port	==>x25s0	
Address	==>3106010761	
Extended Address	==>	
Facilities	==>0101	
Press Enter when finished.		Esc = Cancel

Now, start `x25mon -p -n sx25a0` in another window and make a call to fili using the call function of `xtalk`; here is what we will see on the trace:

```
sx25a0 PS 0x0014 CALL      dN 1a:10 1f:2   1d:1 AA31060107613106010760020101FD
```

If we compare this to the trace without the facility request:

```
sx25a0 PS 0x0014 CALL      dN 1a:10 1f:0   1d:1 AA31060107613106010760000FD
```

we can see that `xtalk` has generated a facility length of 02 and has inserted it with the facility request 0180 between the NUA block and the CUD.

4.2 The x25mon Tool

The `x25mon` command replaces the `xmonitor` command used in previous AIX X.25 products.

4.2.1 Description

The `x25mon` tool traces the frame or packet traffic of an X.25 port, and the report is sent to `stdout`.

When using the `x25mon` application, the display that is used to show the data must keep up with the rate of data being received. If running on a slow display, such as an ASCII terminal, the `x25mon` output should be directed to a file. This keeps system resources from being depleted and terminating the data transfer.

With AIXLink/X.25 Version 1.1.3, two new `x25mon` flags, `-i` and `-d`, are offered and the length of data is indicated via an `l`: in the trace for both packet and frame layers.

4.2.2 Using the x25mon Tool

Running an X.25 line trace via `x25mon` on a system with heavy X.25 traffic can negatively impact X.25 throughput due to additional processing overhead and adapter memory resource constraints. Thus, tracing is recommended for problem isolation activity rather than as a matter of routine.

Prior to the X.25 Version 1.1.3 enhancement, all packet and frame data was saved in the `x25mon` trace information and trace packets were discarded, when necessary, to keep up with the incoming traffic. When this happened, the trace output was flagged with:

```
PK_LMS_IND  1:0 missed: xx
```

where `xx` is the number of packets missed.

With X.25 Version 1.1.3, users can lessen the trace burden by saving only a portion of the contents of the data packets. Two new flags, `-i` and `-d`, allow users to specify the maximum number of frame and packet level data bytes to save in the trace. In addition, all trace data will be captured, regardless of network load. That is, the X.25 device driver will no longer sacrifice trace data in order to service incoming traffic.

The `-i` flag indicates the info frame trace size. Valid value ranges are from 0 to 5003. If this flag is not specified, 512 bytes is assumed. The `-d` flag sets the data packet trace size. Valid value ranges are from 0 to 4096. If not specified, all data is captured in the trace.

The following command will trace the packet and frame layers of the `sx25a0` port, showing the first 5 bytes of data in the packet layer and the first 10 bytes of data within the frame layer. The user will know that the trace shows only a portion of the actual content because bytes beyond the number specified will be annotated with: `".."`.

```
x25mon -p -d 5 -f -i 10 -n sx25a0
```

Limiting the number of bytes saved in the trace is extremely important, as it affects both performance and memory resource utilization. It is highly recommended that frame traces be limited to less than 512 bytes. Running with larger traces may result in severe adapter memory constraints.

Generally, the `x25mon` tool is issued as:

```
x25mon -p -f -n <portname>
```

This will trace both the frame and packet levels. We will show `x25mon` as separate frame and packet traces to better identify the flags.

4.2.2.2 Frame Level

First, open a new window or a new session and start x25mon at the frame level with: x25mon -f -n sx25a0. You should normally get the following:

```
X.25 Monitor sx25a0
10:34:26 sx25a0 FR 0x0000 INFO a:1 p:0 ns:6 nr:0 50040BAA31060107603106010
10:34:26 sx25a0 FS 0x0000 INFO a:3 p:0 ns:0 nr:7 10040F
10:34:26 sx25a0 FR 0x0000 INFO a:1 p:0 ns:7 nr:1 100400544556454E4E4543
10:34:26 sx25a0 FR 0x0000 INFO a:1 p:0 ns:0 nr:1 5004025343524156494720494
10:34:26 sx25a0 FS 0x0000 INFO a:3 p:0 ns:1 nr:1 100441
10:34:26 sx25a0 FR 0x0000 INFO a:1 p:0 ns:1 nr:2 1004130000
10:34:26 sx25a0 FS 0x0000 INFO a:3 p:0 ns:2 nr:2 100417
10:34:27 sx25a0 FR 0x0000 RR a:3 p:0 nr:3

      1      2      3      4      5              6              7
```

The keys to understand this trace are the following:

- Column one:
 - Time stamp
- Column two:
 - X.25 port traced
- Column three:
 - FS Data frame type sent
 - FR Data frame type received
- Column four:
 - Indicates the logical channel number
- Column five:
 - Specifies the frame type
- Column six:
 - Specifies the different flags of the frame level:
 - a : Gives the frame address DCE(3) or DTE(1)
 - p : Gives the setting of the poll/final bit
 - n : Gives the settings of the send and receive counters
- Column seven:
 - Information field of the frame level

4.3 The xroute Tool

The xroute tool is used to manage the COMIO routing table.

4.3.1 The Routing Table

All incoming calls made over COMIO emulation are routed to the applications by the X.25 device driver using the X.25 routing table. If the incoming call matches the criteria defined in this table for a specific application, the call will be routed to that application. A routing table is not needed if you are using only PVCs. The applications not based on COMIO but on the NPI API get their routing with the STREAMS implementation (N_BIND_REQ primitive).

An application listening for an incoming call is associated with an entry name in the routing table. This entry specifies the criteria that must be satisfied for the application program to receive an incoming call.

The call user data (CUD) field of the call packet is generally used for routing but other conditions may also be tested, such as:

- The network address of the caller
- A subaddress in the called DTE address
- The attachment that has received the call

The xroute tool is used to manage the COMIO X.25 routing table so it is usable only for xtalk, SNA based applications and all applications written over the previous X.25 API library or over the previous X.25 device driver supported in AIX V3. For applications that do not use the COMIO emulation interface, like TCP/IP and NPI, the xroute tool is unnecessary. So the xroute tool works with X.25 ports that have COMIO emulation configured.

4.3.2 Updating the X.25 Routing Table with the xroute Tool

The routing table is stored in the /etc/xrt.names file. A default file is provided that may be updated with the xroute command.

To use xroute, log on as root and type:

```
# xroute -s
```

You will get the X.25 Communications XROUTE menu:

X.25 Communications XROUTE				
ADD	BROWSE	CHANGE	DELETE	QUIT
Entry name	Port	Called Subaddress	CUD	
IBMELLC	*	*	C6	
IBMELLC	*	*	CE	
IBMP SH	*	*	C2	
IBMP SH	*	*	CA	
IBMQLLC	*	*	C3	
IBMQLLC	*	*	CB	
IBMSAMP	*	*	*	
IBMTCPO	x25s0	*	CC	
IBMXTALK	x25s0	*	FC	
IBMXTALK	*	*	FD	
		F1 = Help	F3 = Shell	

This is the default routing table with predefined entries for SNA, TCP/IP, the sample program and xtalk. The first six entries correspond to X.25/SNA protocols:

- IBMELLC identifies the SNA-Enhanced logical link control used for peer-to-peer
- IBMPSH identifies the physical services header LLC used with the IBM-5793 network interface adapter (NIA)
- IBMQLLC identifies the SNA-qualified logical link control

The two CUDs associated with each of these protocols identify their version, 1980 or 1984/88.

To change a routing list entry, move the cursor to the entry name you would like to change and type c for change.

The following screen will appear:

X.25 Communications XROUTE		CHANGE
Entry name	==>IBMXTALK	
User name	==>*	
X.25 port	==>*	
Calling Address	==>*	
Called Subaddress	==>*	
Calling Address Ext	==>*	
Called Address Ext	==>*	
Call User Data	==>FD	
	==>	
	==>	
	==>	
Priority (1-3)	==>1	
Action (R,F)	==>R	
Press Enter when finished.		Esc = Cancel
F1 = Help		F3 = Shell

The fields are:

- Entry Name** Name of the entry that will be used by the listening program to refer to this set of criteria.
- Call User Data** This is a part of the data received in an incoming call packet. It can be used for any purpose, but it often specifies the protocol. Only the first 64 bytes of the CUD can be used by xroute.
- User Name** This is the login name of the user who is allowed to start applications listening for incoming calls associated with this entry. An * means that any user can listen for calls using this entry. A user whose login name does not match this entry is not allowed to listen for calls corresponding to it.

For example, your routing list entry contains the user name *marcel*:

```

X.25 Communications XROUTE                                     CHANGE
-----
Entry name          ==>IBMXTALK
User name           ==>marcel

```

If you try to start the command `xtalk -l IBMXTALK` as the user `root`, you will get an error message such as:

```

X.25 Communications XTALK                                     ERROR 121
-----
C10 Status 79 - X25_AUTH_LISTEN
You cannot listen to this name, because the routing list entry has
a userid which excludes the user running the application.

```

- X.25 Port** The name of the X.25 port associated with the application for which the call is intended. This can be set to `*` to indicate any port.
- Calling Address** The application will receive incoming calls only from this calling address. An `*` at the end of this entry indicates that any digits are acceptable in the remainder of the address. Another NUA trying to call this application will be rejected.
- Called Subaddress** You can add additional digits to the end of an NUA up to the 15-digit limit. This subaddress can be used to route a call internally within a node. For example, calling 310601076101 may be routed to fili with a subaddress of 01. This parameter can be set to `*` to indicate any subaddress.
- Calling Address Ext** In X.25 (1984), it is possible to specify up to 40 additional digits of address. This can be set to `*` for any address extension.
- Called Address Ext** See calling address extension.
- Priority (1-3)** If two applications need to listen to the same routing information, the priority field specifies who should receive the call if both are listening. For example, IBMSAMP is defined as listening to any condition `*` in all fields but with the lowest priority (3). Thus, a background daemon could be written that listened to IBMSAMP and logged any incoming call that is not received by another application.
- Action (R,F)** This can be `R` to reject the incoming call, or `F` to forward the incoming call. These conditions apply when no application is listening for the routing list entry which is the best match for the incoming call.

If this parameter is set to `R` (reject) and your application is not running when you try to establish a call, the call will be cleared with cause 0 and diagnostic 0. If it is set to `F` the next best match is selected, and so on until an entry that specifies `R` is selected and the call is rejected.

If a call does not match any entry in the routing table, it is cleared with cause 0, diagnostic 0.

4.4 Other X.25 Commands

The X.25 LPP offers a set of specific commands to manage X.25 configurations on the RISC/6000. We present in this section only the purpose of each command. More details on the syntax and the use of these commands are available in Appendix A of the *AIXLink/X.25 Version 1.1 for AIX: Guide and Reference* manual. Other commands include:

x25ip The x25ip command updates or displays translation information in the IP/X.25 translate table. The x25ip -h rouse -s command displays:

IP/X.25 Switched Virtual Circuit Configuration

IP host name	rouse
Remote DTE address	3106010762
Rcv packet size	
Xmit packet size	
Rcv window size	
Xmit window size	
Closed user group index	
Closed user group index (outgoing)	
RPOA selection	
User-defined facilities	
Call user data	cc

lsx25 The lsx25 command displays the configuration of the X.25 support on the system:

```

*****
* Configuration report for X.25 LPP ports configured *
*****
Machine: strider

*****
* Report by slot number - bus 0
*****
Slot 1 - empty
Slot 2 - empty
Slot 3 - empty
Slot 4, ampx0      X.25 Co-Processor/2 Adapter
Slot 4, twd0      X.25 Streams driver
                  Physical port 0 is x25 port sx25a0 -31060760
Slot 5, tok0      Token-Ring High-Performance Adapter
Slot 6, gda0      Grayscale Graphics Display Adapter
Slot 7 - empty
Slot 8, scsi0     SCSI I/O Controller

*****
* Report by X.25 port's logical location *
*****
X.25          Logical   Logical
Port          Driver    NUA        COMIO      TCP/IP      board      port
sx25a0       twd0        31060760   x25s0      xs0         0         0

*****
* Report by X.25 port's physical location *
*****
X.25          Phys.
Port          Driver  Adapter  Slot  Port  Interface
sx25a0       twd0   ampx0    4    0    cable selectable

*****
* Report by NUA *
*****
NUA          X.25 Port
31060760     sx25a0

*****
* Report of COMIO emulators *
*****
COMIO        X.25 Port
x25s0        sx25a0

*****
* Report of X.25 TCP/IP (xs) interfaces *
*****
TCP/IP        addr          X.25 Port
xs0           10.2.0.1      sx25a0

```

x25status A new command, x25status, has been added to allow customers to view the current status of their ports. It will display the current state of the packet layer on all defined ports. The following is a list of possible packet layer states:

- Disconnected
- Restart in Progress
- Registration in Progress
- Line Cleanup in Progress
- Packet Layer Connected

The packet layer states actually reflect the state of the link between the DTE and DCE.

Each time a virtual circuit is established, the Active SVCs or Active PVCs count is incremented depending on the type of virtual circuits the port uses.

Link status for kili

Port	Packet State	Active SVCs	Active PVCs
sx25a0	PACKET LAYER CONNECTED	1	0
sx25a1	PACKET LAYER CONNECT		
sx25a2	DISCONNECT		

In this example, sx25a0 has a link with the DCE and an established virtual circuit. Sx25a1 has a link, but a virtual circuit has not been established, and port sx25a2 is not active.

sx25debug This command is a debugging aid. It has two basic functions: to read debug and error messages from the microcode and to display them to stdout.

1. Use the `-b` flag to specify a logical board number. This number can be attained by executing the `lsx25` command.
2. When the `-i` flag is specified, this is interactive mode and allows characters to be entered from the keyboard. In this mode the microcode is polled to ensure it is operational. The response of "pse_gdebug: Alive" indicates all is OK.

You must have root authority in order to run this command.

mksx25 The `mksx25` command initializes the attributes of the X.25 port specified.

chsx25 The `chsx25` command re-initializes the attributes of the X.25 port specified.

rmsx25 The `rmsx25` command removes an X.25 port.

mkpvc The `mkpvc` command creates or modifies a non-default permanent virtual circuit (PVC) on an X.25 port.

lspvc The `lspvc` command displays the non-default permanent virtual circuit (PVC) attribute information for an X.25 port.

4.5 Configuration Tools

The AIX X.25 LPPs include shell scripts to backup, remove and restore your X.25 configuration.

4.5.1 The backupx25 Script

This script is used to save the configuration of the X.25 LPP to a set of files. Those files can be used later to reconfigure the system. The files contain information to restore:

- Adapter configuration
- Device driver configuration
- Port configuration
- COMIO configuration
- TCP/IP configuration
- PAD profiles

The script is executed by entering:

```
/usr/bin/backupx25
```

Use the `-d` flag to specify the directory for the files. The default is the current directory.

The `-v` option, for verbose mode, will display key messages during execution.

4.5.2 The restorex25 Script

This is a shell script to restore the configuration of the LPP from the backup files created by the backupx25 script.

The script is executed by entering:

```
/usr/bin/restorex25
```

Use the `-d` flag to specify the directory for the files. The default is the current directory.

The `-v` option, for verbose mode, will display key messages during execution.

4.5.3 The removex25 Script

This is the script to remove the configured instances related to the X.25 LPP.

The script is executed by entering:

```
/usr/bin/removex25
```

The flags for the script are:

- q assume "yes" to removal
- v verbose
- d directory of files from backupx25
- f remove even if no backup is found

Chapter 5. Packet Assembler Disassembler (PAD)

In this chapter, we will present the PAD implementation in the AIX/6000 X.25 LPPs. For further information, see *AIXLink/X.25 Version 1.1 for AIX: Guide and Reference Manual*.

5.1 Using a PAD

X.25 defines the relationship between a DTE node and a network's DCE. The CCITT also defines the attachment of asynchronous start-stop devices to the PSDN in recommendations X.3, X.28 and X.29. (See 5.2.1, "X.3, X.28 and X.29 Standards" on page 93.) The asynchronous device is typically a low-cost terminal such as an IBM 3151 or 3152. As these terminals lack the intelligence of a full-function node on the network, they rely on a device called a Packet Assembler Disassembler (PAD).

A PAD is a protocol converter interfacing asynchronous terminals with an X.25 network or an X.25 network with applications written for asynchronous terminals. Low-cost asynchronous terminals are connected by the public switched telephone network to a local public outgoing PAD, often simply referred to as a PAD. The PAD takes the ASCII data stream coming from the terminal, buffers it, converts it into a properly formatted X.25 packet and sends it over the X.25 PSDN, addressed to the desired DTE.

Data from the terminal that has been divided into packets and shipped by a PAD to the DTE node must be received by some incoming PAD software (also referred to as an X.29 interface). This incoming PAD unpacks the data and passes it to the application (typically, the TTY driver) for processing, as if it were coming from directly attached asynchronous terminals. The other way, data coming from the application is first put into packets by the X.29 PAD and transferred over the network. When the PAD receives a packet addressed to the terminal, it reverses the process. This presents the data in the form that the terminal can accept.

The computer industry has expanded the definition of PAD to include protocol conversion between X.25 and various teleprocessing protocols. Some users talk about SDLC PADs, BSC-3270 PADs and so on. Users must write programs to support incoming packets from these PADs on AIX hosts.

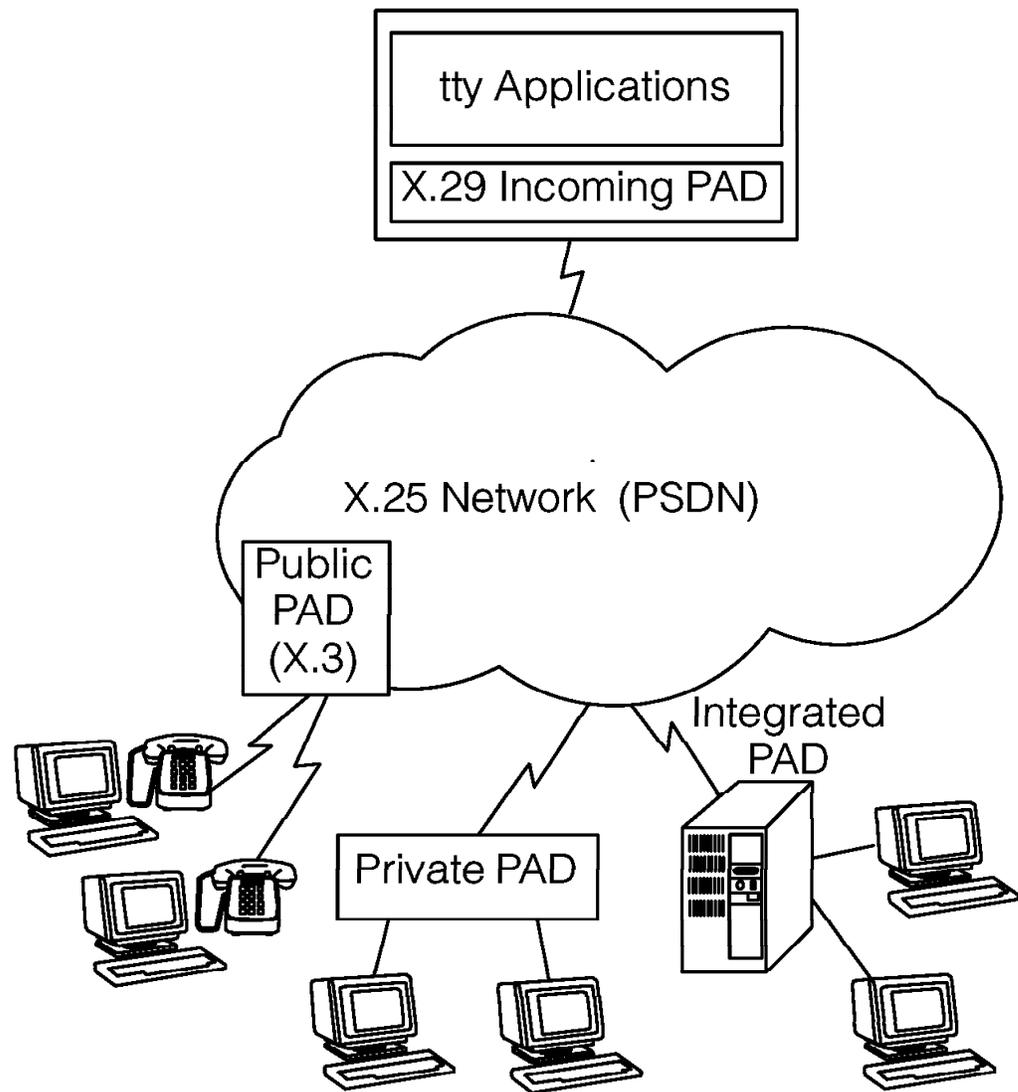


Figure 28. The Various Types of PADs

5.2 Function of the PAD

Basic functions include:

- Assembly of characters into packets (asynchronous-to-X.25)
- Disassembly of the packet's user data field (X.25-to-asynchronous)
- Handling of virtual call setup, clearing, resetting and interrupt procedures
- Generation of service signals
- Forwarding packets when the proper conditions exist
- Transmitting data characters, including start, stop and parity elements, as appropriate to the start-stop terminal
- Handling a break signal from the start-stop terminal

- Editing of PAD command signals
- Setting and reading the current value of PAD parameters

Optional functions include:

- A mechanism for selection of a standard profile
- A mechanism for automatic detection of data rate, code, parity and optional characteristics
- A mechanism for a remote DTE to request a virtual call between the start-stop mode DTE and another DTE

5.2.1 X.3, X.28 and X.29 Standards

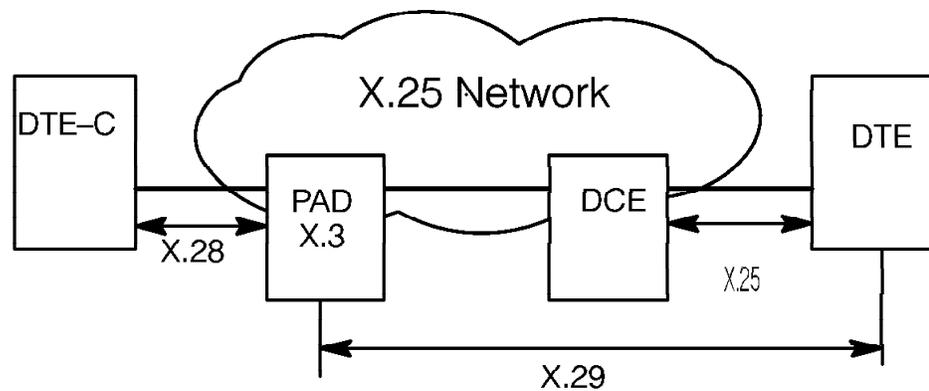


Figure 29. X.3, X.28 and X.29 Standards

The above figure illustrates the components and standards used to attach an asynchronous terminal to a remote host via X.25.

The asynchronous terminal DTE-C attaches to a packet assembler/disassembler (PAD), the function of which is to handle the mapping of characters from the terminal side into X.25 packets on the network side.

X.3 defines a set of parameters that the PAD uses to identify and control the attached terminals. A complete set of parameters is called a profile, and each DTE-C has its own profile selected or set for use with the PAD.

X.28 defines the control procedures used to establish the physical connection to the PAD, the commands the user sends to the PAD and the service signals sent by the PAD to the terminal user. Simply, X.28 defines how a terminal user can control the X.3 PAD parameters. It specifies the command and response formats, as well as status indicators.

X.29 defines the way in which a PAD and a remote DTE (or another PAD) exchange control messages on an X.25 virtual circuit. The messages are formatted as special X.25 packets called qualified data packets. For example, the packet mode DTE may use an X.29 message to change one of the internal X.3 parameters in the PAD. X.29 messages will only be effective when an X.25 call is established. The user at the terminal is not explicitly aware of the X.29 communication between the PAD and the remote DTE. The user at the terminal

logs into the host DTE in the same way as the user of an ASCII terminal locally attached to the same host.

5.2.2 Outgoing PADs

PADs used to establish connections from asynchronous devices are outgoing PADs. These PADs can be provided by different sources and may be physical devices or emulated by software.

5.2.2.1 Public PAD

A public PAD is a PAD provided by the Post Telegraph Telephone (PTT) administration that you can access by a local call on the public switched network. Once your terminal is connected to the public PAD, you enter the NUA of your host DTE and the PAD will establish the communications.

5.2.2.2 Private PAD

A private PAD is a hardware device with several asynchronous connectors and an X.25 attachment. This commercially available device can be connected to a public or private X.25 PSDN and has the same capabilities as a public PAD.

5.2.2.3 Integrated PAD Software

Integrated PAD software is a software product emulating a PAD. You can install it on a computer supporting an X.25 connection, (xspad does this function.) It generally provides an asynchronous terminal emulator or can use existing ones such as ATE or CU. With the X.25 LPP, the xspad function performs this role. Integrated PADs are software emulations of private PADs.

Figure 30 compares a public PAD and an integrated private PAD.

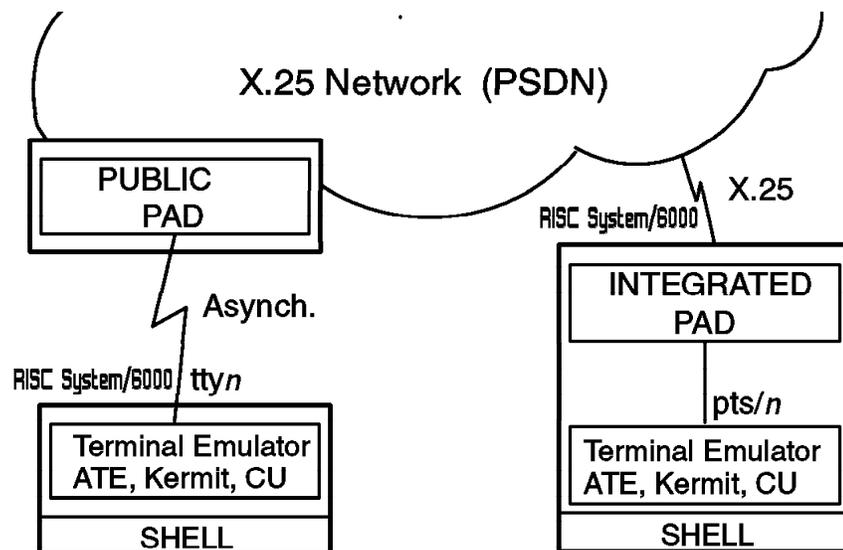


Figure 30. Public PAD vs. Integrated Private PAD

5.2.3 Incoming PADs

Private PADs are often reversible and may be used to provide X.25 communications capability between devices with only asynchronous attachments.



Figure 31. Private PADs Used to Carry Asynchronous Connections over a PSDN

An incoming PAD can be implemented either as an X.29 daemon or as an X.29 device driver. The X.25 LPP software uses the X.29 daemon implementation and STREAMS modules.

5.3 CCITT PAD Interfaces

The PAD performs a number of functions and exhibits operational characteristics. Some of the functions allow the terminal, the X.25 host or both to configure the PAD so that its operation is adapted to the terminal characteristics and, if possible, to the application.

5.3.1 PAD Parameters

The operation of the PAD depends on the value of the set of internal variables called PAD parameters. An independent set of parameters exists for each start-stop mode DTE. The current value of each PAD parameter defines the operational characteristic of its related function.

Normally, the values of the parameters are chosen by the selection of one of the available profiles. Subsequent changes for a given terminal session are usually done under the control of the session's tty subsystem on the host PAD. When the application being run requires a change to the terminal characteristics, it will modify the session's stty structure which will in turn cause the X.29 protocol to issue a change to the session's X.3 parameters held in the terminal PAD.

The PAD parameters are listed in F.1, "PAD Parameters" on page 287. They are referenced as PAR 1 to PAR 22. Here are descriptions of some of the parameters:

PAR 1 - PAD recall

This parameter specifies the character that a user can type at the terminal to interrupt the data flow with the X.25 host in order to send commands to the PAD. This parameter operates like a break key.

Possible values are:

- 0 Data flow may not be interrupted; the user may not enter PAD commands.
- 1 The escape character is CTRL-P.

32-126 Value of the PAD recall character in decimal.

Note: If this parameter value is changed to 0, the user will no longer be able to change PAD parameters.

PAR 2 - echo

This parameter specifies whether characters received from the terminal are to be transmitted back to the terminal.

Possible values are:

- 0** No echo by the PAD
- 1** Local echo provided by the PAD

When working in line mode, this parameter should be set to 1 to get an echo from the PAD, and the echo provided by the host should be disabled with the `stty -echo` command.

PAR 3 - data forwarding signal

This parameter specifies the characters that will cause the data in the buffer to be forwarded over the network to the host, as defined in X.25.

Possible values are:

- 0** Data forwarding not controlled by a character
- 1** Data forwarding controlled by alphanumeric characters (A-Z, a-z, 0-9)
- 2** Data forwarding on receipt of CR
- 4** Data forwarding on receipt of ESC, BEL, ENQ, ACK
- 8** Data forwarding on receipt of DEL, CAN, DC2
- 16** Data forwarding on receipt of EXT, EOT
- 32** Data forwarding on receipt of HT, LF, VT, or FF.
- 64** Data forwarding on receipt of any character from columns 0 and 1 of the ASCII code page

If this parameter is set to 0, every character typed at the terminal is sent by the PAD in an individual packet. For the other settings, the PAD acts as a buffer and sends a packet only when the specified character is entered. If the terminal is in line mode, for example entering UNIX commands, this parameter may be set to 18. Thus, packets will only be sent to the host on pressing Enter, Ctrl-c or Ctrl-d.

PAR 4 - idle timer delay

This parameter specifies a timeout period for the reception of characters from the terminal. After this timeout expires, characters already received by the PAD are formatted into a packet and sent across the X.25 network to the host.

Possible values are:

- 0** No timeout period is used
- 1-255** Timeout period expressed in units of 0.05 seconds

If using line mode, this parameter would be set to 0. When a non-zero value is used, this parameter can improve the performance of applications transferring data from the terminal to the host, as the time delay between the PAD receiving two consecutive characters is lower than the timeout period. The PAD would accumulate the data

and only send it when it has a full packet or when the file transfer application has stopped sending characters.

PAR 20 - echo mask

This parameter, used in conjunction with PAR 2, controls which characters are echoed back to terminal, if local echo is enabled.

Possible values are:

- 0** All characters are echoed. There is no mask.
- 1** CR character is not echoed.
- 2** LF character is not echoed.
- 4** VT, HT, FF characters are not echoed.
- 8** BEL and BS characters are not echoed.
- 16** ESC and ENQ characters are not echoed.
- 32** ACK, NAK, STX, SOH, EOT, ETB, ETX characters are not echoed.
- 64** Editing characters designated by parameters 16-18 (described in Appendix F, "PAD Parameters and Commands" on page 287) are not echoed.
- 128** No echo of all other characters in columns 0 and 1, except Delete or those characters mentioned above.

These parameters can be changed either from the terminal or by the X.29 incoming PAD. The terminal user will use PAD commands (see 5.3.2, "PAD Commands") to display or alter these parameters. The parameters can be stored in a PAD profile, a set of predefined parameter values stored in the PAD. The incoming PAD will use the X.29 protocol to change the parameters of the outgoing PAD.

5.3.2 PAD Commands

When connected via an X.3/X.28 terminal PAD session, various commands can be issued to the PAD. Some of these commands can only be issued after a connection has been established with the remote X.25 host. Depending on the PAD's profile, the commands understood by the PAD are either based on the CCITT standard, using parameter numbers, or interpreted in advanced mode, using parameter names.

The start/stop mode DTE user may interact with the PAD using *PAD commands*. These commands are provided for:

- Establishing and clearing of virtual calls
- Selecting PAD profiles
- Displaying the PAD parameter values
- Changing the PAD parameter values
- Sending an interrupt or a reset packet

To enter a PAD command at the terminal, the user enters the command at the PAD prompt. If the user is not connected to a remote host, the PAD prompt, which is the asterisk character (*), will be displayed. To get the PAD prompt when connected to a remote session, press **Ctrl-p**, the default character for PAR 1.

From the prompt, the following commands can be issued:

BREAK Send a break character to the PAD.

CALL	Establish a connection to the remote X.25 host, for example call 3106010761.
CLEAR	Clear the connection with the remote X.25 host. The clear is sent from the PAD immediately. (See the iclear command.)
HELP	Request help text. Each PAD command, parameter and profile has help text associated with it, for example help call. To list the help topics, use help list. To change the language of the help text, use the language command.
ICLEAR	Send an invitation to clear to the remote X.25 DTE. This allows the remote host to send any pending data before clearing the call.
INTERRUPT	Send an interrupt packet. The packet contents cannot be user-specified.
LANGUAGE	Set the language for PAD help text to English, French or Spanish, for example language french.
PAR	Display the PAD parameter values. PAR? Displays all parameters. PAR? 2, 3 Displays parameters 2 and 3. In advanced mode, read or parameter can be used instead of par.
PROFILE	Display which profiles are available or change the profile. Enter profile followed by the profile number, such as profile51.
READ	See par.
RPAR/RREAD	Read the remote PAD parameters in implementations where a remote PAD DTE is supported.
RSET/RSETREAD	Set the remote PAD parameters in implementations where a remote PAD DTE is supported.
SET	Sets one or more of the PAD parameters: set 2:1, 14:2 In advanced mode set echo:1, lfpad:2 Note: The default profile does not allow the parameters to be changed locally, as they are being controlled remotely through the TTY subsystem and the stty command. Xspad does not require the use of the set command or any of the remote PAD commands (rread, rpar, rset and rsetread).
STATUS	Indicate whether a connection to the remote X.25 host is active. Returns ENGAGED if it is active or FREE if the connection has not been established.

5.3.2.1 Establishing Connections

To establish a connection with a remote X.25 host, a call must be made to it. This call must contain the remote DTE's address and any call user data or facilities being requested.

Facility code	Function requested	Value (text that follows the facility code)
C	Charging information	
D	Call User Data	ASCII CUD string to be added to the standard PAD call CUD. Must be the last part of the command string
E	Called Address Extension	Extended address
F	Fast select with no restriction	
G	Closed User Group	CUG number
N	Network User Identification	NUI string
Q	Fast select with restrictions on response	
R	Reverse charging	
-	Facility end marker	End of facilities marker, to then allow CUD

Calls can be placed from the command line, using `xspad -a`, or from the PAD prompt. For example:

To establish a call to NUA 3106010761:

```
call 3106010761
```

To also request fast select and reverse charging:

```
call 3106010761 F, R-
```

To also have ASCII base CUD user1:

```
call 3106010761 F, R- D user1
```

After a call has been made and accepted by the remote host, a connection has been established. The terminal is now under the control of the tty subsystem of the remote host. Typically, a login panel will be presented, and after the user has logged in, it can be used in the same way as other attached ttys.

5.3.2.2 Terminating Connections

If for some reason the call establishment is rejected by the network or the remote host, a clear packet will be received with a message similar to:

```
CLEAR DTE 0 241 - Call cleared, by remote device, data may be lost
```

This type of message will also be received on termination of the login shell used to log in to the remote X.25 host. In these cases, the session was terminated above the PAD layer and was unexpected, so the PAD could not determine if the

termination was user initiated. If it had been user initiated, no data would have been lost.

Another way for a user to initiate a `clear` is to use the `iclear` command from the PAD prompt. `Iclear` causes the local terminal PAD to send a request to the host PAD to clear the connection. The host PAD would then issue a `clear` but would disregard any applications that might be running under this login. As the host PAD software issued the clear, the diagnostics would reflect it as being an expected clear and result in a message similar to:

```
CLEAR PAD 0 0 - Call cleared, remote request
```

In this example, the additional text shown is given only when advanced mode is enabled through the profile; otherwise, only the base clearing reason is given.

5.3.2.3 Clearing Codes

When a call is cleared, the information passed in the X.25 packet is displayed. If advanced mode has been selected, a more-detailed explanation of the `clear` is given.

Code	Explanation
OCC	Remote DTE busy
NC	Network congestion
INV	Invalid facility
NA	Access barred
ERR	Local procedure error
RPE	Remote procedure error
NP	Number not assigned
DER	DTE out of order
PAD	DTE clearing
DTE	DTE device clearing
RNA	Reverse charging rejected
ID	Incompatible destination
SA	Ship cannot be contacted
FNA	Fast select rejected
ROO	Cannot route as requested

Appendix F, "PAD Parameters and Commands" on page 287 lists these commands.

5.3.3 PAD Profiles

A *PAD profile* is a predefined set of PAD parameters. The terminal PAD in the X.25 LPP implements four standard, predefined profiles:

- 50** Standard profile with minimum textual information
- 51** Standard profile with extended textual information

- 90 CCITT simple profile
- 91 CCITT transparent profile

AIXLink/X25 provides three other predefined profiles.

- 10 Default for incoming PAD
- 20 Default for outgoing PAD
- 30 Default for PAD printing
- 91 CCITT transparent profile

Table 10. Parameter Settings for Predefined PAD Profiles

PAR	50	51	90	91
1	1	1	1	0
2	1	1	1	0
3	2	2	126	0
4	0	0	0	20
5	0	0	1	0
6	5	16	1	0
7	21	21	2	2
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	14	14	14	14
12	1	1	1	0
13	5	5	0	0
14	0	0	0	0
15	1	1	0	0
16	8	8	127	127
17	3	3	24	24
18	18	18	18	18
19	2	2	1	1
20	64	64	0	0
21	0	0	0	0
22	0	0	0	0

A particular profile can be selected using the prof PAD command. To list the available profiles, use the help prof command. For AIXLink X.25 V1.1.3, the default profile is in /etc/sx25pad/x28parm and can be seen as profile #20, currently named titn_test.

<i>Table 11. Parameter Settings for Default PAD Profiles</i>			
PAR	10	20	30
1	1	1	0
2	0	1	0
3	0	2	126
4	1	0	1
5	0	0	2
6	16	16	0
7	4	21	2
8	0	0	0
9	0	0	0
10	0	0	0
11	n/a	14	n/a
12	0	1	1
13	0	5	0
14	0	0	0
15	0	1	0
16	8	8	127
17	24	3	24
18	18	18	18
19	2	2	1
20	64	64	0
21	0	0	0
22	0	0	0

If your system is used as an integrated PAD, profiles can be defined to set the PAD parameters to commonly used values. These profiles can be used to set PAD parameters for outgoing calls or those of a remote PAD receiving incoming calls.

The profiles are stored as an ASCII file, `/etc/sx25pad/x28parm`, that can be modified with a text editor.

5.4 Configuring the PAD

Once the PAD software is enabled on the system, it can be used as an outgoing terminal PAD, supporting local terminal users, or as an incoming host PAD, working with remote terminal PADs. Both modes of operation can be run at the same time.

For use as a terminal PAD, each terminal or terminal emulator that requires a PAD session would execute the `xspad` command. Xspad connects the terminal to an application that provides the standard X.28 interface. See 5.6, "PAD Usage" on page 112 for more details on the use of terminal PAD sessions.

When the system is to be used as a host PAD, then no user interaction takes place on the system itself. Once the system is configured to enable the PAD,

users can connect from terminals connected to remote terminal PADs. The system running the host PAD will allow login and applications as for any other type of remote ASCII terminal.

5.4.1 PAD for AIXLink/X.25 1.1.2 Running on AIX V.4

There are three basic PAD features that are controlled by four PAD configuration files located in the `/etc/sx25pad` directory. These configuration files allow changes to be made to the environment for *incoming* PAD calls only.

- Automatic login of incoming calls based on calling address
- Automatic selection of TTY and PAD parameters for incoming calls per calling address and userid
- Restriction of outgoing calls to specific addresses based on calling user

The `x29d` daemon must be running for these features to work. The daemon can be started with flags to allow the use of these features. The four PAD configuration files are:

tty	This file defines the tty attributes. The default tty attributes are in the "cooked" mode (for example, local echo , editing , etc..)
profile	This specifies the PAD profile to be used. The default profile is shown in Table 11 on page 102.
address	Selects features based on address. The default uses the "cooked" tty mode and the "default" profile.
user	Selects features as per the user name. The default uses the "cooked" tty mode, the "default" profile and all outgoing calls for all users.

5.4.1.1 Automatic Login

The automatic login feature can be implemented by defining the `login_user` attribute in the address file with an existing user name. When a call is received from an authorized address, the call is automatically logged in as that user. The normal system authentication screen/method is NOT invoked.

If the attribute in the address file reads:

```
31*:  
    login_user = user1
```

This causes any call from a NUA starting 31* to be automatically logged into the machine as user1.

5.4.1.2 Automatic TTY and PAD Parameter Selection

The `tty` and `profile` files can be used to define sets of parameters by name. These are then pointed to by the `in_tty` and `in_profile` attributes in either the user or address files. This allows the selection of any combination of host tty and remote PAD parameters based on calling address, user or both. Since these attributes can appear in either of the files, it is necessary that one file should have priority over the other, should the attributes be set in both files. By default, the user files attributes take priority unless the default prioritization is overridden by the setting on the attribute stanza.

```
attribute = value;priority
```

If the `in_tty` attribute is set in both the user and address files as shown below but the address file has the priority set, then the tty attributes are set to the values defined by `tty2` as specified in the address profile for the particular user.

```
user file
  user1:
    in_tty = tty1
address file
  addr1:
    login_user = user1
    in_tty = tty2;0
```

5.4.1.3 Restriction of Outgoing Calls

The `out_allow` and `out_deny` attributes can be defined in the user file to control which addresses a user can place PAD calls to. The values of this attribute can consist of patterns so that complex sets of addresses can be specified to tailor the outgoing calls. For an outgoing call to succeed it **MUST** match the `out_allow` attribute and **MUST NOT** match the `out_deny` attribute. By default the `out_allow` attribute is set to the asterisk character, which allows all users without explicit entries in the user file to place outgoing calls to any address.

```
user1:
  out_allow = 31*
  out_deny = 31(3-5)*
```

User1 in the above example can call any address starting with 31, with the exception of addresses starting with 313, 314 and 315.

5.4.2 PAD for AIXLink/X.25 1.1.3 (and later) on AIX V.4

The number of configuration files for this version of the AIXLink/X.25 product has increased to allow greater tailoring to take place. The files are:

x28parm This file defines the profiles and its values for the PAD outgoing module. It also contains the standard values of 50, 51, 90 and 91. The three new values given are:

- 10 (default); this is really only for X.29 calls and appears also in the `x29parm` file
- 20 (`titn_test`), same as "51" and is the outgoing PAD `xspad` default
- 30 (print)

x29parm This file defines the profiles and values for the PAD incoming module. The values for the five preset profiles are:

- 90 (CCITT Standard Simple Profile)
- 91 (CCITT Standard Transparent Profile)
- 10 (Default Profile)
- 20 (AIX/TITN Extended Compatibility Profile)

- 30 (Remote Printing Profile)
- x29tty** This file defines the profile and its attributes for the PAD incoming. The default for this is "cooked" mode. For PAD printing, "raw" mode should be used.
- x28user** This file selects the values for an outgoing user; the default is the #20 (51) profile and allows all calling addresses.
- x29user** Selects the values for an incoming user; the default is "cooked" tty mode and the default (#10) profile.
- x29access** This files selects key features based on the address. The default uses the "cooked" tty mode and the default profile.
- qdata** Defines the values needed for locally initiated PAD printing.

5.4.3 Default Initial Application

This allows an AIX host user, connected via X.25 and X.29, to select the initial application which is presented. The criteria that must be met are configurable and based on the X.25 calling address. This is not configurable by any user but is configured for each by a highly privileged AIX user, such as root. There are three kinds of initial applications that can be started upon acceptance of the incoming X.25 call using the X.29 protocol ID. They are logged user, trusted user and selective user.

5.4.3.1 Logged User

For a logged user, an AIX login is done in the normal way and conventional password validation and security processing are performed by the user. There can be multiple "logged users" per X.25 calling address. This kind of initial application processing is based on the fact that the X25 calling address is not configured in the x29access file -- or that the address is configured but the access criteria are not satisfied. A "logged" user invokes "login" ONLY as the default if nothing is specified or if a field value is incorrect or NOT satisfied. A "logged user" who supplies an AIX login has a non-default initial tty-X.29/X.3 profile which is set under two conditions:

- The user has an X.25 address in the x29access file and the attributes for that address reference a valid tty-X.29/X.3 profile set.
- The user's X.25 address is in the x29user file and the attributes for that address reference a valid tty-X.29/X.3 profile set.

5.4.3.2 Trusted User

A trusted user is when an AIX login is done so that it circumvents the usual userid and password validation. This is restricted to one login ID per X.25 calling address. This processing is started when the X.25 calling address is configured in the x29access file. The userid specified as the trusted user must be a valid AIX user with a valid password. The initial application is AIX "login" which executes without performing standard authentication including password authentication.

NOTE: The x29d daemon must be restarted to ensure that the x29access file is re-read when any changes have been made to the file.

5.4.3.3 Selective User

A "selective user" starts an application preconfigured in the x29access file. The application is responsible for the security arrangements. Additional criteria linked to the X.25 calling address may need to be satisfied before the initial application is started and the "selective user" actually exists.

A "selective user" must satisfy the criteria as specified by the access_class value: **remote**, **user_data**, **sub_address** or **user_sub_address** values. Look at the section below for more about the specific criteria. The "selective user" will cause the initiation of the application as specified with the initial_application field (instead of the default of login). This application will be executed as the user specified by the login_id field.

5.4.3.4 access_class values

For each X.25 calling address in the x29access file there is an access_class attribute that specifies one of the following values:

- LOGGED** Connections from the corresponding address undergo standard authentication (the criteria set to be satisfied) before AIX host access is granted. The initial application is AIX "login", which performs the authentication. Typically, a shell is spawned if authentication succeeds. (The call initiator is known as a "logged" user).
- TRUSTED** Connections from the corresponding address are not authenticated (no additional criteria need to be satisfied) before AIX host access is granted. However, the initial application is AIX "login", which executes without performing standard authentication including password authentication. In other words, the user name specified in the login_id keyword is logged in automatically.
- REMOTE** Connections from the corresponding address are granted AIX host access without satisfying additional criteria. (The call initiator is known as a "selective" user). The initial application must be specified by the initial_application attribute and the login_id specified must be that of a valid AIX user with a valid AIX password.
- SUB_ADDRESS** Connections from the corresponding address are granted AIX host access only if the sub-address of the X.25 call packet called address matches that in the sub_address attribute. The match MUST be exact. (The call initiator is known as a "selective" user). The initial application must be specified by the initial_application attribute, and the login_id specified must be that of a valid AIX user with a valid AIX password.
- USER_DATA** Connections from the corresponding address are granted AIX host access only if a user data segment from the X.25 call packet matches that specified by the user_data attribute. The matching conditions are specified in the user_data attribute description. (The call initiator is known as a "selective" user). The initial application must be specified by the initial_application attribute, and the login_id specified must be that of a valid AIX user with a valid AIX password.
- USER_SUB_ADDRESS** Connections from the corresponding address are granted AIX host access only if both the criteria specified in the sub_address and user_data attribute descriptions are satisfied. (The call initiator is known as a "selective" user). The initial application must be specified by the initial_application attribute, and the login_id specified must be that of a valid AIX user with a valid AIX password.

The following three attributes were added after AIXLink/X.25 Version 1.1.3 was released and are available via a PTF upgrade. Please contact your local AIX Systems Support Centre for the PTF. At the time of updating this redbook, the PTF was U443807.

TRUSTED_USER_DATA Connections from the corresponding address are granted AIX host access via the `login_id` specified only if a user data segment from the X.25 call packet matches that specified by the `user_data` attribute. (The matching conditions are specified in the `user_data` attribute description.) No authentication (additional criteria) needs to be satisfied before AIX host access is granted. The initial application, AIX "login", executes without performing standard authentication, including password authentication. Whatever is set up as the user's initial program is spawned by login. (The call initiator is known as a "trusted" user with special criteria needing to be satisfied.)

TRUSTED_SUB_ADDRESS Connections from the corresponding address are granted AIX host access via the `login_id` specified only if the `sub_address` of the X.25 call packet called address matches that in the `sub_address` attribute. The match must be exact. No authentication (additional criteria) needs to be satisfied before AIX host access is granted. The initial application -- which executes without performing standard authentication, including password authentication -- is spawned by login. (The call initiator is known as a "trusted" user with special criteria needing to be satisfied.)

TRUSTED_USER_SUB Connections from the corresponding address are granted AIX host access via the `login_id` specified only if both the criteria specified in the `sub_address` and `user_data` type are satisfied (`user_data` and `sub_address`). No authentication (additional criteria) needs to be satisfied before AIX host access is granted. The initial application is AIX "login", which executes without performing standard authentication, including password authentication. (The call initiator is known as a "trusted" user with special criteria needing to be satisfied.)

5.4.4 Selectable Profile

The selectable profile mechanism allows for the selection of non-default profiles for PAD users. For outgoing PAD application connections, the new functionality allows non-default X.28/X.3 parameter profiles to be selected by name or number. For incoming connections, both non-default X.29/X.3 profiles and tty parameter profiles can be selected. The tty-X.29/X.3 profile sets are selected based on the configuration data found in the `x29access` and `x29user` files.

5.4.4.1 Incoming X.29/X.25 Connection Profile Selection

The method of selecting non-default tty-X.29/3 profile sets is based on the setting of the `x3_profile` field value name and `tty_profile` field value name. There are several ways these profiles are selected depending on how the data is configured in the `x29access` and `x29user` files and the user's AIX login. In all cases, a member-type profile must be referenced in order to select a non-default profile set. The `x29parm` file contains the members of the X.29/X.3 parameter-type. The `x29tty` file contains the members of the tty characteristic-type. Together, these types compose the profile sets.

5.4.4.2 Outgoing X.28/X.25 Profile Selection

The PAD application can select a non-default initial X.28/X.25 profile in three ways:

- Using the **xspad -p** profile, where profile is a selected name or a defined numeric profile number, (#20).
- Preconfiguring a non-default profile in the x28user file. It is prerequisite that the file contain a data set for the user's AIX login. Then, the x3_profile field can be used to specify a profile.
- Using the standard profile command, (PROF).

The available profiles are all stored in the x28parm file.

5.4.4.3 Distribution of Profile Data

All of the X.28/X.3 parameter profiles are stored in the x28parm file. However the xspad command does not directly extract profiles from the x28parm file. Before any profile selection occurs, the X.28 STREAMs module is pushed on X.25. It is at this time that the x28parm file is downloaded in the STREAMs module. This is because a new non-initial profile can be selected at a later stage of xspad execution.

5.4.5 Configurable Profile

The x28parm file contains all the X.3 parameter profiles and is used to tailor the characteristics of the outgoing X.28/X.25 PAD sessions. The x29parm file also contains X.3 parameter profiles and along with the x29tty file, which contains tty characteristic profiles, is used to tailor the characteristics of the incoming X.29/X.25 sessions.

Both the x28parm and x29parm files have the same format. Each profile contains all 22 standard CCITT X.3 parameters along with descriptive information. Each profile has a unique name and numerical identifier, and all the fields must be present even if they are NOT used. The fields can be in any order, but it is recommended that they are kept in ascending numerical order. There is one value that is not used: That is the x3_11_dte_speed parameter, which is read only and ignored.

Each profile in the x29tty file has a unique name. The field names are similar to those used in the normal AIX tty structures. It is easy to set the tty characteristics as almost all of them are either enabled or disabled by setting the field to either on or off. All the fields must be present and in the correct order.

5.4.6 Security Features

Security for outgoing X.28/X.25 PAD sessions is on a per-user basis. This is achieved by restricting the X.25 addresses to which a particular user can connect.

Security for incoming X.29/X.25 sessions depends on the user's X.29 access category:

logged user

Uses the normal AIX login security, (user must type in user name and password).

trusted user

Controlled by the X.25 calling address and data in the x29access file.
The implementation changes DO NOT change the security method.

selective user

Controlled by entries in the x29access file.

5.5 PAD printing

To allow PAD printing, you must have the following filesets installed.

- printers.rte
- printers.rte.(printer type being used)
- printers.hpJetDirect.attach

The printers.hpJetDirect.attach package is important in that the AIXLink/X.25 LPP does not create its own pioin and pioout applications; it uses the standard ones which come with this package. With the exception of AIXLink/X.25, the other packages mentioned are shipped as part of the BOS fileset.

PAD printing can be accomplished by one of two methods. Both methods require an X.25/X.29 connection between the AIX host and the PAD to which the printer is attached:

Remote initiated

The remote PAD printer can poll (initiate the CALL) the AIX host for print jobs queued for it.

Host/Locally initiated

The AIX host can automatically transmit queued jobs (initiate the CALL) to the PAD printer. This method can only be used if the PAD printer has a unique NUA and if the VC has been established between the PAD and the AIX host. This setup should be done in a manner that will not interfere with other devices attached to the PAD.

To enable PAD printing using either method, the AIX host needs substantial configuration and software modules to be installed (as mentioned above). There are two categories of configuration data. The first describes printer attributes, and the second associates NUAs and AIX print queues with actual printers.

The software functionality specific to the host-initiated method is the establishment of an X.25/X.29 connection.

For the remote-initiated method, it is as follows:

- Acceptance of an incoming X.25/X.29 connection
- Association of a destination NUA subaddress with a print queue
- Configuration of X.3 parameters
- Activation of the queue

The software functionality for both PAD printing methods:

- Obtaining jobs from the front of a print queue
- Obtaining "control" of an X.25/X.29 remote printer connection
- Transmission of a job at a rate that does not overrun the printer

- Releasing a X.25/X.29 connection when piobe stops passing jobs

5.5.1 Setting up the PAD Printer Queue

The printer queue for PAD printing is set up in exactly the same way as for normal AIX printing:

- Define the printer device after IPL as "Print Spooling".
- Define a print queue. (This MUST be hpJetDirect - Network printer.)
- Select the printer type.

Once these are configured, you need to enter the queue name and the hostname associated with the printer. This will create an entry in /etc/qconfig with the print queue name that has been chosen. The remaining configuration is dependent on whether you choose to use remotely or locally initiated printing.

5.5.2 PAD Printing Process

PAD printers are defined as AIX print queues and are available to all AIX users.

There are two processes which interact with the printer qdaemon to accomplish PAD printing. They are:

- piox25
- piox25start

Piox25 is started indirectly each time PAD printing is invoked and the qdaemon removes a job from the print queue. The job is piped from piobe to piox25, which transmits it to the network through the X.25 connection. Pacing and formatting are handled by piobe, hence the need for the printers.hpJetDirect.attach fileset to be installed.

5.5.3 Configuration for Remotely Initiated Printing

The X.25 calling address of the printer needs to be configured as follows in /etc/sx25pad/x29access:

1. The access_class must be set to one the selective types:
 - REMOTE
 - SUB_ADDRESS
 - USER_DATA
 - USER_SUB_ADDRESS

The user_data, sub_address specified or user_sub_address must have the criteria satisfied via the appropriate fields.

2. The initial_application parameter must be set to /usr/lpd/piox25start QNAME, where QNAME is the name of the print queue associated with the remote printer.
3. The login_id parameter must be set to a valid AIX user. (This user MUST have a password set up.)
4. The tty_profile parameter should be set to raw, and the x3_profile should be set to an X.3/X.29 profile consistent with remote printing. Currently, one is available in the x29parm file as number 30 or the name "print".

5. The `/etc/qconfig` file entry for the queue must have the backend field so that it points to `/usr/sbin/piox25remote`.

```
pad_q:
    device=hp@kili
hp@kili:
    file=/var/spool/lpd/pio/@local/dev/hp@kili#hpJetDirect
    header=never
    trailer=never
    access=both
    backend=/usr/sbin/piox25remote pad_q
```

Where `kili` = hostname
`pad_q` = print queue name

When `x29d` receives an X.25/X.29 call from the specified calling address indicating the criteria were satisfied, it spawns **`piox25start`**. The `piox25start` process does two things: It activates the specified print queue using the `enq` command, and it indicates the name of the device over which the X.28/X.29 connection is established.

5.5.4 Configuration of Locally Initiated Printing

To configure locally initiated printing do the following:

1. In the `/etc/xs25pad/qdata` file ,add `QNAME LDEVICE DEST`.

Where `QNAME` is a print queue name; `LDEVICE` is a logical device name, such as `sx25a#`; and `DEST` is the X.25 called address of the hardware PAD to which the local printer is connected. (This would also include the subaddress if set up as such.)

2. In the `/etc/qconfig` file, the backend field in the configuration must be `/usr/sbin/piox25local`.

```
pad_q:
    device=hp@fili
hp@fili:
    file=/var/spool/lpd/pio/@local/dev/hp@fili#hpJetDirect
    header=never
    trailer=never
    access=both
    backend=/usr/sbin/piox25local pad_q
```

Where `fili` = hostname
`pad_q` = print queue name

3. You will need to preconfigure the X.3 parameters for the remote hardware PAD. Features such as parity checking and generation need to be disabled, and ancillary device control needs to be set for the type of printer.

5.6 PAD Usage

This section covers how to start an X.28/X.3 terminal PAD session. For information on how to start X.29 on the system, refer to 5.7, "Incoming PAD Setup."

To start a terminal PAD session, run the `xspad` command. On systems where the PAD is configured, this will start a terminal PAD session and provide the PAD prompt (*). At the prompt, the PAD commands can then be used. For information on how to start a terminal PAD session, refer to 5.8, "Setup of an Integrated PAD" on page 114.

A typical PAD session would consist of the following steps:

- Ensure the PAD is configured on the system
- Start the terminal PAD session on an X.25 port:
`xspad -l <port>`
- At the PAD prompt, issue the call to the remote X.25 host:
`call <NUA>`
- Log on to the X.25 host and run the desired application
- Exit off host
- Press **Ctrl-K** to exit out of the `xspad` command

To avoid any interaction by the user (or to limit the interaction), the invocation of the `xspad` command could be as such:

```
xspad -l <port> -a <DEST_NUA> (, <SRC_NUA>) -x
```

where `<DEST_NUA>` is the NUA being called and `<SRC_NUA>`, which is optional, is the calling AIX address. The `-x` option will terminate the `xspad` command once exiting from the host.

5.7 Incoming PAD Setup

To allow the RISC System/6000 to accept incoming calls from remote PADs, the only thing that needs to be done is to ensure that the X.25 port has been configured and that the X.29 daemon is running on the system. You *do not* have to set up routing tables; this is all handled by the X.25 LPP device driver.

5.7.1 Listing the Configured Ports

This can be performed through SMIT as follows:

Devices

- ~~C~~ommunications
 - ~~X~~.25 Co-Processor/2 or Multiport/2 Adapter
 - ~~A~~ppar Adapter
 - ~~M~~anage Device Drivers for X.25 Co-Processor/2 or Multiport/Adapt
 - ~~M~~anage X.25 LPP Device Driver
 - ~~M~~anage X.25 Ports
 - ~~L~~ist All Defined Ports

Or use the fastpath: `smit cx25str_mp`.

5.7.2 Stopping the X.29 Daemon

This can be performed through SMIT as follows:

Devices

- ~~-~~Communications
- ~~-~~X.25 Co-Processor/2 or Multiport/2 Adapter
- ~~-~~Adapter
 - ~~-~~Manage Device Drivers for X.25 Co-Processor/2 or Multiport/Adapt
 - ~~-~~Manage X.25 LPP Device Driver
 - ~~-~~Manage the X.25 Triple-X PAD
 - ~~-~~Stop the Triple-X PAD X.29 Daemon

Or use the fastpath: `smit x25str_pad`

The name of the X.29 daemon is `x29d`.

5.7.3 Starting the X.29 Daemon

The X.29 daemon (incoming PAD) is the software that handles the incoming calls from PADs and interfaces with tty-based applications. The X.29 daemon must be running before a remote caller can access the system.

The X.29 daemon is started, by default, by the `/usr/lib/drivers/pse/x29` entry in the `/etc/rc.net.x25` file. If this entry includes the `-n` option, incoming PAD calls will not be allowed. This entry may be modified or deleted to determine if and how the PAD daemon is automatically started.

Verify that this command is invoked correctly. If the daemon has been started with the `-n` option, the daemon must be stopped and restarted to allow incoming PAD calls. See 5.7.2, "Stopping the X.29 Daemon" to stop the daemon.

The daemon can be started at the command line (via the root user) by invoking the following command:

```
/usr/lib/drivers/pse/x29d
```

This can be performed through SMIT as follows:

Devices

- ~~-~~Communications
- ~~-~~X.25 Co-Processor/2 or Multiport/2 Adapter
- ~~-~~Adapter
 - ~~-~~Manage Device Drivers for X.25 Co-Processor/2 or Multiport/Adapt
 - ~~-~~Manage X.25 LPP Device Driver
 - ~~-~~Manage the X.25 Triple-X PAD
 - ~~-~~Start the Triple-X PAD X.29 Daemon

Or use the fastpath: `smit x25str_pad`.

The name of the X.29 daemon is `x29d`.

There have been some additional flags added to the `x29d` which allow greater flexibility in the usage of the daemon. The default `x29d` mode can run as an outgoing PAD call validation daemon and an X.29 listener daemon.

If the daemon is started using the `-n` flag, the daemon will only allow outgoing PAD calls; all incoming calls will be ignored.

If problems are experienced with incoming calls being ignored, verify that the `-n` option was specified. Stop and restart the `x29d` daemon.

5.7.3.1 User-Invoked x29d Options

The following options are passed to the `x29d` daemon after the daemon has been started. They must be issued by a user who has logged into the AIX system via the PAD.

Issuing the `x29d` command with the `-a` or `-p` options will not start the `x29d` daemon.

- p user** This sets the initial tty and X.3 attributes of an X.29 session by AIX user ID. This flag cannot be used with the `-n` flag. This requires that `x29d` is already running as the X.29 listener daemon. Once set, the only way to turn this option off is to reload the `x29d` daemon.
- a** This is an informational command to `x29d`. It outputs the calling X.25 address of the system initiating the X.29 session to stdout. `x29d` runs and obtains the X.25 address; an AIX userid is not required as NO X.3 or tty attributes are related. This flag cannot be used with the `-n` flag and needs the `x29d` to be already running as a listener daemon.

5.8 Setup of an Integrated PAD

The PAD function is implemented in the X.25 LPP. The `xspad` command provides:

1. The X.3 PAD emulation, allowing commands to be entered to make or clear a call to a remote incoming PAD system
2. The emulation of an asynchronous terminal connected to the PAD through a virtual tty

5.8.1 Starting the PAD Program

The `xspad` program can be started as follows:

```
xspad -l <port>
```

Once the PAD prompt (*) is received, you can proceed with the connection as described in 5.3.2.1, "Establishing Connections" on page 99.

An example of the screen format is as follows:

```
# ./xspad -l sx25a0
TWICE(r) X.3/X.28, (c)Copyright TITN Inc, 1984-1991, All Rights Reserved
*
```

From here, you would use the PAD commands as described in 5.3.2.1, "Establishing Connections" on page 99 to proceed.

5.8.2 Exiting

From the PAD command prompt, enter **Ctrl-K** to terminate the xspad program. The xspad application must not have a call established when attempting to terminate xspad.

5.9 X.25 PAD and C-Kermit

C-Kermit is freely available communication software used to transfer files. The AIX X.25 PAD does not provide file transfer capability. This leads to C-Kermit being used for file transfer. The AIX system can be the client, server or both.

IBM neither supplies nor supports C-Kermit.

5.9.1 Requirements

Since the programs are not IBM-supplied and the local or remote system may not be an AIX system, the requirements are rather vague.

Local and remote hosts require a version of C-Kermit that supports the current X.25 implementation. This is significant since AIXLink/X.25 for AIX Version 4.1 requires a different version of C-Kermit than the AIX X.25 LPP for AIX Version 3.2.5.

C-Kermit is normally distributed in source format. The program must be compiled, renamed `kermit` and placed in a directory named in the `path` variable.

X.25 PAD must be properly configured.

For AIX X.25 to be a C-Kermit client, TCP/IP must be started.

5.9.2 Documentation

C-Kermit documentation is supplied with the source code.

5.9.3 AIX X.25 PAD as a C-Kermit Client

The AIX X.25 PAD does not provide the `pty` functions that C-Kermit requires. C-Kermit has to be started on the local machine using the following command:

```
kermit -j local_host_name
```

Where `local_host_name` is the local TCP/IP host name.

At the C-Kermit prompt, configure the C-Kermit environment using the following commands:

```
set buffers 300000 30000
set windows 31
set receive packet-length 9000
set send packet-length 9000
set filetype binary
connect
```

These commands may be stored in the `.kermrc` file.

Log on to the local host again. Start X.25 PAD and connect to the C-Kermit X.25 PAD server using the following commands:

```
$ xspad <port_name>
* call <NUA>
```

Once connected to a remote host, start C-Kermit using the following command:

```
kermit -ix
```

Enter the C-Kermit escape character to access the C-Kermit menu and select the option for command mode. These will be defined in the C-Kermit documentation. At the C-Kermit prompt, enter the command to send or receive files:

```
send <local_file_name> <remote_file_name>
receive <remote_file_name> <local_file_name>
```

5.9.4 AIX X.25 PAD as a C-Kermit Server

Connect to the AIX system using X.25 PAD and start C-Kermit using the following command:

```
kermit -ix
```

Enter the C-Kermit escape character to access the C-Kermit menu, and select the option for command mode. These will be defined in the C-Kermit documentation. At the C-Kermit prompt, enter the command to send or receive files:

```
send <local_file_name> <remote_file_name>
receive <remote_file_name> <local_file_name>
```

5.9.5 Session Statistics

After file transfer, the user may enter `stat` at the C-Kermit prompt to display session statistics for the most recent file transfer.

Chapter 6. APIs: COMIO, NPI and DLPI

This chapter presents the APIs available in the X.25 LPP. These APIs can be used to write applications tailored to specific needs. Since the new APIs, NPI and DLPI now available with the LPP are based on the STREAMS environment, first we will review the STREAMS concepts and mechanisms. Then, we describe the NPI, DLPI and COMIO APIs. NPI stands for Network Provider Interface, DLPI stands for Data Link Provider Interface and COMIO stands for Common Input/Output, which is an emulation of the previously device driver interface supported in AIX V3.

6.1 Implementation of X.25 in the STREAMS Environment

Figure 32 details the X.25 implementation in the STREAMS environment of the RS/6000. This figure shows where each kind of application (NPI, DLPI and COMIO) is interfaced.

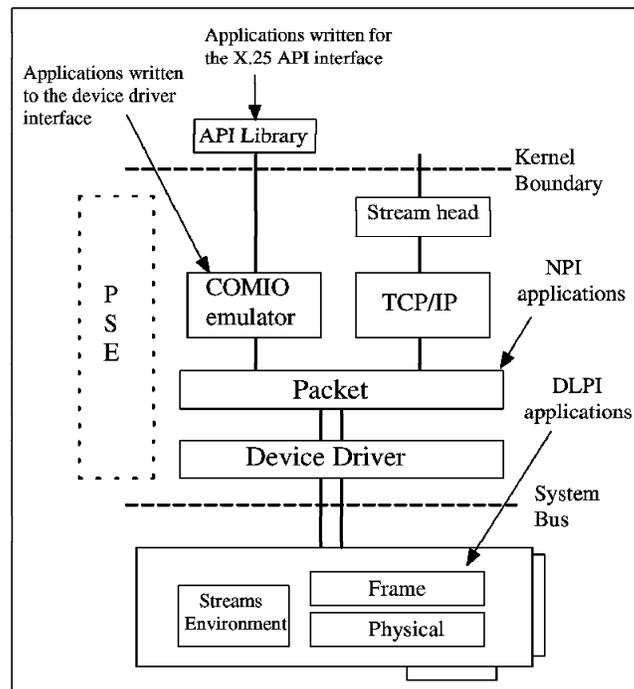


Figure 32. X.25 In the STREAMS Environment

6.2 STREAMS

STREAMS defines standard interfaces for character input/output within the system kernel and between the kernel and the rest of the system.

6.2.1 STREAMS Definition

STREAMS is a collection of system calls, kernel resources and kernel utility routines for developing system communication services. STREAMS is used by developers to provide services ranging from complete networking protocol suites to individual device drivers.

The tools that compose the STREAMS environment allow you to create, use and dismantle a stream. A stream is a full-duplex processing and data transfer path between a driver in kernel space and a process in user space.

6.2.2 STREAMS Components

As shown in Figure 33, a stream is made of three main components:

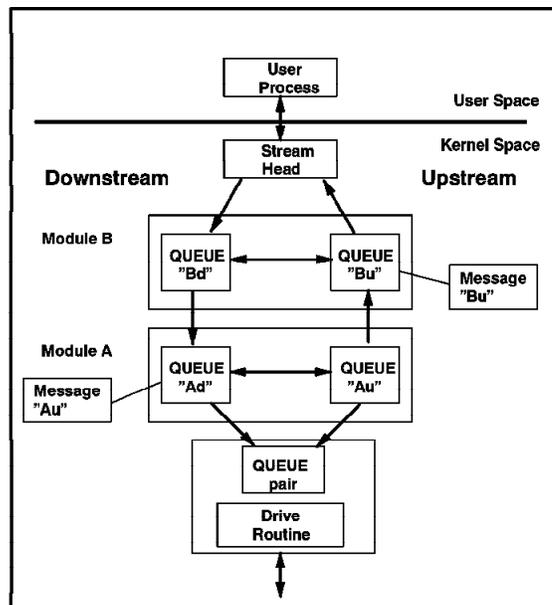


Figure 33. Components of a Stream

Stream Head

Provides the interface between the stream and user processes. Its principal function is to process STREAMS-related system calls.

Module

Processes data that travels between the stream header and the driver. Modules may be optional when the driver performs all the work needed.

Stream End

Provides the services of an external input/output device or an internal software driver. The internal software driver is commonly called a pseudo device driver.

Using the tools, STREAMS passes data between a driver and the stream head in the form of messages. Messages that are passed from the stream head toward the driver are said to travel *downstream* while messages passed in the other direction travel *upstream*.

6.2.2.1 Stream Head

The stream head provides the interface between an application program and the stream for the transfer of data between the data space of a user process and STREAMS kernel data space. Data sent to a driver from a user process is packaged into STREAMS messages and transmitted downstream. Downstream messages arriving at the stream head are processed, and the data is copied from user buffers. These messages contains data, status and control information, or a combination of both. Each message includes a specified message type indicator that identifies the contents. The stream head processes STREAMS-related operations from the application and performs the bidirectional transfer of data and information between the application and messages queue.

6.2.2.2 Modules

STREAMS can insert one or more modules into a stream between the stream head and the driver to process data passing through the stream. Each module is constructed from a pair of queue structures required to implement the bidirectional and symmetrical attributes of a stream. One queue performs functions on messages passing upstream through the module. The other queue performs another set of functions on the downstream messages. Each of the two queues in a module generally have distinct functions, that is, unrelated processing procedures and data. The queues operate independently, but if the developer specifically programs the module functions to perform sharing, messages and data can be shared. Each queue in a module can contain or point to:

- Messages
- Processing procedures
- Data

Although depicted as distinct from modules, the stream head and the stream end also contain a pair of queues.

6.2.2.3 Stream End

The stream end is a module in which the module processing procedures are the driver routines. The procedures in the stream end are different from those in other modules because they are accessible from an external device and the STREAMS mechanism allows multiple streams to be connected to the same driver (see Figure 37 on page 122).

The driver can be a device driver providing interface between kernel space and an external communication device or an internal pseudo device driver. A pseudo device driver is not directly related to any external device, and it performs functions internal to the kernel. Device drivers must transform all data and

status or control information between STREAMS message format and their external data representation.

6.2.3 Benefits of STREAMS

The STREAMS-standard-based interfaces provide a portable environment.

6.2.3.1 Two Main Benefits

Creating Service Interfaces: STREAMS allows easy creation of modules that offer standard data communications services to any neighboring application program, module or device driver. This service interface consists of a specified set of messages and the rules for allowable sequences of these messages across the boundary of the modules. A module that implements a service interface will receive a message from a neighbor and respond with an appropriate action based on the specific message received and the preceding sequence of messages.

STREAMS Modularity: As we have just seen, the basic components in a STREAMS implementation are referred to as modules. These modules, which reside in the kernel, offer a set of processing functions and associated service interfaces. From the user level, modules can be dynamically selected and interconnected to provide any rational processing sequence. This STREAMS modularity allows:

- User-level programs that are independent of underlying protocols and physical communication media
- Network architectures and high-level protocols that are independent of underlying protocols, drivers and physical communication media
- High-level services that can be created by selecting and connecting low-level services and protocols
- Enhanced portability of protocol modules, resulting from STREAMS' well-defined structure and interface standard

6.2.3.2 Consequences of these Two Main Benefits

Protocol Portability: This architecture allows you to use the modules with different drivers on different machines. Figure 34 shows an X.25 protocol module used with different drivers on different machines by implementing compatible service interfaces.

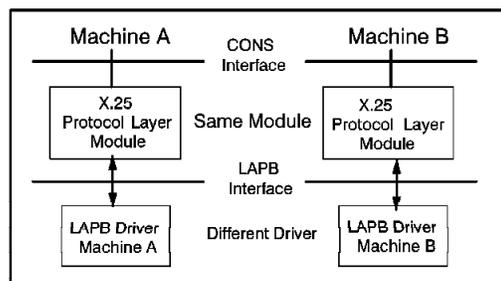


Figure 34. The Portability of the STREAMS Architecture

Protocol Substitution: Alternative protocol modules and device drivers can be interchanged on the same machine if they are implemented to equivalent service interfaces.

Protocol Migration: If the modules are written to comply with industry-standard service interfaces (called, in that case, *protocol modules*), it allows easier migration. An illustration is shown in Figure 35 where functions are migrated between kernel software and front-end microcode. We can see in this example that the transport protocol module can be connected identically to either an X.25 module or X.25 driver that has the same service interface. So this mechanism allows you to rapidly incorporate technology advances since only one layer of the software architecture has to be modified.

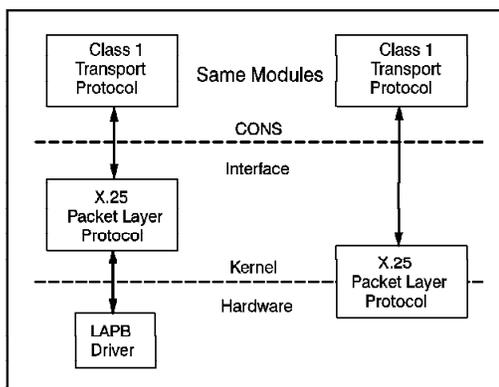


Figure 35. Easy Migration with the STREAMS Architecture

Module Reusability: Modules may be reused in different streams.

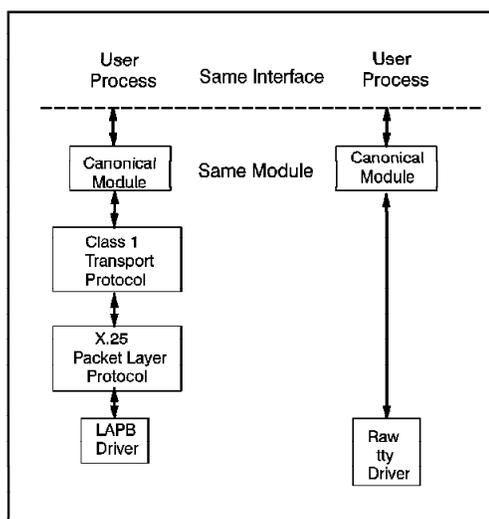


Figure 36. Reusability of STREAMS Modules

6.2.3.3 STREAMS Facilities

STREAMS provides developers with integral functions, a library of utility routines and facilities that expedite software design and implementation:

- Buffer management: Maintains STREAMS' own and independent buffer pool
- Scheduling: Incorporates STREAMS' own scheduling mechanism
- Asynchronous operation of STREAMS and user processes: Allows STREAMS-related operations to be performed efficiently from user level
- Flow control: Conserves STREAMS memory and processing resources
- Error and trace logs: Allow for debugging and administrative functions
- Multiplexing: Allows for processing interleaved data streams such as those that occur in X.25 (SVCs and PVCs). See Figure 37.

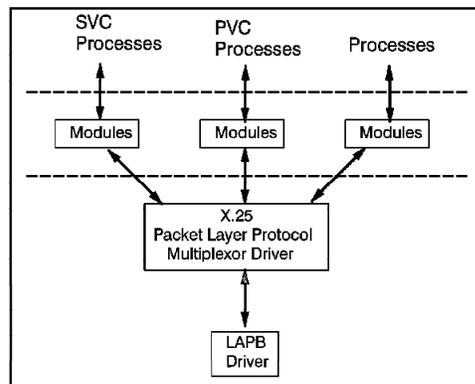


Figure 37. Multiplexing Facility of the STREAMS

6.2.4 How to use STREAMS

Application programmers can take advantage of the STREAMS facilities by using a set of system calls, subroutines, utilities and operations. The following is a description of the tools available. Examples of use will be found in the later sections, where NPI and DLPI APIs are described.

6.2.4.1 Subroutines

STREAMS subroutines include:

- open** Opens the stream to the specified driver. A stream is created on the first open subroutine to a character special file corresponding to a STREAMS driver. For instance, with the X.25 LPP, the /dev/x25pkt packet driver is already available, and so you have only to open the stream with the open subroutine (dev=open"/dev/x25pkt",0_RDWR).
- close** Closes a stream (closedev).
- read** Reads data from a stream. Data is read in the same manner as character files and devices.
- write** Writes data to a stream. Data is written in the same manner as character files and devices.
- poll** Notifies the application program when selected events occur on a stream.

6.2.4.2 System Calls

STREAMS system calls include:

getmsg Receives the message at the stream head. We will see in the next section how this system call will be used to get the primitive message in the stream.

Example:

```
rc = getmsg(dev, &ack_ctl, (struct strbuf *)NULL, &getflags);
```

putmsg Sends a message downstream. As above, we will see in the next section how this system call will be used to put the primitive message on the stream.

Example:

```
rc = putmsg(dev2, &req_ctl, (struct strbuf *)NULL, 0);
```

getpmsg Receives the priority message at the stream head.

putpmsg Sends a priority message downstream.

6.2.4.3 Operations

Operations include:

ioctl The `ioctl` subroutine allows you to perform functions specific to a particular device. A set of generic STREAMS `ioctl` operations (named streamio operations) support a variety of functions for accessing and controlling streams. The two streamio operations used in the samples described below are the `I_PUSH` and `I_POP`, which respectively allow to push and pop a module in a stream already opened. The modules manipulated in this manner are called pushable modules, in contrast to the modules contained in the stream head and stream end. For example, to push a new module in a stream, you do the following streamio operation: `ioctl(dev, I_PUSH, "npi")`. See Figure 39 on page 124 for an illustration of the STREAMS pushing mechanism.

6.3 NPI API

To better understand how to use the NPI, before we describe the NPI samples, we briefly present the NPI implementation on the RS/6000. More details are available on the NPI in the *AIXLink/X.25 Version 1.1 for AIX: Guide and Reference manual*.

6.3.1 NPI Implementation on the RS/6000

The Network Provider Interface (NPI) consists of a set of primitives defined as STREAMS messages that provide access to the network layer services. The primitives are transferred, as STREAMS messages, between the network service (NS) user entity (data space of a user process) and the NS provider (STREAMS kernel data space).

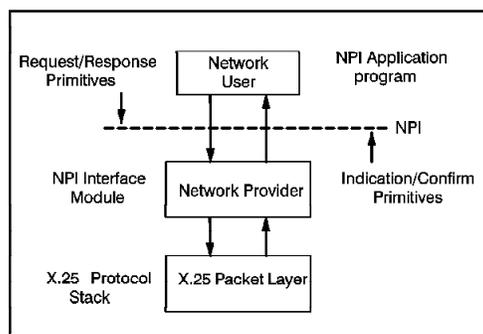


Figure 38. NPI Components

A NPI primitive can be one of the following types:

User-originated

These primitives make requests to the NS provider or respond to an event of the NS provider. They are processed by the putmsg system call.

Provider-originated

These primitives are either confirmations of a request or are indications to the NS user that the event has occurred. They are processed by the getmsg system call.

The open subroutine makes the stream available:

- `open("/dev/x25pkt", 0_RDWR);`

The streamio operation pushes the NPI module in the stream:

- `ioctl(dev, I_PUSH, "npi");`

The building of a stream is illustrated by the Figure 39.

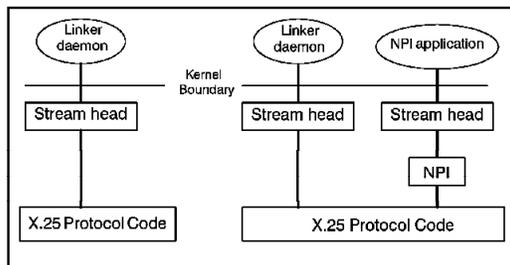


Figure 39. The Building of a NPI Stream

The NPI allows you to access the packet layer (layer 3 in the OSI Reference Model) and provides a connection-mode communication, which allows you to build:

- Virtual circuits
- Data transfer by a pre-established path
- Data transfer reliability

Since the interface is standard, multiple NPI applications can access the X.25 protocol code.

6.3.2 NPI Primitives Available on the RS/6000

A set of primitives is available to establish a connection, transfer data and terminate a connection:

1. Local Management Primitive

These primitives are called to open or close a stream connected to the NPI module. They also manage options supported by the NPI module and report information on its supported parameter values.

2. CONS Primitive Format and Rules

A network connection, once established, can be seen as a pair of queues linking two network addresses or X.25 hosts. There is one queue for each direction of information flow, and each queue provides a control function for its information flow. The different groups of CONS primitives are:

- Connection Establishment Primitives
- Normal Data Transfer Primitives
- Receipt Confirmation Service Primitives
- Expedited Data Transfer Service Primitives
- Reset Service Primitives
- Network Connection Release Primitives

More details on the use of these primitives are available in the *AIXLink/X.25 Version 1.1 for AIX: Guide and Reference*.

6.3.3 Use of the Primitives

Each time you want to use a primitive, you must perform the following steps:

1. Space management

Allocate and zero out space for the primitive you will invoke.

This is an example for the N_BIND_REQ primitive :

```
listen_call = (char *) malloc(NPI_MAX_CTL);
bzero(listen_call, NPI_MAX_CTL);
```

If needed, allocate and zero out space for the data associated with the primitive.

2. Structures

Build the structure(s) needed for the primitive you are going to invoke:

- Build the primitive by filling up the structure associated with the primitive. This kind of structure must be filled when the primitive is user-originated (putmsg()).

- Build the control structure. This structure must be filled when the primitive is user-originated (`putmsg()`) and provider-originated (`getmsg()`).
- Build the data structure. This structure must be filled when the primitive is provider-originated (`getmsg()`).

3. System calls

Call the `putmsg()` system call to put the primitive message on the stream if the primitive is user-originated.

Call the `getmsg()` system call to get the primitive message in the stream if the primitive is provider-originated.

4. Verification

If you have just done a `getmsg()` system call, verify that you have received the right primitive.

5. Deallocation of memory

Free up the memory allocated earlier:

```
free(listen_call);
```

Once the X.25 software is installed on a system, the NPI module is made available, and then applications that link to the streams library are able to access the NPI module.

6.3.4 Description of the NPI Samples Available with the LPP

The following two diagrams describe the structure of the two programs which allow you to transfer data between two X.25 hosts by using the NPI. Once you have set up the right values for the line parameter, the called/calling NUA and the data to send, you first run `npiserver` and then `npiclient`.

These programs are located in the `/usr/lpp/sx25/samples/npi` directory on AIX Version 3 machines and in the `/usr/samples/sx25/npi` directory on AIX Version 4 machines.

npiserver.c

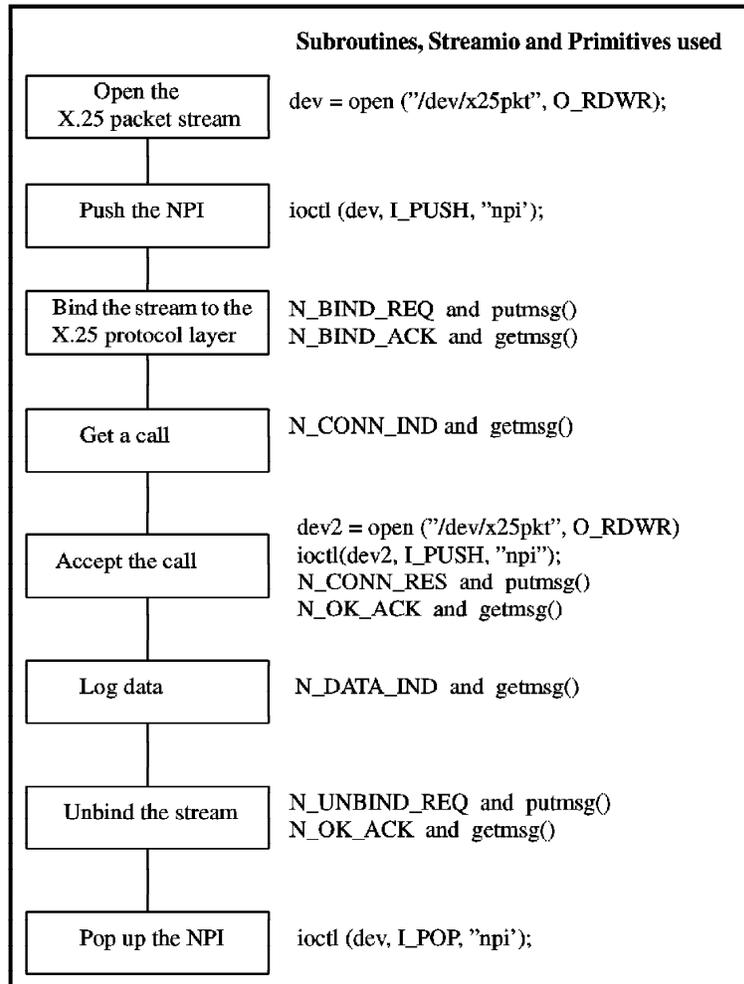


Figure 40. Structure of the npiserver.c Program

npiclient.c

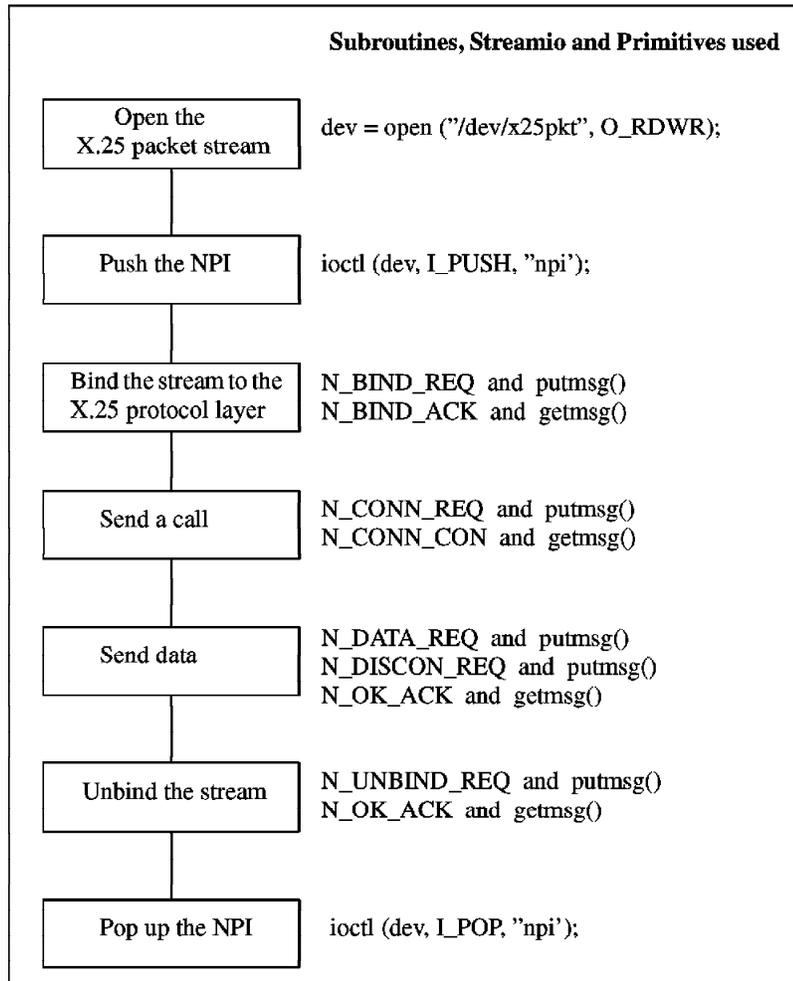


Figure 41. Structure of the *npiclient.c* Program

6.4 DLPI API

To better understand how to use the DLPI, before we describe the DLPI samples we briefly present the DLPI implementation on the RS/6000. More details are available on the DLPI interface in the *AIXLink/X.25 Version 1.1 for AIX: Guide and Reference*.

Note: To use the DLPI layer on a port, the Enable DLPI interface ONLY parameter must be set to yes using the Change / Show X.25 General Parameters SMIT menu. When this is done, only the frame and physical levels can be used. Since the packet level will not be activated, the port can no longer be used for packet level applications such as TCP/IP, PAD and others.

6.4.1 DLPI Implementation on the RS/6000

The data link layer (layer 2 in the OSI model) is also referred to as the frame or LAP-B layer. This layer is responsible for the transmission and error-free delivery of bits of information over a physical communication medium.

The Data Link Provider Interface (DLPI) provides point-to-point data communication and is designed for two systems connected back to back over a communication link. In the X.25 LPP, this is provided through a connection-oriented subset of the DLPI specification. It specifies an interface to the services of the data link layer as Figure 42 illustrates.

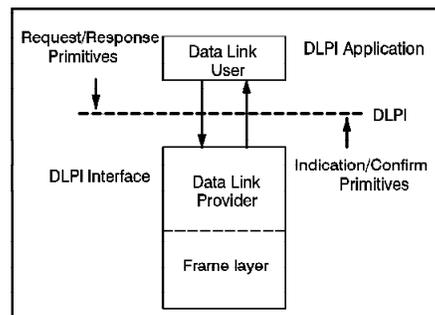


Figure 42. DLPI Components

The data link interface is the boundary between the network and data link layer of the OSI Reference Model. The network layer entity, which is usually the X.25 layer, is the user of the services of the data link interface. This user is DLPI's user application and is sometimes referred to as the Data Link Service (DLS) user. The DLPI layer which provides the programming interface is referred to as the DLS provider. This interface consists of a set of primitives that provide access to the link layer services. The service primitives that make up kernel-level interfaces are defined as STREAMS messages that are transferred between the user and provider of the service. DLPI is targeted for STREAMS protocol modules that either use or provide data link services.

6.4.2 Establishment of a Stream to DLPI

The first step you have to do when you want to use the direct access to the frame layer on a port is to enable the DLPI, so you have to set the Enable DLPI interface only field of the General Parameters screen to yes when you set up the port parameters. This disables the packet layer X.25 access, and regular X.25 applications cannot use the port. Neither a COMIO nor a TCP/IP interface is authorized.

Set one of the ports in passive-wait mode; this is done by setting the frame attributes from SMIT.

The user has to establish a stream to DLPI before DLPI primitives can be used by the application. To establish a stream, the user binds to the frame layer. Only one application per port can bind to the frame layer. Once bound to a stream, the application can send primitives to DLPI to establish a connection and to

transfer data. When an application wishes to terminate, it disconnects the link and unbinds from its stream.

6.4.3 DLPI Primitives Available on the RS/6000

There are two groups of DLPI primitives.

6.4.3.1 Local Management Service Primitives

This subset of primitives are in charge of the information reporting, attach and bind/unbind services of DLPI. Once a stream has been opened by a DLS user, these primitives initialize the stream and prepare it for use.

6.4.3.2 Connection Mode Service Primitives

These primitives are in charge of the connection-mode service of the data link layer. They may be split in four different subsets:

- Connection Establishment Primitives
- Data Transfer Primitives
- Connection Release Primitives
- Reset Primitives

6.4.4 Description of the DLPI Samples Available with the X.25 LPP

The two following diagrams describe the structure of the two programs which allow you to transfer data between two hosts using the frame-level DLPI interface.

The DLPI client and server sample programs have the following syntax:

```
dlpiclient [-b #] [-p #] [-n #]
           [-?] [-h]
```

```
dlpiserver [-b #] [-p #] [-n #]
           [-?] [-h]
```

The flags specify the following:

- b board number
- p physical port number
- n number of frames to be sent
- ?/-h returns usage statement

If no command line arguments are passed, the programs default to the values specified in the common.h file. The size of the info frames can be modified by changing the buf_sz value in common.h.

These programs are located in the /usr/lpp/sx25/samples/npd directory on AIX Version 3 machines and in the /usr/samples/sx25/npd directory on AIX Version 4 machines.

dlpiserver.c

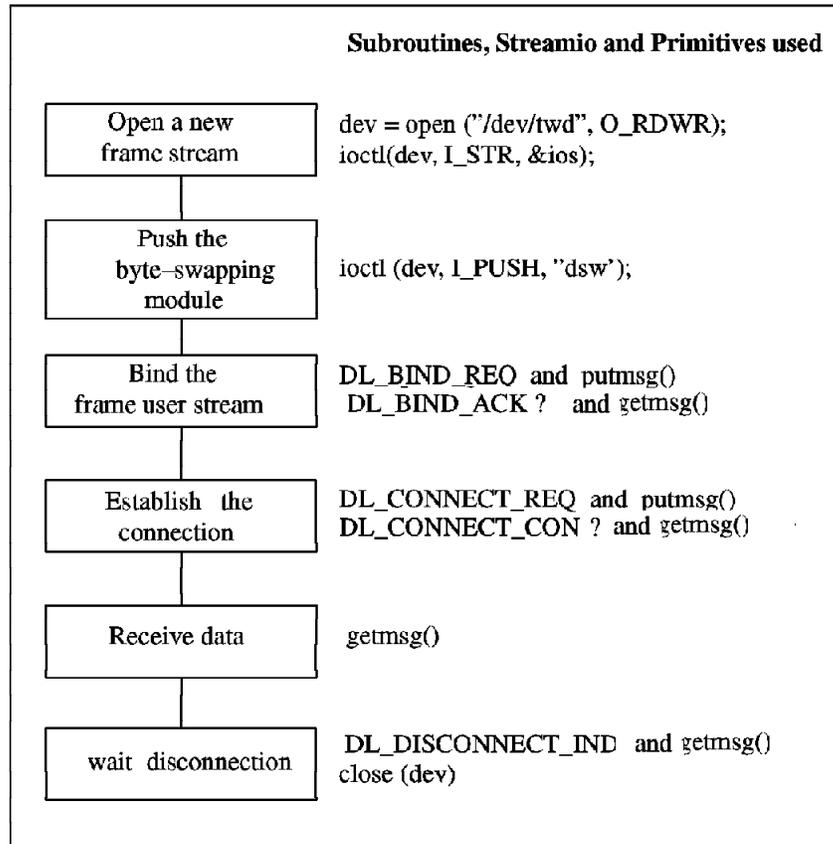


Figure 43. Structure of the dlpiserver.c Program

dlpiclient.c

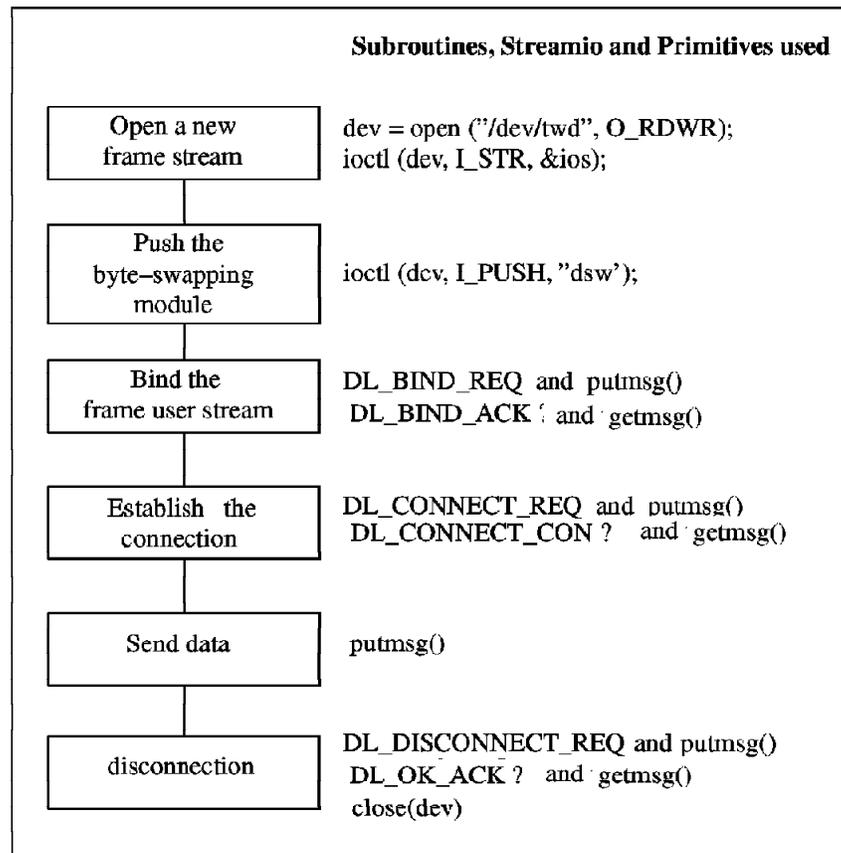


Figure 44. Structure of the *dlpiclient.c* Program

6.5 COMIO API

The X.25 LPPs provide the COMIO interface to maintain compatibility with the previous AIX X.25 product.

6.5.1 COMIO Definition

COMIO stands for Common Input/Output. COMIO emulates the device driver interface supported in previous releases of AIX X.25. This allows the user-space library API (available in AIX V.3 X.25 support) and the codes written to the device driver interface to work unchanged. As the library API is still available in the X.25 LPP, all the applications written over this library work without recompilation.

Figure 45 on page 133 illustrates COMIO emulation.

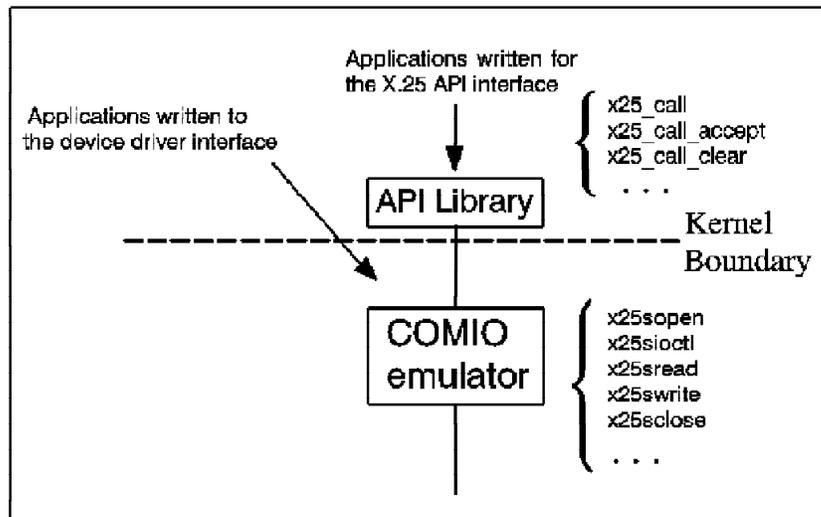


Figure 45. COMIO Emulation

The COMIO emulation is not intended for new program development but more for compatibilities.

This library API and the device driver interface are detailed in Chapter 12 of the *AIXLink/X.25 Version 1.1 for AIX: Guide and Reference* manual. So we shall only briefly describe the use of the API and the samples made available at X.25 LPP installation.

The X.25 application programming interface functions are at a level slightly above the packet level.

6.5.2 Using COMIO

Once COMIO is configured for an X.25 port, you can use the `xtalk` tool or a user-developed application.

6.5.2.1 Enabling COMIO

To allow use of this API, the COMIO emulation function must be configured on the X.25 ports (`sx25an`). To do this, select the Add Comio Interface to Port in the Manage X.25 Ports menu. It will create a device entry with the same name that would have been generated by the base AIX Version 3 X.25 support (`x25sn`).

6.5.2.2 COMIO Subroutines

You need access to a C compiler because the X.25 API includes a library of C subroutines that use the services of the COMIO emulator. Applications call these subroutines to access X.25 functions.

The subroutines are split into four groups:

- Initialization and termination subroutines
- Network subroutines
- Counter subroutines
- Management subroutines

The X.25 API subroutines are kept in the `/usr/lib/libx25s.a` library. The API subroutines are called using standard C conventions.

Each subroutine is detailed in the *Technical Reference: Communications* manual, Volume 3, Chapter 11.

6.5.2.3 COMIO structures

The subroutines use a number of structures to pass information between the X.25 functions and the application program. The `/usr/include/x25sdefs.h` header file describes all the structures used (also described in the *Files Reference* manual).

6.5.2.4 X.25 Error Codes

The X.25 subroutines set the `x25_errno` and `errno` flags to indicate error conditions.

If an error condition results from an X.25 API subroutine call, it is indicated in one of the following ways:

- For X.25-specific error conditions, the `x25_errno` flag indicates the error, for example, `X25ACKREQ`. The `errno` flag is not set in these conditions.
- For other error conditions, the `x25_errno` flag is set to `X25SYSERR` and the `errno` flag indicates the error, for example, `EFAULT`.

The API Error Codes section, on page 13-19 of the *Communication Programming Concept* manual, lists the error codes that may be returned by X.25 subroutines.

6.5.3 X.25 Example Programs

To help you learn how to use the X.25 subroutines, there are two pairs of example programs. One pair demonstrates the use of a switched virtual circuit, and the other the use of a permanent virtual circuit.

These programs are located in the `/usr/lpp/sx25/samples/comio` directory on AIX Version 3 and in the `/usr/samples/sx25/comio` directory on AIX Version 4 machines:

- `svcxmit.c`: Make a call using an SVC
- `svcrcv.c`: Receive a call using an SVC
- `pvcxmit.c`: Send data using a PVC
- `pvcrcv.c`: Receive data using a PVC

Each of the example programs has variables to which values are assigned at the start: `calling_addr`, `called_addr`, `link_name` and `log_chan_num`. You should customize these values in accordance to your own environment before running the programs.

To compile the example programs, use the following commands:

```
# cd /usr/samples/sx25/comio (or cd /usr/lpp/sx25/samples/comio)
# cc -o svcxmit svcxmit.c -lx25s
# cc -o svcrcv svcrcv.c -lx25s
# cc -o pvcxmit pvcxmit.c -lx25s
# cc -o pvcrcv pvcrcv.c -lx25s
```

To run a program, type the name of the executable file at the shell prompt.

Note: Run the programs in pairs: svcxmit with svcrcv and pvcxmit with pvcrcv. (These two last programs can be run only if you use permanent virtual circuits.) The programs which receive data must be run first: svcrcv and pvcrcv.

For more details about these programs, please refer to *AIXLink/X.25 Version 1.1 for AIX: Guide and Reference* manual.

Chapter 7. TCP/IP Setup

This section documents the steps required to configure TCP/IP on an X.25 network.

7.1 Introduction

AIX Version 3.2 and AIX Version 4.1 implement the 4.4 BSD Reno level of TCP/IP and support the following Internet Request For Comments (RFC):

RFC 1356 IP over X.25

This RFC specifies a standard adopted by CSNET, the VAN gateway and other organizations for the transmission of IP datagrams over the X.25-based public data networks.

An X.25 virtual circuit is opened on demand when a datagram arrives at the network interface for transmission. A virtual circuit is closed after some period of inactivity. A virtual circuit may also be closed if the interface runs out of virtual circuits.

Standards:

1. The first octet in the Call User Data field (the first data octet in the Call Request Packet) is used for protocol demultiplexing. The value of hex CC (binary 11001100, decimal 204) is used to mean Internet Protocol.
2. IP datagrams are sent as X.25 *complete packet sequences*. That is, datagrams begin on packet boundaries and the M-bit (More data) is used for datagrams that are larger than one packet. There are no additional headers or other data in the packets.
3. Either site may close a virtual circuit. If the virtual circuit is closed or reset while a datagram is being transmitted, the datagram is lost.

This chapter describes how to configure TCP/IP over X.25 on AIX Version 3.2.5 and AIX Version 4.1 with the X.25 LPP. You must have successfully completed the X.25 LPP configuration described in Chapter 3, "X.25 LPP Installation and Setup" on page 43 before trying to use TCP/IP.

The steps are described in the subsequent sections. The main tool used in the setup is SMIT, but some AIX commands will also be used at the proper time.

Setup and Test of TCP/IP Connection with SVC

1. Add entries for the local and remote hosts in /etc/hosts with
smit mkhostent
2. Initialize and start the IP/X.25 interface:
 - Start smit mkinet1xs.
 - Select the sx25a0 X.25 LPP Port.
 - Enter the local X.25 IP address and the IP network mask.
3. Add a route for each remote host connected via X.25 to IP networks different from the IP network ID of the X.25 interface:
 - Start smit mkroute
 - Enter:
 - The destination type as host
 - The remote X.25 hostname as DESTINATION address
 - The local X.25 hostname as GATEWAY address
 - The number 0 as the METRIC or number of hops to destination gateway.
4. For each remote host, create an entry in the IP hostname to NUA translation table:
 - Start smit mksx25
 - Enter the remote HOSTNAME, and its NUA
5. Verify that everything is correct with netstat -r.
6. Test the connection to a remote host with ping.

If you plan to use a PVC instead of an SVC for the IP communications, change item 4 to:

4. For each remote host, create an entry in the translation table from IP hostname to PVC number:
 - Start smit mksx25p.
 - Enter the remote IP host name, the corresponding logical channel number and the X.25 LPP port number.

7.2 TCP/IP X.25 Connection Setup and Test

Figure 46 on page 139 shows our two hosts kili and fili. We have assigned them the X.25 hostnames: kilix and filix, and IP addresses: 9.3.5.205 and 129.35.1.93. You have probably noticed that our two hosts do not belong to the same IP network and are not even in the same IP network address class. This is deliberate, because IP hosts connected through X.25 generally belong to different IP networks.

We will describe the customization process for kili. It is similar for fili.

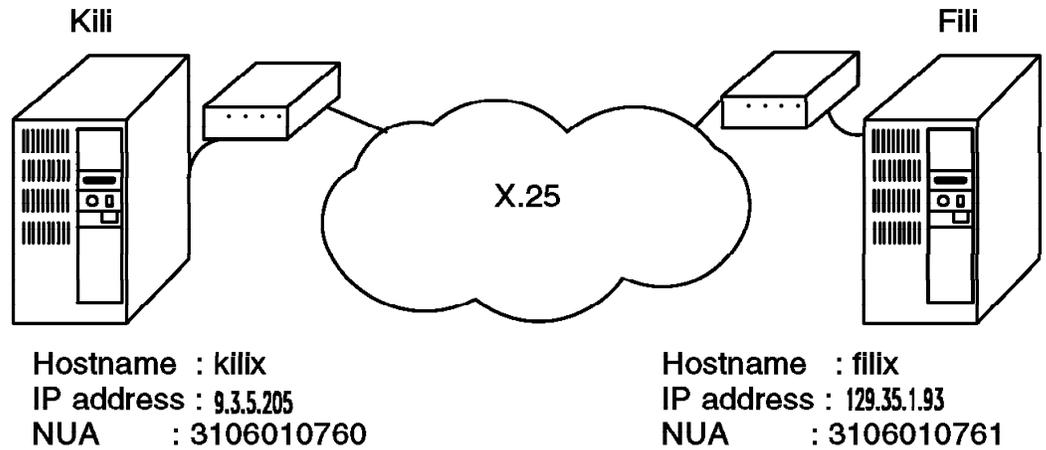


Figure 46. Test Configuration for TCP/IP

7.2.1 Creating Entries in /etc/hosts

This step can be skipped only if DNS is used and AIX is connected to the nameserver on a LAN. IF the network initialization script /etc/rc.net starts building the IP to NUA translation table before the connection to the nameserver is established, it will not find all the X.25 remote hostnames in /etc/hosts, it fails and IP communications over X.25 will not be possible.

Create an entry for the local X.25 host (kilix) as well as for the remote X.25 hosts by using the following SMIT menu sequence:

```

Communications Applications and services
  -TCP/IP
    -Further Configuration
      -Name Resolution
        -Host Table
          -Add a Host
  
```

Or use the fastpath: `smit mkhostent`.

Add a Host Name

Type or select values in entry fields.
 Press Enter AFTER making all desired changes.

<p>* INTERNET ADDRESS (dotted decimal)</p> <p>* NAME</p> <p>ALIAS(ES) (if any - separated by blank space)</p> <p>COMMENT (if any - for the host entry)</p>	<p>[Entry Fields]</p> <p>[9.3.5.205]</p> <p>[kilix.itsc.austin.ibm.></p> <p>[kilix]</p> <p>[kili on X.25]</p>
--	--

This example was for the local X.25 host kilix. We have also added the hostname filix at address 129.35.1.93 corresponding to the remote host fili.

7.2.2 Initializing and Starting the IP/X.25 Interface

If you have not defined any other TCP/IP attachment, use the fastpath `smit mktcpip` to configure TCP/IP and start an interface. In the other cases, just add an interface by using the following sequence of SMIT menus:

```
Communications Applications and Services
- TCP/IP
  - Further Configuration
    - Network Interfaces
      - Network Interface Selection
        - Add a Network Interface
          - Add a X.25 LPP Network Interface
```

Or with the fastpath: `smit mkinet1xs`, select your X.25 LPP Port, and then configure the interface.

```

X.25 LPP Port

Move cursor to desired item and press Enter.

sx25a0 Available 00-04-00-00 X.25 Port
```

```

Add a X.25 LPP Network Interface

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]

* Internet ADDRESS (dotted decimal)      [9.3.5.205]
Network MASK (hexadecimal or dotted decimal) [255.255.255.0]
```

The X.25 LPP does not attempt a connection to the X.25 network while configuring the IP/X.25 interface `xs0`. You can get an OK message from SMIT even if your X.25 connection is down.

To check if the `xs0` interface is up and running try:

```
# ifconfig xs0
xs0: flags=80a0043<UP,BROADCAST,RUNNING,ALLCAST,MULTICAST>
      inet 9.3.5.205 netmask 0xfffff00 broadcast 9.3.5.255
```

On AIX Version 3.2.5 systems try:

```
# netstat -r
Routing tables
Destination      Gateway          Flags  Refcnt  Use      Interface
Netmasks:
(root node)
(0)0 ff00 0
(0)0 ffff ff00 0
(root node)

Route Tree for Protocol Family 2:
(root node)
default          eze              UG      0      27066   tr0
9.3.1            kili             U        14     18963   tr0
```

```

9.3.5          kilix          U      0      181    xs0
127             loopback        U       0       13      lo0
(root node)

```

```

Route Tree for Protocol Family 6:
(root node)
(root node)

```

On AIX Version 4.1 systems try:

```

# netstat -r
Routing tables
Destination      Gateway          Flags  Refcnt Use      Interface
Netmasks:
255.255
255.255.255

```

```

Route Tree for Protocol Family 2:
default          eze              UG      0      27066   tr0
9.3.1           kili             U       14     18963   tr0
9.3.5         kilix          U      0     181    xs0
127             loopback        U       0       13      lo0

```

7.2.3 Adding Routes for the Remote Systems

When we defined the xs0 attachment, we created an implicit route for the IP network corresponding to the IP address and the netmask we have set. In our case, the network 9.3.5 is associated with xs0. If we want to communicate with systems not belonging to this network, we must add *routes*. In this particular case, the gateway is the local host itself, kilix.

```

                                Add Static Route

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
* DESTINATION TYPE                [host]
* DESTINATION Address              [filix]
  (dotted decimal or symbolic name)
* GATEWAY Address                  [kilix]
  (dotted decimal or symbolic name)
* METRIC (number of hops to destination gateway) [0]

```

We can now verify with netstat -r that the route to filix is correct:

```

# netstat -r
Routing tables
Destination      Gateway          Flags  Refcnt Use      Interface
:
default          eze              UG      0      27066   tr0
filix          kilix          UH     0     0      xs0
:

```

Create a route, in the same way, on fili to access kilix.

7.2.4 Mapping IP Addresses to X.25 NUAs

7.2.4.1 Attachment to the Defense Data Network (DDN)

AIX 3.2 and AIX 4.1 conform to RFC 1236 "IP to X.121 Address Mapping for DDN," which defines an algorithm that automatically determines the NUA based on the Internet address. Choosing the "DDN" network identifier invokes this algorithm in place of the x25ip tables. With the X.25 LPP, you can use Class A, B and C Internet addresses when working with DDN (the RFC lists Class B and C as optional).

7.2.4.2 For All Other Network Identifiers

For all other network types, the x25ip table maps TCP/IP Internet addressing to the X.121 Network User Address (NUA). This table is maintained via the SMIT X.25 LPP IP/X.25 Host Entry menu.

We now create, for each remote host, an entry in this table. From the first menu of the SMIT interface, select:

```
Communications Applications and Services
-CP/IP
  -Further Configuration
    -Network Interfaces
      -Network Interface Selection
        -X.25 LPP IP Host Configuration
          -Add a X.25 LPP IP Host Entry
            -Add a Switched Virtual Circuit (SVC) X.25 LPP IP Host Entry
```

Or use the fastpath: `smit mksx25s`.

```

                                     Add INTERNET / X.25 SVC Host Entry

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                     [Entry Fields]
* Remote HOSTNAME                     [filix]
* Remote DTE Address                   [3106010761] #
----- Optional X.25 Facilities -----
RECEIVED data PACKET size              [ ] +
TRANSMITTED data PACKET size           [ ] +
RECEIVED data WINDOW size              [ ] #
TRANSMITTED data WINDOW size           [ ] #
CLOSED USER GROUP selection            [ ] #
CLOSED USER GROUP WITH OUTGOING ACCESS selection [ ] #
Recognized Private Operating Agency (RPOA) [ ]
User-Defined Facilities                 [ ]

----- CALL USER Data -----
Note: RFC-1356 (supersedes RFC-877) mandates
the first byte of call user data is 0xcc. If you
do not put 'cc' as the first byte, SMIT
will put it there for you.
Call User Data                          [ ]
```

Also create an entry for kilix on the Internet/X.25 SVC translation table of filix. This is necessary even if only kili makes the calls.

Note: It may happen that the translation table gets corrupted and contains duplicate entries. The simplest solution is to remove this table and create another. Here is how to remove it:

```
# cd /etc/objrepos
# odmdrop -o X25ip
```

This table will be re-created automatically with the default values. Simply add your new entries with SMIT.

You may also save the file `/etc/objrepos/X25ip`. This file can be restored and the table reinitialized by entering the command:

```
# x25ip
```

7.2.5 Testing the Connection

To test the TCP/IP connection, use the `ping` command. This command will start to send packets of 64 bytes across the network. To stop this command, hit the Interrupt key, usually **Ctrl-C**

```
# ping filix
PING filix.itsc.austin.ibm.com: (129.35.1.93): 56 data bytes
64 bytes from 9.3.5.62: icmp_seq=1. time=229. ms
64 bytes from 9.3.5.62: icmp_seq=2. time=106. ms
64 bytes from 9.3.5.62: icmp_seq=3. time=106. ms
```

You should receive a message with zero percent of packets lost. This demonstrates that a good connection has been established. A small percentage of packet loss can be attributed to call completion time and should not be of concern.

For the last test, if reasonable, shut down and reboot the system. Then try the `ping` command again. If it doesn't work, you have a problem with the execution of the `/etc/rc.net` script, which is reported in `/tmp/rc.net.out`. For more information on TCP/IP problem determination, see 9.8, "Diagnosing TCP/IP Problems" on page 191. This verifies your TCP/IP X.25 configuration.

Notes:

- A virtual call is made when the first session with a remote system is established.
- Several IP sessions can share the same virtual circuit. Once an X.25 communication has been established, all IP traffic between two systems will share the same virtual circuit. Only one SVC is needed for communication between two systems.
- The X.25 virtual circuit is left connected for a period of time after the last IP session has been closed. This is so the cost of call establishment is not incurred if the remote system is contacted again in the near future (which is reasonably likely). This time delay, which is approximately 20 minutes, is modifiable with the Network Options (no) command using the `arpt_killc` option. For example, to clear the virtual circuit after five minutes of inactivity:

```
# no -o arpt_killc=5
```

Note: The `no` command is used to modify some network parameters until the system is rebooted. If you want your new values to remain

unchanged after each system boot, you have to add the command at the end of the `/etc/rc.tcpip` file.

- To clear a TCP/IP X.25 virtual connection, both SVC and PVC, use the `arp -d` command. For example, to clear an existing virtual connection between `kilix` and `filix`, type on `filix`:

```
# arp -d filix
```

- To check the state of PVC or SVC connections, enter:

```
# arp -a
```

If the `arp` entries for the PVCs or SVCs are missing, enter the following to restore:

```
# x25ip
```

- If the IP address of a remote host is changed at the name server, enter the following command to update the IP/X.25 translate table:

```
# x25ip
```

- X.25 packets are buffered by TCP/IP using system *mbufs*. A shortage of *mbufs* may result in a performance degradation. You can display the number of *mbufs* assigned with:

```
# lsattr -E -l sys0
```

Or look at the `thewall` parameter with:

```
# no -a
```

You can display the usage of *mbufs* with:

```
# netstat -m
```

To change the maximum number of *mbufs*, use:

```
# chdev -l sys0 -a maxmbuf=.....
```

Or:

```
# no -o thewall=.....
```

7.2.6 Suppressing the X.25 LPP IP Interface

If you want to modify some X.25 parameters and you have defined an `xs0` interface, SMIT will display the following message:

```
Method error (/etc/methods/chgsx25):
```

```
0514-029 Cannot perform the requested function because a
child device of the specified device is not in a correct state.
```

To avoid this problem, you will have to remove the `xs0` interface with the `fastpath smit rminet`. When the modification is done, you re-create the IP/X.25 attachment with `fastpath smit mkinet1xs`. Verify also that there is no other application using the X.25 line, such as `x29d`.

Note: At your own risk, you may also bring the interface down by using the following command:

```
# rmdev -lxs0
```

To re-create it:

```
# mkdev -lxs0
# x25ip
```

Remember that, in this case, the attachment definition will be reset after the next system reboot.

7.3 Requesting the Use of a Facility with TCP/IP

It is possible to customize TCP/IP so that it uses optional X.25 facilities. For each host entry in the x25ip table, you can configure the use of X.25 facilities. From the SMIT menu, select:

```
Communications Applications and Services
-CP/IP
  -Further Configuration
    -Network Interfaces
      -Network Interface Selection
        -X.25 LPP IP Host Configuration
          -Change / Show a X.25 LPP IP Host Entry
```

Or use the fastpath: `smit chinetsx25`.

```
Change/Show an X.25 LPP IP SVC Host Entry

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
* Remote HOSTNAME                [filix]
* Remote DTE Address              [3106010761] #
----- Optional X.25 Facilities -----
RECEIVED data PACKET size        [ ] +
TRANSMITTED data PACKET size     [ ] +
RECEIVED data WINDOW size       [ ] #
TRANSMITTED data WINDOW size     [ ] #
CLOSED USER GROUP selection      [ ] #
CLOSED USER GROUP WITH OUTGOING ACCESS selection [ ] #
Recognized Private Operating Agency (RPOA) [ ]
User-Defined Facilities           [ ]

----- CALL USER Data -----
Note: RFC-1356 (supersedes RFC-877) mandates
the first byte of call user data is 0xcc. If you
do not put 'cc' as the first byte, SMIT
will put it there for you.
Call User Data                    [ ]
```

As you can see in this panel, you don't need to code the more common facility requests yourself. For the most common facilities, such as non-standard packet size or window size request, just enter the corresponding parameters and SMIT will build the request for you. If the facility you want to use is not listed, enter the facility request in hexadecimal in the User-Defined Facilities field.

The example shows how we have changed the packet sizes and requested to reverse the charge each time we call the TCP/IP host filix:

```

Change/Show an X.25 LPP IP SVC Host Entry

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                     [Entry Fields]
* Remote HOSTNAME                    [filix]
* Remote DTE Address                  [3106010761]      #
----- Optional X.25 Facilities -----
RECEIVED data PACKET size            [256]          +
TRANSMITTED data PACKET size         [256]          +
RECEIVED data WINDOW size            []              #
TRANSMITTED data WINDOW size         []              #
CLOSED USER GROUP selection          []              #
CLOSED USER GROUP WITH OUTGOING ACCESS selection []      #
Recognized Private Operating Agency (RPOA) []
User-Defined Facilities               [0101]

----- CALL USER Data -----
Note: RFC-1356 (supersedes RFC-877) mandates
the first byte of call user data is 0xcc. If you
do not put 'cc' as the first byte, SMIT
will put it there for you.
Call User Data                        []

```

Here is the x25mon trace of the call packet that is generated when we do a ping:
 sx25a0 PS 0x0014 CALL dN 1a:10 1f:5 1d:1 AA31060107613106010760054208080101CC

You will notice that the length of the facilities is 05, facility request for input and output packet size of 256 is 420808 and the reverse charging request is 0101.

7.4 Changing TCP/IP CUD Values

TCP/IP RFC 877 defines CC as the standard TCP/IP CUD value. By default, the X.25 LPP sends CUD CC in all outgoing TCP/IP call packets and routes all incoming call packets with CC to TCP/IP. Some hosts send CC000000 as the CUD value. For example, in cases where a remote host expects the RISC System/6000 to generate a CUD of CC000000, the user must customize the CUD value in the X.25 LPP IP Host Entry (x25ip). From the first menu of the SMIT interface, select:

```

Communications Applications and Services
-CP/IP
  -Further Configuration
    -Network Interfaces
      -Network Interface Selection
        -X.25 LPP IP Host Configuration
          -Change / Show a X.25 LPP IP Host Entry

```

Or use the fastpath: smit chinetsx25.

```

Change/Show an X.25 LPP IP SVC Host Entry

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                     [Entry Fields]
* Remote HOSTNAME                    [filix]
* Remote DTE Address                  [3106010761]      #
----- Optional X.25 Facilities -----
RECEIVED data PACKET size            []             +
TRANSMITTED data PACKET size         []             +
RECEIVED data WINDOW size            []             #
TRANSMITTED data WINDOW size         []             #
CLOSED USER GROUP selection          []             #
CLOSED USER GROUP WITH OUTGOING ACCESS selection []     #
Recognized Private Operating Agency (RPOA) []
User-Defined Facilities                []

----- CALL USER Data -----
Note: RFC-1356 (supersedes RFC-877) mandates
the first byte of call user data is 0xcc. If you
do not put 'cc' as the first byte, SMIT
will put it there for you.
Call User Data                        [CC000000]

```

7.5 Configuring SNMP

We assume that the `snmpd` daemon is running and all the necessary software needed to ensure that the `snmpd` is properly functioning is installed and working. You can find the X.25 proxy agent `x25smuxd` and the MIB definitions file `x25smuxd.defs` in the `/usr/sbin` directory.

Data collection from X.25 for SNMP is enabled and disabled by the configuration methods.

- Enabled at the time the first port is configured
- Disabled at the time the last port is unconfigured

Before invoking `x25smuxd` by the configuration methods at port configuration time, use the following installation procedure.

1. Log in as root
2. Check to see if the `/usr/sbin/x25smuxd.defs` file is installed.
3. In the `/usr/sbin/x25smuxd.defs` file, find the following line:

```
-- object definitions compiled from RFC1381-MIB { iso 3 6 1 2 1 }
```

Append every line from this line to the end of the file to the bottom of the `/etc/mib.defs` file.
4. Add the following entry to the bottom of the `/etc/snmpd.peers` file:

```
"x25smuxd" 1.3.6.1.2.1.10.16 "x25smuxd_password"
```
5. Add the following entry to the bottom of the `/etc/snmpd.conf` file:

```
smux 1.3.6.1.2.1.10.16 x25smuxd_password #x25smuxd
```

6. Refresh the snmpd daemon so that it rereads the /etc/snmpd.conf file with the following command:

```
refresh -s snmpd
```

Warning:

1. Only run x25muxd when you are logged in with root authority.
2. Never start more than one instance of x25muxd as it can cause conflicts with the interprocess communication mechanism.

Chapter 8. Accessing an SNA Network with X.25

In this chapter, we will use the X.25 connection defined in Chapter 3, "X.25 LPP Installation and Setup" on page 43, to connect our system kili to an SNA network and access the S/390 host. To make the connection, we will be using the SNA Server/6000 V3.1 licensed program product. This product is now marketed as "Communications Server for AIX". The 3270 terminal emulation program we will be using is 3270 Host Connection Program/6000 (HCON) Version 2.1.

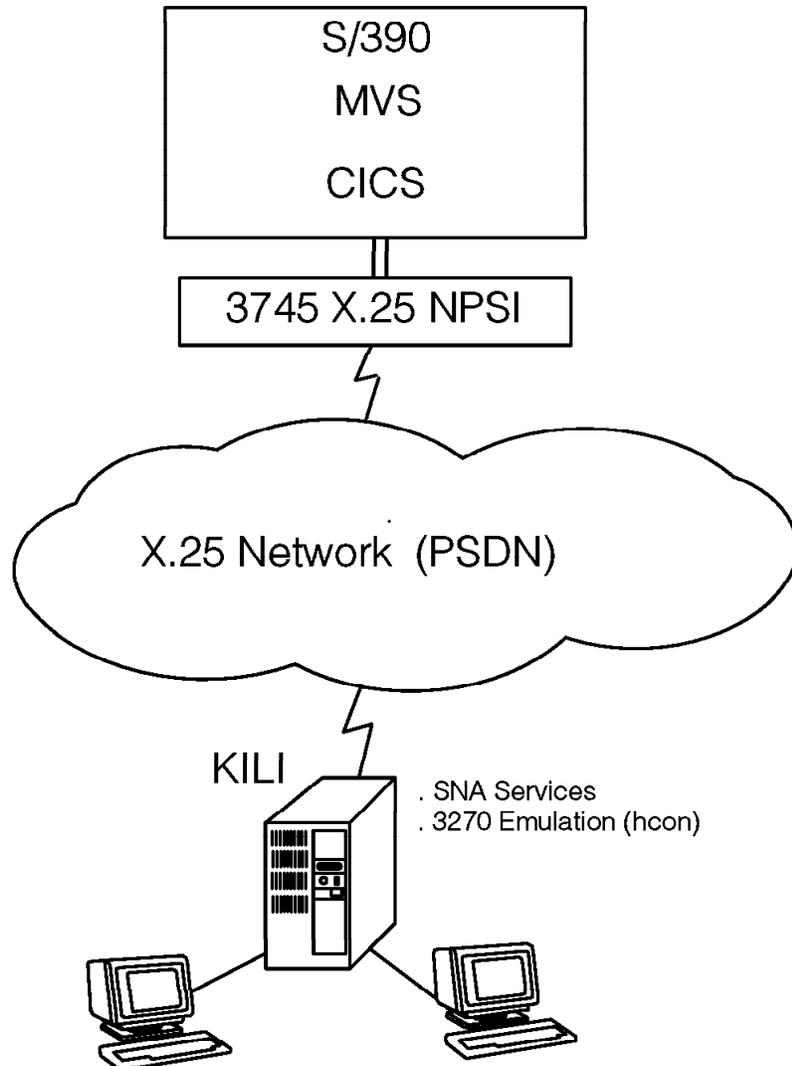


Figure 47. Scenario for an SNA Connection Using X.25

Before we start to configure our system, we will need to understand some background information on using SNA over X.25, along with the structure and functions of the SNA product on the RISC System/6000.

This chapter is intended for system administrators responsible for:

- Installing AIX SNA Server/6000
- Configuring the system to match the network to which it is connected

- Keeping the system running

SNA administrators should be familiar with an AIX node and operating procedures for the AIX operating system. Also, they should know the network to which the system is being connected and the general concepts of SNA. In addition, individuals assuming administrative responsibilities should know how to use the Systems Management Interface Tool (SMIT). Some of the tasks related to system administration are:

- Customizing AIX SNA
- Starting and stopping SNA
- Defining network security
- Getting network status

The SNA Server/6000 product provides menu dialogs, commands and procedures for system administration purposes.

8.1 QLLC with Reference to RISC System/6000 X.25 Support

Qualified Logical Link Control (QLLC) is only for SNA support. It permits the use of additional link control information that SNA needs, but X.25 does not. QLLC architecture specifies the mapping between Synchronous Data Link Control (SDLC) frames and X.25 packets.

When SNA is used over X.25, it uses the Qualifier-bit (Q-bit) in the X.25 packet header to indicate special link control information. This information is relevant for SNA control between the two systems communicating with each other, but it is of no concern to X.25 link control. These qualified packets help SNA to determine who is calling whom between the two communicating systems, and it will indicate maximum message sizes and so on.

QLLC must be used in the following situations:

- When two RISC System/6000s are communicating with each other using SNA over X.25
- When a RISC System/6000 is communicating with a host, for example a System/390, using SNA over X.25

Some systems want to control the segmenting and de-segmenting of messages themselves, rather than leaving it to the X.25 protocol. They use additional control on top of X.25 to perform this. The RISC System/6000 does not support this additional control of Logical Link Control (LLC) headers, imbedded in X.25 packets, to segment messages. The RISC System/6000 will leave it to the X.25 protocol to handle segmenting.

8.2 Introduction to SNA Server/6000

AIX SNA Server/6000 implements two SNA components that control the operation of the local node in the network. These components are:

- Physical Unit (PU)
 - SNA Server/6000 provides the capabilities of a PU Type 2 node.
- Logical Units (LUs) classified as one of the following types:

LU 0 Program-to-program communications

This logical unit is used by certain host database systems (such as IMS/MVS) and some point-of-sale systems for the retail and banking industries. Current releases of these products also support communications using LU 6.2, which is the preferred protocol for new applications.

SNA Server/6000 supports up to 256 LU 0 type LUs, with primary and secondary server capabilities:

- LU 0 primary server supports communication with PU type 2 nodes. It provides simple functions that communicate directly with the SDLC device driver. LU 0 primary servers can only be used on EIA232D and EIA422A links.
- LU 0 secondary server supports dependent communication with a host system, emulating a PU type 2 node. It is an application written to the generic SNA device driver. LU 0 secondary servers can be used on any of the links supported by SNA Server/6000.

LU 1 Host program to RJE workstation

This is used for application programs and single and multiple device data processing workstations communicating in an interactive, batch-data transfer or distributed-data processing environment.

SNA Server/6000 supports LU 1 for dependent communication using the SNA character string or Document Character Architecture (DCA) data streams. This enables it to provide workstation emulation through Transaction Programs (TPs). It typically supports printing, card reading and card punching devices. You can write TPs for LU 1 using the Operating System Subroutines or Library Subroutines Application Programming Interfaces (APIs).

LU 2 Host program to display workstation

Used by application programs and display workstations to communicate in an interactive environment. In addition, it is used by some programs to communicate with hosts that normally provide output to 3270 display devices. SNA Server/6000 supports LU 2 for file transfers and for dependent communication using the SNA 3270 data stream.

LU 3 Host program to 3270 printer

Used by application programs and printers using the SNA 3270 data stream. SNA Server/6000 supports LU 3 for dependent communications, which enables the SNA server to provide printer emulation through TPs.

LU 6.2 Advanced Program-to-Program Communications (APPC)

Can be used for communication between two type-5 nodes, a type-5 node and a type-2.0 or type-2.1 node, or two type 2.1 nodes. This LU type provides more function and greater flexibility than any previous LU type. Unless you are constrained by existing hardware or software, LU 6.2 is the logical choice when developing new applications.

SNA Server/6000 supports independent LU 6.2 for communication in peer networks and dependent LU 6.2 for communication in subarea networks. You can write TPs for LU 6.2 using the CPI Communications, Operating System Subroutines or Library Subroutines APIs. You can also use the same configuration profiles and TPs for both independent and dependent LU 6.2.

8.2.1 SNA Application Programming Interfaces (APIs)

SNA Server/6000 provides several APIs, each consisting of a set of subroutines that enables TPs to communicate with LUs.

Transaction programmers choose APIs depending on the type of LU for which a TP is written and the requirements of the application. Each TP uses a single API, but TPs written for the same LU type, using different APIs, can still communicate with each other.

SNA Server/6000 provides the following APIs. With the exception of CPI Communications (CPI-C) API, the SNA Server/6000 APIs are supported only on SNA Server/6000 nodes.

8.2.1.1 CPI Communications API

This API provides a common interface across different SNA platforms. TPs written to this API can run without modification on nodes that support IBM Systems Application Architecture (SAA) CPI-C.

This API is used only for LU 6.2 TPs.

8.2.1.2 Operating System Subroutines API

This API uses AIX subroutines to support TPs running under SNA Server/6000. It can be used to write TPs for LU types 1, 2, 3 and 6.2.

8.2.1.3 Library Subroutines for TP Conversations

This API uses SNA Server/6000 subroutines to support TPs running under SNA Server/6000. It can be used to write TPs for LU types 1, 2, 3 and 6.2.

8.2.1.4 LU 0 APIs

Two APIs are available for use with LU 0: LU 0 Primary API and LU 0 Secondary API.

- The LU 0 Primary API can be used to write TPs to communicate with an LU 0 TP on a PU type 2 node.
- The LU 0 Secondary API can be used to write TPs to communicate with an LU 0 TP running on a host.

8.2.1.5 Generic SNA API

This API provides subroutines that interact directly with the PU services component of SNA Server/6000. These subroutines can be used to program SNA functions that are not available from SNA Server/6000.

8.2.1.6 Network Management APIs

SNA Server/6000 includes two sets of subroutines for network management: Library Subroutines API and Management Services API.

- The Library Subroutines API includes network management subroutines that can be used on a RISC System/6000 workstation in a subarea network.
- The Management Services API enables a RISC System/6000 workstation configured as a node in an APPN network to function as a Management Services (MS) entry point.

For more information about SNA Server/6000 APIs, see *AIX SNA Server/6000: Transaction Program Reference*.

8.2.2 SNA Configuration Profiles

AIX SNA Server/6000 keeps all network information in structured files called profiles. It stores the profiles as database files that are controlled by system data management routines.

The following figure shows the profiles needed to use the 3270 host connection program with X.25:

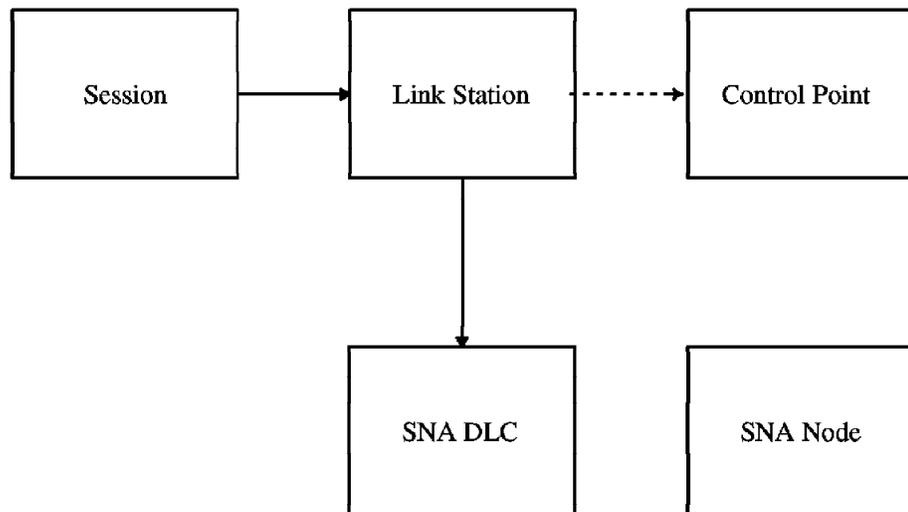


Figure 48. Generic AIX SNA Server/6000 Profiles for HCON

The profiles and their functions are as follows:

SNA Node: Describes the operating parameters for SNA Server/6000 on the local node. This profile is created automatically during SNA Server/6000 installation, but it can be modified to suit your needs. There can only be one SNA Node Profile per RISC System/6000.

Control Point: Provides identifying information for the control point (CP) on the local system - for example the CP name, network name, XID node ID - and indicates the role of the node in the network. This profile is also created automatically during SNA Server/6000 installation, but it must be modified to suit your needs. There can only be one control point profile per RISC System/6000.

Link Definitions: SNA Server/6000 requires information about each link to the network. The attributes of a link can be divided into two parts:

- Physical and logical characteristics of a port
- Control characteristics of the link station

You specify the attributes for each link using the SNA DLC and Link Station Profiles.

SNA DLC: Identifies the adapter device driver and specifies the adapter characteristics. This profile also provides APPN routing and recovery procedures for all dynamic link stations.

Link Station: Describes the control characteristics that support a link to the network.

Session: Specifies the characteristics of a local LU and parameters for a session with a remote LU over a specific link station.

8.2.3 SNA Configuration Database and Commands

SNA Server/6000 maintains two separate Object Database Management (ODM) databases: the *working database* and the *committed database*. The working database is used as a working area in which profiles can be added and altered. The committed database contains the set of profiles that SNA Server/6000 uses while running. The SNA Server/6000 commands are used to maintain the validity of the two separate databases.

When configuration profiles are added or changed, they are stored in the working configuration database. Before these profiles can be used, the `verifysna` command must be used to ensure that the profiles are correct and internally consistent. See Figure 49 on page 155.

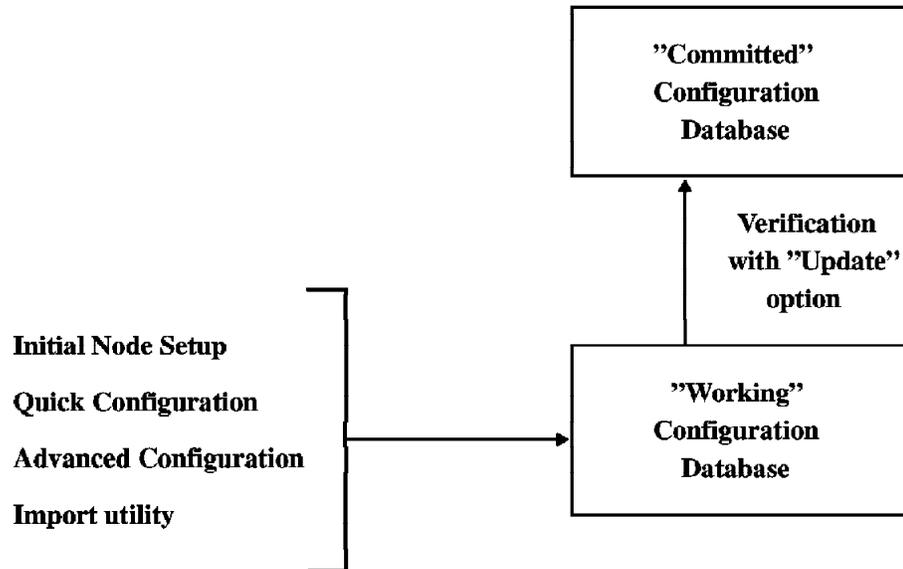


Figure 49. AIX SNA Server/6000 Configuration Process

A set of configuration commands is provided with AIX SNA Server/6000 to allow you to set up the network. These commands can be accessed directly from the AIX command line or through the SMIT interface. The commands explained here are only those used for our example LU 2 setup. For more information on configuration, see *AIX SNA Server/6000: Configuration Reference*.

mksnaobj

Creates a profile in the working configuration database.

chsnaobj

Changes the values in an already defined profile in the working configuration database.

rmsnaobj

Removes a profile.

qrysnaobj

Retrieves the values of all fields, in stanza format, in an already defined profile.

helpsnaobj

Provides online help for the mksnaobj and chsnaobj commands.

lssnaobj

Lists the profiles in the configuration database that meet a specified criteria.

verifysna

Verifies that all the profiles in the working database are consistent and error-free. If the update option is used, the committed database is changed, provided that all the profiles in the working database verify. This command does not change the working database in either case.

sna

When used with the options below, controls the SNA server. When used without arguments, it displays help information for the command.

- display** Displays SNA status information, including information about sessions and links, and local directory and topology databases
- setlogs** Configures the SNA service files (files containing trace and error information)
- start** Starts SNA, link stations and sessions using various criteria
- stop** Stops SNA, link stations and sessions using various criteria

For more detailed information on these and other SNA Server/6000 commands, see *AIX SNA Server/6000: Command Reference*.

8.2.4 SNA Server Components

AIX SNA Server/6000 is set of AIX utilities that includes the following components:

- Device drivers
- Kernel extensions
- Daemon processes

SNA Server/6000 utilities enable TPs running on the RISC System/6000 workstation to access SNA network resources using the AIX input/output (I/O) device interface.

8.2.4.1 SNA Server/6000 Directories

This section lists the AIX directories that can contain SNA Server/6000 files. The files may be command files, executable files, configuration files or script files.

/etc/

An AIX directory which contains the rc.sna shell script, used for automatically starting SNA.

/etc/objrepos/sna/work/

SNA Server/6000 directory containing the configuration profile for the working database.

/etc/objrepos/sna/verified/

SNA Server/6000 directory containing the configuration profile for the committed database.

/usr/adm/ras/

An AIX directory containing AIX trace and error logging information.

/usr/bin/
An AIX directory containing AIX and SNA Server/6000 commands.

/usr/include/
An AIX directory containing files to support SNA Server/6000 applications programming.

/usr/lib/
An AIX directory containing libraries to support SNA Server/6000 application programming.

/usr/lib/drivers/
An AIX directory containing SNA Server/6000 device drivers.

/usr/lib/methods/
An AIX directory containing SNA Server/6000 kernel extensions configuration commands.

/usr/lpp/lu0/
SNA Server/6000 directory containing files to support LU 0 application programming.

/usr/lpp/lu0/bin/
SNA Server/6000 directory containing LU 0 executable files.

/usr/lpp/lu0/samples
SNA Server/6000 directory containing sample files for LU 0.

/usr/lpp/msg/En_US/
An AIX directory containing the SNA Server/6000 message catalogs.

/usr/lpp/sna/
SNA Server/6000 directory containing installation files and general information files.

/usr/lpp/sna/bin/
SNA Server/6000 directory containing executable files, daemons and installation programs.

/usr/lpp/sna/samples/
SNA Server/6000 directory containing sample TPs.

/usr/lpp/sna/samples/bin/
SNA Server/6000 directory containing APPC sample TPs.

/usr/lpp/sna/samples/doc/
SNA Server/6000 directory containing notes on sample TPs and configuration profiles provided with the LPP.

/usr/lpp/sna/samples/xmpprof/
SNA Server/6000 directory containing sample SNA Server/6000 configuration profiles.

/var/lu0/
SNA Server/6000 directory containing LU 0 configuration data and trace results.

/var/sna/
SNA Server/6000 directory containing internal error log and trace data.

8.3 AIX SNA Server/6000 Setup

This section gives a step-by-step description of how to customize SNA Server to use 3270 emulation over an X.25 network. The setup will correspond to the host definitions listed in 8.3.3, "Getting Host Definitions" on page 160.

The steps listed in the box are described in the subsequent sections.

AIX SNA Server/6000 Setup

1. Install the SNA Server/6000 LPP
2. Install the appropriate DLC device driver
3. Get host definitions
4. Perform initial node setup
5. Define the SNA profiles for LU 2 setup:
 - Change the SNA control point profile
 - Define the SNA DLC profile
 - Define the SNA link station profile
 - Define the SNA X.25 optional facilities profile
 - Define the SNA session profiles
6. If appropriate, verify the SNA profiles with the update option
7. Start SNA
 - Start the SNA server
 - Start the SNA link station
 - List the current status of SNA

8.3.1 Installation of the AIX SNA Server/6000 LPP

The SNA Server/6000 licensed program product, program number 5765-247, consists of the following parts:

sna.sna.obj Contains the SNA Server/6000 base program.

sna.lu0.obj Contains the SNA Server/6000 LU 0 facility. Sna.sna.obj is a prerequisite for this component.

snam/language.msg Contains the messages and help information in the specified language for the run-time environment. If you install multiple languages for a product, ensure that you install the preferred (or primary) language first. For example, to install U.S. English as the primary language, you would install snamEn_US.msg. Sna.sna.obj is a prerequisite for this component.

Also available as options in the SNA Server/6000 product family are the following orderable LPPs.

gw.sna.obj

SNA Gateway/6000 Product – requires sna.sna.obj to be installed.

sna.brwsr.obj

DynaText browser utility – used to view SNA Server/6000 softcopy publications; sna.sna.obj and X11rte.obj are prerequisites for this component.

sna.usdoc.obj

AIX SNA Server/6000: User's Guide – softcopy publication (sna.sna.obj and sna.brwsr.obj are prerequisites).

sna.crdoc.obj

AIX SNA Server/6000: Configuration Reference – softcopy publication (sna.sna.obj and sna.brwsr.obj are prerequisites).

sna.dgdoc.obj

AIX SNA Server/6000: Diagnosis Guide and Messages – softcopy publication (sna.sna.obj and sna.brwsr.obj are prerequisites).

sna.cmdoc.obj

AIX SNA Server/6000: Command Reference – softcopy publication (sna.sna.obj and sna.brwsr.obj are prerequisites).

sna.tpdoc.obj

AIX SNA Server/6000: Transaction Program Reference – softcopy publication (sna.sna.obj and sna.brwsr.obj are prerequisites).

sna.gwdoc.obj

AIX SNA Gateway/6000: User's Guide – softcopy publication (sna.sna.obj and sna.brwsr.obj are prerequisites).

The licensed program may also contain update files.

8.3.2 Installation of the Data Link Control

Each data link control interface has a special file in the /dev directory.

To show the predefined data link controls:

```
# lsdev -P -H -c dlc
class type          subclass description
dlc tokenring       dlc      Token-Ring Data Link Control
dlc sdlc            dlc      SDLC Data Link Control
dlc x25_qllc        dlc      X.25 QLLC Data Link Control
dlc fddi            dlc      FDDI Data Link Control
dlc ethernet       dlc      Standard Ethernet Data Link Control
dlc IEEE_ethernet dlc      IEEE Ethernet (802.3) Data Link Control
```

Each of the above corresponds to one of the physical interfaces supported by SNA communications in AIX.

To install the X.25 QLLC Data Link Control for the X.25 Co-Processor, from the first menu of the SMIT interface, select:

```
Devices
-Communication Devices
  -X.25 Co-Processor/2 or Multiport/2 Adapter
    -Services
      -Data Link Controls
        -Add a QLLC Data Link Control
```

Or use the fastpath: smit cmddlc_qllc_mk

The AIX-generated command is:

```
mkdev -c dlc -s dlc -t x25_qllc
```

If using the Portmaster Adapter:

Devices

 -Communication Devices

 -Portmaster Adapter/A

 -Services

 -Data Link Controls

 -Add an SDLC Data Link Control

Or use the fastpath: `smit cmdclc_sdclc_mk`

The AIX-generated command is:

```
mkdev -c dlc -s dlc -t sdlc
```

8.3.3 Getting Host Definitions

Before configuring SNA Server/6000, some information must be requested from the VTAM administrator for the SNA node to which we will connect. The following table is the minimum information required for this example:

VTAM Parameter	Value	RISC System/6000 Profile
Network User Address	3106001984	Attachment
IDBLK	071	3270 LU profiles
IDNUM	06000	3270 LU profiles
SSCP ID	20	3270 LU profiles
LOCADDR	2 3	3270 LU profiles

Network User Address: The X.25 address, 3106001984, uniquely identifies the X.25 line to which the S/370 host is attached.

This parameter will be used in the attachment profile definition.

IDBLK The three hexadecimal digits of this parameter provide an identifier, or block number, that is unique to each product on the network. For the RISC System/6000, this number is 071.

IDNUM The five hexadecimal digits of this *ID number* distinguish a specific piece of equipment from all others of a similar kind on the network. The number usually is given by the VTAM administrator. In our case, an ID number of 06000 was assigned to our RISC System/6000.

SSCPID The ID of the controlling System Services Control Point (SSCP) in the SNA network (decimal value).

LOCCADR These are the local addresses of the 3270 displays and printers in the 3270 cluster. One decimal value is used for each display or printer connected.

8.3.4 SNA Profiles for an X.25 LU 2

Figure 50 shows the profiles needed to configure a logical unit type 2 (LU 2):

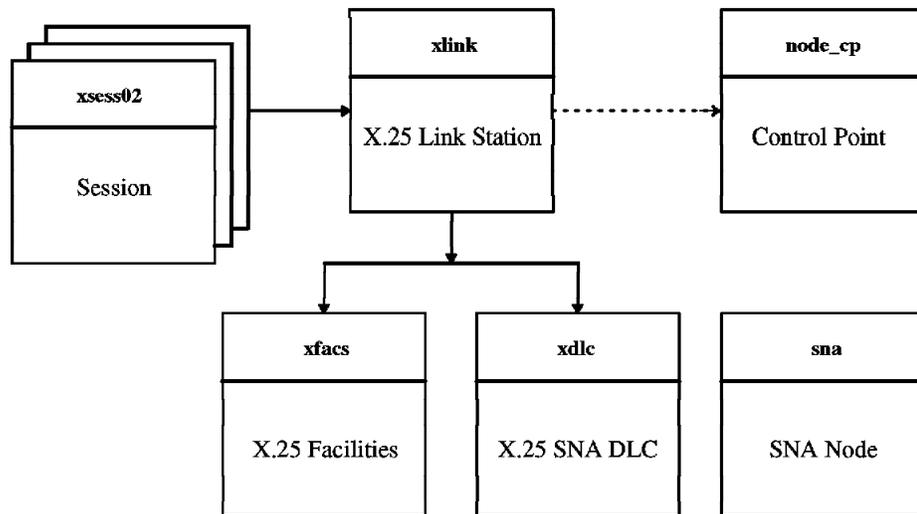


Figure 50. X.25 AIX SNA Server/6000 Profiles for HCON

8.3.5 Performing Initial Node Setup

Initial Node Setup configures the node control point and a single link station. It enables you to create the basic configuration of SNA Server/6000 while providing a minimal amount of information. Although we are going to use advanced configuration to configure the control point, the information we enter in the initial node setup is required to verify profiles and run SNA Server/6000, so we must still perform this part.

To enter the initial node setup information from the first menu of SMIT, select:

```
Communications Applications and Services
  -SNA Server/6000
    -Configure SNA Profiles
      -Initial Node Setup
```

Or use the fastpath: `smit _snainit`.

This displays the Initial Node Setup screen:

```

                                Initial Node Setup

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
Control Point name                [AUSCP]
Control Point type                appn_end_node      +
Local network name                [USIBMRA]
XID node ID                       [*]

Optional link station information:

Link station type                 x.25_svc
Link station name                 []
* Calling link station?           yes              +
Remote X.25 station address       [3106001984]      X

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do

```

Here we have defined:

Control Point name:

If you are connecting to a host, the value in this field should match the CPNAME= entry in the host's VTAM PU statement.

Control Point type:

Determines whether this machine is to act as an intermediate session routing node or not. In this case, we have defined that the local system accesses network services through an associated network node. It maintains directory information only for the local system.

XID node ID:

If connecting to a host over a switched connection, enter a value to support the LU definitions. The value should consist of the IDBLK= (a three digit hexadecimal) and IDNUM= (a five digit hexadecimal) entries in the VTAM PU definition.

Link station type:

Describes the DLC type for a link station for this node.

Calling link station?:

This value is required if using X.25 link stations using switched virtual circuits, as well as for token-ring and ethernet link stations.

Remote X.25 station address:

Address of the remote X.25 link station. Must be specified if using switched virtual circuits.

8.3.6 Setting Up the SNA profiles for LU 2

SNA LU 2 setup requires the modification of several profiles.

8.3.6.1 Changing the Control Point Profile

The control point profile describes the characteristics of the control point on the local RISC System/6000. A RISC System must have one (and only one) control point profile. The control point supports multiple independent LU sessions and a single dependent LU session. When the node participates in an APPN network, it supports APPN network node and end node functions. The SNA Server/6000 control point acts as both a logical unit (LU) and a physical unit (PU):

- As an LU, it identifies itself to the CP of the adjacent SNA node during CP-CP sessions, enabling the node to access APPN support services such as directory searches and route selection.
- As a PU, the CP automatically assumes control over all available link stations on the RISC System/6000 and uses them to support APPN operations with adjacent SNA nodes.

To enter the control point profile information from the first menu of SMIT, select:

```
Communications Applications and Services
- SNA Server/6000
  - configure SNA Profiles
    - Advanced Configuration
      - Control Point
        - Change/Show a Profile
```

Or use the fastpath: `smit _snacpch`.

This displays the Change/Show Control Point Profile screen:

```
Change/Show Control Point Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
* Profile name                    node_cp
XID node ID                       [*]
Network name                      [USIBMRA]
Control Point (CP) name           [AUSCP]
Control Point alias               [AUSCP]
Control Point type                 appn_end_node      +
Maximum number of cached routing trees [500]          #
Maximum number of nodes in the TRS database [500]      #
Route addition resistance          [128]              #

Comments                          []

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do
```

Here we have defined:

XID Node ID:

If connecting to a host over a switched connection, enter a value to support the LU definitions. The value should consist of the IDBLK= (a three digit hexadecimal) and IDNUM= (a five digit hexadecimal) entries in the VTAM PU definition.

Network name:

Distinguishes this network from other networks to which the machine can be connected. This name combined with the control point name, forms a unique identifier (the fully qualified name) for this control point.

Control Point (CP) Name:

If you are connecting to a host, the value in this field should match the CPNAME= entry in the host's VTAM PU statement.

Control Point type:

Determines whether this machine is to act as an intermediate session routing node or not. In this case, we have defined that the local system accesses network services through an associated network node. It maintains directory information only for the local system.

Cached routing trees:

This field applies only to network nodes.

Nodes in the TRS database:

This is the maximum number of APPN CPs that can be stored in the APPN topology routing services (TRS) database. If this CP is an end node (as in this example), this value should be greater than or equal to the number of network nodes, end nodes or LEN nodes adjacent to this node.

Route addition resistance:

This field applies only to network nodes.

8.3.6.2 Defining the Data Link Control Profile

The Data Link Control (DLC) is a set of communications protocols that supports orderly exchanges of data over a link. The DLC manages a communication adapter. SNA Server/6000 requires you to supply information about DLC characteristics in the SNA DLC profile. This profile describes the characteristics of a communication port on the RISC System/6000 workstation. A single port can support one or more link stations.

To enter the DLC profile information, from the first menu of SMIT, select:

```

Communications Applications and Services
  -SNA Server/6000
    -Configure SNA Profiles
      -Advanced Configuration
        -links
          -X.25
            -X.25 SNA DLC
              -Add a Profile

```

Or use the fastpath: `smit _snaX25linkmk`.

This displays the Add X.25 SNA DLC Profile screen:

```

                                Add X.25 SNA DLC Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                [Entry Fields]
* Profile name                        [xd1c]
Data link device name                  [x25s0]          +
Force disconnect time-out (1-600 seconds) [120]           #
User-defined maximum I-Field size?     no              +
    If yes, Max. I-Field size (265-30729) [1417]         #
Max. num of active link stations (1-255) [100]          #
    Number reserved for inbound activation [0]            #
    Number reserved for outbound activation [0]            #
Local X.25 network address              [3106010760]   #
Receive window count (1-127)            [7]             #
Transmit window count (1-127)           [7]             #

Secondary and Negotiable Stations
    Secondary inactivity time-out (1-255 sec) [30]        #

Primary and Negotiable Stations
    Primary repoll frequency (1-255 seconds) [30]          #
    Primary repoll count (1-255)            [10]           #

Link Recovery Parameters
    Retry interval (1-10000 seconds)         [60]             #
    Retry limit (0-500 attempts)            [20]             #

Comments                               []
[BOTTOM]

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit        Enter=Do

```

8.3.6.3 Defining the Link Station Profile

A link station provides control function for a single link that connects the local node to an adjacent node in the SNA network. The link station performs the following tasks:

- Controls link activations and deactivations
- Informs the node CP of link status
- Communicates with the adjacent link station across the physical medium

Link stations can either call or listen:

- A calling link station initiates the activation of a link
- A listening link station waits for a link activation request from a calling link station

To enter the link station profile information from the first menu of SMIT, select:

Communications Applications and Services

→SNA Server/6000

→Configure SNA Profiles

→Advanced Configuration

→links

→X.25

→X.25 Link Station

→Add a Profile

Or use the fastpath: `smit _snaX25attcmk`.

This displays the Add X.25 Link Station Profile screen:

Add X.25 Link Station Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

```

[TOP]
* Profile name [xlink]
  Use APPN Control Point's XID node ID? yes +
    If no, XID node ID [07100123]
* SNA DLC Profile name [xdlc] +
  Stop link station on inactivity? no +
    If yes, Inactivity time-out (0-10 minutes) [0] #
  LU address registration? no +
    If yes, LU Address Registration Profile name [] +
  Trace link? no +
    If yes, Trace size long +
  X.25 level 1984 +
  Station type secondary +

Adjacent Node Identification Parameters
  Verify adjacent node? no +
  Network ID of adjacent node []
  CP name of adjacent node []
  XID node ID of adjacent node (LEN node only) [*]
  Node type of adjacent node learn +

Link Activation Parameters
  Solicit SSCP sessions? yes +
  Initiate call when link station is activated? yes +
  Virtual circuit type switched +
    If permanent,
      Logical channel number of PVC (1-4095) [1] #
    If switched,
      Listen name [IBMQLLC]
      Remote station X.25 address [3106001984] #
      X.25 Optional Facilities Profile name [xfacs] +
  Activate link station at SNA start up? no +
  Activate on demand? no +
  CP-CP sessions supported? no +
    If yes,
      Adjacent network node preferred server? no +
      Partner required to support CP-CP sessions? no +
      Initial TG number (0-20) [0] #

Restart Parameters
  Restart on activation? no +
  Restart on normal deactivation? no +
  Restart on abnormal deactivation? no +

Transmission Group COS Characteristics
  Effective capacity [9600] #
  Cost per connect time [128] #
  Cost per byte [128] #
  Security public_switched_network> +
  Propagation delay packet_switched_network> +
  User-defined 1 [128] #
  User-defined 2 [128] #
  User-defined 3 [128] #

```

Comments []
[BOTTOM]

```

F1=Help      F2=Refresh    F3=Cancel    F4=List
F5=Reset     F6=Command   F7=Edit     F8=Image
F9=Shell     F10=Exit     Enter=Do

```

Here we have defined:

Use APPN Control Point's XID node ID?

Determines if the link station is controlled by the node's APPN CP. If specifying no, you must specify the XID node ID for the PU represented by this link station.

XID node ID:

If connecting to a host over a switched connection, enter a value to support the LU definitions. The value should consist of the IDBLK= (a three digit hexadecimal) and IDNUM= (a five digit hexadecimal) entries in the VTAM PU definition.

Stop link station on inactivity:

Specifies whether SNA Server/6000 stops the link station when there is no traffic on the links for a specified period of time. This choice can affect system performance, as each open link station uses system resources.

LU address registration?

Used to indicate if generic LU addresses are registered to this link station. Generic SNA is used to create special SNA functions that are not available from SNA Server/6000.

X.25 level:

Indicates, by date, the release of the CCITT X.25 Recommendation that is supported. SNA Server/6000 V2.1 currently only supports 1980 and 1984, although the X.25 LPP also supports 1988.

Adjacent Node Identification Parameters:

This set of fields enables you to identify the adjacent node to which this link station profile provides a link.

Link Activation Parameters:

This set of fields describes the behavior of the link station within the network.

Listen name:

Specifies the name configured in the routing table for routing incoming calls to Qualified Logical Link Control (QLLC).

Transmission Group COS Characteristics:

This set of fields shows adapter-specific settings for the communications devices used by this link station. You can, however, override these settings and specify a nonstandard COS or set of TG characteristics for a link station by changing the values displayed in these fields.

To specify non-standard characteristics for a series of link stations or for specialized communications adapters, you should consider changing the SNA Server/6000 adapter definition for the communications device in question.

8.3.6.4 Defining X.25 Optional Facilities Profile

This enables you to specify optional facilities, services and restrictions for X.25 link stations.

Several different X.25 link stations can share the same X.25 optional facilities profile, so you could specify a set of customized X.25 functions in a single profile and then apply it to an array of X.25 link stations.

To enter the X.25 optional facilities profile information from the first menu of SMIT, select:

```

Communications Applications and Services
  -SNA Server/6000
    -Configure SNA Profiles
      -Advanced Configuration
        -links
          -X.25
            -X.25 Optional Facilities
              -Add a Profile
  
```

Or use the fastpath: `smit _snaX25optsmk`.

This displays the Add X.25 Optional Facilities Profile screen:

```

                                Add X.25 Optional Facilities Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                [Entry Fields]
* Profile name                        [xfacs]
Throughput class for received data    9600          +
Throughput class for transmitted data 9600          +
Closed user group?                    no           +
  If yes, Index to closed group       [0]          #
Closed user group outgoing access?    no           +
Network user identification?          no           +
  If yes,
    Network user ID name               []
    Network user ID in hex?           no           +
Reverse charging?                     no           +
RPOA?                                  no           +
  If yes, Data network ID codes       []
Packet size for received data         128          +
Packet size for transmitted data      128          +

Comments                              []
[BOTTOM]

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command    F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do
  
```

8.3.6.5 Defining the LU 2 Session Profile

The session profile describes the characteristics of a dependent LU 1, 2 or 3. Each type of LU on the local node must be defined to SNA Server/6000 in a session profile. SNA Server/6000 uses this information to process inbound link requests for the local LU that come from remote hosts or peer systems.

To enter the LU 2 session profile information from the first menu of SMIT, select:

```

Communications Applications and Services
  -SNA Server/6000
    -Configure SNA Profiles
      -Advanced Configuration
        -Sessions
          -LU 2
            -Add a Profile

```

Or use the fastpath: `smit _snasess2mk.`

This displays the Add LU 2 Session Profile screen:

```

                                Add LU 2 Session Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
* Profile name                    [xsess02]
Local LU name                     [S4080502]
* Local LU address (1-255)        [2] #
System services control point
  (SSCP) ID (*, 0-65535)          [*]
Link Station Profile name         [xlink] +
Network name                      []
Remote LU name                    []
Maximum number of rows           [24] #
Maximum number of columns        [80] #

Comments                          []

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do

```

Here we have defined:

Local LU name:

If you are connecting to a host system, the Local LU name is the assigned name of the resource found in the VTAM LU definition statement.

Local LU address:

This is the address that remote systems use as the destination address when they send information to the local LU. On a VTAM system, the local LU address is defined by the `LOCADDR=` keyword in the VTAM LU definition statement.

SSCP ID: This is the ID of the System Services Control Point (SSCP) of the host system that controls this LU in the SNA network. This value should be in *decimal*. The system uses this value to generate a four-digit hexadecimal value that corresponds to the `SSCP ID=` parameter in the VTAM `START` statement on the host.

If you require that any host be able to initiate sessions with this LU, leave it as the default value of `*` (an asterisk). That way, any host

whose local LU address matches that given in this profile, can initiate sessions.

Maximum number of rows:

The maximum number of displayed rows that the LU should permit. If this is smaller than the number of lines of output that the host expects, SNA Server/6000 rejects bind requests with this LU. VTAM requires a screen size of 24 rows; using a value less than 24 can cause a VTAM error.

Maximum number of columns:

The maximum number of displayed columns that the LU should permit. If this is different to the number of lines of output that the host expects; some hosts can reject bind requests with this LU. VTAM requires a screen size of 80 columns; using a value less than 80 can cause a VTAM error.

8.3.6.6 Defining the LU 3 Session Profile

To enter the LU 3 Session Profile information from the first menu of SMIT, select:

```
Communications Applications and Services
-SNA Server/6000
  -Configure SNA Profiles
    -Advanced Configuration
      -Sessions
        -LU 3
          -Add a Profile
```

Or use the fastpath: `smit _snasess3mk`.

This displays the Add LU 3 Session Profile screen:

```

                                     Add LU 3 Session Profile
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                     [Entry Fields]
* Profile name                       [xsess09]
Local LU name                         [S4080509]
* Local LU address (1-255)           [9] #
System services control point
(SSCP) ID (*, 0-65535)               [*]
Link Station Profile name            [xlink] +
Network name                         []
Remote LU name                       []
Maximum number of rows               [24] #
Maximum number of columns            [80] #
Comments                             []

F1=Help      F2=Refresh    F3=Cancel    F4=List
F5=Reset     F6=Command   F7=Edit     F8=Image
F9=Shell     F10=Exit      Enter=Do
```

Here we have defined:

Local LU name:

If you are connecting to a host system, the Local LU name is the assigned name of the resource found in the VTAM LU definition statement.

Local LU address:

This is the address that remote systems use as the destination address when they send information to the local LU. On a VTAM system, the local LU address is defined by the LOCADDR= keyword in the VTAM LU definition statement.

SSCP ID: This is the ID of the System Services Control Point (SSCP) of the host system that controls this LU in the SNA network. This value should be in *decimal*. The system uses this value to generate a four-digit hexadecimal value that corresponds to the SSCP ID= parameter in the VTAM START statement on the host.

If you require that any host be able to initiate sessions with this LU, leave it as the default value of * (an asterisk). That way, any host whose local LU address matches that given in this profile can initiate sessions.

Maximum number of rows:

The maximum number of displayed rows that the LU should permit. If this is smaller than the number of lines of output that the host expects, SNA Server/6000 rejects bind requests with this LU. VTAM requires a screen size of 24 rows; using a value less than 24 can cause a VTAM error.

Maximum number of columns:

The maximum number of displayed columns that the LU should permit. If this is different to the number of lines of output that the host expects, some hosts can reject bind requests with this LU. VTAM requires a screen size of 80 columns; using a value less than 80 can cause a VTAM error.

8.3.7 Verifying SNA Profiles

After you have created or modified configuration profiles, they remain in a working database. They are not used by SNA Server/6000 until you verify and promote them to the committed database that SNA Server/6000 uses when it is running.

In addition, you must choose to update the existing configuration database when you verify profiles. If verification is successful, the verified profiles can be moved to the committed database.

The following update options are available when you verify the configuration database:

No update

Profiles are verified, but the configuration database is not updated.

Normal update

This can only be performed when the SNA Server/6000 is not active. Following a successful normal update, configuration profiles are used by SNA Server/6000 when it is next started.

Dynamic update

This promotes profiles to the committed database while SNA Server/6000 is running. If this option is used while the server is not running, a normal update is performed.

Dynamic updates affect any new SNA Server/6000 control structures that are initiated following the update. Resources that were already active prior to the update will continue to use control structures that were created prior to the update. For an active resource to use updated information, you must stop and restart the resource.

Dynamic update modifies the Local LU (also Partner LU and Mode Profiles if using LU 6.2) only if there are no active sessions using these profiles. If there is an active session using a profile, the update for that profile does not take effect until you restart SNA Server/6000, or until you stop any sessions using the profile and perform another update.

The following profiles are not updated until SNA Server/6000 is restarted:

- SNA node
- Control point
- SNA DLC
- Link station that uses a new SNA DLC

To verify the SNA profiles from the first menu of the SMIT interface, select:

```
Communications Applications and Services
  →SNA Server/6000
    →Configure SNA Profiles
      →Advanced Configuration
        →Verify Configuration Profiles
```

Or use the fastpath: `smit verifysna`.

This displays the Verify Configuration Profiles screen:

```

Verify Configuration Profiles

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Update action if verification successful      [Entry Fields]
                                             none      +
If normal_update or dynamic_update,
Backup file for committed database         []
Backup security file for committed database []

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit      Enter=Do

```

The verification checks your local SNA Server/6000 profile database for inconsistencies to ensure that all system profiles referred to exist in the database and that their values do not conflict. Only profiles that exist on the local system are verified for syntax and consistency.

8.3.8 Starting SNA Server/6000 Server

Before any application, such as HCON, can make use of SNA Server/6000, the SNA server must be running or *active*.

You can use operator commands to start SNA Server/6000, link stations and LU sessions. You can also use several configuration options to control starting SNA Server/6000 resources. In practice, resources can start in several different ways:

- Link stations can be configured to start when SNA Server/6000 starts.
- SNA Server/6000 starts automatically if you start a link station when SNA Server/6000 is not already active.
- Starting the link station also causes the session to start if the link station is associated with a dependent LU and is configured to solicit SSCP sessions.

If necessary, the server can be started from SMIT by selecting:

```

Communications Applications and Services
-SNA Server/6000
  -Manage SNA Resources
    -Start SNA Resources
      -Start SNA

```

Alternatively, from the command line, you could enter:

```

# sna -start    or
# sna -s

```

8.3.9 Starting SNA Server/6000 Link Station

To start a link station from SMIT, use:

```
Communications Applications and Services
-SNA Server/6000
-Manage SNA Resources
  -Start SNA Resources
    -Start an SNA Link Station
```

Alternatively, from the command line, you could enter:

```
# sna -start l -p<link station profile name> or
# sna -s l -p<link station profile name>
```

8.3.10 Listing Current Status of SNA Server/6000

To show the current status of the SNA server from the SMIT interface, use:

```
Communications Applications and Services
-SNA Server/6000
-Manage SNA Resources
  -Display SNA Resources
```

Or use the fastpath: `smit _snastatserv`.

This displays the following screen:

```

                                     Display SNA Resources
Move cursor to desired item and press Enter.

Display the Status of SNA
Display SNA Global Information
Display Session Information
Display Active Link Information
Display APPN Topology Database
Display APPN Directory Database
Display SNA Gateway Information

F1=Help      F2=Refresh    F3=Cancel    F8=Image
F9=Shell     F10=Exit      Enter=Do
```

Then select the desired resource.

Alternatively, status information about SNA can be obtained using the following command:

```
# lsrc -s sna
```

To display more status information including SNA Server/6000, sessions, links, global information about the local directory and topology databases, use the commands with the appropriate options:

```
# sna -displayr
# sna -d
```

See *AIX SNA Server/6000: Command Reference* for the available options.

8.4 3270 Host Connection Program/6000

The AIX 3270 Host Connection Program/6000 (HCON) is a software package that allows a RISC System/6000 to communicate with one or more IBM System/390 computer systems.

8.4.1 Overview

HCON operates in *sessions*, periods of interaction with a host computer. There can be display sessions and printer sessions.

A printer session prints files from the host to a local printer or stores printable files on the local system.

At the beginning of a display session, the emulator acts as if you have started a 3278/79 screen. From this display session, you can run commands and applications interactively or perform file transfers.

You can also write programs that communicate with the mainframe host using High-Level Language Application Programming Interface (HLLAPI).

Before you can use HCON, someone with root user authority must register you as an HCON user. Next, you must create at least one session profile. This specifies parameters for the emulator sessions, such as:

- Type of host
- Emulation type (display or printer)
- Mode of connection to the host
- Local printer name (if a printer session)

The `smit hcon` or the `mkhconu` commands may be used to add a new HCON user.

Each user of HCON can have a maximum of 26 sessions, each allowing one or more simultaneous invocations of HCON to communicate with one or more hosts using different session characteristics and communication protocols.

Session profiles can be created with the `smit hcon` or `mkhcons` commands.

The following list is a classification of HCON commands:

1. Setting up HCON:
 - startsrc -s hcon** Starts the `hcondmn` daemon.
2. Working with HCON users:
 - mkhconu** Registers HCON users.
 - lshconu** Lists HCON users.

rmhconu Removes an HCON user.

3. Working with session profiles:

mkhcons Creates an HCON session profile.

lshcons Lists the characteristics of a session.

lshconp Lists all of a user's session profiles.

chhcons Changes a session profile.

rmhcons Removes a session profile.

4. Working with HCON session:

e789 n Starts one or more specified HCON sessions.

e789cln Removes IPC resources and HCON processes left over from one or more abnormally terminated sessions. This will also terminate all other active sessions for the user issuing this command.

fxfer Implicit file transfers between the RISC System/6000 and the host (you can also use hconutil for a full-screen interface).

5. The hconutil command:

Provides a menu which allows the creation of a binary color definition table, a binary keyboard definition table, explicit file transfer, logon/logoff facilities and access to the genprof command used to create/modify automatic logon profiles.

See the *Commands Reference RISC System/6000* for more information about HCON commands.

8.4.2 Setup and Use of 3270 Host Connection Program/6000

Once we have set up all the related SNA Server/6000 profiles, we will follow these steps to start our connection:

Setup and Use of 3270 HCON/6000

1. Define valid HCON users:
smit mkhconu
2. Define HCON SNA display sessions:
smit mkhcons_st
3. Start an HCON session:
e789 Session_Name

8.4.2.1 Defining HCON Users

First, we add a user named edgar as a valid HCON user:

```
# smit mkhconu
```

Add HCON User

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

* AIX LOGIN name [Entry Fields]
[edgar]

8.4.2.2 Defining HCON SNA Sessions

Next, we define a session for user edgar. A letter a will be assigned to uniquely identify this session. We also specify the SNA connection profile, xsess02, that is used for this session.

```
# smit mkhcons_st
```

```

                                Add SNA Display Session

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
* HCON USER name                [edgar] +

```

```

                                Add SNA Display Session

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                            [Entry Fields]
HCON user name                    edgar
SESSION name                      a +
Session USE                       [X.25 CONNECTION]
* SNA Profile                      [xsess02] +
COUNTRY                          United States +
* KEYBOARD table                  [/usr/lib/hcon/e789_ktb>
* COLOR table                     [/usr/lib/hcon/e789_ctb>
* File used by SAVES key          [/u/edgar/e789_saves]
* File used by REPLS key         [/u/edgar/e789_repls]
Host TYPE                         CMS +
Host LOGIN ID                    []
Autolog NODE ID                  []
Autolog TRACE                    no +
Autolog TIMEOUT (seconds)       [0] #
[MORE...5]

F1=Help          F2=Refresh      F3=Cancel      F4=List
F5=Undo          F6=Command      F7=Edit        F8=Image
F9=Shell         F10=Exit        Enter=Do

```

8.4.2.3 Starting a HCON Session

Now, start a 3270 session with the e789 command:

```
# e789 a
```

The letter a is the name of the session we just have created for user edgar.

8.4.3 SNA Problems

It can be very difficult for a novice to determine where problems lie in SNA configurations. Section 9.9, "Diagnosing SNA Server Problems on AIX V4" on page 194, contains information on the problem determination procedures that should be followed to help speed up resolution.

8.4.4 HCON Problems

If the file transfer times out on a System/390 host, the emulator screen will display the message 0789-114.

One of the reasons this can occur is that the PSERVIC field of the MODEENT macro has not been set correctly for HCON. (See the VTAM listings given in 8.3.3, “Getting Host Definitions” on page 160 to find the format.) If it has been set correctly, check that the log mode table has been refreshed since changing the PSERVIC value.

AIX Communications Concepts and Procedures contains more information on troubleshooting with HCON.

Chapter 9. X.25 Problem Determination

This chapter contains information that is intended to be helpful, accurate and useful as a guide of how to accomplish a given system task. The provided explanations, techniques and procedures have been reviewed for technical accuracy and applicability but *have not* been tested *in every* possible environment or situation. Normal precautions should be taken in adopting these same techniques and procedures, because as product and system interfaces change, so would the use of this information.

9.1 X.25 Problem Diagnosis

Before investigating any problem, ensure that X.25 communications are set up correctly. These commands may help diagnose the problem:

- The x25mon command
- The xtalk command
- The lxx25 command
- The x25status command
- The sx25debug command
- X.25 clear and reset codes (See Appendix D, "CCITT Causes and Diagnostics" on page 269.)

See the following for further discussion of X.25 problems and solutions:

- 9.2, "Diagnosing Problems with Connecting to the X.25 Network"
- 9.3, "Diagnosing Problems with Making an Outgoing X.25 Call" on page 182
- 9.4, "Diagnosing Problems with Receiving an Incoming X.25 Call" on page 183
- 9.5, "Diagnosing X.25 Packet Problems" on page 183
- 9.6, "Diagnosing X.25 Command Problems" on page 184
- 9.7, "Diagnosing xtalk Problems" on page 185
- 9.8, "Diagnosing TCP/IP Problems" on page 191
- 9.9, "Diagnosing SNA Server Problems on AIX V4" on page 194
- 9.12, "Collecting Information for Resolution of X.25 Problems" on page 206
- 9.13, "How to Resolve an X.25 'Device Busy' Condition" on page 207

9.2 Diagnosing Problems with Connecting to the X.25 Network

Problem

When adding a port, adapter or driver, the add fails due to:

A device is already configured at the specified location.

Suggestion

The lxx25 command will show how the X.25 system is configured. Other software might be configured to use a given adapter, in which case the adapter would appear in the configuration listing, but there would be no adapter driver.

Problem

Attempts to connect an X.25 port fail when making the port using mksx25 or through SMIT. The physical and the frame layers remain down. The x25mon command shows that there is no line activity.

Suggestions

- The cable is loose or missing.
- There is no carrier (DCD or RLSD) from the network terminating unit (NTU).
- There is a cabling problem such as an extra null.
- The X.25 adapter is not seated correctly.

Problem

Attempts to connect an X.25 port fail. The physical layer is established for several seconds, as shown by the lights on the NTU, but then goes down again. The `x25mon` command indicates that there is no line activity.

Suggestions

- The cable is loose, causing the clock pin to be disconnected.
- The X.25 adapter is expecting a clock signal and not receiving one. Adjust the NTU to provide clocking.

Problem

Attempts to connect an X.25 port fail. The physical layer is connected, but the frame layer fails to come up. The `x25mon` command monitoring the frame layer may show a string of SABMs.

Suggestions

- The type of line attribute is DCE rather than DTE for this X.25 adapter. Ensure that the DTE/DCE switching configured in SMIT is suitable for the devices being attached.
- The connection mode attribute is set incorrectly. Use SMIT to change it.

9.3 Diagnosing Problems with Making an Outgoing X.25 Call

Problem

Incoming calls are arriving, but outgoing calls cannot be made on a switched virtual circuit (SVC).

Suggestions

- The configuration of the SVC logical channel numbers is incorrect. Check your network subscription and use SMIT to check the network attributes.
- Use SMIT to check that outgoing calls are allowed.

Problem

A call is being cleared with a cause code from 1 to 127.

Suggestion

Either the network or the adapter is clearing the call. The diagnostic code gives details of why the call is being cleared. Common causes are that the network user address (NUA) is not known, the X.25 line is not connected or unsupported optional facilities have been requested.

Note: Cause codes 80 to FF are set by SNA. See 9.7.2, "X.25 Protocol Problems" on page 186.

9.4 Diagnosing Problems with Receiving an Incoming X.25 Call

Problem

The `x25mon` command indicates that an incoming call has arrived at the adapter, but it is not being routed to the application that is running.

Suggestions

- If the call is being cleared with cause 0 and diagnostic 0, it may be that the application is not listening to a name in the routing list that matches the incoming call. There may be another name in the routing list that is a better match to the call and has reject (R) specified as the action. The routing list is used only by COMIO applications.
- The incoming call may have requested an optional facility that is not allowed by the current configuration.
- The incoming call may have arrived on an invalid logical channel. Check your network subscription and the network configuration attributes in your SMIT configuration.
- Use SMIT to check that incoming calls are allowed.
- Check the cause and diagnostic codes with the standard list of codes and any network-specific codes.
- A different application may be listening to criteria that are a better match those which your application is listening. Check for another program using the X.25 adapter.

9.5 Diagnosing X.25 Packet Problems

Problem

Whenever a packet is sent on a permanent virtual circuit (PVC), the network sends a clear-indication packet.

Suggestion

The logical channel number ranges are set incorrectly. Check your network subscription and your SMIT configuration.

Problem

When sending a data packet with the D-bit set on a switched virtual circuit (SVC), a reset-request packet is sent instead.

Suggestion

The intention to use D-bits must be made clear when the call is originally established, either in the call-request packet or in the call-accepted packet.

Problem

When sending a data packet with the D-bit set on a permanent virtual circuit (PVC), a reset packet is sent instead.

Suggestion

Use SMIT to configure the PVC to use D-bits.

Problem

When sending an interrupt packet with more than one byte of user data, a reset packet is sent instead.

Suggestion

The 1980 version of X.25 supports exactly one byte of user data in the interrupt packet. The 1984 and 1988 versions support up to 32 bytes. Check the subscription and the value of the support attribute in SMIT.

Problem

When sending large packets on slow lines, the link sometimes gets restarted.

Suggestion

The CCITT timer, T1, may have expired. Use SMIT to increase the value of the attribute, or use smaller packets.

Problem

While trying to connect the line, the x25mon command shows reject packets coming from the DCE instead of restart packets.

Suggestion

The most likely cause is that the packet modulo parameter you have chosen is not appropriate and must be changed with `smit x25str_mp_csp_p_sel`.

9.6 Diagnosing X.25 Command Problems

Problem

Attempts to start up the x25mon command on the X.25 port fail.

Suggestion

Only root may start the X.25 monitoring.

Problem

The xmanage, xcomms and xmonitor commands are not in Version 4.

Suggestion

These commands are not shipped with the V4 LPP. The x25mon command replaces xmonitor.

Problem

None of the X.25 commands get past the title panel.

Suggestion

Use the `echo $TERM` command to find out your terminal-type setting and make sure that it matches your actual terminal type.

Problem

The number of virtual circuits cannot be configured past an upper limit.

Suggestion

The maximum number on VCs supported depends on the license of the product that was purchased.

Problem

COMIO emulation ports fail to configure on available x25 ports (sx25an).

Suggestion

AIX V3 users must ensure that their Portable Streams Environment has been updated to the latest available PTF level.

9.7 Diagnosing xtalk Problems

You can use xtalk as a tool to diagnose COMIO problems. Common problems when using xtalk:

Problem

You may get ERROR 22 when using xtalk.

You cannot use the Talk function because no X.25 adapters have been configured

Suggestion

You must configure a port first.

Problem

Sometimes you get this message:

The call has been cleared with cause 00 and diagnostic 81

Suggestion

Please check that the listening machine (the DCE one) has xtalk running. Start xtalk first on the DCE machine before calling.

9.7.1 Device Driver Problems

You have a device driver problem when xtalk displays a message starting with a CIO status such as this one:

X.25 Communications XTALK	ERROR 110
CIO Status 68 - X25_NAME_USED The name is already being listened to.	
Esc = Cancel	

These messages, corresponding to return codes of the device driver, are listed in Appendix G, "CIO and X.25 Device Driver Error Codes" on page 293. Since you have already tested the connection to the network, the errors that are most likely to appear are those related to the routing table:

CIO Status 68 - X25_NAME_USED The name is already being listened to. Another copy of xtalk is probably running in the background and listening to IBMXTALK.

Remember to always use QUIT to leave xtalk (and not F3).

CIO Status 77 - X25_TABLE

Could not update routing list. A copy of xroute is probably still running, or a lock file xroute.lck has been left in /etc/locks.

CIO Status 73 - X25_NO_NAME

There is no such name in the routing list. The -l parameter does not match an entry in the routing list.

Two other problems are also related to the routing:

- The call has been cleared with cause 00 and diagnostic 00.

This means that the call has reached the remote system but has not been transmitted to an application. Either the application (xtalk) is not loaded, it has not been started with the -l IBMXTALK flag or the IBMXTALK entry in the routing table does not exist or does not contain a CUD matching the one in the incoming call packet (FD).

- The call has been cleared with cause 00 and diagnostic F4.

This is a diagnostic generated by xtalk on the called DTE meaning that xtalk received the call but was not ready to accept it (not on the main menu).

9.7.2 X.25 Protocol Problems

If you get a message like the following, you have an X.25 protocol problem:

X.25 Communications XTALK	INFORMATION 006
The call has been cleared with cause 13 and diagnostic 43.	
	Esc = Cancel

The cause and diagnostic codes reported by xtalk are hexadecimal values indicating the reason for the problem; x25mon reports decimal values. These codes may be generated by the local system, the PSDN or the remote DTE and are explained in Appendix D, "CCITT Causes and Diagnostics" on page 269.

Notes:

- The microcode on the X.25 adapter initially assumes that an incoming call where the first byte of the Call User Data is greater than 0xC0 is coming from an SNA system. (All X.25 SNA implementations such as QLLC use such codes - they are defined in the architecture manuals.) Thus, if the adapter does not like the incoming call packet for some reason, it will reject the call using SNA cause and diagnostic codes. (See D.5, "SNA Diagnostic Codes" on page 271.) SNA codes are only assumed until the application is past the incoming call, when it is allowed to specify which codes should be used for the call.
- If the Call User Data is less than 0xC0, then CCITT cause and diagnostics are used.
- Some networks will also generate diagnostic codes which are not in the standards. A clear packet was generated by the network if its cause code

was between 0x01 and 0x7F. To interpret codes generated by the network, you will need to ask your network provider for additional documentation.

9.7.2.1 Normal Virtual Circuit Establishment

In some cases, the information given by the cause and diagnostic codes is sufficient to determine the nature of the problem and the way to fix it, but often you will have to look at the x25mon packet trace on both systems to determine what really happened.

The normal sequence of packets to establish an SVC connection is shown in Figure 51.

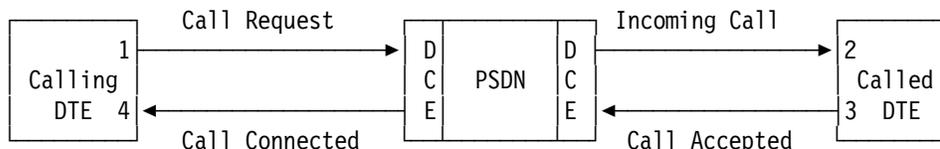


Figure 51. Establishing a Switched Virtual Circuit

This is what you can see on the traces when the virtual call is established. (The trace on the calling machine is on the left. The trace on the called machine is on the right. Only the beginning of each line is shown.):

```
# x25mon -p -n sx25a0
X.25 Monitor sx25a0
16:11:21 sx25a0 PS 0x0014 CALL.. -Call request
                                     Incoming call
                                     16:11:22 sx25a0 PR 0x0001 CALL...
                                     16:11:27 sx25a0 PS 0x0001 CF CALL
                                     Call accepted
16:11:27 sx25a0 PR 0x0014 CF CALL.. ←connected
    ↑      ↑      ↑      ↑      ↑
    1      2      3      4      5
```

Column one contains a time stamp; column two contains the name of the X.25 attachment used.

Column three contains PS when the packet is sent to the network, PR when it is received from the network.

Column four contains the logical channel number displayed in hexadecimal. The calling DTE always uses the highest available logical channel, 20 (14 in hex) in our case, to establish a call. The DCE always uses the lowest available logical channel, one in our case, to transmit an incoming call to the called DTE.

The contents of column five tell you the type of packet: CALL for the call request or incoming call, CF CALL for the call confirmation packets call accepted and call connected.

Let us now look at the contents of the call packets. The format is the same for the different call packets and is described in Figure 52 on page 188:

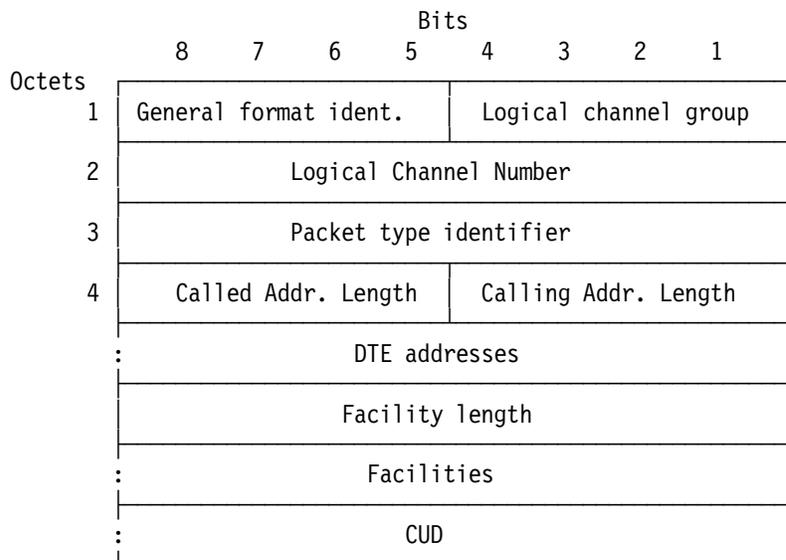


Figure 52. Call Packet Structure

Calling address length	0 to 15, binary value coded on 4 bits
Called address length	0 to 15, binary value coded on 4 bits
Called NUA	0 to 15 decimal digits; each digit is coded on 4 bits (BCD Binary Coded Decimal)
Calling NUA	0 to 15 decimal digits; each digit is coded on 4 bits (BCD Binary Coded Decimal)
Facilities length	0 to 109, binary value coded on one octet (8 bits)
Facilities	0 to 109 octets; the way to decode this field is explained in Appendix E, "Facilities" on page 275.
CUD	Call user data, up to 16 octets of application-defined data (up to 128 if the fast select facility has been subscribed to and has been selected). The CUD is generally used to identify the protocol being used.

Note: The calling NUA is optional in the call packets. Some networks even forbid it in the call request packet (the French TRANSPAC network for example) and generate it automatically in the incoming call packet. In this case, use `smit x25str_mp_csp_g_sel` to change the calling address in the call request packet parameter to `forbid`.

This is what we get on the trace. (We have now suppressed the beginning of each line.):

On the calling side:

```
# x25mon -p -n sx25a0
X.25 Monitor sx25a0
PS 0x0014 CALL      dN 1a:10 1f:0   1d:1 AA3106010761310601076000FD
PR 0x0014 CF CALL   dN 1a:0  1f:5   1d:0 000502AA420A0A
```

On the called side:

```
# x25mon -p -n sx25a0
X.25 Monitor sx25a0
PR 0x0001 CALL      dN 1a:10 1f:7   1d:1 AA310601076131060107600702AA0100420A0AFD
PS 0x0001 CF CALL   dN 1a:0  1f:0   1d:0 0000
```

The information given by the above is the following:

dN No D bit (Delivery confirmation). A dY code would mean that the D bit is present.
la: Length of the addresses (called+calling) in octets (decimal)
lf: Length of the facilities
ld: Length of the CUD

The call request can be interpreted like this:

AA Calling address length 10, called address length 10
3106010761 Called NUA
3106010760 Calling NUA
00 Length of facilities
FD CUD, used in the routing table entry IBMXTALK to send this call to xtalk.

If we look now at the incoming call, we see that the network has modified the packet and has inserted 7 bytes of facilities before the CUD. Using Appendix E, "Facilities" on page 275, they can be interpreted as follows:

07 Facilities length, 7 octets
02AA Throughput class 9600/9600
0100 Reverse charging or fast select, none selected
420A0A Packet size 1024/1024

The receiving DTE can accept these values, negotiate them or reject them by clearing the call. In our case, they are accepted.

The call accepted packet has an address size of 0, a facilities size of 0 and no CUD. The call connected packet has facilities only.

9.7.2.2 Virtual Circuit Establishment Problems

The first thing to do to analyze a virtual circuit establishment problem is to use Appendix D, "CCITT Causes and Diagnostics" on page 269, to understand the cause and diagnostic codes displayed by xtalk (when they exist), then to look at the call packet in the trace to check that there is no obvious format problem. Then you can check for a clear packet on the trace, which should look like this:

```
⋮  
16:33:02 sx25a0 PS 0x0001 CLEAR c:9 d:114
```

where c:9 is the cause and d:114 is the diagnostic, both expressed in decimal.

Before going further, let us look at the case where no clear packet shows up in the trace.

Nothing on the trace: In some cases, xtalk displays a cause and a diagnostic code, but nothing shows up on the packet trace. This happens when you specify the use of facilities in the call packet (see 3.7.1, "Facilities Requested by the DTE" on page 68) and these facilities are either invalid or not enabled in the device driver. (For further details see 3.7.1, "Facilities Requested by the DTE" on page 68.) The device driver does not send the call request on the network but reports the problem to xtalk using the same codes as they would in a clear packet.

Call is sent but not received by the remote DTE: If the call is hanging and you can only see a call request packet on the trace, look at the logical channel number used to make the call (column 4 of the x25mon output). Check with your

subscription and make sure that it is not higher than the lowest logical channel number used for SVCs, plus the number of SVCs subscribed to, minus 1. If necessary, modify the corresponding parameters in your system after having terminated all the programs using the X.25 attachment. (See 3.3.4, "Updating the Number of Virtual Channels" on page 58.)

Incoming call received by the DTE but not acknowledged: If the DTE doesn't answer an incoming call, check the two-way SVC - lowest logical channel number you have customized and make sure that it is not higher than the one in your subscription.

Incoming call received, reset packet sent back: If the DTE sends a reset packet instead of a call accepted when it receives an incoming call, there is probably a PVC that is assigned in the SVC logical channel range. Check your subscription and correct the logical channel assignments with `smit x25str_mp_csp_g_sel`.

Clear packet in the trace: A clear packet can be sent by any of the DTEs or by the PSDN during the various stages of call establishment:

- By the PSDN as a response to the call:

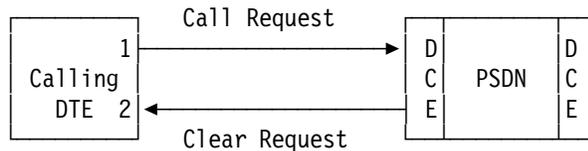


Figure 53. Call Cleared by the DCE

- By the called DTE as a response to the incoming call:

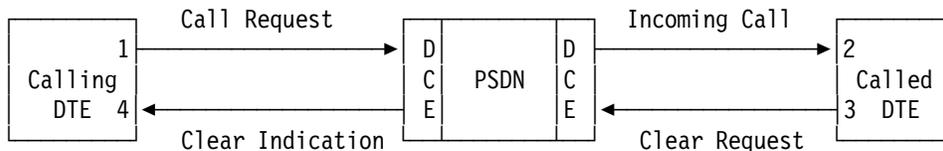


Figure 54. Call Cleared by the Remote DTE

- Or even by the calling DTE as a response to the call confirmation:

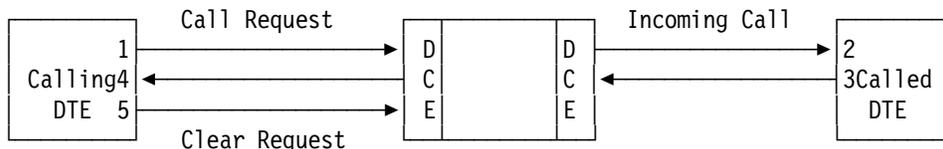


Figure 55. Call Cleared by the Calling DTE

There are many cause and diagnostic codes:

- Invalid facility request cause 03
 - This error may be generated by the attachment, the DCE or the remote DTE. It may occur even if you have not specified facilities in your call packet, in which case these are the facilities inserted in the incoming call and call connected packet by the DCE that are rejected by all of the DTEs.
- Diagnostics CCITT 65 (0x41) or SNA 228 (0xE4) or 231 (0xE7) mean that the facility carried by a call packet either:
 - Does not exist

- Has been disabled on the RISC System/6000 (see 3.7.1, "Facilities Requested by the DTE" on page 68)
- Has not been subscribed to
- Is incompatible with the level of CCITT support (1988)
- Diagnostics CCITT 66 (0x42) or SNA 230 (0xE6) mean that the facility parameters either:
 - Are invalid
 - Are not compatible with the RISC System/6000 X.25 customization:
 - Request for a packet size is larger than the maximum
 - Request for a throughput class is larger than the default

9.8 Diagnosing TCP/IP Problems

To ease the determination of TCP/IP problems, first suppress any default gateways you may have. This will avoid routing IP packets to a gateway rather than using the X.25 attachment.

Also disable the nameserver, if you have one, to avoid the discrepancies with `/etc/hosts`.

9.8.1 Preliminary Checks

- Verify with the `host` command that the hostname and IP addresses are correct on both the local and remote systems. If this is not the case, update `/etc/hosts` and execute the `x25ip` command.
- Verify with `smit chinetsx25` that the IP-to-NUA translation table entry for the remote host is correct. Do the same verification on the remote host.
- Use `netstat -r` to verify that there is a route to your remote system. This route can be either an implied route or a static route.
 - The implied route gives access to all the hosts belonging to the same network as the attachment. This network is defined by applying either the default netmask corresponding to the network class or an optional netmask to the IP address of the attachment. In the `netstat` output, the X.25 implied route has a U in the Flags field and `xsn` in the Interface field. In our case this is what we got:

Destination	Gateway	Flags	Refcnt	Use	Interface
⋮					
9.3.5	kilix	U	0	181	xs0

This route gives access to all the hosts belonging to the IP network 9.3.5 obtained by applying to `filix`'s address our netmask 255.255.255.0.

- Now check the static routes identified by UH in the Flags field and `xsn` in the interface field. You should see the remote system X.25 hostname in the Destination field.

Destination	Gateway	Flags	Refcnt	Use	Interface
⋮					
filix	kilix	UH	0	328	xs0

If an IP address appears in the Destination field, this means that the IP address that you have entered in the route command cannot be resolved

using /etc/hosts. Either delete this route and re-create it using the route command, or update /etc/hosts.

9.8.2 Testing with ping

After you have done these preliminary tests, start x25mon at the packet level in one window or session:

```
# x25mon -f -p -n sx25a0
```

In another window or session, ping the remote host. If you do not get the expected result, there are four possible scenarios:

1. No error message, nothing transmitted
2. Error message, nothing transmitted
3. No error message, the trace shows attempts to establish a virtual circuit
4. No error message, data transmitted

No error message, nothing transmitted: You have a gateway that ping used to send your packets. Verify with netstat -r (not with ifconfig) that the X.25 connection is active. Recheck the static routes and, if necessary, remove and replace the X.25 routes that have a G-flag in the netstat output (see above).

Error message, nothing transmitted: If the error message is:

```
sendto: Network is unreachable
```

Or:

```
sendto: No route for this host
```

You have a problem either with the attachment definition or with the route.

If the error message is:

```
ping: sendto: Can't assign requested address
```

```
Sendto: The socket name is not available
```

There is a problem with the x25ip translation table. Change it with smit chinetsx25 or add a correct entry if necessary with smit mksx25.

If you run the ping command and the output looks similar to:

```
#ping -c 1 nohostx
PING nohostx: (11.3.5.192): 56 data bytes
0821-069 ping: sendto: The socket name is not available on this system.
ping: wrote nohostx 64 chars, ret=1
```

```
----nohostx PING Statistics----
```

```
1 packets transmitted, 0 packets received, 100% packet loss
```

the TCP/IP X.25 interface driver could not find *nohostx* in its X25ip translation table. Run the x25ip -s command and see if it lists the remote host you are trying ping. If not, use the procedure described in chapter 7, to add an entry to the X25ip translation database.

Other TCP/IP debugging tips:

Check your IP routing table. When configuring IP, you must use a unique network ID for EVERY interface on your system.

If your ping fails without an error message, examine the ARP table by entering the following command:

```
arp -a
```

You should see a listing similar to this:

```
foox (11.3.5.165) at X.25 permanent
archiex (8.3.5.206) at 70700 X.25
rascalx (8.3.5.207) at (incomplete)
b905names.itso.ibm.com (123.22.143.253) at 10:0:5a:c9:28:fe (token ring) 840:b991:
```

Note: The b905names entry is for token ring and is not relevant for this example. However, you are likely to see entries for other networks in the arp table.

The entry for *foox* is typical for a PVC. (Note the permanent descriptor at the end of the line.) The entry for *archiex* is typical for a SVC. This line indicates that the X.25 call to *archiex* was completed (the VC exists) and TCP/IP data can be transmitted between *archiex* and the local host.

The *rascalx* line is a typical ARP table entry that does not know the hardware address for the host (*rascalx*). This is true for any type of TCP/IP interface driver that uses the ARP table. If the host shown is reached by way of X.25, it indicates that the X.25 call was not completed. This implies that an X.25 call packet has gone out, but the system has not received a response, either to confirm or clear. This could be the result of a slow network or misconfiguration. If this occurs, check the X25ip database entry using:

```
x25ip -s -h <hostname>
```

The output should look similar to the following:

```
IP/X.25 Switch Virtual Circuit Configuration
IP host name          rascalx
Remote DTE address    55545
Rcv packet size
Xmit packet size
Rcv window size
Xmit window size
Closed user group index
Closed user group index (outgoing)
RPOA selection
User-defined facilities      cc
Call user data
```

Assure that the entry has the correct NUA, packet sizes and so on. If the parameters are correct, and then use *x25mon* to trace the outgoing line in more detail. Use the section in this book on *x25mon* to interpret the trace.

Attempts to establish a virtual circuit: If you see in the trace a series of call packets showing unsuccessful attempts to open a virtual circuit, please refer to 9.7.2, "X.25 Protocol Problems" on page 186 for the problem determination procedure.

If the incoming call is rejected by the remote system with the diagnostic 241, calling address missing, the calling address was not in the translation table. Add it and use x25ip on the remote host to regenerate it.

Data sent to the remote host but nothing received: In this case, a virtual circuit is established and data packets are sent to the remote host, but the remote host does not reply. The packets are correctly transmitted to the remote host, but the remote host IP address they carry does not match the actual IP address of the remote host. Configure the IP addresses so that both the local and remote host agree. This can be done on either host.

9.8.3 Function Keys not Working Properly

When using telnet or rlogin over X.25, the remote system does not always respond correctly to use of the function keys, especially if using Esc-1 instead of F1. The ESCDELAY environment variable controls the timeout period after which the screen handling applications will consider the Esc key to be separate from the following 1. The ESCDELAY variable only works if the applications are written to Curses or Extended Curses. When running over X.25, the Esc key is often put by IP into a separate data packet from the following key. The gap between the two data packets on the receiving system is often farther apart than the default timeout. The default value of ESCDELAY (or if it is not set at all) is 500.

Try setting ESCDELAY to a very high value as follows:

1. Insert the following in the /etc/environment file
ESCDELAY=5000
2. Log completely out of the system, and then log back in to make sure the new ESCDELAY value is set.
3. Verify via the following command:
echo \$ESCDELAY

Once your test is completed, remember to either adjust the value down to a more reasonable value (between 1000 and 2000) or, if your problem is not resolved, remove this line completely from the /etc/environment file. If setting this ESCDELAY parameter does not fix your problem, then your application is not written to extended curses and the application must be modified to await the entire escape sequence. For example, the AIX SMIT application is written to curses, but it will work over a slow network regardless of ESCDELAY because SMIT was written to wait indefinitely for the entire escape sequence. Thus, even though the iptrace tool clearly shows the separation of the Esc and the remainder of the escape sequence, the escape keys work correctly in SMIT.

9.9 Diagnosing SNA Server Problems on AIX V4

When you experience a problem with SNA Server on the RISC System/6000, certain information is necessary to investigate your problem. The purpose of this section is to inform you of what information is available and how to create that information. The more information that is available when you report the problem, the quicker the problem can be resolved.

All of the traces that are generated are memory traces. It is therefore necessary to turn off tracing to ensure that the last buffer gets written to the trace file. Stopping SNA will also flush the buffers. The logs can then be formatted.

It is also very important that all information be collected at the same time. Use the date command to ensure that all machines are synchronized. This is important for determining the timing of events in the traces and logs.

When a problem is reported, there are certain things that you should be prepared to send to the support center. The information should be put into a single directory. That directory should either be sent to the support center in compressed tar format on a diskette or tape, or uploaded to a mainframe (using binary format) and sent to the support center over the network.

One problem we encountered with SNA was when we tried to start linkstation, we got the following error :

```
0105-2920 Start link station cannot find the specified remote station
```

This is because the machines are not configured as one DTE and the other as DCE. To do this, please follow the procedure in Chapter 3.3.5 to Change/Show Packet Parameters, and be sure that the Type of line parameter shows DTE to one machine, and DCE to the other. The other thing to check is the SNA linkstations; it is important that one is call and one is listen. In this case, the DCE X.25 machine was the listen SNA linkstation. This was discovered by running a x25mon trace which is below:

```

sx25a0 FR 0x0000 INFO      a:3 p:0 ns:0 nr:1 10090BAA3106000002310600000100CB
sx25a0 PR 0x0009 CALL      dN 1a:10 1f:0 1d:1 AA3106000002310600000100CB
sx25a0 FS 0x0000 INFO      a:1 p:0 ns:1 nr:1 1009130081
sx25a0 PS 0x0009 CLEAR     c:0 d:129
sx25a0 FR 0x0000 INFO      a:3 p:0 ns:1 nr:2 100917
sx25a0 PR 0x0009 CF CLEAR
sx25a0 PS 0x0009 CALL      dN 1a:10 1f:0 1d:1 AA3106000001310600000200CB
sx25a0 FS 0x0000 INFO      a:1 p:0 ns:2 nr:2 10090BAA3106000001310600000200CB
sx25a0 FR 0x0000 INFO      a:3 p:0 ns:2 nr:3 1009130081
sx25a0 FS 0x0000 INFO      a:1 p:0 ns:3 nr:3 100917
sx25a0 PR 0x0009 CLEAR     c:0 d:129
sx25a0 PS 0x0009 CF CLEAR
sx25a0 FR 0x0000 RR        a:1 p:0 nr:4

```

Below is a checklist which may help, should you see the above error:

<i>Table 13. SNA Problem Checklist</i>	
System 1	System 2
DCE	DTE
Secondary	Primary
Listen SNA Linkstation	Call SNA Linkststation
Unique NUA	Unique NUA

Always have the "listening linkstation" in starting state before starting the call linkstation.

9.10 Basic Information Required

By default, all SNA Server/6000 failure information is captured in the SNA Server/6000 service log. The exception to this is if you have redirected SNA Server/6000 failure data to the system event trace log and the `trcstop` command was used to stop system tracing. (This stops the writing of SNA Server/6000 trace information to the system event trace log, along with other AIX trace information.)

When collecting your data, only information pertaining to your problem should be captured in the traces and logs.

If you need to re-create logs for a particular problem, perform the following:

- Before re-creating the problem, remove all instances of the following logs:
 - Service log
 - All trace logs
 - Failure log
 - Ensure the system error log is also clear
- Activate data collection
- Reproduce the problem
- Deactivate data collection

Note: When collecting data for a problem, make sure that all traces are from the same problem occurrence and not from different occurrences of the same problem.

If the problem involves multiple RISC System/6000s, ensure that the traces are cleared for each one.

9.10.1 Problem Definition

One of the most important things required is a clear definition of the problem. This should be included in a file called `README` on the media. This file should also include a list of the files on the media and a brief description of those files. Be sure to answer the following questions in the problem description:

1. What happened?
2. Can it be reproduced? If so, list the steps to re-create it.
3. What were the exact error messages that were generated? This should include messages from all machines involved (other RISC Systems, AS/400, PS/2, mainframes).
4. What release of the SNA Server/6000 program is active and committed?
5. What release of the AIX operating system is active and committed?
6. What release of the HCON program is active and committed?
7. What fixes have been applied to the system?
8. Has it ever worked? If so, what changes occurred before it stopped working?
9. Get SNA trace of the problem and package up using the `/usr/lpp/sna/bin/getsnapd` command. This produces the following file:
`/var/sna/pd.tar.Z`

9.10.2 Profiles

You should always provide your profiles when you have a problem. This can be done using SMIT as follows:

```

Communications Applications and Services
-SNA Server/6000
  -Configure SNA Profiles
    -Advanced Configuration
  -Export Configuration Profiles
  
```

Alternatively, you could use the command:

```
# exportsna -A -f <FullPathName>
```

Table 14 lists the traces that would need to be provided to the support organization that you are dealing with when you report the problem. They are listed by problem category.

<i>Table 14 (Page 1 of 2). Information to Gather for Specific Problem Categories</i>	
Problem Category	Information Needed
Transaction Programming API Calls (such as open, close, read, write)	SNA Server/6000 failure log Link station trace SNA Server/6000 configuration profiles SNA API trace AIX system error log
Unable to activate a session	SNA Server/6000 failure log Link station trace SNA Server/6000 configuration profiles AIX system error log
Configuration	SNA Server/6000 failure log SNA Server/6000 configuration profiles
System crash	SNA Server/6000 failure log AIX system error log Kernel dump and running kernel (/unix) Information summary file from running crash
Core dump	Core file in /usr/lpp/sna/bin Executable file that caused the dump
Generic SNA API or generic device driver (GDD)	Generic SNA API trace Link station trace SNA Server/6000 failure log AIX system error log
SNA Server/6000 hang (system does not respond to input)	SNA Server/6000 failure log Link station trace SNA API trace Kernel dump and running kernel (/unix) AIX system error log /etc/drivers/sna_sysx when using emergency fixes
Install	AIX system error log
linktest command	SNA Server/6000 failure log AIX system error log

Table 14 (Page 2 of 2). Information to Gather for Specific Problem Categories

Problem Category	Information Needed
Logical unit flows (such as ACTLU, DACTLU, BIND, UNBIND)	SNA Server/6000 failure log Link station trace SNA Server/6000 configuration profiles SNA API trace AIX system error log
LU 0 primary LU	LU 0 API trace (primary) LU 0 server trace SNA Server/6000 configuration profiles AIX system error log
LU 0 secondary LU	LU 0 API trace (secondary) LU 0 server trace Generic SNA API trace Link station trace SNA Server/6000 failure log SNA Server/6000 configuration profiles AIX system error log
Link flows (such as XID exchanges)	SNA Server/6000 failure log Link station trace SNA Server/6000 configuration profiles AIX system error log
Starting or stopping link stations	SNA Server/6000 failure log Link station trace SNA Server/6000 configuration profiles AIX system error log
Starting and stopping sessions	SNA Server/6000 failure log Link station trace SNA Server/6000 configuration profiles AIX system error log

If you are using HCON, you may also be required to provide HCON trace information.

9.10.3 SNA Server/6000 Traces

There are two main types of traces that can be produced with SNA Server/6000. They are described in the following sections.

9.10.3.1 Link Station Traces

Link station traces show all data sent or received by the RISC System/6000 at the Data Link Control layer from the time the trace is turned on. (To include the XID exchanges that start the link station, the trace should be turned on before the link station is activated.)

After a link station trace is activated for a specific link station, the system generates a log of all link activities:

- Link activity timeouts
- Link opens
- Send data
- Receive data
- Link closes

You can use link station traces to examine the flow of SNA Server/6000 commands between two systems and to analyze SNA protocol data.

This trace will stop automatically when the link station is deactivated.

9.10.3.2 SNA Server/6000 Traces

SNA Server/6000 traces capture information pertaining to APIs and internal data. API traces log entry and exit points for calls made to a programmable interface. There are three API traces for SNA Server/6000:

- SNA API
- Generic SNA API
- CPI Communications API

SNA Server/6000 internal traces capture information about important processing events and low-level details of internal function flows.

If you wanted to mix SNA Server/6000 trace information with that of other products in the same system trace file, you could do so by activating traces with hook IDs. The hook IDs for SNA Server/6000 are:

271	SNA API trace
27A	Event and flow traces
27B	SNA failures
281	Generic SNA API trace
390	CPI Communications API trace

9.10.3.3 Starting and Stopping Traces

The traces can be controlled from SMIT, as follows:

```
Communications Applications and Services
-SNA Server/6000
  -Problem Determination Aids
    -Control SNA Traces
      -Start and Stop Selected SNA Traces
```

Or use the fastpath: `smit _snatrace`.

This screen indicates which traces are currently enabled. These values can be changed to turn on specific traces.

```

                                Start and Stop Selected SNA Traces

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
Link Station trace profile name(s)      []          +
API trace                               off         +
Generic API trace                       off         +
CPI-C trace                             off         +
Event trace (list events)               [mbuftrc]   +
Flow trace (select depth)               off         +

F1=Help      F2=Refresh      F3=Cancel    F4=List
F5=Reset     F6=Command     F7=Edit     F8=Image
F9=Shell     F10=Exit       Enter=Do

```

9.10.3.4 Stopping All SNA Traces

You can turn off individual traces using the above method, but you can also turn off all traces using the following SMIT path:

```

Communications Applications and Services
-SNA Server/6000
  -Problem Determination Aids
    -Control SNA Traces
      -Stop All SNA Traces

```

Or use the fastpath: `smit _snatraceoff`.

This will stop all active SNA traces.

9.10.3.5 Formatting SNA traces

Having turned off your traces, they must be formatted before they can be viewed. These formatted traces can be viewed from SMIT as follows:

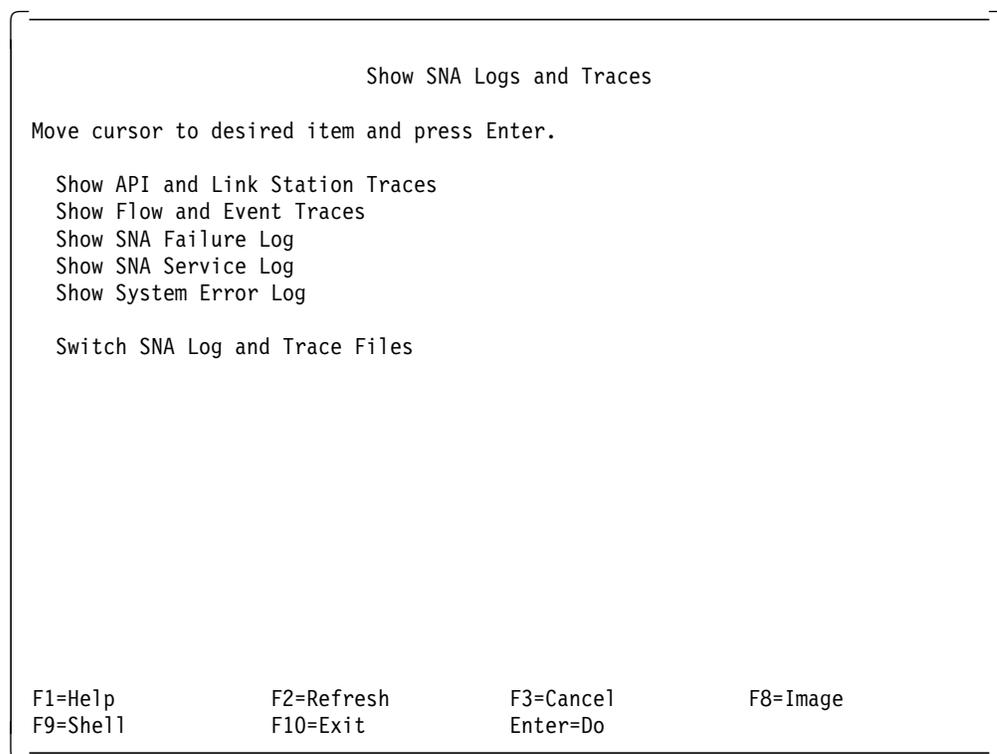
```

Communications Applications and Services
-SNA Server/6000
  -Problem Determination Aids
    -Show SNA Logs and Traces

```

Or use the fastpath: `smit _snashow`.

This will display the following screen.



From here, you can select which trace you would like to see. Selecting Show API and Link Station Traces provides a further two options:

- Show API, Generic API or CPI-C Trace
- Show Link Station Trace

Table 15 shows which hook IDs are used when creating the trace reports, using the options mentioned above.

<i>Table 15. Trace Reports and Associated Hook IDs</i>	
Report option	Hook IDs
Show API, Generic API or CPI-C	271, 281, 390
Flow and Event	27A
SNA Failure	27B
SNA Service	271, 27A, 27B, 281, 390.

9.10.4 LU 0 Information

LU 0 information can be obtained through traces.

9.10.4.1 LU 0 API Traces

LU 0 API traces can be controlled from the configuration profiles. When tracing is enabled, SNA Server/6000 records trace information during each LU 0 API subroutine call. The API trace is stopped when the application calls the lu0closes or lu0closep subroutine.

9.10.4.2 LU 0 Server Traces

To start an LU 0 primary server trace, use the following command:

```
#lu0 -p PrimaryLineProfileName -t -b &
```

This command starts the LU 0 primary server as a background process and starts the primary server trace.

To start an LU 0 secondary server trace, use the following command:

```
#lu0 -s LinkStationProfileName -t -b &
```

This command starts the LU 0 secondary server as a background process and starts the secondary server trace.

The server traces stop automatically when the LU 0 server is stopped. These traces do not need formatting to be viewed.

9.10.5 System Error Log

To ensure that all the data in the system error log is relevant, it is advisable to first clear out the error log.

9.10.5.1 Clearing the System Error Log

Use the following SMIT path to clear the system error log:

```
Communications Applications and Services
-SNA Server/6000
-Problem Determination Aids
-Control SNA Traces
-Clear System Error Log
```

Or use the fastpath `smit _snaerrclr`.

Note: This will clear all errors from the system error log, not just the SNA errors.

9.10.5.2 Showing the System Error Log

Use the following SMIT path to show the system error log:

```
Communications Applications and Services
-SNA Server/6000
-Problem Determination Aids
-Show SNA Logs and Traces
-Show System Error Log
```

Or use the fastpath `smit _snasyserr`.

9.10.6 HCON Trace

HCON has a trace capability built into the product. This can be used by:

```
# startsrc -s sna /* Start sna
# e789 -D hcon.trace Sessionid
*****
* Reproduce Problem * /* Run Test Case
*****
# stopsrc -s sna /* Stop SNA services
# cp hcon.trace /tmp/pmr/hcon.trace /* Save it
```

9.10.7 SNA_ABEND

The SNA_ABEND file is created any time that SNA Server processes an exit unexpectedly. Stopping SNA Services with the -c option will also create this file.

If the /var/sna/SNA_ABEND file is being created when the problem occurs, it will need to be sent to the support organization.

9.10.8 Trace Hooks for X.25

If it is required to trace the internal workings of the X.25 stack, the following hook IDs are available. These traces, along with a line trace from x25mon and a system configuration table from lxsx25, would help IBM diagnose the code's behavior.

25C	Packet layer
329	X.25 TCP/IP interface
32A	NPI
32B	X.25 system utilities
32C	Triple-X PAD
33B	COMIO emulation
33C	Adapter driver

The traces produced from these hook IDs are not in a form that is directly useful to a system user or system administrator. They are designed to allow defect support to diagnose code flow.

9.11 System Errors

Severe X.25 failures may invoke system errors. Included here are some indications of system errors and information to collect for problem determination.

9.11.1 Flashing 888

A flashing 888 in the three-digit display indicates that there is a message to be displayed, encoded as a string of three-digit display values:

- Turn the Key Mode Switch to the Normal position or the Service position.
Note: Every time you press the Reset button, hold it for about one second to allow the program to sense the change.
- Press the Reset button to display the first value in the string.
- Record this value.
- Repeat the above two steps until a flashing 888 is displayed again.

Up to 41 three-digit display values may be included in the string. If necessary, you can display the entire string of values again by repeating the above procedure.

The first value following the 888 indicates the type of information contained in the remainder of the string. Find the first value in the following, and take the action given:

9.11.1.1 Flashing 888-102

An initial value of 102 indicates an unexpected system halt during normal operation. For unexpected system halts, the string of three-digit display values has the following format:

888 102 mmm ddd

where mmm is a value indicating the cause of the halt and ddd is a value indicating whether or not a system dump was obtained.

Refer to the Hardware Problem Determination Procedures in the *IBM RISC System/6000 Diagnostics Programs: Operator Guide*. If these procedures return an SRN, record that SRN in item four of the Problem Summary Form and report the problem to your service organization. If Diagnostics does not detect a problem, record SRN 101-mmm in item four of the Problem Summary Form and report the problem to your service organization. If a system dump was obtained, copy the dump to removable media and be prepared to make it available to your service organization.

The following list gives the possible values of mmm, the second value following the 888, and the cause of the system halt invoking that value:

- 200 Machine check due to memory bus error (RAS/CAS Parity)
- 201 Machine check due to memory timeout
- 202 Machine check due to memory card failure
- 203 Machine check due to address exception: address out of range
- 204 Machine check due to attempted store into ROS
- 205 Machine check due to uncorrectable ECC due to address parity
- 206 Machine check due to uncorrectable ECC
- 207 Machine check due to undefined error
- 300 Data storage interrupt - processor type
- 32x Data storage interrupt - input/out exception - IOCC. The number represented by x is the BUID
- 38x Data storage interrupt - input/output exception - SLA. The number represented by x is the BUID
- 400 Instruction storage interrupt
- 500 External interrupt - Scrub - memory bus error (RAS/CAS Parity)
External interrupt - DMA - memory bus error (RAS/CAS Parity)
External interrupt - undefined error
- 52x External interrupt - IOCC type - channel check
External interrupt - IOCC type - bus timeout
External interrupt - IOCC type - keyboard external
The number represented by "x" is the IOCC number
- 700 Program interrupt
- 800 Floating point unavailable

The value of ddd, the third value following the 888, indicates the current dump status. The possible values and meanings of ddd are:

- 0c0 Dump completed successfully.
- 0c4 Partial dump completed.
- 0c5 Dump failed to start. An unexpected error occurred while the system was attempting to write to the dump device.
- 0c8 Dump failed. No primary dump device is configured.

9.11.1.2 Flashing 888-103

An initial value of 103 indicates a diagnostic message. Diagnostic messages appear in the three-digit display when the console display is not present, when the console display is unavailable because of a display or adapter failure or when a failure is detected that prevents the completion of IPL.

The string of three-digit display values identifies the SRN and up to four Field Replacement Units (FRUs). The string of three-digit display values has the following format:

- 888 103 nnn nnn c01 1ee 2ee 3dd 4dd 5 ss 6ss 7ff 8ff
- c02 1ee 2ee 3dd 4dd 5ss 6ss 7ff 8ff
- c03 1ee 2ee 3dd 4dd 5ss 6ss 7ff 8ff
- c04 1ee 2ee 3dd 4dd 5ss 6ss 7ff 8ff

The two values nnn nnn represent the SRN. The values c01, c02, c03 and c04 indicate the first, second, third and fourth FRUs, respectively. For each FRU, the value sequence 1ee 2ee 3dd 4dd 5ss 6ss 7ff 8ff is the location code. Refer to the *IBM RISC System/6000 Diagnostics Programs: Operator Guide* for information on interpreting these location codes.

Record the SRN in item 4 of the Problem Summary Form and the location codes in item 6 of the Problem Summary Form. Then, report the problem to your service organization.

9.11.2 System Dump

The AIX dump facility records the state of the system at the time of the failure by automatically copying selected operating system data areas to the dump device when an unexpected system halt occurs.

If the system crashes or if you have invoked a dump, you need both the system dump file and the /unix file to debug the problem. The system dump file is usually put on the primary dump device, which in AIX Version 4 is now /dev/hd6.

Note: Invoking a system dump overwrites a previous dump or other data stored on the dump device.

9.11.2.1 Starting a System Dump

A system dump can be invoked in a number of ways:

- Turn the key lock switch to the Service position and press the Reset button. This will create a dump on the primary dump device.
- Turn the key lock switch to the Service position, hold down the Ctrl and Alt keys and at the same time press the 1 key on the number pad. This will create a dump on the primary dump device.
- Turn the key lock switch to the Service position, hold down the Ctrl and Alt keys and at the same time press the 2 key on the number pad. This will create a dump on the secondary dump device.
- Use the sysdumpstart command to invoke a dump on either the primary or the secondary dump device.

Note: If the system has already halted and performed a dump, the Reset button will scroll through the LEDs rather than trigger another dump.

9.12 Collecting Information for Resolution of X.25 Problems

Previous experience has shown us that specific information is needed for technical support personnel to address defects correctly. This information will help clarify customer scenarios, so all members of the defect support team involved in a specific problem will know all the related data for a given X.25 environment.

9.12.1 List of Elements to Gather

1. Provide a copy of the network subscription (usually sent by fax).

Note: This is a document (usually 1 or 2 pages in length) that describes exactly what the customer has purchased from the network provider.

2. Provide a copy of the x25mon output for each port by entering:

- To start x25mon:

```
# x25mon -f -p -n sx25a? > x25mon.out &
```

- To stop xmonitor:

```
# kill `ps -e | grep x25mon | cut -c0-6`
```

Note: Be sure to have the problem occur between the time x25mon started and the time x25mon is stopped.

Warning: Do not use kill -9 to kill the x25mon command.

3. Provide a copy of the X.25 traces by entering:

- To start tracing:

```
# trace -a -j 225,227,25c,329,32a,32b,32c,33b,33c,41f
```

- To stop tracing:

```
# trcstop
```

- To format the trace:

```
# trcrpt /usr/adm/ras/trcfile > trace.out
```

Note: Be sure to have the problem occur between the time tracing is started and the time tracing is stopped.

4. Provide a file called "README" and put the following information in it:

- A very detailed problem description
- A description of what the trace data should be showing, that is, what type of failure
- A description of the hardware configuration
- A detailed description of the network configuration
- A picture of the network topology, all machines, routers, switches, etc.
- A copy of any error messages sent to the screen
- A description of all applications and/or software running on the system
- A description of what events lead up to the failure
- Time between failures, how often failures occur, etc.
- What X.25 applications are running, SNA, SSI, TCP/IP, API, NPI, DLPI, other....

5. Provide a copy of the device information output by entering:

```
# lsdev -C > devices.out
```

6. Provide a copy of the AIX software levels by entering:

```
# ls1pp -ha > levels.out
```

7. Provide a copy of the X.25 attributes by entering the following:

```
# lsattr -E -l sx25a? > lsattr1.out
```

Where ? is the number of the port

```
# lsx25 > lsx25.out and if running TCP/IP:
```

```
# lsattr -E -l xs? > lsattr2.out
```

Where ? is the number of the device

8. Provide a copy of the system error log report by entering:

```
# errpt -a > errpt.out
```

Note: Be sure to clear the error log before the problem occurs. To clear the error log:

```
# errclear 0
```

9.13 How to Resolve an X.25 'Device Busy' Condition

While trying to change the configuration of the X25 device, you receive the error "device not in correct state". If you follow the steps below, you should be able to change the configuration.

9.13.1 Problem Identification

When trying to update the X25 configuration via SMIT, the following message is issued:

```
method error (/etc/methods/chgx25)
0514-029 Cannot perform the requested function because a child
device of the specified device is not in a correct state.
```

9.13.2 Problem Solution

To fix this problem, the user must stop all processes using the X.25 LPP. The user cannot change attachment characteristics while a process is using X.25 because, technically, this process is holding the device open.

Some things to check:

- Kill any x25mon or xtalk processes in the process table.
Use `kill -15` on the process ID. Kill levels 2 and 3 are okay too. NEVER use `kill -9` to stop x25mon.
- Stop any device driver tracing.
- Bring the TCP/IP xsn interface to a defined state using the command `rmdev -l xsn`.
- Stop SNA (via SMIT or with the `sna -stop sna -t normal` command).
- Stop the IBM Triple-X PAD daemon (x29d).
- Stop the SNMP daemon (x25smuxd).
- Stop anything running on COMIO, and then remove COMIO emulation from the port to free it. Do this via SMIT and make sure you remove the definitions

from the database. Otherwise, when you re-create them, they use the next available port name.

COMIO emulates the AIX V3 BOS X.25 device driver (/dev/x25s0) and allows pre-LPP applications to run unchanged. Thus, check for all applications.

- Stop any user-written NPI or DLPI applications.

The user should now be able to change the definitions of the X.25 device.

9.13.3 If All Else Fails

If all else fails and you cannot identify the culprit, the following workaround can be used. Although this is a way around the situation, it is not an ideal solution to the problem, as the system has to be rebooted.

1. Rename the adapter microcode found in directory /usr/lib/asw. The Portmaster/A adapters use the sx25pma.cod file, and the X.25 Co-Processor adapters use sx25cyc.cod.
2. Shut down and reboot the system.
3. When the system comes back up, make the desired changes via SMIT.

At this point, the adapter (apm0 or ampx0) should be available, but port (xs25a#) should be defined. A user can modify a port's characteristics as long as that port is in a defined state. It is not necessary to disable all of the ports on a single adapter or in a system. When using this workaround, however, all ports of this type will become defined.

4. Change the microcode back to the correct name.
5. Either reboot or use SMIT "Configure a Defined Port" in the "Manage X.25 Ports" menu.
6. Restart any applications that were stopped in the previous steps.

9.14 Improving FTP Performance over X.25

If you experience problems when ftp'ing files over X.25 networks, here are a few things to check that are most likely causing the problems.

- Packet size - (1024 is a better size)
- X25 Window Size - (this should be at least 7)
- X25 packet size = MTU size

9.15 ARTIC960 Software Problems

ARTIC960 users must install the IBM ARTIC960 AIX Support Program for RS/6000 software shipped on a diskette packaged with the adapter. This software must be at V1.1.3 or later.

Please make sure that this diskette is kept safe.

If in doubt as to the level of the software, you can check using the following command:

- `lspp -l devices.artic960.rte`

The output should look something like this:

- devices.artic960.rte 1.1.3.0

If the software is installed in the wrong order, then follow the recovery procedures as documented in the Hardware Installation section of the *AIXLink/X25 1.1 for AIX: Guide and Reference* manual.

If you install the ARTIC960 AIX Support Program for RS/6000 V1.1.3 after installing AIXLink/X25, the following error may occur when you attempt to add a "twd" AIXLink/X25 device driver to the ARTIC960 adapter:

```
Method error (/usr/lib/methods/define):  
0514-022 The specified connection is not valid.
```

If you encounter this error, then run the following script which should solve the problem.

```
/usr/lpp/sx25/inst_root/sx25.rte.config
```

9.16 Examples of X.25 PAD Problem Determination

This section provides some further examples of X.25 problem determination performed in real situations.

This may be of benefit to those who are unsure at what position in the communication stack the problem lies.

Initial diagnosis of an X.25 problem should follow a methodical path that will lead to a corrective action. There are many variables that constitute the X.25 network. Any incorrect value may result in an inoperative network. Often, a large multiuser site communicating with users and printers via X.25 will not tolerate downtime of the network. A few minutes spent isolating the source of the problem at the time of failure can reduce subsequent downtime as the correct party to resolve the problem can be called in. For example, your network provider may not have facilities to modify the AIX X.25 configuration, and similarly, your local AIX support organization may not have facilities to modify the X.25 network settings. Transient problems can be resolved by judicious tracing selection prior to the event occurring again.

9.16.1 Providing Initial Statement of Fault

The statement 'X.25 Network Inoperative' does not reveal much to either your X.25 network provider or to IBM specialists who are resolving the problems.

Work through the following table until you have found a problem. Skipping applicable steps is not recommended, as they will only be repeated by your support organization. Even when a problem would appear to be at a higher level (that is, cannot print via X.25, cannot ping etc.), a lower-level problem often is the cause. If a problem is reported to IBM, ensure the contact details of the X.25 network provider are available, as often the two parties will work together to resolve a problem.

X.25 Network Provider

Do you have the contact details of your X.25 network provider?

You will often need the network provider on call to verify X.25 network configuration and provide additional traces.

Table 16. X.25 Problem Checklist

Symptom	Layer	Refer
Physical	Hardware	9.16.2, "X.25 Hardware Connection"
Devices	AIX Device Driver	9.16.3, "Device Driver" on page 211
Connection State	Physical Frame or Packet	9.16.4, "Connection State" on page 211
Total Line Termination	Physical Frame or Packet	9.16.5, "Total Line Termination or Failure to Start" on page 212
No Traffic Flow	Physical Frame or Packet	9.16.6, "Connected to X.25 Network No Traffic Flow" on page 214
Transient Dropouts	Packet	9.16.7, "Transient Dropouts" on page 214
Hangs or Crashes	System	9.16.8.2, "System Hangs/Crashes Caused by X.25 Software" on page 216
Processes using X.25	System	9.16.8.1, "System Problems Introduced by Disassociated X.25 Processes" on page 216
Printing	Application	9.16.9, "Printing" on page 216

9.16.2 X.25 Hardware Connection

Ensure physical integrity of X.25 cables connecting your network to your RS/6000 communication adapters.

Often, the cables connected to the RS/6000 communication adapter are large with many pins. Ensure the cable head is screwed into the adapter and all pins are intact.

If you have multiple ports, are you connected to the correct port?

Does your X.25 communications adapter support your X.25 network connection speed?

9.16.3 Device Driver

The `lscfg`, `lsx25` and `lsdev` commands should be used to ensure your X.25 communications adapter is known to the system and in the available state. For example, following is a fragment of output from the `lscfg` command on a system with one 6-port artic adapter with four X.25 ports configured:

```
+ apm0          00-02          6-Port Portmaster Adapter/A X.21
* twd0          00-02-00
* sx25a0        00-02-00-00
* x25s0         00-02-00-00-00
* sx25a1        00-02-00-01
* x25s1         00-02-00-01-00
* sx25a2        00-02-00-02
* x25s2         00-02-00-02-00
* sx25a3        00-02-00-03
* sx25a4        00-02-00-04
```

The following output shows a fragment from the `lsx25` command:

```
*****
* Report by X.25 port's logical location *
*****
X.25          Logical Logical
Port  Driver  NUA      COMIO  TCP/IP  board  port
sx25a0 twd0     1200621 x25s0  n/a     0      0
sx25a1 twd0     1200621 x25s1  n/a     0      1
sx25a2 twd0     1200621 x25s2  n/a     0      2
sx25a3 twd0     1200621 n/a     n/a     0      3
sx25a4 twd0     1200621 n/a     n/a     0      4
```

```
*****
* Report by X.25 port's physical location *
*****
X.25          Phys.
Port  Driver  Adapter Slot  Port  Interface
sx25a0 twd0     apm0     2     0     X.21
sx25a1 twd0     apm0     2     1     X.21
sx25a2 twd0     apm0     2     2     X.21
sx25a3 twd0     apm0     2     3     X.21
sx25a4 twd0     apm0     2     4     X.21
```

Examine each port and driver via the `lsdev -C -l <driver>` command to ensure it is in the available state.

9.16.4 Connection State

Prior to establishment of any communication, the physical, frame and packet layers of the X.25 network connection must be up.

Use the `x25status` command to view the port state:

```
x25status
```

```
Link status for bfpvill01
```

```
-
Port          Packet State          Active Active
                SVCs      PVCs
sx25a0        PACKET LAYER CONNECTED  23      0
```

sx25a1	PACKET LAYER CONNECTED	2	0
sx25a2	DISCONNECTED	0	0

In the example, the port sx25a2 is in a disconnected state. No communication will be possible via that port. Examine the link parameters and/or trace the line to find the cause of the problem.

9.16.5 Total Line Termination or Failure to Start

If you have a symptom involving all X.25 traffic being simultaneously terminated, then suspect a network DISC is being sent from the X.25 network.

Following is sample output from the x25mon command:

```
x25mon -fpm sx25a0
```

```
14:59:33 sx25a0 FR 0x0000 DISC   a:3 p:1
14:59:33 sx25a0 FS 0x0000 UA     a:3 p:1
14:59:39 sx25a0 FS 0x0000 SABM   a:1 p:1
14:59:39 sx25a0 PS 0x0000 RESTART c:0 d:161
14:59:39 sx25a0 FR 0x0000 UA     a:1 p:1
14:59:39 sx25a0 PR 0x0000 RESTART c:7 d:0
14:59:39 sx25a0 FR 0x0000 INFO   a:3 p:0 ns:0 nr:0 1000FB0700
14:59:39 sx25a0 PR 0x0001 CALL dN
```

The trace shows the X.25 network has sent us a disconnect request. This would disrupt all channels using the X.25 network.

Why is the network sending a DISC?

An x25mon trace of the few seconds leading up to the disconnect reveal the answer:

```
14:59:25 sx25a0 FS 0x0000 INFO   a:1 p:0 ns:1 nr:6 100C7A1B3D2122DA
14:59:25 sx25a0 FS 0x0000 INFO   a:1 p:0 ns:2 nr:6 101B61
14:59:25 sx25a0 FS 0x0000 INFO   a:1 p:0 ns:3 nr:6 101B6236
14:59:25 sx25a0 FR 0x0000 INFO   a:3 p:0 ns:6 nr:0 101C01
14:59:25 sx25a0 FR 0x0000 INFO   a:3 p:0 ns:7 nr:0 101C21
14:59:25 sx25a0 FS 0x0000 INFO   a:1 p:0 ns:4 nr:0 101B641B3D2E24
14:59:25 sx25a0 FR 0x0000 INFO   a:3 p:0 ns:0 nr:1 100C85
14:59:25 sx25a0 FR 0x0000 RR     a:1 p:0 nr:2
14:59:25 sx25a0 FS 0x0000 INFO   a:1 p:0 ns:5 nr:1 1022E1
14:59:26 sx25a0 FS 0x0000 INFO   a:1 p:0 ns:6 nr:1 102B41
14:59:26 sx25a0 PS 0x002b RR     pr:2
14:59:27 sx25a0 PS 0x0031 RR     pr:2
14:59:27 sx25a0 FS 0x0000 INFO   a:1 p:0 ns:7 nr:1 103141
14:59:28 sx25a0 FS 0x0000 RR     a:1 p:1 nr:1
14:59:29 sx25a0 PS 0x000f RR     pr:4
14:59:30 sx25a0 PS 0x002e RR     pr:4
14:59:31 sx25a0 FS 0x0000 RR     a:1 p:1 nr:1
14:59:32 sx25a0 PS 0x0021 RR     pr:3
14:59:33 sx25a0 FR 0x0000 DISC   a:3 p:1
```

As you can see, the first thing shown is us sending ns = 1 with a bunch of data. Then, after the ns = 1 is sent by us, we send frames with ns = 2 and ns = 3. These will be listed as outstanding frames. We receive ns = 6 and ns = 7 from the network, and we ack them when we send a frame of ns = 4, via the nr = 0 (this acks 6 and 7 sent from the network). The network sends us a frame with ns

= 0 and acks a previous frame via the nr = 1 (which will ack our frame 0 sent). Then, the network acks the first thing listed above via the nr = 2 (our frame sent with ns = 1). Then, we send frames of ns = 5 and ns = 6. The packet layer sends two RRs. Then, we send a frame of ns = 7. At this point, our window becomes *full*, and we have frames 2-7 that have yet to be acked by the network. So, because the *last* acknowledgement we received was at 14:59:25, 3 seconds after that, the T1 timer (which can be seen in the Change/Show Frame Parameters for the port and should match that of the network provider's T1 timer) pops (14:59:28) – and thus, the poll that is sent by us showing correctly that we have responded up to nr = 1 (packet sent to us with ns = 0). I am reiterating this because this shows that we have NO outstanding packets to ack that were sent by the network. Three seconds after that, at 14:59:31, we again send a poll. (We will do this 10 times, as specified by the N2 timer again in the Change/Show Frame Params, before we will send a DISC ourselves). Then, as shown at the very end of the trace, (14:59:33) a DISC is received.

The network provider increased the physical level idle timer to resolve the problem.

Note: If we had forced the connection down ourselves, (rmdev -l sx25a0) we would have seen a DISC being *sent* by us, not received, from x25mon:

```
13:49:29  sx25a1 FS 0x0001 DISC          a:1 p:1
```

The following command sequence can be used to trace the establishment of connection to the X.25 network.

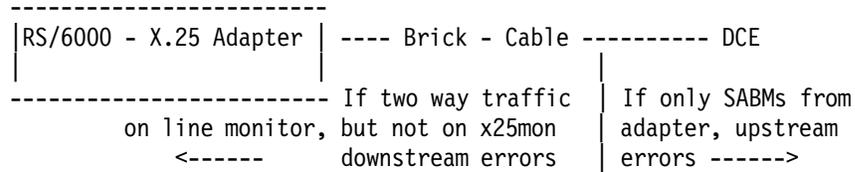
1. rmdev -l sx25a0
2. x25mon -fjn sx25a0 > /tmp/x25mon.out &
3. mkdev -l sx25a0

Does the trace show incoming or outgoing calls after establishment of the connection?

A successful network connection should show a set asynchronous balanced mode frame (SABM) followed by an unnumbered acknowledgement (UA). You should then see polls 'RR' and, hopefully, incoming call data (in the case of PAD traffic).

```
23:32:00  sx25a0 FS 0x0000 SABM          a:1 p:1
23:32:00  sx25a0 FR 0x0000 SABM          a:3 p:1
23:32:00  sx25a0 PS 0x0000 RESTART        c:0  d:161
23:32:00  sx25a0 FS 0x0000 UA            a:3 p:1
23:32:00  sx25a0 FR 0x0000 UA            a:1 p:1
23:32:00  sx25a0 FS 0x0000 INFO          a:1 p:0 ns:0 nr:0 1000FB00A1
23:32:00  sx25a0 FR 0x0000 INFO          a:3 p:0 ns:0 nr:0 1000FB0700
23:32:00  sx25a0 PR 0x0000 RESTART        c:7  d:0
23:32:00  sx25a0 FR 0x0000 RR            a:1 p:0 nr:1
23:32:01  sx25a0 FS 0x0000 RR            a:3 p:0 nr:1
23:32:02  sx25a0 FR 0x0000 INFO          a:3 p:0 ns:1 nr:1
10010BC7120062114170000000200001000000
23:32:02  sx25a0 PR 0x0001 CALL          dN 1a:10 1f:0 1d:4
```

If you still cannot establish network connection, use a hardware line monitor to watch the data flow.



9.16.6 Connected to X.25 Network No Traffic Flow

Examine an x25mon trace and look for a call confirm packet:

```
17:53:51  sx25a0 PR 0x0002 CALL          dN 1a:10 1f:0 1d:4
          C7120062114170000000200001000000
17:53:51  sx25a0 PS 0x0002 CF CALL     dN 1a:10 1f:0 1d:0
```

Note that the call confirm corresponds to the call on the same logical channel number (LCN). In this example, the LCN is 0x0002.

For PAD users, one reason for calls not being confirmed is the x29 daemon is started in outbound mode only.

Locate the command used to start the x29 daemon in the file:

```
/etc/rc.x25
```

The x29 daemon should not be started with -n flag:

```
x29d -n
```

This will prevent incoming calls from being confirmed. Remove the -n flag if incoming calls are desired.

Check the cause and diagnostic values for calls that are prematurely cleared.

9.16.7 Transient Dropouts

If your symptom involves an individual user or application terminating unexpectedly while others continue to talk across the network, then suspect a CLEAR packet is being sent or other application problem. You might see this entry in the x25mon trace:

```
CLEAR c:0 d:241
```

With a site with many users and large amounts of X.25 traffic, you will need to judiciously perform a trace to capture the problem event. It becomes overly difficult trying to correlate problem events with trace files of the entire day's traffic.

For example, the following shell script can be used to start a trace then have it stop automatically once a problem event is detected. The script will restart every 30 minutes using new trace files – so, when a problem is found, the resultant trace files will be at most only 30 minutes' worth of trace data.

```
#!/bin/ksh
# Detect a problem

cd /trace # some large File system

while :
```

```

do

D=`date +%H%M%S%d%m` # timestamp each file

T=./trace.out.$D
X0=./x25mon.0.out.$D
U=./x25status.out.$D

# this traces X.25 hooks in a large trace buffer
trace -a -j 14e,255,277,25c,32c -l -T 2000000 -L 500000000 -o $T

x25mon -fjn sx25a0 > $X0 &
XPID0=$!

x25status > $U

m=0
while [ $m -lt 30 ] # 30 minutes
do
if grep CLEAR $X0 # Change the expression to whatever you need
then
    trcstop
    kill $XPID0
    echo "Caught"
    exit 1
fi
    sleep 5
    let m=$m+1
done

trcstop
kill $XPID0
date
echo "stop"

done

```

This can be used to trap disconnects (DISC), clears (CLEAR) or other problem events you may be interested in.

9.16.7.1 User-Originated Clear Packets

The detection of a CLEAR packet in a x25mon trace may not necessarily indicate a network error. Here are two examples of situations where a CLEAR packet received with a diagnostic code of 241 related to user activity.

TERM: In the \$HOME/.profile file, the \$TERM environment variable was not set correctly for particular PAD users. When the user was using a full screen ASCII application, sometimes incorrect ASCII escape sequences were being generated that simulated a break character, forcing the user off the network. Correct the value of the \$TERM environment variable to match the terminal type in use.

User-Generated CLEAR: In a large multiuser site, a number of frequent CLEARS in the x25mon trace were noted. These were traced to users who were typing in incorrect passwords once too many times, and the application was logging the user out; this had the effect of clearing the logical channel associated with that user.

For other diagnostics codes received with a CLEAR packet examine the appropriate diagnostic value.

Another common reason for a CLEAR is that the number of LCN (logical channel numbers) is exceeded and the X.25 connection cannot support additional users.

9.16.8 Application

X.25 applications may disrupt system operation in several ways.

9.16.8.1 System Problems Introduced by Disassociated X.25 Processes

If a network problem disrupts the X.25 network connection for a PAD user, the user application will receive a SIGHUP signal. If the user application ignores or catches this signal, then the application may continue to run in the background, disassociated from its controlling terminal. In this case, the TTY column from the output of the `ps -ef` command will show a '-' field instead of the usual X.25 PAD pseudo terminal. This is not an X.25 problem as such, as the application will need to clean itself up. Otherwise, the system administrator will need to detect these rogue processes and take the appropriate action. Usually, this would involve killing the process.

9.16.8.2 System Hangs/Crashes Caused by X.25 Software

A severe system problem induced by X.25 will not be trivial to detect, as a kernel dump will need to be examined together with auxiliary traces. Typically, the stack trace of running processes is examined to see if any X.25 kernel extensions are present. In any case, we suggest you make the kernel dump available to your support organization as per procedure in *GG24-2513 AIX V4 Software Problem Debugging and Reporting*

Look for these X25 kernel extensions in the kernel stack trace:

- twd
- x25pkt

If the event is readily repeatable, then it is suggested you run a kernel trace using a pinned memory buffer. If a kernel dump is gathered after this event, then valuable information regarding system activity leading up to the crash/hang are stored in the kernel dump.

To start this trace at system boot time, place the following command near the end of the `/etc/rc` system startup file:

```
trace -a -j 25C -L 1000000000 -o /trace/trace.out
```

Indicate this fact to your support organization so employees there can extract the trace data from the kernel dump. You may be asked to add other kernel trace hooks depending on the suspected region of code.

9.16.9 Printing

Problem

Cannot cancel remote job to X.25 network attached printer. The remote printer did not appear to be printing.

Suggestions

Restart qdaemon:

```
stopsrc -s qdaemon  
lssrc -s qdaemon # ensure it is stopped  
startsrc -s qdaemon
```

Ensure /var and / filesystems are not at 100% usage via the df command.

Problem

During high volumes of print traffic, the printer appears to hang or one working printer fails to print.

Suggestion

Perform an x25mon trace on the line to examine traffic flow.

The direction of traffic will depend on the configuration of the X.25 printer – that is, locally or remotely initiated.

Chapter 10. Performance and Tuning

This chapter will explain performance considerations relevant when using X.25 networks with AIX in real situations.

The RISC System/6000 supports communications on an X.25 PSDN via V.24, V.35, V.36 and X.21 interfaces. Overall performance depends upon a number of factors, the major ones being network speed, network configuration parameters and the application.

10.1 Line Speed

The speed within the network is under the control of the network provider. The CCITT standards define network access up to 64 Kbps. So the maximum obtainable line speed depends upon the type of connection. V.24, V.35 and X.21 line connections support speeds up to 19.2 Kbps, 56 Kbps and 64 Kbps, respectively.

The following table gives a summary of the performance of the different line speeds. Of course, this is the ideal and does not take into account such factors as overhead for packetizing.

Interface	V.24	V.35	X.21	
Kbps	9600	19200	64000	64000
Bytes p/s	1200	2400	7000	8000
Packets p/s (Packet size = 128)	9	18	54	62

The V.36 and X.21 interfaces on the ARTIC960 adapter support speeds up to 2 Mbps.

10.2 Transit Delay Versus Data Throughput

A common error when calculating the throughput of a network between two points is to first determine the time taken by a block of data to be transmitted and acknowledged by a remote system, then to divide the transmitted block size by the transmission time to obtain a throughput in bytes per second.

This method of calculation, which assumes that only one block of data is traveling at a time on the network, is correct for protocols like BSC on a leased line. It is not correct with an X.25 network, where the sender can send several packets before the receiver has received the first one.

For an X.25 network, the transmission time is the sum of the following:

- Sending application processing time
- Packet assembly processing time by the X.25 adapter
- Transmission time to the DCE

- Transit time in the network
- Transmission time from the sending DCE to the receiving DTE
- Packet disassembly processing time by the X.25 adapter
- Receiving application processing time

If the data transfer application is properly designed, the sender will not need any acknowledgment of the previous packet from the receiver to send the next packet. In this case, all these elementary delays are overlapped. When the application prepares the data to be transmitted:

- The X.25 adapter processes packet number n and transmits packet n-1 to the DCE
- A few packets may be travelling in the network
- One packet is being sent from the remote DCE to the remote DTE
- One packet is being processed by the adapter
- Received data is processed by the application

In this case, the throughput of the data transfer will be limited by the slowest part of the network.

10.3 Factors That May Limit Throughput

As explained in the previous section, the end-to-end maximum throughput is limited by the throughput of the slowest part of the network. This is analogous to a highway that has a throughput of 24000 cars per hour but may be limited to 2400 cars per hour if there is a section with road repairs in progress.

In the following section, we will describe the various factors that may influence the X.25 performance:

- X.25 adapter performance
- DTE-to-DCE line speed and throughput class
- Application design

10.3.1 X.25 Adapter

In some circumstances, the X.25 adapter attachment may limit the throughput. The time taken by the attachment to process a packet is almost constant and independent of the size of the packet. So if the packets are small and both the DTE-to-DCE line speed and throughput class are high, the X.25 adapter may spend more time assembling or disassembling packets than in sending them.

10.3.2 DTE-to-DCE Line Speed and Throughput Class

If we consider only the speed of the lines between the DCEs and DTEs and the throughput class measuring the maximum throughput of the network itself, the end-to-end throughput will be limited by the smallest of these two values. For example:

- If we have DTE-to-DCE lines at 19200 bps and the throughput class subscribed to is 2400 bps, the theoretical throughput of the network will be 2400 bps.

- If we have DTE-to-DCE lines at 2400 bps and a throughput class of 9600 bps, the theoretical throughput of the network will also be 2400 bps.

The throughput class, which is the class of service you are paying for, and the DTE-to-DCE clocking speed of the modems are apparently unrelated. However, the previous example shows that the best price/performance ratio is achieved when they both have the same value. Also, some networks do not implement throughput class.

10.3.3 Influence of the Application Design

The way you write the data transfer application is also very important. Two factors may limit the performance: the processing speed of the application that cannot keep up with the transmission speed of the network, or a trade-of-design like requesting an end-to-end acknowledgment of each packet before sending the next. Let us consider, for example, a network with access lines at 19200 bps and the same value for the throughput class. Suppose that the transit time in the network is 200 ms (typical value) and you are sending 100-byte packets. If you are waiting for acknowledgment from the other end before sending the next packet, this will take 400 ms. In this case, the throughput will be only $100/4 = 250$ bytes per second.

10.4 Network Configuration Parameters

When you customize your network parameters, you must consider the following points:

- Since there is overhead with every packet, maximize the packet size and send full packets whenever possible. Also, try to eliminate unnecessary protocol-only packets for the higher level protocols like TCP/IP and SNA.
- Do not use D-bit acknowledgement because it generates an acknowledgement for each packet sent.
- Maximize frame window size. Keep in mind that when a packet is lost, all the packets of the window where the loss has occurred will be sent again. So, when your network is reliable, you can set a big window size.

10.5 Applications

Performance considerations are mainly related to the applications. As a result, some of the most commonly used applications are terminal emulators. We should be aware of how terminal emulators behave in an X.25 network.

10.5.1 X.25 LPP PAD Versus TCP/IP

X.25 LPP PAD will always perform better than TCP/IP. The reason is that TCP/IP, in general, and telnet, in particular, do not utilize the network very efficiently. TCP/IP does the same thing on a PSDN that it does on a LAN, but because the PSDN offers less bandwidth and the PSDN suppliers typically charge on a per-packet basis, efficient network utilization can become very important to X.25 users.

Because PADs typically service dumb ASCII terminals (for example 3101, 3151, PCs running asynchronous terminal emulation programs), they must provide intelligence on behalf of the terminal. As the user types, the data is buffered by the PAD (or RISC System/6000) until the data forwarding signal is received. This

signal is configurable (please refer to 5.1, "Using a PAD" on page 91) and typically, it might just be a carriage return. When this signal is received, the packet is made with the buffered data addressed to the target host and sent across the network. Notice that you may not send a full packet, but you are certainly sending more than one character here, so that is better than telnet. In addition, you can set the PAD to provide the echo to the terminal so there is no echo back from the remote host. There is no acknowledgement (X.25 provides error detection/correction so TCP/IP's is redundant) saving even more packets than when using telnet.

Note: When a virtual circuit is established for an application such as X.25 LPP PAD or TCP/IP between two sites, all traffic for that application flows over that single virtual circuit. It allows you to minimize the number of SVCs.

10.5.2 TCP/IP

The telnet application tends to cause the transmission of many small packets. This is not good utilization of X.25. The ftp application tends to send larger packets and better utilizes X.25.

Telnet is an interactive application which generally sends a single character per packet. To make matters worse, each character must also be echoed from the remote system back to the user's screen, then acknowledged. This means each user keystroke typically generates three packets with telnet.

Choosing a batch-mode application like ftp means that the user will send full packets for the most part. For maximum efficiency, users should consider the relationship between the TCP/IP datagram size (the Maximum Transmission Unit or MTU) and the X.25 packet size.

By definition, the maximum size of an IP datagram is 65536 bytes. Assuming a 20-byte IP header, this leaves up to 65516 bytes for the data in the datagram. If the IP datagram exceeds the size of the underlying MTU, fragmentation occurs. Every TCP/IP implementation must support a minimum IP datagram size of 576 bytes. A network can have an MTU smaller than 576 bytes, but any host must be able to reassemble the fragmented IP packets into at least a 576-byte IP datagram.

Since UDP packets are transmitted using IP, it is especially important to note that if the sum of the UDP header and the IP header causes the datagram to exceed the network's MTU size, fragmentation will occur.

It is not the same for TCP. TCP breaks the data up into segments. The size of the segment is negotiated between the two processes when the connection is established.

To determine the MTU size, issue the following command:

```
# netstat -i
```

Here is an example of output:

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
lo0	1536	<Link>		20	0	20	0	0
lo0	1536	127	loopback	20	0	20	0	0
xt0	576	<Link>		67	0	60	0	0
xt0	<u>576</u>	9.3.5	killix	67	0	60	0	0

To modify the MTU size, you can use the `ifconfig` command or go through SMIT:

```
smit articmpx or smit portmaster
  -User Applications
    -TCP/IP
      -Network Interfaces
        -X25 LPP Network Interface Drivers
```

Or use the fastpath: `smit chifsx25`

```

X.25 LPP Network Interface Drivers

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Network Interface                                [Entry Fields]
                                                xs0

Maximum IP PACKET SIZE for THIS DEVICE      [1500]      #

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Undo      F6=Command      F7=Edit      F8=Image
F9=Shell     F10=Exit      Enter=Do
```

If two machines on the same network have different MTU sizes set and they start communicating across the network, problems can occur. This would happen when a machine with a large MTU size has a packet to send which fits within its MTU size but is larger than the MTU size of the receiving system. It is best to set all machines on the network to the same MTU size.

The `iptrace` command is helpful in understanding how TCP/IP is utilizing the network. The utilization characteristics are the same whether you are on token ring, Ethernet or X.25. However, do not leave it running for longer than a couple of commands because the trace information grows quickly. The procedure is:

1. To start `iptrace`:
`# iptrace -d <server_name> -b /tmp/iptrace.out`
2. Execute a couple of commands.
3. To stop the trace:
`# kill `ps -e | grep iptrace | cut -c0-6``
4. To obtain a readable report of the trace:
`# ipreport /tmp/iptrace.out > /tmp/ipreport`
5. Then you can edit the resulting file: `/tmp/ipreport`.

You can also look at `x25mon` to see network utilization. Use the procedure:

1. To start xmonitor:

```
# x25mon -f -n -n sx25a0 > /tmp/x25mon_trace &
```

2. Run a couple of commands.

3. To stop xmonitor:

```
# kill `ps -e | grep x25mon | cut -c0-6`
```

4. Then you can edit the resulting file: /tmp/xmon_trace.

When reading the trace, a full packet with M-bit (also called the “MORE bit”) set, followed immediately by a small packet, should signal users to review the balance between packet and MTU sizes. The M-bit indicates that X.25 has to fragment the data into more than one packet. It sets the M-bit to say that more data follows. When very small packets always seem to follow a full packet, the user must either decrease the application’s MTU size, or ask the network provider for a larger X.25 packet size. The goal is to fill most of the packets, either in a single packet or in successive packets using the M-bits. Note that you can subscribe to up to 1KB packets in CCITT 1980 and 4KB packets in CCITT 1984 networks, but your per-packet charge is higher.

10.5.3 SNA

The options that we discussed earlier for TCP/IP also apply for SNA. Users must balance the SNA Request/Response Unit (RU) size with the X.25 packet size. You can use SNA and x25mon traces to visualize the traffic and customize the RU size in case of problems.

10.6 Memory Buffers

The X.25 device driver (and all communications drivers) use mbufs for storing data between the actual communications link and the application requesting the data. Usually, there are not many mbufs available, as they are defined for temporary storage. If a customer’s application is trying to send large data sequences, it is very likely that the system will run out of mbufs in which to store the incoming data. This will not kill the system, but it will probably disrupt all communication facilities until you reboot the machine.

You can show the memory buffers using the command:

```
# netstat -m
```

Here is an example of the output of the netstat command:

```
369 mbufs in use:
  262 mbufs allocated to data
   1 mbufs allocated to packet headers
  36 mbufs allocated to socket structures
  52 mbufs allocated to protocol control blocks
   8 mbufs allocated to routing table entries
   7 mbufs allocated to socket names and addresses
   3 mbufs allocated to interface addresses
0/18 mapped pages in use
164 KBytes allocated to network (56% in use)
0 requests for memory denied
0 requests for memory delayed
0 calls to protocol drain routines
```

If you have a value other than 0 in the two highlighted lines (request for memory denied/delayed) then you will probably have to increase the number of mbufs:

- With the no command

The no -a command will give you a list of all network options. The parameter which deals with the mbufs is thewall. To change it, issue:

```
# no -o thewall=<new_value>
```

- With the chdev command

Use first the lsattr command to show the value of the mbufs parameter:

```
# lsattr -E -l sys0
```

Here is an example of what we have on our system:

```
⋮  
maxmbuf      2048    Maximum KBytes of real memory allowed for MBUFS True  
⋮
```

To change the value, issue:

```
chdev -l sys0 -a maxmbuf=<new_value>
```

To simulate the overrunning of the mbufs, you can do a spray across the xs0 interface. This command uses TCP/IP, and it floods the adapter with datagrams of the size and number you specify. This test, however, will not hang the system. Rather, it will time out when TCP/IP decides that the network (the link) is no longer responding.

Appendix A. Summary of the SMIT Main Menu Used for Configuration

This appendix contains the SMIT configuration screens for the X.25 LPP base, PAD, TCP/IP and SNA Server/6000 support. The screens shown are those for AIX V4. For AIX V3.2.5, the screens may vary slightly.

A.1 X.25 Adapter Configuration

The following table contains all the SMIT screens that are used to configure an X.25 adapter.

Table 17 (Page 1 of 4). SMIT Menus for the Base X.25 Configuration

After having installed the X.25 LPP, if you are using the X.25 CoProcessor/2, you need to install the device driver.

Devices

- Communications
 - X.25 CoProcessor/2 or Multiport/2 Adapter
 - Adapter
 - Manage Device Drivers for X.25 ...
 - Manage X.25 LPP Device Driver
 - Add a Device Driver

Fastpath = smit cx25str_dd

```
Manage X.25 LPP Device Driver

Move cursor to desired item and press Enter.

List All Configured Device Drivers to a CoProcessor/2 Adapter
Add a Device Driver
Remove a Device Driver
Configure a Defined Device Driver
Manage X.25 Ports - A -
Manage the X.25 Triple-X PAD - B -

-----
Parent Adapter
-----
Move cursor to desired item and press Enter.

ampx0 Available 00-04 X.25 CoProcessor/2 Adapter

F1=Help          F2=Refresh      F3=Cancel
F8=Image         F10=Exit       Enter=Do
F1| /=Find      n=Find Next
F9 -----
```

Fastpath = smit cx25str_mp

```
Manage X.25 Ports

Move cursor to desired item and press Enter.

List All Defined Ports
Add Port - 1 -
Move Port Definition
Change / Show Characteristics of Port - 2 -
Remove Port
Configure a Defined Port
Add Comio Interface to Port
Remove Comio Interface from Port
Add TCP/IP Interface to Port
Remove TCP/IP Interface from Port

F1=Help          F2=Refresh      F3=Cancel      F8=Image
F9=Shell         F10=Exit       Enter=Do
```

A - Manage X.25 Ports

Table 17 (Page 2 of 4). SMIT Menus for the Base X.25 Configuration

Fastpath = smit mksx25c

```

                                Add an X.25 Port

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
Parent adapter driver                twd0
* PORT number                        [0]                +
* Local network user address (NUA)   [3106010760]       #
* Network identifier                 [other private]     +
Country prefix                       []                #

F1=Help      F2=Refresh    F3=Cancel    F4=List
F5=Reset     F6=Command   F7=Edit     F8=Image
F9=Shell     F10=Exit     Enter=Do
    
```

Fastpath = smit x25str_mp_csp

```

                                Change / Show Characteristics of Port

Move cursor to desired item and press Enter.

Change / Show General Parameters
Change / Show Packet Parameters
Change / Show Frame Parameters
Change / Show Default for Permanent Virtual Circuits
Manage Non-Default Permanent Virtual Circuit

F1=Help      F2=Refresh    F3=Cancel    F8=Image
F9=Shell     F10=Exit     Enter=Do
    
```

This screen allows access to the detailed parameters of the X.25 Configuration.

Table 17 (Page 3 of 4). SMIT Menus for the Base X.25 Configuration

Fastpath = smit x25str_mp_csp_g_sel

```

Change / Show X.25 General Parameters

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                     [Entry Fields]
Port name                                sx25a0
Local network user address (NUA)         [3106010760] #
Calling address in call request/accept packet [allow] +
Enable DLPI interface ONLY               [no] +
PVC - lowest logical channel number      [0] +
PVC - Number of logical channels         [0] +
Incoming SVC - lowest logical channel number [0] +
Incoming SVC - number of logical channels [0] +
Two-way SVC - lowest logical channel number [1] +
Two-way SVC - number of logical channels [64] +
Outgoing SVC - lowest logical channel number [0] +
Outgoing SVC - number of logical channels [0] +

X.32 Configuration
*****
Use X.32 XID Exchange                    [no] +
X.32 XID Identity                        []
X.32 XID signature                       []

Dial-Up Configuration
*****
Connection Type                         [Direct/Leased] +
V25bis Call Establishment Method         [Addressed] +
Phone Number or Address to Call         []
Maximum Connection Delay                 [10] #
Enable/Disable DSR Polling              [disable] +
DSR polling timeout                      [30] #
[BOTTOM]

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do
    
```

Fastpath = smit x25str_mp_csp_f_sel

```

Change / Show X.25 Frame Parameters

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Port name                                [Entry Fields]
T1 retransmission timer                  [3] +
T2 acknowledgement timer                 [1] +
T3 idle timer                            [15] +
T4 activity timer                        [0] +
N2 retransmission count                  [10] +
Frame modulo                             [8] +
Frame window size                        [7] +
Allow automatic DTE/DCE detection        [yes-automatic] +
Connection mode                          [active-sendsABM] +

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do
    
```

Fastpath = smit x25str_mp_csp_p_sel

```

Change / Show X.25 Packet Parameters

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                     [Entry Fields]
Port name                                sx25a0
CCITT support                            [1988] ++
Packet modulo                             [8] ++
Type of line                              [DTE] +
Packet layer DTE/DCE line alteration policy [per OSI 8208 procedure] +
Disconnect on inactivity                  [no] +
Registration                              [no] +
A-bit                                     [off] +
Network intermediate data unit (nidu) size [200] ++

Default Attributes for SVCs
*****
Default receive packet size               [128] ++
Default transmit packet size              [128] ++
Default receive packet window             [3] ++
Default transmit packet window            [3] ++
Default receive throughput class           [64000] ++
Default transmit throughput class          [64000] ++

Maximum Negotiable Attributes for SVCs
*****
Maximum receive packet size               [1024] ++
Maximum transmit packet size              [1024] ++
Maximum receive packet window             [5] ++
Maximum transmit packet window            [5] ++

ISO8208-Defined Timers
*****
T20 restart timer                        [180] ++
T21 call timer                           [200] ++
T22 reset timer                           [180] ++
T23 clear timer                           [180] ++
T24 window status transmission timer      [60] ++
T25 window rotation timer                 [200] ++
T26 interrupt timer                       [180] ++
T27 reject response timer                  [60] ++
T28 registration timer                    [300] ++
R20 retransmission count restart timer    [3] ++
R22 retransmission count reset timer      [3] ++
R23 retransmission count clear timer      [3] ++
R25 retransmission count window rotation timer [0] ++
R27 retransmission count reject timer      [0] ++
R28 retransmission count registration timer [1] ++

Packet-Level Features
*****
Fast select mode                          [allow] +
D-bit                                     [allow] +
Reverse charging                           [allow] +
Maximum facility field length              [109] ++
Interrupt data size                        [32] ++
[BOTTOM]

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do
    
```

Table 17 (Page 4 of 4). SMIT Menus for the Base X.25 Configuration

These screens are used for PVC configuration only.

Fastpath = smit x25str_mp_csp_d_sel

```

Change / Show X.25 Default PVC Parameters

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                     [Entry Fields]
Port name                             sx25a0
PVC receive packet size                [128]      ++
PVC maximum transmit packet size       [128]      ++
PVC maximum receive packet window      [3]        ++
PVC maximum transmit packet window     [3]        ++
PVC D-bit                              [allow]    +

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do
    
```

Fastpath = smit x25str_mp_mnd_a_sel

```

Add X.25 Non-Default PVC Parameters

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                     [Entry Fields]
Port name                             sx25a0
* PVC number                           []          #
PVC receive packet size                [128]      ++
PVC maximum transmit packet size       [128]      ++
PVC maximum receive packet window      [3]        ++
PVC maximum transmit packet window     [3]        ++
PVC D-bit                              [allow]    +

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do
    
```

Fastpath = smit x25str_pad

```

Manage the X.25 Triple-X PAD

Move cursor to desired item and press Enter.

Start the Triple-X PAD X.29 Daemon
Stop the Triple-X PAD X.29 Daemon

F1=Help      F2=Refresh      F3=Cancel      F8=Image
F9=Shell     F10=Exit       Enter=Do
    
```

B - Manage the X.25 Triple-X PAD

This screen is used for the PAD function only.

A.2 TCP/IP Configuration

The following is a quick way to customize TCP/IP over the X.25 link. Table 18 only shows the main SMIT screens used for this configuration.

Note: Of course, only the screens for one RISC System/6000 are described.

Table 18 (Page 1 of 3). SMIT Menus for the TCP/IP Configuration

Communications
→TCP/IP
→Minimum Configuration & Startup

Fastpath = smit mktcpip

```

                                Minimum Configuration & Startup
                                -----
                                To Delete existing configuration data, please use Further Configuration menus

                                Type or select values in entry fields.
                                Press Enter AFTER making all desired changes.

                                [Entry Fields]
                                * HOSTNAME                               [kili]
                                * Internet ADDRESS (dotted decimal)    [9.3.5.205]
                                Network MASK (dotted decimal)          [255.255.255.0]
                                * Network INTERFACE                     xt0
                                NAMESERVER
                                  Internet ADDRESS (dotted decimal)  [9.3.1.74]
                                  DOMAIN Name                          [itsc.austin.ibm.com]
                                Default GATEWAY Address                 [9.3.1.74]
                                  (dotted decimal or symbolic name)
                                START TCP/IP daemons Now               no          +

                                F1=Help      F2=Refresh    F3=Cancel    F4=List
                                F5=Reset     F6=Command   F7=Edit     F8=Image
                                F9=Shell    F10=Exit    Enter=Do

```

Communications
→TCP/IP
→Further Configuration

Fastpath = smit configtcp

```

                                Further Configuration
                                -----
                                Move cursor to desired item and press Enter.

                                Hostname
                                Static Routes                - 3 -
                                Network Interfaces
                                Name Resolution                - 1 -
                                Client Network Services
                                Server Network Services
                                Manage Print Server
                                Select BSD style rc Configuration

                                F1=Help      F2=Refresh    F3=Cancel    F8=Image
                                F9=Shell    F10=Exit    Enter=Do

```

Table 18 (Page 2 of 3). SMIT Menus for the TCP/IP Configuration

1 - Create an entry in /etc/hosts for the local and remote hosts.

Fastpath = smit mkhostent

```

                                Add a Host Name

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
* INTERNET ADDRESS (dotted decimal)      [9.3.5.205]
* HOST NAME                               [kili]
ALIAS(ES) (if any - separated by blank space) []
COMMENT (if any - for the host entry)     []

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do
    
```

Communications
 ->TCP/IP
 ->Further Configuration
 ->Network Interfaces

Fastpath = smit inet

```

                                Network Interface Selection

Move cursor to desired item and press Enter.

List All Network Interfaces
Add a Network Interface
Change / Show Characteristics of a Network Interface - 2 -
Remove a Network Interface
X.25 LPP IP Host Configuration - 4 -

F1=Help      F2=Refresh      F3=Cancel      F8=Image
F9=Shell     F10=Exit       Enter=Do
    
```

2 - Initialize and start IP/X.25 interface.

Fastpath = smit chinnet or smit mkinetxt

```

                                Change / Show an X.25 LPP Network Interface

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
Network Interface Name                    xt0
INTERNET ADDRESS (dotted decimal)        [9.3.5.205]
Network MASK (hexadecimal or dotted decimal) [255.255.255.0]
Current STATE                             up      +

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do
    
```

Table 18 (Page 3 of 3). SMIT Menus for the TCP/IP Configuration

3 - Add a static route for each remote host if you use two different IP network addresses.

Fastpath = smit mkroute

```

                                Add a Static Route

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
Destination TYPE                               host          +
* DESTINATION Address                          [129.35.1.93]
  (dotted decimal or symbolic name)
* GATEWAY Address                              [9.3.1.74]
  (dotted decimal or symbolic name)
* METRIC (number of hops to destination gateway) [1]           #
  Network MASK (hexadecimal or dotted decimal) [255.255.255.0]

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do
    
```

4 - For each remote host, create an entry in the IP hostname to NUA translation table.

Fastpath = smit mksx25s

```

                                Add X.25 LPP IP SVC Host Entry

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]
[TOP]
* Remote HOSTNAME                             [fili]
* Remote DTE Address                          [3106010761] #
----- Optional X.25 Facilities -----
RECEIVED data PACKET size                    +
TRANSMITTED data PACKET size                  +
RECEIVED data WINDOW size                    [] #
TRANSMITTED data WINDOW size                  [] #
CLOSED USER GROUP selection                   [] #
CLOSED USER GROUP WITH OUTGOING ACCESS selection [] #
Recognized Private Operating Agency (RPOA)    []
User-Defined Facilities                       []
----- CALL USER Data -----
Note: RFC-1356 (supersedes RFC-877) mandates
the first byte of call user data is 0xcc. If you
do not put 'cc' as the first byte, SMIT
will put it there for you.
Call User Data                                []
[BOTTOM]

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do
    
```

A.3 SNA Configuration

First, you must add the X.25 QLLC Data Link Control for the X.25 Co-Processor. From the first menu of the SMIT interface, select:

```
Devices
-Communication
  -X.25 Coprocessor/2 or Multiport/2 Adapter
    -Services
      -X.25 QLLC Data Link Controls
        -Add a QLLC Data Link Control
```

Or use the fastpath: `smit cmdllc_qllc_mk`.

If using the Portmaster Adapter:

```
Devices
-Communication Devices
  -Portmaster Adapter/A
    -Services
      -Data Link Controls
        -Add an SDLC Data Link Control
```

Or use the fastpath: `smit cmdllc_sdllc_mk`.

The system gives you a `dlcqllc` as available.

Then you can go on with the SNA configuration profiles that are required to use SNA over X.25. (See Table 19 on page 235.)

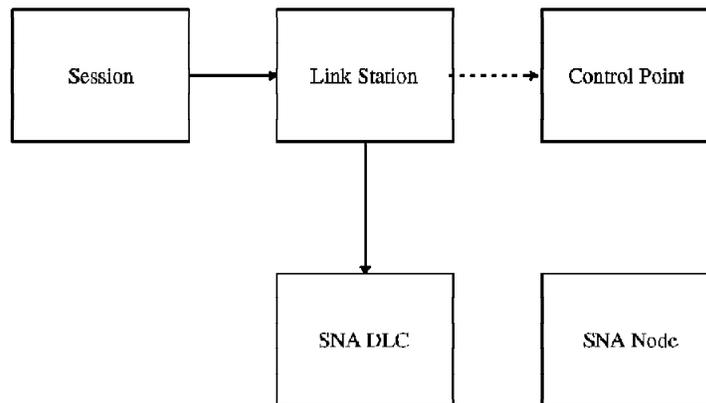


Figure 56. AIX SNA Services/6000 Main Profiles

Table 19 (Page 1 of 4). SMIT Menus for the SNA Configuration

To perform the Initial Node Setup, from the SMIT main menu select:

- Communications Applications and Services
 - SNA Server/6000
 - Configure SNA Profiles
 - Initial Node Setup

Fastpath = smit _snainit

```

Initial Node Setup

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Control Point name          [Entry Fields]
Control Point type          [AUSCP]
Local network name          appn_end_node      +
XID node ID                 [USIBMRA]
                             [*]

Optional link station information:

Link station type           x.25_svc
Link station name           []
* Calling link station?     yes                +
Remote X.25 station address [3106001984]      X

F1=Help      F2=Refresh  F3=Cancel   F4=List
F5=Reset     F6=Command  F7=Edit     F8=Image
F9=Shell     F10=Exit   Enter=Do
  
```

To change the Control Point Profile from the SMIT main menu select:

- Communications Applications and Services
 - SNA Server/6000
 - Configure SNA Profiles
 - Advanced Configuration
 - Control Point
 - Change/Show a Profile

Fastpath = smit _snaopc

```

Change/Show Control Point Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

* Profile name              [Entry Fields]
XID node ID                 node_cp
                             [*]
Network name                [USIBMRA]
Control Point (CP) name     [AUSCP]
Control Point alias         [AUSCP]
Control Point type          appn_end_node      +
Maximum number of cached routing trees [500]      #
Maximum number of nodes in the TRS database [500]      #
Route addition resistance    [128]          #

Comments                    []

F1=Help      F2=Refresh  F3=Cancel   F4=List
F5=Reset     F6=Command  F7=Edit     F8=Image
F9=Shell     F10=Exit   Enter=Do
  
```

Table 19 (Page 2 of 4). SMIT Menus for the SNA Configuration

To create the SNA DLC Profile from the SMIT main menu, select:

- Communications Applications and Services
 - SNA Server/6000
 - Configure SNA Profiles
 - Advanced Configuration
 - links
 - X.25
 - X.25 SNA DLC
 - Add a Profile

Fastpath = smit _snaX25linkmk

```

                                Add X.25 SNA DLC Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                [Entry Fields]
* Profile name                        [xdlc]
Data link device name                 [x25s0]
Force disconnect time-out (1-600 seconds) [120]
User-defined maximum I-Field size?    no
  If yes, Max. I-Field size (265-30729) [1417]
Max. num of active link stations (1-255) [100]
Number reserved for inbound activation [0]
Number reserved for outbound activation [0]
Local X.25 network address            [3106010760]
Receive window count (1-127)         [7]
Transmit window count (1-127)        [7]

Secondary and Negotiable Stations
  Secondary inactivity time-out (1-255 sec) [30]

Primary and Negotiable Stations
  Primary repoll frequency (1-255 seconds) [30]
  Primary repoll count (1-255)           [10]

Link Recovery Parameters
  Retry interval (1-10000 seconds)       [60]
  Retry limit (0-500 attempts)          [20]

Comments                               []

[BOTTOM]

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image
F9=Shell     F10=Exit       Enter=Do
  
```

Table 19 (Page 3 of 4). SMIT Menus for the SNA Configuration

To create a link station profile from the SMIT main menu, select:

- Communications Applications and Services
 - SNA Server/6000
 - Configure SNA Profiles
 - Advanced Configuration
 - links
 - X.25
 - X.25 Link Station
 - Add a Profile

Fastpath = smit _snaX25attcmk

```

Add X.25 Link Station Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                     [Entry Fields]
* Profile name                            [xlink]
Use APPN Control Point's XID node ID?    yes +
  If no, XID node ID                       [07100123]
* SNA DLC Profile name                     [xdlc] +
Stop link station on inactivity?          no +
  If yes, Inactivity time-out (0-10 minutes) [0] #
LU address registration?                   no +
  If yes, LU Address Registration Profile name [] +
Trace link?                               no +
  If yes, Trace size                       long +
X.25 level                                 1984 +
Station type                              secondary +

Adjacent Node Identification Parameters
Verify adjacent node?                     no +
Network ID of adjacent node               []
CP name of adjacent node                  []
XID node ID of adjacent node (LEN node only) [*]
Node type of adjacent node                learn +

Link Activation Parameters
Solicit SSCP sessions?                   yes +
Initiate call when link station is activated? yes +
Virtual circuit type                      switched +
  If permanent,
  Logical channel number of PVC (1-4095)   [1] #
  If switched,
  Listen name                              [IBMQLLC]
  Remote station X.25 address              [3106001984] #
  X.25 Optional Facilities Profile name     [xfacs] +
Activate link station at SNA start up?    no +
Activate on demand?                      no +
CP-CP sessions supported?                no +
  If yes,
  Adjacent network node preferred server? no +
  Partner required to support CP-CP sessions? no +
  Initial TG number (0-20)                 [0] #

Restart Parameters
Restart on activation?                    no +
Restart on normal deactivation?           no +
Restart on abnormal deactivation?         no +

Transmission Group COS Characteristics
Effective capacity                        [9600] #
Cost per connect time                     [128] #
Cost per byte                             [128] #
Security                                  public_switched_network> +
Propagation delay                         packet_switched_network> +
User-defined 1                            [128] #
User-defined 2                            [128] #
User-defined 3                            [128] #

Comments                                  []
[BOTTOM]

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command      F7=Edit       F8=Image
F9=Shell     F10=Exit        Enter=Do

```

Table 19 (Page 4 of 4). SMIT Menus for the SNA Configuration

To create an X.25 optional facilities profile, from the SMIT main menu, select:

```

Communications Applications and Services
- SNA Server/6000
  - Configure SNA Profiles
    - Advanced Configuration
      - Links
        - X.25
          - X.25 Optional Facilities
            - Add a Profile
  
```

Fastpath = smit _snaX25optsmk

```

Add X.25 Optional Facilities Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[ ] [Entry Fields]
* Profile name [xfacs]
Throughput class for received data 9600 +
Throughput class for transmitted data 9600 +
Closed user group? no +
  If yes, Index to closed group [0] #
Closed user group outgoing access? no +
Network user identification? no +
  If yes,
    Network user ID name [ ]
    Network user ID in hex? no +
Reverse charging? no +
RPOA? no +
  If yes, Data network ID codes [ ]
Packet size for received data 128 +
Packet size for transmitted data 128 +

Comments [ ]

[ ] [BOTTOM]

F1=Help      F2=Refresh   F3=Cancel    F4=List
F5=Reset     F6=Command   F7=Edit      F8=Image
F9=Shell     F10=Exit    Enter=Do
  
```

To create an LU 2 session profile using the manual method, from the SMIT main menu, select:

```

Communications Applications and Services
- SNA Server/6000
  - Configure SNA Profiles
    - Advanced Configuration
      - Sessions
        - LU 2
          - Add a Profile
  
```

Fastpath = smit _snasess2mk

```

Add LU 2 Session Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[ ] [Entry Fields]
* Profile name [xsess02]
Local LU name [S4080502]
* Local LU address (1-255) [2] #
System services control point
(SSCP) ID (*, 0-65535) [*]
Link Station Profile name [xlink] +
Network name [ ]
Remote LU name [ ]
Maximum number of rows [24] #
Maximum number of columns [80] #

Comments [ ]

F1=Help      F2=Refresh   F3=Cancel    F4=List
F5=Reset     F6=Command   F7=Edit      F8=Image
F9=Shell     F10=Exit    Enter=Do
  
```

An LU 2 session profile will need to be created for each display session required over X.25.

To create an LU 3 session profile using the manual method, from the SMIT main menu, select:

```

Communications Applications and Services
- SNA Server/6000
  - Configure SNA Profiles
    - Advanced Configuration
      - Sessions
        - LU 3
          - Add a Profile
  
```

Fastpath = smit _snasess3mk

```

Add LU 3 Session Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[ ] [Entry Fields]
* Profile name [xsess09]
Local LU name [S4080509]
* Local LU address (1-255) [9] #
System services control point
(SSCP) ID (*, 0-65535) [*]
Link Station Profile name [xlink] +
Network name [ ]
Remote LU name [ ]
Maximum number of rows [24] #
Maximum number of columns [80] #

Comments [ ]

F1=Help      F2=Refresh   F3=Cancel    F4=List
F5=Reset     F6=Command   F7=Edit      F8=Image
F9=Shell     F10=Exit    Enter=Do
  
```

An LU 3 session profile will need to be created for each printer session required over X.25.

Appendix B. Differences Between X.25 LPP and AIX V3 Base X.25 Support

If you are migrating from the AIX V3 base X.25 support to the X.25 LPP you will likely want to know what are the main differences between them. In this chapter we will explain what hardware, functionality, configuration and setup procedures have been changed with the new X.25 LPP.

B.1 Hardware Differences

The X.25 LPP supports:

- The X.25 Co-Processor/2 adapter (FC 2960)
- The X.25 Co-Processor ISA-bus adapter (FC 6753), only on AIX Version 4.1 machine with an ISA-bus and AIXlink/X.25 LPP
- The ARTIC Portmaster Adapter/A with 1MB (FC 7006)
- The ARTIC Portmaster Adapter/A with 2MB (FC 7008)
- The ARTIC960 Adapter with 8-port RS232 daughter card (FC 2929)
- The ARTIC960 Adapter with 6-port V.36/V.35 daughter card (FC 2936)
- The ARTIC960 Adapter with 8-port X.21 daughter card (FC 2938)
- The ISA Portmaster Multiport/2 (PS/2 FC 6590 (USA), FC 5309 (EMEA))

The ARTIC Portmaster Adapters with 1MB and 2MB can now be used to connect to an X.25 network. They are Portmaster adapters which support V.24, V.35 and X.21 interfaces.

Table 20. X.25 Supported Adapters

Adapter	AIX V3 Base X.25 support	X.25 LPP
X.25 Co-Processor/2 Micro Channel	supported	supported
X.25 Co-Processor ISA-bus	not supported	AIXLink Only
ARTIC Portmaster Adapter/A 1MB	not supported	supported
ARTIC Portmaster Adapter/A 2MB	not supported	supported
ARTIC960 Adapter	not supported	AIXLink V1.1.3 or later
ISA Multiport/2 Adapter	not supported	AIXLink V1.1.3 or later

B.2 Functionality Differences

The X.25 LPP provides the following new features:

X.25 LPP New Features

- Support of the International Telegraph and Telephone Consultative Committee (CCITT) 1988 X.25
- Packet Layer Programming Interface Network Provider Interface (NPI)
- Frame Layer Programming Interface Data Link Provider Interface (DLPI)
- Compatibility application program interface (API) for applications written to the base Version 3 X.25 support
- Triple-X (X.3, X.28, X.29) Packet Assembler/Disassembler (PAD)
- PAD supported printing
- Configurable PAD access controls
- Dial up on demand facility
- Simple Network Management Protocol (SNMP) support for data items from the machine instruction buffers (MIBs) for the packet and frame layers
- V25bis support
- Supports up to 512 logical channels per line
- Automatic DTE configuration
- Supports an aggregate sustained rate of 200 128-bytes packets per second (measured at the packet layer API) for each adapter in the system

Table 21. Functional Differences

Functionality	AIX V3 base X.25 support	X.25 LPP
CCITT Conformance	1984	1988
NPI	NO	YES
DLPI	NO	YES
API Library	YES	YES•
PAD Support	NO•	YES
SNA Support	YES	YES
TCP/IP Support	YES	YES
SNMP Support	NO	YES
Automatic DTE configuration	NO	NO
Logical channels per line	64	512
Packets per second (128-byte packets)	35	200

Note:

- Applications based on the API library are only supported when the COMIO emulator is configured on the port.
- Only with third-party software.

B.2.1 Differences between (CCITT) 1988 and 1984 X.25

The 1988 recommendations further refine and expand the CCITT X.25.

B.2.1.1 Packet Layer

The main differences between CCITT 1994 and 1988 X.25 recommendations at the packet level are:

Network User Identification (NUI): Facilities related to Network User Identification (NUI) have been divided into three parts:

- NUI_subscription
- NUI_override
- NUI_selection

DTE/DTE Operation: DTE-to-DTE operation without an intervening network has been defined. In this situation, one DTE must act as DCE. The DTE acting as DCE at packet layer may be acting as DTE at Data Link Layer and vice versa. This is an optional facility.

Circuit-switched Connection without Prior Agreement: A circuit-switched connection without prior agreement (such as electronic mail-order) has been defined and default values specified for all applicable parameters. This is an optional capability.

Throughput Class of 64000 bits/s: A new throughput class of 64000 bits per second has been defined. This is an optional capability.

Address Block Definition: A new Address Block has been defined for call setup and clearing packets which allows addresses of 12 or 15 digits. This is an optional capability.

TOA/NPI Address Subscription: A new facility, TOA/NPI_Address_Subscription, has been added to accommodate E.164 (ISDN) addresses of up to 17 digits in length. This addition results in a redefinition of the address block and the consequent definition of new formats for the CALL_REQUEST, CALL_ACCEPTED, CALL_CONNECTED, CLEAR_REQUEST, CLEAR_INDICATION and CLEAR_CONFIRMATION packets. This is an optional capability.

Call Deflection: Call_Deflection_selection facilities whereby the DTE forwards calls after receiving an INCOMING_CALL packet (unlike CALL_REDIRECTION, which is handled in the network and the originally called DTE never receives an INCOMING_CALL packet) has been added. There are three call deflection facilities:

- Call_Deflection_Subscription, which enables the DTE to request Call_Deflection_Selection.
- Call_Deflection_Selection, which may be used on a per-virtual-call basis only if Call_Deflection_Subscription has been subscribed to.
- Call_Deflection_Notification, which informs the alternate DTE that the call has been forwarded.

These are optional user facilities.

Priority Facility: A Priority Facility has been added to specify the priority of data on a connection and the priority to keep a connection. This is an optional capability.

Protection Facility: A Protection Facility has been added to specify the protection of data on a connection.

Maximum Size of Called and Calling Address Extension: The maximum size of the called and calling address extension fields is extended from 32 to 40 digits, and an OSI/non-OSI indicator has been added. Support of the larger address is optional. However, a test of the OSI/non-OSI indicator is required to determine the size of the called and calling address.

Recognized Private Operating Agency: RPOA-related facilities have been subdivided into:

- RPOA_Subscription, which applies to all virtual calls
- RPOA_Selection, which applies to a given virtual call and does not require RPOA_Subscription

These are optional capabilities.

Mandatory Address Length Fields in CALL_ACCEPTED Packets: The use of the address-length fields in CALL_ACCEPTED packets is mandatory, even if they are set to zero.

Mandatory Facility Length Fields in CALL_ACCEPTED packets: The use of the Facility Length Fields in CALL_ACCEPTED packets is mandatory, even if they are set to zero.

Virtual Circuit Clearing/Resetting Failure: When a CLEAR_REQUEST packet is not confirmed within time-limit T23, the DTE will retry the call clearing procedure up to R23 times, at T23 intervals, before notifying the higher layer (virtual circuit user) of the failure; this leaves the logical channel in the DTE_CLEAR_REQUEST state (p6) rather than placing the logical channel in an inoperative state, as specified by early versions.

When a RESET_REQUEST packet is not confirmed within time-limit T22, the DTE will retry the resetting procedure up to R22 times, at T22 intervals, before notifying the higher layer (virtual circuit user) of the failure; this leaves the logical channel in the DTE_RESET_REQUEST state (d2) rather than placing the logical channel in an inoperative state, as specified by early versions.

B.2.1.2 Frame Layer

The main differences between CCITT 1994 and 1988 X.25 recommendations at the frame level are:

DTE/DTE Operation: Although not specified in CCITT Recommendation X.25, International Standard Organization ISO 7776 supports communication between two DTEs without an intervening network. Since there is no intervening network, link layer characteristics must be made by bilateral agreement rather than at subscription time. This is an optional capability but is required if communicating using Open Systems-Interconnect (OSI).

Clearing a FRMR Condition at the DCE: After the DCE has transmitted an FRMR response, the frame rejection condition is cleared when the DCE receives an FRMR response (in addition to when a SABM/SABME, DISC or DM is sent or received).

Maximum Number of Outstanding I-Frames: American National Standards ANS X3.100 specifies that all networks must support $k=7$. K is the maximum number of outstanding I-frames.

B.3 Configuration and Setup Differences

Installation and Setup Differences

- You need to order and install the X.25 LPP.
- You do not need to install the adapter microcode.
There is one exception to this, and that is the microcode for the ARTIC960 adapter, which is supplied on diskette with the adapter.
- You do not need to use xmanage to connect to the network.
- Several SMIT fastpaths have changed.
- Several attribute names have changed.

Since the X.25 LPP is a Licensed Program Product, your first step in your configuration and setup procedure is to install the X.25 code. You do not need to install the microcode from diskette anymore, it is now installed with the LPP. Your next step is to configure the device driver and the X.25 port. As soon as your X.25 port is configured and available, the X.25 LPP software tries to bring up the frame and packet layer, which means that you don't have to use xmanage to connect to the X.25 network. The command xmanage is not supported by the X.25 LPP.

You'll probably need to change some attributes such as number of virtual circuits, throughput, etc. If you are going to use SMIT fastpaths, be aware that many of them have changed.

Table 22. SMIT Fastpaths Differences

Parameter	Old SMIT Fast Path	New SMIT Fast Path
Change / Show X.25 General Parameters	x25csg	x25str_mp_csp_g_sel
Change / Show X.25 Frame Parameters	x25csf	x25str_mp_csp_f_sel
Change / Show X.25 Packet Parameters	x25csp	x25str_mp_csp_p_sel

The xroute command works only with X.25 ports that have COMIO emulation configured and must be used only when you have applications which use this emulation (xtalk, SNA, ...). The new TCP/IP implementation does not use the COMIO emulation, so you no longer need to add xroute entries when using more than one X.25/IP interface.

The X.25 LPP allows you to enable/disable only these two facilities:

- Fast Select

- Reverse Charging

B.4 Command Differences

Many commands that were used for management and configuration purposes have changed.

Command Differences Summary

- The `x25mon` command is now used for tracing an X.25 port.
- The `x25ip` command is now used for managing the NUA/IP table.
- The `lsx25` command has been added for listing the configuration of the X.25 support on the system.
- The `x25status` command has been added to display the status of the connection between the DTE and the DCE.
- The `x25debug` command has been added to enable more in-depth diagnostics to be carried out on the device driver and microcode.

The `xmonitor` command does not exist anymore. The `x25mon` command is now used to trace an X.25 port.

Table 23. Differences Between xmonitor and x25mon

	xmonitor	x25mon
Frame Layer	-frame	-f
Packet Layer	-packet	-p
Port	x25sn	-n sx25an

The `x25ip` command has the same syntax than the `x25xlate` command.

The `lsx25` command uses information available from the system configuration database to display the relationship between adapters, drivers, ports, etc. that are configured to use X.25.

The `x25status` command is a partial replacement for the `xmanage` function in the BOS X.25 facility. It enables the system administrator the ability to interigate the status of the X.25 ports. However, it does NOT enable the ports to be manipulated like the `xmanage` facility.

B.5 ODM/SMITAttributes Differences

Table 24 shows the differences in attribute names.

Table 24 (Page 1 of 2). SMIT Attribute Names

AIX V3 X.25 Support	AIX X.25 LPP
num_in_out_svc	bi_vc_num
in_out_svc	bi_vc_start
ccitt_support	ccitt

<i>Table 24 (Page 2 of 2). SMIT Attribute Names</i>	
AIX V3 X.25 Support	AIX X.25 LPP
connection_mode	connect_seq
d_bit	d_bit_accept
pvc_d_bit	def_pvc_d_bit
def_rx_pkt_size	def_rx_size
def_rx_through	def_rx_th
def_rx_pkt_win	def_rx_win
def_tx_pkt_size	def_tx_size
def_tx_through	def_tx_th
def_tx_pkt_win	def_tx_win
frame_modulo	f_modulo
fast_select	fs_mode
num_in_svcs	in_vc_num
in_svc	in_vc_start
line_type	line_type
local_nua	local_nua
max_rx_pkt_size	max_rx_size
max_rx_pkt_win	max_rx_win
max_tx_pkt_size	max_tx_size
max_tx_pkt_win	max_tx_win
n2_counter	n2
network_id	network_id
num_out_svcs	out_vc_num
out_svc	out_vc_start
pkt_modulo	pkt_modulo
num_of_pvcs	pvc_num
pvc_channel	pvc_start
rev_charging	rev_charge
t1_timer	t1
t21_timer	t21
t22_timer	t22
t23_timer	t23
t24_timer	t24
t25_timer	t25
t26_timer	t26
t4_timer	t4
zero_address	zero_address

B.6 Default Values of Important Parameters

The default values of several important parameters have changed. Table 25 shows the differences between the AIX V3 base X.25 support and the X.25 LPP.

<i>Table 25. Default Values of Important Parameters</i>		
Parameter	AIX V3 base X.25 Defaults	X.25 LPP Defaults
Frame window size	7	7
Frame modulo	8	8
Packet modulo	8	8
CCITT support	1980	1984
Default receive packet size	128	128
Default transmit packet size	128	128
Default receive packet window	2	3
Default transmit packet window	2	3
Default receive throughput class	9600	64000
Default transmit throughput class	9600	64000

Appendix C. X.25 Cables and Connectors

This appendix will be divided into two sections. Section C.1, "IBM X.25 Co-Processor and IBM X.25 Co-Processor/2" describes the cables and connectors used with the IBM X.25 Co-Processor and the IBM X.25 Co-Processor/2. These adapters use the same cables and connectors. Any reference to the X.25 Co-Processor or Co-Processor applies to both adapters.

Section C.2, "IBM Portmaster Adapter/2 and Multiport Model 2" on page 257 describes the cables and connectors used with the IBM ARTIC Portmaster Adapter/2, Multiport Model 2 and ARTIC960 adapters.

C.1 IBM X.25 Co-Processor and IBM X.25 Co-Processor/2

This section describes the cables and connectors used with the IBM X.25 Co-Processor and the IBM X.25 Co-Processor/2. These adapters use the same cables and connectors. Any reference to the X.25 Co-Processor, X.25 Co-Processor/2 or Co-Processor applies to both adapters.

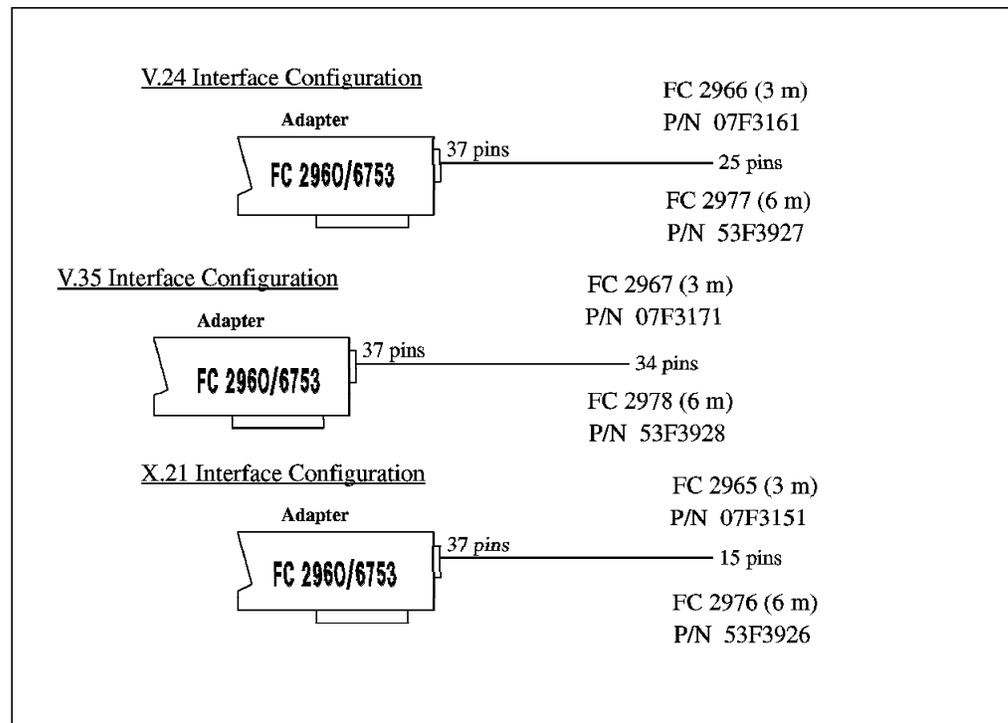


Figure 57. Co-Processor Adapters

C.1.1 X.25 Co-Processor 37-Pin Connector Pin Assignment

Table 26. X.25 Co-Processor 37-Pin Connector Pin Assignment

Pin	Circuit Designation	Symbol	X.21bis /V24	X.21bis /V35	X.21
1	Reserved				
2	Transmitted data	TXD	x		
3	Received data	RXD	x		
4	Request to send	RTS	x	x	
5	Clear to send	CTS	x	x	
6	Data set ready	DSR	x	x	
7	Signal ground	GND	x	x	x
8	Carrier detect	CD	x	x	
9	Cable ID 0	ID0	x		x
10	Transmitted data (A)	T (A)			x
11	Control (A)	C (A)			x
12	Received data (A)	R (A)			x
13	Indication (A)	I (A)			x
14	Transmit clock (A)	S (A)			x
15	Cable ID 1	ID1	x	x	
16	Receive clock (B)	RX CLK (B)		x	
17	Transmitted data (B)	TXD (B)		x	
18	Transmit clock (B)	TX CLK (B)		x	
19	Receive data (B)	RXD (B)		x	
20	Data terminal ready	DTR	x	x	
21	Remote loopback test	RLBT	x		
22	Call indicate	CI	x	x	
23	Reserved				
24	Transmit clock	TX CLK	x		
25	Test indicate	TI	x		
26	Receive clock	RX CLK	x		
27	Local loopback test	LLBT			
28	Transmitted data (B)	T (B)			x
29	Control (B)	C (B)			x
30	Received data (B)	R (B)			x
31	Indication (B)	I (B)			x
32	Transmit clock (B)	S (B)			x
33	Reserved				
34	Receive clock (A)	RX CLK (A)		x	
35	Transmitted data (A)	TXD (A)		x	
36	Transmit clock (A)	TX CLK (A)		x	
37	Received data (A)	RXD (A)		x	

C.1.2 X.25 Co-Processor Modem Attachment Pin Assignment

Supported modem types include the V.11, V.24/X.21bis and V.35/X.21bis attachments.

C.1.2.1 X.21 Pin Assignment

<i>Table 27. Pin Assignment of V.11 Type of Interface Circuits to 15-Pin Connectors</i>		
PIN NUMBER	CIRCUIT ASSIGNMENT	Circuit Description
1•	-	-
2	T(a)•	Transmit
3	C(a)	Control
4	R(a)	Receive
5	I(a)	Indication
6	S(a)	Signal element timing
7	B(a)/X(a)	Byte timing/Modem transmit signal (not used)
8	G	Signal ground or common return
9	T(b)•	Transmit
10	C(b)	Control
11	R(b)	Receive
12	I(b)	Indication
13	S(b)	Signal element timing
14	B(b)/X(b)	Byte timing/Modem transmit signal (not used)
15	Future use	Future use

Note:

- Assigned for connecting the shields between tandem sections of shielded interface cables.
- (a) and (b) indicate pins which are associated to form pairs.

G - Signal ground (or common return):

In the case of interchange circuits according to Recommendation V.11, it interconnects the zero volt reference points of a generator and a receiver to reduce environmental signal interference, if required.

T - Transmit:

The binary signals originated by the system to be transmitted during the data transfer phase via the data circuit to one or more remote systems are transferred on this circuit to the modem.

This circuit also transfers the call control signals originated by the system to be transmitted to the modem in the call establishment and other call control phases as specified by the relevant recommendations for the procedural characteristics of the interface.

The modem monitors this circuit for detection of electrical circuit fault conditions, according to the specifications of the electrical characteristics of the interface. A circuit fault is interpreted by the modem as defined in the recommendation for the procedural characteristics of the interface.

R - Receive:

The binary signals sent by the modem as received during the data transfer phase from a remote system are transferred on this circuit to the system.

This circuit also transfers the call control signals sent by the modem as received during the call establishment and other call control

phases as specified by the relevant recommendation for the procedural characteristics of the interface.

The system monitors this circuit for detection of electrical circuit fault conditions, according to the specifications of the electrical characteristics of the interface. A circuit fault is to be interpreted by the system as defined in the recommendations for the procedural characteristics of the interface.

C - Control:

Signals on this circuit control the modem for a particular signalling process.

Representation of a control signal requires additional coding of circuit T-Transmit as specified in the relevant recommendation for the procedural characteristics of the interface. During the data phase, this circuit must remain ON. During the call control phases, the condition of this circuit must be as specified in the relevant recommendation for the procedural characteristics of the interface.

The modem monitors this circuit for detection of electrical circuit fault conditions, according to the specifications of the electrical characteristics of the interface. A circuit fault is to be interpreted by the modem as defined in the recommendation for the procedural characteristics of the interface.

I - Indication:

Signals on this circuit indicate to the system the state of the call control process.

Representation of a control signal requires additional coding of circuit R-Receive, as specified in the relevant recommendation for the procedural characteristics of the interface. The ON condition of this circuit signifies that signals on circuit R contain information from the distant system. The OFF condition signifies a control signalling condition which is defined by the bit sequence on circuit R as specified by the procedural characteristics of the interface.

The system monitors this circuit for detection of electrical circuit fault conditions, according to the specifications of the electrical characteristics of the interface. A circuit fault is to be interpreted by the system as defined in the recommendation for the procedural characteristics of the interface.

S - Signal element timing:

Signals on this circuit provide the system with signal element timing information from the modem. The condition of this circuit is ON and OFF for nominally equal periods of time. However, for burst asynchronous operations, longer periods of the OFF condition may be permitted equal to an integer odd number of the nominal period of the ON condition as specified by the relevant procedural characteristics of the interface.

The system must present a binary signal on circuit T-Transmit and a condition on circuit C-Control, in which the transitions nominally occur at the time of the transitions from the OFF to ON condition of this circuit.

The modem presents a binary signal on circuit R-Receive and a condition on circuit I-Indication in which the transitions nominally occur at the time of the transitions from the OFF to ON condition of this circuit.

The modem transfers signal element timing information on this circuit across the interface whenever the timing source is capable of generating this information.

C.1.2.2 V.24/X.21bis Pin Assignment

Table 28. V.24/X.21bis Pin Assignment to 25-Pin Connector (Speeds up to 20 Kbps)

Pin	CCITT Circuit	CCITT Circuit Designation
1		
2	103	Transmitted data
3	104	Received data
4	105	Request to send
5	106	Ready for sending
6	107	Data set ready
7	102	Signal ground
8	109	Received line signal detector
9	-	N
10	-	N
11	-	N
12	-	N
13	-	F
14	-	F
15	114	Tx signal element timing from modem
16	-	F
17	115	Rx signal element timing from modem
18	141	Local loopback
19	-	F
20	108	Data terminal ready/ Connect data set to line
21	140	Local loopback
22	125	Calling indicator
23	-	N
24	-	F
25	142	Test indicator

Note:

N Pin permanently reserved for national use.

F Pin reserved for future international standard and should not be used for national use.

C.1.2.3 V.35/X.21bis Pin Assignment

Table 29. V.35/X.21bis Pin Assignment to 34-Pin Connector (Speeds above 20 kbps)

Pin	CCITT Circuit	CCITT Circuit Designation	Direction
A			
B	102	Signal ground or common return	Common
C	105	Request to send	From system
D	106	Ready for sending	To system
E	107	Data set ready	To system
F	109	Data channel received line sig. detect.	To system
H	108.2	Data terminal ready	From system
J	125	Calling indicator	To system
K	141	Local loopback	Not Used
L	-	F	-
M	-	F	-
N	140	Loopback/Maintenance test	Not Used
P	103	Transmitted data A-wire	From system
R	104	Received data A-wire	To system
S	103	Transmitted data B-wire	From system
T	104	Received data B-wire	To system
U	113	Tx sig. elem. timing from system A-wire	Not Used
V	115	Receiver signal element timing A-wire	To system
W	113	Tx sig. elem. timing from system B-wire	Not Used
X	115	Receiver signal element timing B-wire	To system
Y	114	Tx sig. elem. timing from modem A-wire	To system
Z	-	F	-
AA	114	Tx sig. elem. timing from modem B-wire	To system
BB	-	F	-
CC	-	F	-
DD	-	F	-
EE	-	F	-
FF	-	F	-
HH	-	N	-
JJ	-	N	-
KK	-	N	-
LL	-	N	-
MM	-	F	-
NN	142	Test indicator	Not Used
Note:			
N Pin permanently reserved for national use.			
F Pin reserved for future international standard and should not be used for national use.			

C.1.2.4 V.36 Pin Assignment

Table 30. V.36 Pin Assignment to 37-Pin Connector

Pin	CCITT Circuit	Signal Name	Direction
04	103	TXDnA	From system
22	103	TXDnB	From system
06	104	RXDnA	To system
24	104	RXDnB	To system
05	114	TCLKInA	To system
23	114	TCLKInb	To system
08	115	TCLKInA	TO system
26	115	TCLKInB	To system
17	113	TCLKOnA	From system
35	113	TCLKOnB	From system
07	105	RTSn	From system
09	106	CTSn	To system
13	109	DCDn	To system
12	108.2	DTRn	From system
11	107	DSRn	To system
19	102	SigGnd(-)	-
01	-	-	

C.1.3 X.25 Co-Processor Interconnection Cables

This section shows the pin assignments for the Co-Processor-to-modem interconnection cables.

C.1.3.1 X.21 Interface Cable

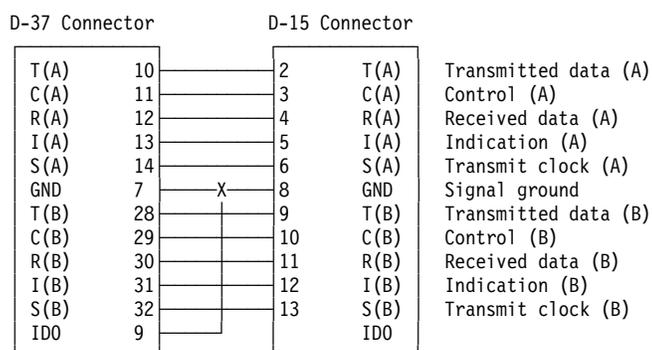


Figure 58. X.21 Interface Cable

C.1.3.2 X.21bis/V.24 Interface Cable

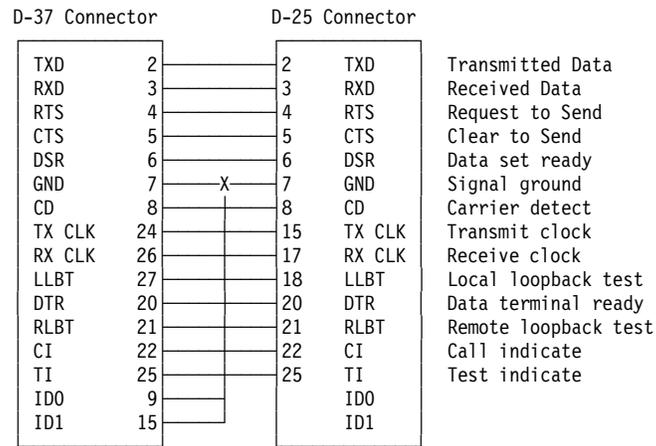


Figure 59. X.21bis/V.24 Interface Cable

C.1.3.3 X.21bis/V.35 Interface Cable

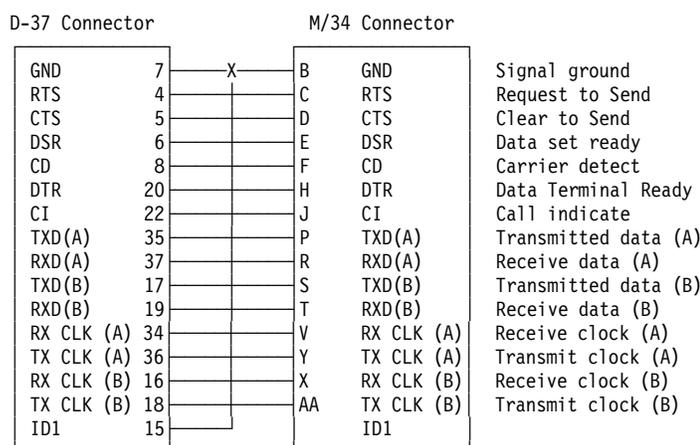


Figure 60. X.21bis/V.35 Interface Cable

C.1.4 X.25 Co-Processor Adapter and Cable Diagnostics

Adapter and cable wrap plugs, used for diagnostics, are automatically included with cables and X.25 adapter orders specifying the RISC/6000 order numbers.

The following wrap plug pin assignments, taken from the *X.25 Co-Processor/2 Hardware Technical Reference manual*, are listed below.

They are supplied for local loopback tests in accordance with CCITT Recommendation X.150:

- D-37 wrap plug
- D-15 wrap plug
- D-25 wrap plug
- M/34 wrap plug

C.1.4.1 D-37 Wrap Plug - Co-Processor

The D-37 wrap plug is used to test local loopback at the D-37 connector on the adapter. It has a cable identifier of ID0=1, ID1=1.

The pin assignments are as follows:

Signal	Pin no.		Pin no.	Signal
T(B)	28	-	30	R(B)
T(A)	10	-	12	R(A)
C(B)	29	-	31	I(B)
C(A)	11	-	13	I(A)
TXD	2	-	3	RXD
RTS	4	-	5	CTS
DTR	20	-	6	DSR
LLBT	27	-	25	TI
RLBT	21	-	22	CI
TXD(A)	35	-	37	RXD(A)
TXD(B)	17	-	19	RXD(B)
ID0	9	-	15	ID1

Figure 61. D-37 Wrap Plug

C.1.4.2 D-15 Wrap Plug - Co-Processor, X.21

The D-15 wrap plug is used to test loopback at the DCE end of the X.21 interface cable.

The pin assignments are as follows:

Signal	Pin no.	-	Pin no.	Signal
T(B)	9	-	11	R(B)
T(A)	2	-	4	R(A)
C(B)	10	-	12	I(B)
C(A)	3	-	5	I(A)

Figure 62. D-15 Wrap Plug

C.1.4.3 D-25 Wrap - Plug Co-Processor, V.24

The D-25 wrap plug is used to test loopback at the DCE end of the X.21bis/V.24 interface cable.

The pin assignments are as follows:

Signal	Pin no.	-	Pin no.	Signal
TXD	2	-	3	RXD
RTS	4	-	5	CTS
DTR	20	-	6	DSR
LLBT	18	-	25	TI
RLBT	21	-	22	CI

Figure 63. D-25 Wrap Plug

C.1.4.4 M/34 Wrap Plug - Co-Processor, V.35

The M/34 wrap plug is used to test loopback at the DCE end of the X.21bis/V.35 interface cable.

The pin assignments are as follows:

Signal	Pin no.	-	Pin no.	Signal
TXD(A)	P	-	R	RXD(A)
TXD(B)	S	-	T	RXD(B)
RTS	C	-	D	CTS
DTR	H	-	E	DSR

Figure 64. M/34 Wrap Plug

C.2 IBM Portmaster Adapter/2 and Multiport Model 2

The cables and connectors described in this section are for use with the IBM Portmaster Adapter/2 and Multiport Model 2. The cables and connectors are interchangeable between the adapters and have the same pin assignments.

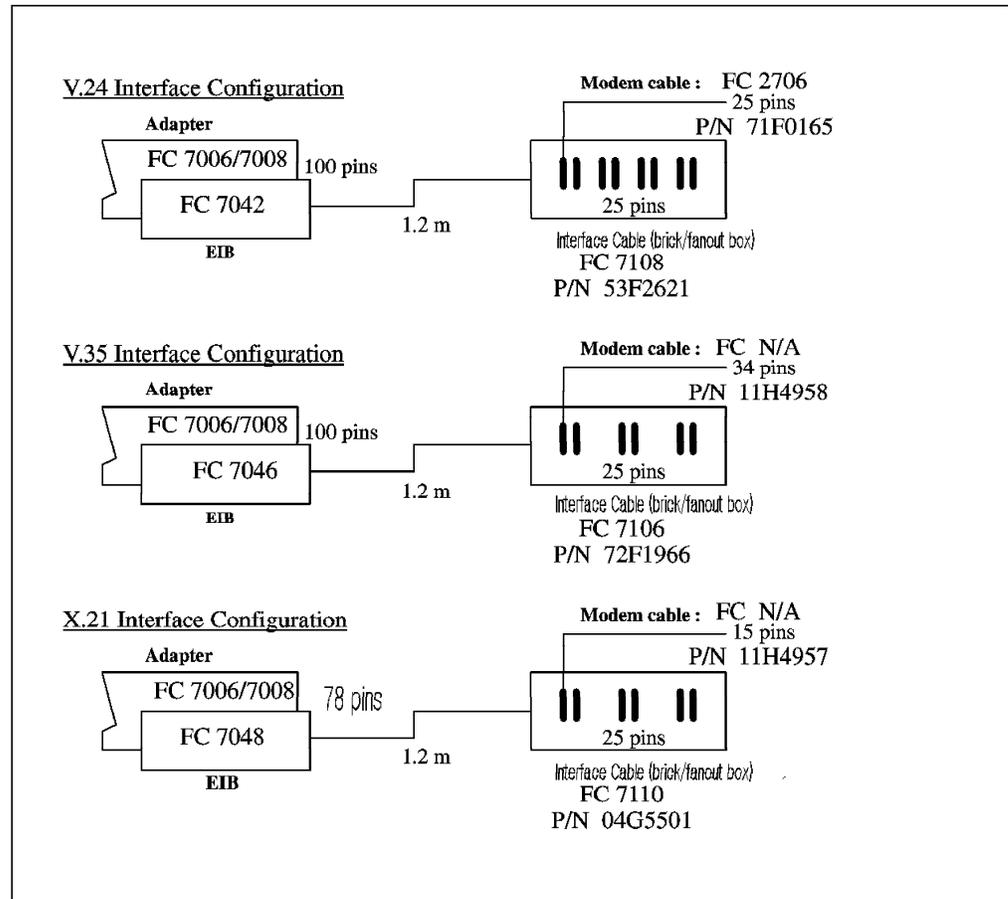


Figure 65. Portmaster and Multiport

C.2.1 IBM ARTIC Portmaster Adapter/A Pin-Out Information

The following tables describe the pin assignments of the interface cables:

C.2.1.1 V.35 Pin Assignment

Table 31 (Page 1 of 2). Portmaster and Multiport: V.35

Mnemonic	I/O	Port 0	Port 1	Port 2	Port 3	Port 4	Port 5	25-Position Connector
TxDA	O	94	21	47	71	72	23	02
TxDB	O	70	46	22	95	96	48	14
RxDA	I	08	54	58	29	28	57	03
RxDB	I	33	78	82	04	03	81	16
TxCA IN	I	76	06	77	56	27	55	15
TxCB IN	I	52	31	53	80	02	79	12

Table 31 (Page 2 of 2). Portmaster and Multiport: V.35

Mnemonic	I/O	Port 0	Port 1	Port 2	Port 3	Port 4	Port 5	25-Position Connector	
RxCA	I	20	41	38	19	32	30	17	
RxCB	I	45	16	13	44	07	05	09	
TxCA OUT	O	24	73	98	25	99	26	24	
TxCB OUT	O	49	97	74	50	75	51	11	
RTS	O	42	43	92	93	37	39	04	
CTS	I	15	65	86	87	59	09	05	
DCD	I	89	40	62	61	35	84	08	
DTR	O	18	91	69	68	14	12	20	
DSR	I	66	90	88	64	60	85	06	
SGND	-	34	17	63	67	01	83	07	
FGND	-	100	shield						01

TxDA, TxDB Transmit Data

RxDA, RxDB Receive Data

TxCA IN, TxCB IN Transmit Clock (DCE clock)

RxCA, RxCB Receive Clock (DCE clock)

TxCA OUT, TxCB OUT Transmit Clock (DTE clock)

C.2.1.2 V.24 Pin Assignment

Mnemonic	I/O	Port 0	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6	Port 7	25-Position Connector
TxD	O	51	54	07	10	13	16	94	48	02/BA
RxD	I	02	05	83	86	89	92	46	74	03/BB
RTS	O	01	04	82	85	88	91	45	73	04/CA
CTS	I	77	80	34	37	40	43	71	24	05/CB
DCD	I	28	31	59	62	65	68	21	99	08/CF
DTR	O	76	79	33	36	39	42	70	23	20/CD
DSR	I	53	56	09	12	15	18	96	50	06/CC
HRS	I	27	30	58	61	64	67	20	98	23/CI
RI	I	03	06	84	87	90	93	47	75	22/CE
TxCLKIN	I	29	32	60	63	66	69	22	100	15/DB
TxCLK	O	52	55	08	11	14	17	95	49	24/DA
RxCLK	I	78	81	35	38	41	44	72	25	17/DD
SGND	-	19	19	26	26	57	57	97	97	07/AB
FGND	Cable Shield									01/AA

Note: x stands for port number

- TxD** Transmit Data
- RxD** Receive Data
- RTS** Request To Send
- CTS** Clear To Send
- DCD** Data Carrier Detect
- DTR** Data Terminal Ready
- DSR** Data Set Ready
- HRS** Half Rate Select
- RI** Ring Indicate
- TxCLK IN** Transmit Clock IN
- TxCLK** Transmit Clock
- RxCLK** Receive Clock
- SGND** Signal Ground
- FGND** Frame Ground

C.2.1.3 X.21 Pin Assignment

Table 33. Portmaster and Multiport: X.21

Mnemonic	I/O	Port 0	Port 1	Port 2	Port 3	Port 4	Port 5	25-Position Connector
TxA	O	40	04	66	69	73	55	02
TxB	O	41	05	19	20	10	13	24
RxA	I	02	64	28	31	54	75	03
RxB	I	62	26	57	77	18	53	17
CxA	O	01	63	27	30	34	16	04
CxB	O	60	24	47	50	35	17	20
IxA	I	61	25	48	51	15	36	05
IxB	I	42	06	68	71	72	33	06
XxA	O	22	45	09	12	74	56	08
XxB	O	03	65	29	32	49	52	22
SxA	I	23	46	78	59	39	14	15
SxB	I	21	44	76	37	38	58	23
Ground	-	43	07	08	67	11	70	07

Note: x stands for port number

TxA, TxB Transmit Data

RxA, RxB Receive Data

CxA, CxB Control

IxA, IxB Indication

XxA, XxB Transmit Clock

SxA, SxB Receive Clock

C.2.2 Modem Attachment Pin Assignment

For modem pin assignments, see C.1.2, "X.25 Co-Processor Modem Attachment Pin Assignment" on page 249.

C.2.3 X.25 Portmaster and Multiport Interconnection Cables

This sections shows the pin assignments for the Portmaster and Multiport to modem interconnection cables.

C.2.3.1 X.21 Interface Cable - 6 Port Portmaster and Multiport

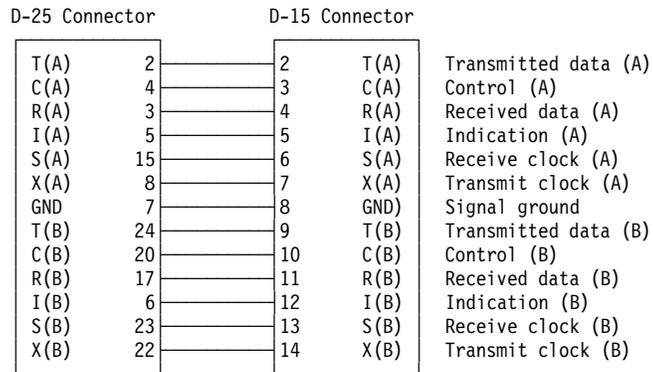


Figure 66. X.21 Interface Cable

C.2.3.2 X.21bis/V.24 Interface Cable - 8 Port Portmaster and Multiport

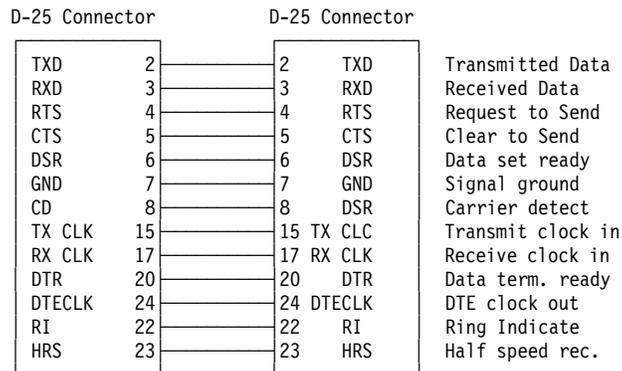


Figure 67. X.21bis/V.24 Interface Cable

C.2.3.3 V.35 Interface Cable - 6 Port Portmaster and Multiport

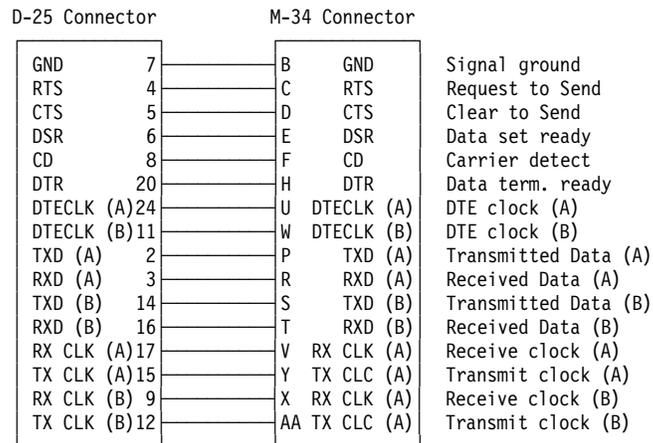


Figure 68. X.21bis/V.35 Interface Cable

C.2.4 X.25 Portmaster, Multiport and ARTIC960 Diagnostic Wrap Plugs

This sections documents the pin assignments for the Portmaster, Multiport and ARTIC960 adapter wrap plugs.

C.2.4.1 D-15 Wrap Plug: X.21

The D-15 wrap plug is used to test loopback at the DCE end of the X.21 interface cable.

The pin assignments are as follows:

Signal	Pin no.	-	Pin no.	Signal
T(B)	9	-	11	R(B)
T(A)	2	-	4	R(A)
C(B)	10	-	12	I(B)
C(A)	3	-	5	I(A)
X(A)	7	-	6	S(A)
X(B)	14	-	13	S(B)

Figure 69. D-15 Wrap Plug

C.2.4.2 D-25 Wrap Plug: EIA 232 V.24

The D-25 wrap plug is used to test loopback at the DCE end of the X.21bis/V.24 interface cable.

The pin assignments are as follows:

Signal	Pin no.		Pin no.	Signal
TXD	2	-	3	RXD
RTS	4 - 8	-	15 - 5	CTS
DTR	20 - 22	-	23 - 6	DSR
DTECLK	24	-	17	RXCLK
RLBT	21	-	22	CI

Figure 70. D-25 Wrap Plug

PIN 8 DCD

Pin 15 TXCLKIN

Pin 22 RI

Pin 23 HRS

C.2.4.3 M/34 Wrap Plug: V.35

The M/34 wrap plug is used to test loopback at the DCE end of the X.21bis/V.35 interface cable.

The pin assignments are as follows:

Signal	Pin		Pin	Signal	
TXD(A)	P	-	R	RXD(A)	
TXD(B)	S	-	T	RXD(B)	
RTS	C	-	D	CTS	
				F	DCD
Signal	Pin		Pin	Signal	
DTR	H	-	E	DSR	
DTECLK(A)	U	-	V	RXCLK(A)	
DTECLK(B)	W	-	X	RXCLK(B)	

Figure 71. M/34 Wrap Plug

C.2.4.4 D-25 Wrap Plug: X.21

The pin assignments are as follows:

Signal	Pin no.		Pin no.	Signal
T(B)	2	-	3	R(B)
T(A)	24	-	17	R(A)
C(B)	4	-	5	I(B)
C(A)	20	-	6	I(A)
X(A)	8	-	15	S(A)
X(B)	22	-	23	S(B)

Figure 72. D-25 Wrap Plug

C.2.4.5 D-25 Wrap Plug: V.35

The pin assignments are as follows:

Signal	Pin	Pin	Signal	Pin	Signal		
TXD(A)	2	-	3	RXD(A)	-	15	TXCLK(A)
TXD(B)	14	-	16	RXD(B)	-	12	TXCLK(B)
RTS	4	-	5	CTS	-	8	DCD
Signal	Pin	Pin	Signal	Pin	Signal		
DTR	20	-	6	DSR	6		
DTECLK(A)	24	-	17	RXCLK(A)			
DTECLK(B)	11	-	9	RXCLK(B)			

Figure 73. D-25 Wrap Plug

C.2.4.6 D-37 Wrap Plug - 6 ARTIC960: V.36

The pin assignments are as follows:

Signal	Pin	Pin	Signal	Pin	Signal		
TXD(A)	4	-	6	RXD(A)	-	5	TCLK(A)
TXD(B)	22	-	24	RXD(B)	-	23	TCLK(B)
RTS	7	-	9	CTS	-	13	DCD
Signal	Pin	Pin	Signal	Pin	Signal		
DTR	12	-	11	DSR			
TCLK(A)	17	-	8	RCLK(A)			
TCLK(B)	35	-	26	RCLK(B)			

Figure 74. D-37 Wrap Plug

C.2.4.7 D-78 Wrap Plug - 6 Port X.21

The pin assignments are as follows:

Board-Wrapped Connections for Ports 0 - 5		
PORT	Connector K9B Pins	Signals wrapped
0	40 → 02	TO(A) → RO(A)
	41 → 62	TO(B) → RO(B)
	01 → 61	CO(A) → IO(A)
	60 → 42	CO(B) → IO(B)
	22 → 23	XO(A) → SO(A)
	03 → 21	XO(B) → SO(B)
1	04 → 64	T1(A) → R1(A)
	05 → 26	T1(B) → R1(B)
	63 → 25	C1(A) → I1(A)
	24 → 06	C1(B) → I1(B)
	45 → 46	X1(A) → S1(A)
	65 → 44	X1(B) → S1(B)
2	66 → 28	T2(A) → R2(A)
	19 → 57	T2(B) → R2(B)
	27 → 48	C2(A) → I2(A)
	47 → 68	C2(B) → I2(B)
	09 → 78	S2(A) → S2(A)
	29 → 76	X2(B) → S2(B)
3	69 → 31	T3(A) → R3(A)
	20 → 77	T3(B) → R3(B)
	30 → 51	C3(A) → I3(A)
	50 → 71	C3(B) → I3(B)
	12 → 59	S3(A) → S3(A)
	32 → 37	X3(B) → S3(B)
4	73 → 54	T4(A) → R4(A)
	10 → 18	T4(B) → R4(B)
	34 → 15	C4(A) → I4(A)
	35 → 72	C4(B) → I4(B)
	74 → 39	S24A) → S4(A)
	49 → 38	X4(B) → S4(B)
5	55 → 75	T25A) → R5(A)
	13 → 53	T5(B) → R5(B)
	16 → 36	C5(A) → I5(A)
	17 → 33	C5(B) → I5(B)
	56 → 14	S5(A) → S5(A)
	52 → 58	X5(B) → S5(B)

Figure 75. D-78 X.21 Wrap Plug

C.2.4.8 D-100 Wrap Plug - 8 Port X.21

The pin assignments are as follows:

PORT	Connector K9B Pins	Signal Wrapped
0	01 - 52	T0(A) - R0(A)
	27 - 76	T0(B) - R0(B)
	03 - 54	TCLK0(A) - RCLK0(A)
	29 - 78	TCLK0(B) - RCLK0(B)
	02 - 53	C0(A) - I0(A)
	28 - 77	C0(B) - I0(B)
1	04 - 55	T1(A) - R1(A)
	30 - 79	T1(B) - R1(B)
	07 - 57	TCLK1(A) - RCLK1(A)
	32 - 81	TCLK1(B) - RCLK1(B)
	05 - 56	C1(A) - I1(A)
	31 - 80	C1(B) - I1(B)
2	08 - 58	T2(A) - R2(A)
	33 - 82	T2(B) - R2(B)
	10 - 60	TCLK2(A) - RCLK2(A)
	35 - 84	TCLK2(B) - RCLK2(B)
	09 - 59	C2(A) - I2(A)
	34 - 83	C2(B) - I2(B)
3	11 - 61	T3(A) - R3(A)
	36 - 85	T3(B) - R3(B)
	13 - 63	TCLK3(A) - RCLK3(A)
	39 - 87	TCLK3(B) - RCLK3(B)
	12 - 62	C3(A) - I3(A)
	37 - 86	C3(B) - I3(B)
4	14 - 64	T4(A) - T4(A)
	40 - 88	T4(B) - R4(B)
	16 - 66	TCLK4(A) - RCLK4(A)
	42 - 90	TCLK4(B) - RCLK4(B)
	15 - 65	C4(A) - I4(A)
	41 - 89	C4(B) - I4(B)
5	17 - 67	T5(A) - R5(A)
	43 - 91	T5(B) - R5(B)
	20 - 69	TCLK5(A) - RCLK5(A)
	45 - 93	TCLK5(B) - RCLK5(B)
	18 - 68	C5(A) - I5(A)
	19 - 92	C5(B) - I5(B)
6	21 - 70	T6(A) - R6(A)
	46 - 94	T6(B) - R6(B)
	23 - 72	TCLK6(A) - RCLK6(A)
	48 - 96	TCLK6(B) - RCLK6(B)
	22 - 71	C6(A) - I6(A)
	47 - 95	C6(B) - I6(B)
7	24 - 73	T7(A) - R7(A)
	49 - 97	T7(B) - R7(B)
	26 - 99	TCLK7(A) - RCLK7(A)
	51 - 100	TCLK7(B) - RCLK7(B)
	50 - 74	C7(A) - I7(A)
	75 - 98	C7(B) - I7(B)

Figure 76. D-100 Wrap Plug

C.2.4.9 D-100 Wrap Plug - 8 Port V.24

The pin assignments are as follows:

PORT	Connector K9B Pins	Signal Wrapped
0	51 - 02 01 - 77 - 28 - 29 52 - 78 76 - 53 - 03 - 27	TXD0 - RXDA0 TRS0 - CTS0 - DCD0 - TXCLKIN0 TXCLK0 - RXCLK0 DTR0 - DSR0 - RI0 - HRS0
1	54 - 05 04 - 80 - 31 - 32 55 - 81 79 - 56 - 06 - 30	TXD1 - RXDA1 TRS1 - CTS1 - DCD1 - TXCLKIN1 TXCLK1 - RXCLK1 DTR1 - DSR1 - RI1 - HRS1
2	07 - 83 82 - 34 - 59 - 60 08 - 35 33 - 09 - 84 - 58	TXD2 - RXDA2 TRS2 - CTS2 - DCD2 - TXCLKIN2 TXCLK2 - RXCLK2 DTR2 - DSR2 - RI2 - HRS2
3	10 - 86 85 - 37 - 62 - 63 11 - 38 36 - 12 - 87 - 61	TXD3 - RXDA3 TRS3 - CTS3 - DCD3 - TXCLKIN3 TXCLK3 - RXCLK3 DTR3 - DSR3 - RI3 - HRS3
4	13 - 89 88 - 40 - 65 - 66 14 - 41 39 - 15 - 90 - 64	TXD4 - RXDA4 TRS4 - CTS4 - DCD4 - TXCLKIN4 TXCLK0 - RXCLK0 DTR4 - DSR4 - RI4 - HRS4
5	16 - 92 91 - 43 - 68 - 69 17 - 44 42 - 18 - 93 - 97	TXD5 - RXDA5 TRS5 - CTS5 - DCD5 - TXCLKIN5 TXCLK5 - RXCLK5 DTR5 - DSR5 - RI5 - HRS5
6	94 - 46 45 - 71 - 21 - 22 95 - 72 70 - 96 - 47 - 20	TXD6 - RXDA6 TRS6 - CTS6 - DCD6 - TXCLKIN6 TXCLK6 - RXCLK6 DTR6 - DSR6 - RI6 - HRS6
7	48 - 74 73 - 24 - 99 - 100 49 - 25 23 - 50 - 75 - 98	TXD7 - RXDA7 TRS7 - CTS7 - DCD7 - TXCLKIN7 TXCLK7 - RXCLK7 DTR7 - DSR7 - RI7 - HRS7

Figure 77. D-100 V.24 Wrap Plug

C.2.4.10 D-100 Wrap Plug - 6 Port V.35 and 6 Port V.36

The pin assignments are as follows:

Board-Wrapped Connections for Ports 0 - 5		
PORT	Connector K9B Pins	Signals wrapped
0	94 - 08 - 76 70 - 33 - 52	TXDA0 → RXDA0 → TXCA0 IN TXDB0 → RXDB0 → TXCB0 IN
	24 - 20 49 - 45	TXCA0 OUT → RXCA0 TXCB0 OUT → RXCB0
	42 - 15 - 89	RTS0 → CTS0 → DCD0
	18 - 66	DTR0 → DSR0
1	21 - 54 - 06 46 - 78 - 31	TXDA1 → RXDA1 → TXCA1 IN TXDB1 → RXDB1 → TXCB1 IN
	73 - 41 97 - 16	TXCA1 OUT → RXCA1 TXCB1 OUT → RXCB1
	43 - 65 - 40	RTS1 → CTS1 → DCD1
	91 - 90	DTR1 → DSR1
2	47 - 58 - 77 22 - 82 - 53	TXDA 2 → RXDA2 → TXCA2 IN TXDB 2 → RXD20 → TXCB2 IN
	98 - 38 74 - 13	TXCA2 OUT → RXC20 TXC20 OUT → RXCB2
	92 - 86 - 62	RTS2 → CTS2 → DCD2
	69 - 88	DTR2 → DSR2
3	71 - 29 - 56 95 - 04 - 80	TXDA 3 → RXDA3 → TXCA3 IN TXDB 3 → RXDB3 → TXCB3 IN
	25 - 19 50 - 44	TXCA3 OUT → RXCA3 TXCB3 OUT → RXCB3
	93 - 87 - 61	RTS3 → CTS3 → DCD3
	68 - 64	DTR3 → DSR3
4	72 - 28 - 27 96 - 03 - 02	TXDA 4 → RXDA4 → TXCA4 IN TXDB 4 → RXDB4 → TXCB4 IN
	99 - 32 75 - 07	TXCA4 OUT → RXCA4 TXCB4 OUT → RXCB4
	37 - 59 - 35	RTS4 → CTS4 → DCD4
	14 - 60	DTR4 → DSR4
5	23 - 57 - 55 48 - 81 - 79	TXDA 5 → RXDA5 → TXCA5 IN TXDB 5 → RXDB5 → TXCB5 IN
	26 - 30 51 - 05	TXCA5 OUT → RXCA5 TXCB5 OUT → RXCB5
	39 - 09 - 84	RTS5 → CTS5 → DC5
	12 - 85	DTR5 → DSR5

Figure 78. D-100 V.36/V.35 Wrap Plug

Appendix D. CCITT Causes and Diagnostics

D.1 CCITT Clear and Reset Causes

The CCITT meanings for the clear cause codes are:

Table 34. List of CCITT Clear Causes

Hex	Dec	Meaning
00	00	Originated by the remote X.25 data terminal equipment (DTE)
01	01	Number busy
03	03	Incorrect facility request
05	05	Network congestion
09	09	Out of order
0B	11	Access barred
0D	13	Not obtainable
11	17	Remote procedure error
13	19	Local procedure error
15	21	RPOA out of order
19	25	Reverse charging acceptance not subscribed
21	33	Incompatible destination
29	41	Fast select acceptance not subscribed
80	128	80 through FF Not defined by CCITT, but used by SNA

The CCITT meanings for the reset cause codes are:

Table 35. List of CCITT Reset Causes

Hex	Dec	Meaning
00	00	Originated by the remote X.25 data terminal equipment (DTE)
01	01	Out of order
03	03	Remote procedure error
05	05	Local procedure error
07	07	Network congestion
09	09	Remote DTE operational
0F	15	Network operational
11	17	Incompatible destination
1D	29	Network out of order
80	128	Originated through FF DTE

D.2 CCITT Diagnostic Codes

Table 36 (Page 1 of 2). List of CCITT Diagnostic Codes

Hex	Dec	Meaning
00	0	Clear or reset generated during restart
01	1	Invalid P(S) in packet from DCE
02	2	Invalid P(R) in packet from DCE
10	16	Invalid packet type
11	17	Invalid packet from DCE in state r1
12	18	Invalid packet from DCE in state r2
13	19	Invalid packet from DCE in state r3
14	20	Invalid packet from DCE in state p1
15	21	Invalid packet from DCE in state p2
16	22	Invalid packet from DCE in state p3

Table 36 (Page 2 of 2). List of CCITT Diagnostic Codes

Hex	Dec	Meaning
17	23	Invalid packet from DCE in state p4
18	24	Invalid packet from DCE in state p5
19	25	Invalid packet from DCE in state p6
1A	26	Invalid packet from DCE in state p7
1B	27	Invalid packet from DCE in state d1
1C	28	Invalid packet from DCE in state d2
1D	29	Invalid packet from DCE in state d3
20	32	Packet not allowed
21	33	Unidentifiable packet received from DCE
22	34	Incoming call received on one-way channel
23	35	Clear or call packet received from DCE on a permanent virtual circuit (PVC)
24	36	Packet received on an unassigned logical channel
25	37	REJECT not subscribed to
26	38	Packet received from DCE was too short
27	39	Packet received from DCE was too long
28	40	Invalid general format identifier (GFI)
29	41	Restart packet received from DCE with non-zero logical channel identifier
2A	42	Invalid fast-select packet received from DCE
2B	43	Unauthorized interrupt confirmation
2C	44	Interrupt packet received from DCE when acknowledgment was still outstanding
2D	45	Unauthorized reject
30	48	Timer expired (or limit surpassed)
31	49	Time-out or retries reached on call response from DCE
32	50	Time-out or retries reached on clear response from DCE
33	51	Time-out or retries reached on reset response from DCE
34	52	Time-out or retries reached on restart response from DCE
35	53	Time expired for call deflection
40	64	Call setup clearing or registration problem
41	65	Facility/Registration code not allowed
42	66	Invalid facility parameter
43	67	Invalid called address
44	68	Address missing in incoming call from DCE
45	69	Invalid facility/registration length field
46	70	Incoming call barred
47	71	No logical channel available
48	72	Call collision
49	73	Duplicate facility requested
4A	74	Non-zero address length in fast-select clear from DCE
4B	75	Non-zero facility length in fast-select clear from DCE
4C	76	Facility not provided when expected
4D	77	Invalid CCITT-specified DTE facility
4E	78	Minimum number of call redirections or call deflections exceeded
50	80	Miscellaneous
51	81	Improper cause code from DTE
52	82	Nonoctet aligned
53	83	Inconsistent Q-bit settings
54	84	NUI problem
70	112	International problem
71	113	Remote network problem
72	114	International protocol problem
73	115	International link out of order
74	116	International link busy
75	117	Transit network facility problem
76	118	Remote network facility problem
77	119	International routing problem
78	120	Temporary routing problem
79	121	Unknown called DNIC
7A	122	Maintenance action (may also apply within a national network)
80	128	Reserved for DTE-defined diagnostic information

D.3 Product-Specific Diagnostic Codes

Table 37. List of Product-Specific Diagnostic Codes

Hex	Dec	Meaning
81	129	No listener for incoming call
82	130	No available LCN for call
83	131	User error
84	132	User call rejection
85	133	Call cleared before accept
86	134	Invalid call reference
87	135	Registration timer expired
88	136	Invalid link layer state

D.4 ISO 8208 Diagnostic Codes

Table 38. List of ISO 8208 Diagnostic Codes

Hex	Dec	Meaning
90	144	Timer expired or retransmission count surpassed
91	145	Timer expired or retransmission count surpassed on interrupt-confirm from DCE
92	146	T25 timer expired, for data packet transmission
93	147	Timer expired or retransmission count surpassed for reject
A0	160	DTE-specific signals
A1	161	DTE operational
A2	162	DTE not operational (level 2) or no application listening (network)
A3	163	DTE resource constraint
A4	164	Fast select not subscribed
A5	165	Invalid partially full data packet received from DCE
A6	166	D-bit procedure not supported
A7	167	Registration or cancellation confirmed
E0	224	OSI Network Service Problem
E1	225	Disconnection (transient condition)
E2	226	Disconnection (permanent condition)
E3	227	Connection rejection -- reason unspecified (transient condition)
E4	228	Connection rejection -- reason unspecified (permanent condition)
E5	229	Connection rejection -- quality of service not available (transient condition)
E6	230	Connection rejection -- quality of service not available (permanent condition)
E7	231	Connection rejection -- NSAP unreachable (transient condition)
E8	232	Connection rejection -- NSAP unreachable (permanent condition)
E9	233	Reset -- reason unspecified.
EA	234	Reset -- congestion
EB	235	Connection rejection -- NSAP address unknown (permanent condition)

D.5 SNA Diagnostic Codes

Table 39 (Page 1 of 3). List of SNA Diagnostic Codes

Hex	Dec	Meaning
00	0	Normal initialization or termination
0C	12	Invalid LLC type
10	16	Invalid packet type (general)
11	17	Invalid packet type for state r1
12	18	Invalid packet type for state r2
13	19	Invalid packet type for state r3

Table 39 (Page 2 of 3). List of SNA Diagnostic Codes

Hex	Dec	Meaning
14	20	Invalid packet type for state p1
15	21	Invalid packet type for state p2
16	22	Invalid packet type for state p3
17	23	Invalid packet type for state p4
18	24	Invalid packet type for state p5
19	25	Invalid packet type for state p6
1A	26	Invalid packet type for state p7
1B	27	Invalid packet type for state d1
1C	28	Invalid packet type for state d2
1D	29	Invalid packet type for state d3
20	32	DCE timer expired (general)
21	33	DCE timer expired: incoming call
22	34	DCE timer expired: clear indication
23	35	DCE timer expired: reset indication
24	36	DCE timer expired: restart indication
30	48	DTE timer expired: (general)
31	49	DTE timer expired: call request
32	50	DTE timer expired: clear request
33	51	DTE timer expired: reset request
34	52	DTE timer expired: restart request
40	64	Unassigned (general)
50	80	QLLC error: (general)
51	81	QLLC error: undefined C-field
52	82	QLLC error: unexpected C-field
53	83	QLLC error: missing I-field
54	84	QLLC error: undefined I-field
55	85	QLLC error: I-field too long
56	86	QLLC error: Frame reject received
57	87	QLLC error: Header invalid
58	88	QLLC error: Data received in wrong state
59	89	QLLC error: Time-out condition
5A	90	QLLC error: Nr invalid
5B	91	QLLC error: Recovery rejected or terminated
5D	93	QLLC error: ELLC time-out condition
60	96	PSH error (general)
61	97	PSH error: sequence error
62	98	PSH error: header too short
63	99	PSH error: PSH format invalid
64	100	PSH error: command undefined
65	101	PSH error: protocol invalid
66	102	PSH error: data received in wrong state
69	105	PAD error: time-out condition
70	112	PAD error: (general)
71	113	PAD error: PAD access facility failure
72	114	PAD error: SDLC FCS error
73	115	PAD error: SDLC time-out
74	116	PAD error: SDLC frame invalid
75	117	PAD error: I-field too long
76	118	PAD error: SDLC sequence error
77	119	PAD error: SDLC frame aborted
78	120	PAD error: SDLC FRMR received
79	121	PAD error: SDLC response invalid
7B	123	PAD error: invalid packet type
7F	127	PAD error: PAD inoperable
80	128	DTE-specific (general)
81	129	DTE-specific: 8100_DPPX-specific
82	130	DTE-specific: INN_QLLC-specific
83	131	DTE-specific: INN_QLLC-specific
84	132	DTE-specific: INN_QLLC-specific

Table 39 (Page 3 of 3). List of SNA Diagnostic Codes

Hex	Dec	Meaning
85	133	DTE-specific: INN_QLLC-specific
86	134	DTE-specific: INN_QLLC-specific
87	135	DTE-specific: INN_QLLC-specific
88	136	DTE-specific: INN_QLLC-specific
89	137	DTE-specific: INN_QLLC-specific
8A	138	DTE-specific: INN_QLLC-specific
8B	139	DTE-specific: INN_QLLC-specific
8C	140	DTE-specific: INN_QLLC-specific
8D	141	DTE-specific: INN_QLLC-specific
8E	142	DTE-specific: INN_QLLC-specific
8F	143	DTE-specific: INN_QLLC-specific
90	144	Network-specific
91	145	Network-specific: DDX-P RNR packet received
A0	160	Packet not allowed (general)
A1	161	Packet not allowed: invalid M-bit packet sequence
A2	162	Packet not allowed: invalid packet type received
A3	163	Packet not allowed: invalid packet on PVC
A4	164	Packet not allowed: unassigned LC
A5	165	Packet not allowed: diagnostic packet received
A6	166	Packet not allowed: packet too short
A7	167	Packet not allowed: packet too long
A8	168	Packet not allowed: invalid GFI
A9	169	Packet not allowed: not identifiable
AA	170	Packet not allowed: not supported
AB	171	Packet not allowed: invalid Ps
AC	172	Packet not allowed: invalid Pr
AD	173	Packet not allowed: invalid D-bit received
AE	174	Packet not allowed: invalid Q-bit received
B0	176	DTE-specific: (NPSI gate/date) (general)
B1	177	DTE-specific: no LU-to-LU session
C0	192	DTE-specific: (general)
C1	193	DTE-specific: termination pending
C2	194	DTE-specific: channel inoperative
C3	195	DTE-specific: unauthorized interrupt confirmation
C4	196	DTE-specific: unauthorized interrupt request
C5	197	DTE-specific: PU (PVC) not available
C6	198	DTE-specific: inactivity time-out
C7	199	DTE-specific: incompatible line configuration
D0	208	Resources: (general)
D1	209	Resources: buffers depleted
D2	210	Resources: PIU too long
E0	224	Local procedure error (general)
E1	225	Local procedure error: packet with LC=0 not received
E2	226	Local procedure error: restart or diagnostic packet on LCI = x'000'
E3	227	Local procedure error: incoming call received on wrong LC
E4	228	Local procedure error: facility not subscribed to
E5	229	Local procedure error: packet not restart or diagnostic on LCI = x'000'
E6	230	Local procedure error: facility parameters not supported
E7	231	Local procedure error: facility not supported
E8	232	Local procedure error: unexpected calling DTE
E9	233	Local procedure error: invalid D-bit request
EA	234	Local procedure error: reset indication on virtual call
EB	235	Local procedure error: invalid protocol identifier
EC	236	Local procedure error: connection identifier mismatch
ED	237	Local procedure error: missing cause or diagnostic code

D.5.1 List of Diagnostic Codes Used by xtalk

The following diagnostic codes are set up by the IBM-supplied application xtalk when clearing connections:

Table 40. List of xtalk Diagnostic Codes

Hex	Dec	Meaning
F1	241	Normal disconnection
F4	244	Connection request rejected. This may be because the program is busy (already connected to someone else), or because the other program is not listening.

D.5.2 Logical Channel States

This is the list of the CCITT logical channel states referred to in D.2, "CCITT Diagnostic Codes" on page 269 and D.5, "SNA Diagnostic Codes" on page 271.

State	Meaning
d1	Flow control ready
d2	DTE reset request
d3	DCE reset indication
p1	Channel ready
p2	DTE call request
p3	DCE incoming call
p4	Data transfer
p5	Call collision
p6	DTE clear request
p7	DCE clear indication
r1	Packet level ready
r2	DTE restart request
r3	DCE restart indication

Appendix E. Facilities

Facilities allow the subscriber greater control of the X.25 environment. X.25 applications use facilities to take advantage of options in the network subscription.

E.1 Supported Facilities for X.25 Communications

Several types of facilities may be requested in a call packet. Standard X.25 facilities are the most common, but you may also find non-X.25 facilities specific to your network or CCITT-defined facilities to be used with the OSI network services. Check with your network provider to see which facilities are available.

E.1.1 Defining Facilities

See 7.3, "Requesting the Use of a Facility with TCP/IP" on page 145 and 8.3.6.4, "Defining X.25 Optional Facilities Profile" on page 168 to see how to define facilities for TCP/IP and SNA interfaces.

E.1.2 Facilities Format

Non-standard facilities are preceded by a *facility marker*: 0x0000, 0x000FF or 0x000F. They are laid out as follows:

X.25 facilities
0x0
0x0
non-X.25 facilities provided by the local network
0x0
0xFF
non-X.25 facilities provided by the remote network
0x0
0x0F
CCITT-specified DTE facilities

Figure 79. Facilities Format

If any section is not required, both it and the preceding facility marker can be left out.

Within each section, the facilities format is defined as a series of facility codes, followed by a number of bytes of arguments.

The number of bytes of arguments is defined by the first two bits of the facility code:

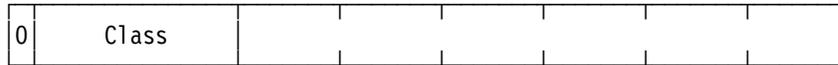


Figure 80. Facility Code Format

The class can have one of the following values:

- 00** Class A. This has a single byte parameter field
- 01** Class B. This has two bytes as a parameter
- 10** Class C. This has three bytes as a parameter
- 11** Class D. The next byte defines how long the parameter is

There is one special facility code, 0xFF, which is reserved for extension of the facility codes. The octet following this one indicates an extended facility code having the format A, B, C or D. Repetition of the facility code 0xFF is permitted resulting in additional extensions.

E.1.3 Index of the Facilities

Use the following index to find the paragraph describing the facility you are looking for.

<i>Table 41. Index of Facilities</i>			
Value	Function	Param. length	Reference
01	Fast Select / Reverse charging	1	E.2.7
02	Throughput class required	1	E.2.3
03	Closed User Group selection required	1	E.2.4
04	Charging Information - Requesting Service	1	E.2.9
08	Called line address modified notification	1	E.2.14
09	Closed User Group with outgoing access selection	1	E.2.5
0A	Quality of Service Negotiation - Minimum throughput class	1	E.3.3
0B	Expedited Data Negotiation	1	E.3.5
41	Bilateral Closed User Group selection required	2	E.2.6
42	Packet Size selection	2	E.2.1
43	Window Size selection	2	E.2.2
44	RPOA selection required (basic format)	2	E.2.13
49	Transit Delay Selection and Notification	2	E.2.16
C1	Charging information - Call Duration	variable	E.2.12
C2	Charging information - Segment Count	variable	E.2.11
C3	Call Redirection Notification	variable	E.2.15
C4	RPOA selection required (extended format)	variable	E.2.13
C5	Charging information - Monetary Unit	variable	E.2.10
C6	Network User Identification	variable	E.2.8
C9	Called Address Extension (OSI)	variable	E.3.2
CA	Quality of Service Negotiation - End-to-end transit delay	variable	E.3.4
CB	Calling Address Extension (OSI)	variable	E.3.1

E.2 X.25 Facilities

The following are the formats of the facilities.

E.2.1 Packet Size Selection

E.2.1.1 Coding in the Call Packet

0	0x42	
1	Reserved	Transmit packet size
2	Reserved	Receive packet size

Figure 81. Packet Size Selection

0x42

Packet size selection.

Transmit packet size

Indicates the requested size for packets transmitted from the calling DTE. Supported values are:

- 0x04 = 16 octets
- 0x05 = 32 octets
- 0x06 = 64 octets
- 0x07 = 128 octets
- 0x08 = 256 octets
- 0x09 = 512 octets
- 0x0A = 1024 octets
- 0x0B = 2048 octets
- 0x0C = 4096 octets

Receive packet size

Requested size for packets transmitted from the called DTE. Supported values are the same as for the transmitted packet size.

E.2.2 Window Size Selection

E.2.2.1 Coding in the Call Packet

0	0x43	
1	Reserved	Transmit window size
2	Reserved	Receive window size

Figure 82. Window Size Selection

0x43

Window size selection

Transmit window size

Requested size for the window for packets transmitted by the calling DTE. This represents the maximum number of packets that can be received without an acknowledgment. Values are in the range from 0x01 to 0x07 inclusive.

Received window size Requested size for the window for packets to be transmitted by the called DTE. Values are in the range from 0x01 to 0x07 inclusive.

E.2.3 Throughput Class

E.2.3.1 Coding in the Call Packet

0	0x02	
1	Outgoing Throughput class	Incoming Throughput Class

Figure 83. Throughput Class

0x02 Throughput class required.

Outgoing Throughput Throughput class requested for data to be sent by the calling DTE. Supported values are:

0x07 = 1200 bit/s
 0x08 = 2400 bit/s
 0x09 = 4800 bit/s
 0x0A = 9600 bit/s
 0x0B = 19200 bit/s
 0x0C = 48000 bit/s

Incoming Throughput Throughput class request for data sent from the called DTE. Supported values are the same as for incoming throughput class.

E.2.4 Closed User Group Selection

E.2.4.1 Coding in the Call Packet

0	0x03	
1	First BCD digit of CUG	Second BCD digit of CUG

Figure 84. Closed User Group Selection (Basic Format)

Or

0	0x47	
1	First BCD digit of CUG	Second BCD digit of CUG
2	Third BCD digit of CUG	Fourth BCD digit of CUG

Figure 85. Closed User Group Selection (Extended Format)

0x03 (0x47) Closed User Group selection required (extended).

CUG Indicates the value of a Closed User Group: 1 to 99 for the basic format, 1 to 9999 for the extended format.

E.2.5 CUG with Outgoing Access

E.2.5.1 Coding in the Call Packet

0	0x09	
1	First BCD digit of CUG	Second BCD digit of CUG

Figure 86. CUG with Outgoing Access (Basic Format)

or

0	0x48	
1	First BCD digit of CUG	Second BCD digit of CUG
2	Third BCD digit of CUG	Fourth BCD digit of CUG

Figure 87. CUG with Outgoing Access (Extended Format)

0x09 (0x48) Closed User Group with outgoing access (extended).

CUG Indicates the value of a closed user group. 1 to 99 for the basic format, 1 to 9999 for the extended format.

E.2.6 Bilateral Closed User Group Selection

E.2.6.1 Coding in the Call Packet

0	0x41	
1	First BCD digit of CUG	Second BCD digit of CUG
2	Third BCD digit of CUG	Fourth BCD digit of CUG

Figure 88. Bilateral Closed User Group Selection

0x41 Bilateral Closed User Group selection required.

CUG Indicates the value of a Closed User Group (1 to 9999).

E.2.7 Reverse Charging and Fast Select

E.2.7.1 Coding in the Call Packet

0	0x01					
1	A					B

Figure 89. Reverse Charging and Fast Select

- 0x01** Fast Select.
- A** Indicates whether a restricted response is required when fast select is also requested. Valid values are:
- 00** Fast select not selected
 - 01** Fast select selected
 - 10** Fast select requested with no restriction on response
 - 11** Fast select requested with restriction on response
- B** Reverse charge required; valid values are:
- 0** No reverse charging requested
 - 1** Reverse charging requested

E.2.8 Network User Identification

E.2.8.1 Coding in the Call Packet

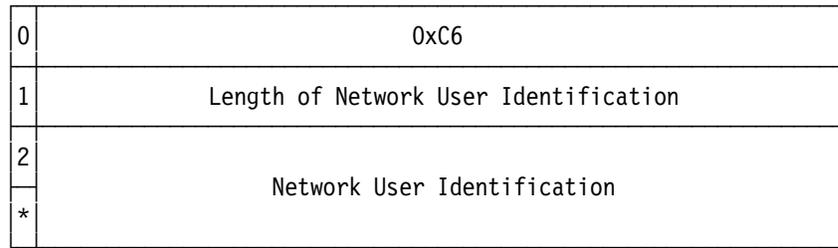


Figure 90. Network User Identification

- 0xC6** Network user identification.
- Length of NUI** The number of bytes given in network user identification data.
- NUI data** Network user identification data in format identified by the network administrator.

E.2.9 Charging Information Request

E.2.9.1 Coding in the Call Packet

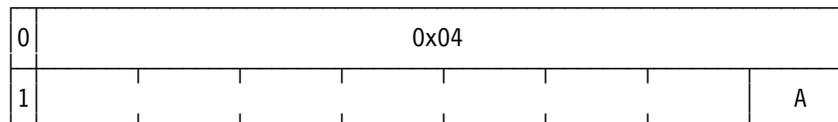


Figure 91. Charging Information Request

- 0x04** Charging Information - Requesting Service can be one of:
- 0** Charging information not requested
 - 1** Charging information requested

E.2.10 Charging (Monetary Unit)

E.2.10.1 Coding in the Call Packet

0	0xC5
1	Length of charging information
2	Charging Identification
*	

Figure 92. Charging (Monetary Unit)

0xC5 Charging information - monetary unit
Length of Charging Info Number of bytes of the charging information
Charging Identification Charging information - monetary unit data

E.2.11 Charging (Segment Count)

E.2.11.1 Coding in the Call Packet

0	0xC2
1	Length of charging information
2	Charging Identification
*	

Figure 93. Charging (Segment Count)

0xC2 Charging information - segment count
Charging Identification Charging information - segment count data

E.2.12 Charging (Call Duration)

E.2.12.1 Coding in the Call Packet

0	0xC1
1	Length of charging information
2	Charging Identification
*	

Figure 94. Charging (Call Duration)

0xC1 Charging information - call duration

Length of Charging Info Length in bytes of charging information
Charging Identification Charging information - call duration data

E.2.13 RPOA Selection

E.2.13.1 Coding in the Call Packet

0	0x44	
1	First BCD digit of RPOA	Second BCD digit of RPOA
2	Third BCD digit of RPOA	Fourth BCD digit of RPOA

Figure 95. RPOA Selection (Basic Format)

Or

0	0xC4	
1	Length of RPOA information	
2	First BCD digit of RPOA #1	Second BCD digit of RPOA #1
3	Third BCD digit of RPOA #1	Fourth BCD digit of RPOA #1
*	First BCD digit of RPOA #n	Second BCD digit of RPOA #n
*	Third BCD digit of RPOA #n	Fourth BCD digit of RPOA #n

Figure 96. RPOA Selection (Extended Format)

0x44 (0xC4) Recognized private operating agency selection required.

RPOA Indicates the requested RPOA transit network identification code, 1 to 9999.

Length of RPOA info. Length in bytes of the RPOA information in the facility.

E.2.14 Called Line Address Modified Notification

E.2.14.1 Coding in the Call Packet

When the redirection is done by the DCE:

0	0x08			
1	0			A

Figure 97. Called Line Address Modified Notification

0x08 Called line address modified notification

A Can have one of the following values:

- 0x7** Call distribution within a hunt group
- 0x1** Call redirection due to original DTE busy
- 0x9** Call redirection due to original DTE out of order
- 0x0F** Call redirection due to prior request from originally called DTE for systematic redirection

Or, when the redirection is DTE originated:

0	0x08	
1	1	B

Figure 98. Called Line Address Modified Notification (CLAMN)

- 0x08** Called Line Address Modified Notification.
- B** Passed from the remote DTE, gives a reason for the redirection.

E.2.15 Call Redirection Notification

E.2.15.1 Coding in the Call Packet

0	0xC3	
1	Length of redirection information	
2	Call redirection reason	
3		Length of called address
4	Called Address (BCD)	
*		

Figure 99. Call Redirection Notification

- 0xC3** Call Redirection Notification.
- Call redirection reason** Contains reason for call redirection.
 - 0x1** Call redirection due to original DTE busy
 - 0x9** Call redirection due to original DTE out of order
 - 0x0F** Call redirection due to prior request from originally called DTE for systematic redirection
- Call redirection address** The original called DTE address coded in BCD.

E.2.16 Transit Delay Selection and Indication

E.2.16.1 Coding in the Call Packet

0	0x49
1	Transit delay in milliseconds (in binary, hi byte first)
2	

Figure 100. Transit Delay Selection and Indication

0x49	Transit delay selection and notification
Transit delay	Transit delay in milliseconds

E.3 CCITT Specified Facilities to Support the OSI Network

The following facilities are defined for OSI networks.

E.3.1 Calling Address Extension

E.3.1.1 Coding in the Call Packet

0	0xCB	
1	Number of bytes following	
2	Use	Length of calling extension address
3	Calling extension address (BCD)	
*		

Figure 101. Calling Address Extension

0xCB	Calling address extension
Use	May may have the following values:
00	To carry an entire calling OSI NSAP address
01	To carry a partial calling OSI NSAP address
10	To carry a non-OSI calling address
11	Reserved

Calling extension address Up to 40 decimal digits coded in BCD containing the calling address extension

E.3.2 Called Address Extension

E.3.2.1 Coding in the Call Packet

0	0xC9	
1	Number of bytes following	
2	Use	Length of called extension address
3	Called extension address	
*	(BCD)	

Figure 102. Called Address Extension

0xC9 Called address extension

Use Values as above

Called address extension Up to 40 decimal digits containing the called address extension coded in BCD

E.3.3 Minimum Throughput Class

E.3.3.1 Coding in the Call Packet

0	0x0A	
1	Calling minimum throughput	Called minimum throughput

Figure 103. Minimum Throughput Class

0x0A Quality of Service Negotiation - minimum throughput class

Calling min throughput Throughput class requested for data to be sent by the calling DTE. Supported values are:
 0x07 = 1200 bit/s
 0x08 = 2400 bit/s
 0x09 = 4800 bit/s
 0x0A = 9600 bit/s
 0x0B = 19200 bit/s
 0x0C = 48000 bit/s

Called min throughput Throughput class request for data sent from the called DTE. Supported values are the same as for the calling minimum throughput class.

E.3.4 End-to-End Transmit Delay Facility

E.3.4.1 Coding in the Call Packet

0	0xCA
1	Length of the following area
2	Cumulative transit delay in milliseconds (in binary, hi byte first)
3	
4	Requested end-to-end delay in milliseconds (in binary, hi byte first)
5	
6	Maximum acceptable transit delay in milliseconds (in binary, hi byte first)
7	

Figure 104. End-to-End Transmit Delay Facility

0xCA	Quality of Service Negotiation - End-to-end transit delay.
Length	The number of values in the stream. This can be one of 1, 2 or 3, as the requested end-to-end delay and maximum acceptable transit delay are optional.
Req end to end delay	Specifies cumulative, requested end-to-end and maximum acceptable transit delays.

E.3.5 Expedited Data Negotiation

E.3.5.1 Coding in the Call Packet

0	0x0B
1	A

Figure 105. Expedited Data Negotiation

0x0B	Expedited Data Negotiation
A	Can be one of:
0	No use of expedited data
1	Use of expedited data

Appendix F. PAD Parameters and Commands

F.1 PAD Parameters

For more detailed descriptions of the PAD parameters, see Table 43 on page 289.

<i>Table 42 (Page 1 of 3). User-Customizable PAD Parameters Defined in X.3</i>			
Number	Name	Value	Meaning
1	PAD recall	0	Data flow may not be interrupted, you cannot enter PAD commands
		1	The escape character is DLE
		32-126	Value of the PAD recall character in decimal
2	Echo	0	No echo by the PAD
		1	Local echo by the PAD
3	Data forwarding signal	0	Data forwarding not controlled by a character
		1	Alphanumeric characters (A-Z, a-z, 0-9)
		2	Character CR
		4	Characters ESC, BEL, ENQ, ACK
		6	Characters CR, ESC, BEL, ENQ, ACK
		8	Characters DEL, CAN, DC2
		16	Characters ETX, EOT
		18	Characters CR, ETX, EOT
4	Idle timer delay	0	No timeout period is used
		1-255	Timeout period expressed in units of 0.05 seconds
5	Ancillary device control	0	No use of X-ON (DC1) and X-OFF (DC3) flow control
		1	Use X-ON and X-OFF (during data transfer)
		2	Use X-ON and X-OFF (during data and command transfer)
6	PAD service signals	0	No service signals are sent to the terminal (start-stop mode DTE)
		1	Service signals other than prompt PAD service are transmitted in standard format
		4	Service signal prompt PAD service is transmitted in standard format
		8-15	PAD service signals are transmitted in a network-dependent format
		16	English extended dialogue mode
		32	French extended dialogue mode
7	Action of PAD on break signal	0	Nothing
		1	Send interrupt packet to correspondent
		2	Reset
		3	= 1 + 2
		4	Send to DTE indication of break PAD message
		5	= 1 + 4
		6	= 2 + 4
		7	= 1 + 2 + 4
		8	Escape from data transfer state
		9-15	Formed from combination of 1, 2, 4, 8
8	Discard output	0	Normal data delivery to terminal
		1	Discard all output to terminal

<i>Table 42 (Page 2 of 3). User-Customizable PAD Parameters Defined in X.3</i>			
Number	Name	Value	Meaning
9	Padding after carriage return	0 1-255	No padding after CR in datastream sent to terminal Specified number of padding characters inserted after CR in datastream sent to terminal
10	Line folding	0 1-255	No line folding Line folding after the specified number of consecutive printable characters
11 (read only)	Line speed	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18	110 bit/s 134.5 bit/s 300 bit/s 1200 bit/s 600 bit/s 75 bit/s 150 bit/s 1800 bit/s 200 bit/s 100 bit/s 50 bit/s 75/1200 bit/s (outbound/inbound) 2400 bit/s 4800 bit/s 9600 bit/s 19200 bit/s 48000 bit/s 56000 bit/s 64000 bit/s
12	Flow control of the PAD	0 1	No use of X-ON and X-OFF Use of X-ON and X-OFF
13	Linefeed after carriage return	0 1 2 3 4 5 6 7	No linefeed insertion Insertion in datastream to terminal Insertion in datastream from terminal = 1 + 2 Insertion in datastream after echo of CR to terminal = 1 + 4 = 2 + 4 = 1 + 2 + 4
14	Padding after linefeed	0 1-255	No padding characters inserted after LF sent to terminal, in data transfer state Specified number of padding characters inserted after LF sent to terminal, in data transfer state
15	Editing	0 1	No use of editing in data transfer state Use of editing in data transfer state
16	Character delete	0-127	International Alphabet 5 (IA5) character that causes character delete
17	Line delete	0-127	IA5 character that causes line delete
18	Line display	0-127	IA5 character that causes line display
19	Editing PAD service signals	0 1 2 8, 32-126	No editing of PAD service signals Editing for printing terminals Editing for display terminals Editing using a character from IA5
20	Echo mask	0 1 2 4 8 16 32 64 128	All characters echoed (no mask) No echo of CR No echo of LF No echo of VT, HT and FF No echo of BEL, BS No echo of ESC, ENQ No echo of ACK, NAK, STX, SOH, EOT, ETB, ETX No echo of editing characters designated by parameters 16-18 above No echo of all other characters in columns 0 and 1 not mentioned above and DEL

Number	Name	Value	Meaning
21	Parity treatment	0 1 2 3	No parity checked or generated Parity checking Parity generation = 1 + 2
22	Page wait	0 1-255	Page wait disabled Number of line feed characters considered by the PAD for the page wait function

F.2 Detailed Description of PAD Parameters

Number	Name	Explanation
1	PAD recall	This parameter specifies the character that a user can type at the terminal to interrupt the data flow with the X.25 host and send commands to the PAD. Note: If this parameter value is changed to 0, the user will no longer be able to change PAD parameters.
2	Echo	This parameter specifies whether characters received from the terminal are to be transmitted back to the terminal, as well as being interpreted by the PAD. When working in line mode, this parameter should be set to 1 to get an echo from the PAD, and the echo provided by the host should be disabled with the <code>stty -echo</code> command.
3	Data forwarding signal	This parameter specifies the conditions under which a string of input characters from a terminal will be converted to a data packet and forwarded over the network, as defined in X.25. If this parameter is set to 0, every character typed at the terminal is sent by the PAD in an individual packet. For the other settings, the PAD acts as a buffer and sends a packet only when the specified character is entered. If the terminal is in line mode, for example entering UNIX commands, this parameter may be set to 18. Thus, packets will only be sent to the host on pressing Enter, Ctrl-C or Ctrl-D.
4	Idle timer delay	This parameter specifies a time-out period for the reception of characters from the terminal. After this time-out expires, characters already received by the PAD are formatted into a packet and sent across the X.25 network. If using line mode, this parameter would be set to 0. When a non-zero value is used, this parameter can improve the performance of applications transferring data from the terminal to the host, as the time delay between the PAD receiving two consecutive characters is lower than the time-out period. The PAD accumulates the data and only sends it when it has a full packet or when the file transfer application has stopped sending characters.
5	Ancillary device control	This parameter allows for flow control between the PAD and the terminal. The PAD indicates whether or not it is ready to accept characters from the terminal by transmitting special characters (the IA5 characters used to switch an ancillary transmitting device on and off).
6	PAD service signals	This parameter determines whether or not, and in what format, PAD service signals are transmitted. It controls the service signals and the dialogue mode for the terminal.
7	Action of PAD on break signal	This parameter specifies the action to be taken by the PAD when it receives a break signal from the terminal.

Number	Name	Explanation
8	Discard output	This parameter determines whether data should be disassembled and transmitted to the terminal, or discarded.
9	Padding after carriage return	This parameter determines if the PAD will automatically add padding characters into the character stream being sent to the terminal after receiving a CR.
10	Line folding	This parameter determines how many characters can be sent to the terminal without inserting an appropriate formatting character.
11	Line speed	This parameter indicates the line speed at which the terminal is attached to the PAD. This parameter cannot be changed; it can only be read.
12	Flow control of the PAD	This parameter allows for the control of flow between the terminal and the PAD. The terminal indicates whether or not it is ready to accept characters from the PAD by transmitting special characters, which in IA5 are used to switch ancillary transmitting devices on and off.
13	Linefeed after carriage return	This parameter provides for the automatic insertion, by the PAD, of a linefeed character into the character stream to or from the terminal, or after each CR character. This function only applies in the data transfer state.
14	Padding after linefeed	This parameter determines if the PAD will automatically add padding characters into the character stream being sent to the terminal after a linefeed character. This function only applies in the data transfer state.
15	Editing	This parameter controls the use of local editing in the PAD command state and the data transfer state. During the PAD command state, the editing function is always available. For example, the terminal is able to edit the PAD command signals sent to the PAD before they are processed. Parameters 15 - 19 control these editing functions.
16	Character delete	This parameter determines the character from the IA5 that is to request character deletion for the PAD editing buffer.
17	Line delete	This parameter determines the character from the IA5 that causes the PAD to delete the contents of its edit buffer.
18	Line display	This parameter determines the character from the IA5 that causes the PAD to display the contents of its edit buffer.
19	Editing PAD service signals	This parameter controls how the PAD service signals can be edited.
20	Echo mask	This parameter selects which characters are echoed if local echo is enabled.
21	Parity treatment	This parameter allows the PAD to check parity in the data stream from the terminal and/or generate parity on the data stream to the terminal.
22	Page wait	This parameter controls screen scrolling. It allows the PAD to suspend the transmission of characters to the terminal after a specified number of linefeed characters have been sent from the PAD.

F.3 Commands Entered from the PAD Prompt

PAD command	Explanation
BREAK	Send a break character to the PAD.

<i>Table 44 (Page 2 of 2). Commands that can be Issued from the PAD Prompt</i>	
PAD command	Explanation
CALL	Establish a connection to the remote X.25 host (call 3106010761)
CLEAR	Clear the connection with the remote X.25 host. The clear is sent from the PAD immediately - see ICLEAR.
HELP	Request help text, see 5.3.2, "PAD Commands" on page 97 for more details.
ICLEAR	Sends an invitation to clear to the remote X.25 DTE. This allows the remote host to send any pending data before clearing the call.
INTERRUPT	Send an interrupt packet. The packet contents cannot be user specified.
LANGUAGE	Set the language for PAD help text to English, French or Spanish. (language french)
NUI	This command is not implemented in the PAD.
PAR	Read the PAD parameters. PAR? Displays all parameters PAR? 2, 3 Displays parameter 2 and 3 In advanced mode, READ or PARAMETER can be used instead of PAR
PROFILE	Displays which profiles are available, or to change the profile enter PROFILE followed by the profile name, for example, PROFILE PROFILE_51
READ	See PAR above
RREAD	READ function for remote PAD
RSET	SET function for remote PAD
SET	Sets one or more of the PAD parameters: set 2:1, 14:2 or in advanced mode set echo:1, lfpad:2 Note: The default profile does not allow the parameters to be changed locally, as they are being controlled remotely through the TTY subsystem and the stty command.
STATUS	Gives whether a connection to the remote X.25 host is active. Returns ENGAGED if it is active or FREE if the connection has not been established.

F.4 Messages from the PAD to the Terminal

Just as the user can send commands to the PAD, the PAD can send indications to the user. These indications are called PAD service signals. The PAD service signals are summarized in the following table.

Table 45. Service Signals Sent from the X.3 PAD to the X.28 Terminal

PAD service signal	Explanation
COM	Call successful
CLR OCC	The called DTE is occupied
CLR NC	Network congestion
CLR INV	Invalid facility
CLR NA	Access barred (for example because of CUG)
CLR ERR	Local procedure error
CLR RPE	Remote procedure error
CLR NP	Called number does not exist
CLR DER	The called number is out of order
CLR PAD	The call has been cleared as an answer to an invitation to clear from the remote DTE
CLR DTE	The remote DTE has cleared the call
CLR CONF	Clear confirmation (response to CLR PAD command)
RESET DTE	Remote DTE
RESET ERR	Local procedure error
RESET NC	Network congestion
ERROR	Error in PAD command
XXX	Line delete function complete

Appendix G. CIO and X.25 Device Driver Error Codes

Symbol	Hex.	Dec.	Cause
CIO OK	00	00	Operation was successful
CIO BAD_MICROCODE	01	01	Invalid microcode
CIO BUF_OVFLW	02	02	Data too large for buffer
CIO HARD_FAIL	03	03	Hard failure
CIO LOST_DATA	04	04	Data were lost
CIO NOMBUF	05	05	No mbuf available
CIO NOT_STARTED	06	06	Start not performed
CIO TIMEOUT	07	07	Operation timed out
CIO NET_RCVRY_ENTER	08	08	Enter network recovery
CIO NET_RCVRY_EXIT	09	09	Exit network recovery
CIO NET_RCVRY_MODE	0A	10	Network recovery mode
CIO INV_CMD	0B	11	Invalid command
CIO BAD_RANGE	0C	12	Bad address range
CIO NETID_INV	0D	13	Invalid net ID
CIO NETID_DUP	0E	12	Duplicate net ID
CIO NETID_FULL	0F	15	All net are used
CIO TX_FULL	10	16	Transmit queue is full
X25 BAD_CALL_ID	41	65	The X.25 Reject operation completed
X25 CLEAR	42	66	The session has been closed
X25 INV_CTR	43	67	The counter is invalid
X25 NAME_USED	44	68	The name is already being listened to
X25 NOT_PVC	45	69	Not defined as a Permanent Virtual Circuit (PVC)
X25 NO_ACK	46	70	No acknowledgement has been issued
X25 NO_ACK_REQ	47	71	No packets currently require acknowledgement
X25 NO_LINK	48	72	The link is not configured or could not be connected
X25 NO_NAME	49	73	There is no such name in the routing list
X25 PROTOCOL	4A	74	An X.25 protocol error occurred, probably caused by trying to transmit data on a circuit which has already been cleared
X25 PVC_USED	4B	75	The Permanent Virtual Circuit (PVC) has already been allocated
X25 RESET	4C	76	The virtual circuit has been reset
X25 TABLE	4D	77	Could not update routing list
X25 TOO_MANY_VCS	4E	78	No virtual circuits are free on the port
X25 AUTH_LISTEN	4F	79	You cannot listen to this name because the routing list entry has a userid which excludes the user running the application
X25 BAD_PKT_TYPE	50	80	Invalid packet type
X25 BAD_SESSION_TYPE	51	81	

Appendix H. Country Networks Default Parameters

Table 46. Default Parameters for the Various Networks

Country	Code	Network	Conn. Mode	CCITT Support	Thruput Range	Packet Range
Argentina	722	Other public	Passive	1980	75-48000	16-1024
Australia	505	Other public	Active	1980	75-48000	16-1024
Austria	232	DATEX	Passive	1980	300-9600	64-512
Austria	232	Other public	Passive	1980	300-9600	64-512
Belgium	206	Other public	Passive	1980	75-48000	16-1024
Brazil	724	Other public	Active	1980	75-48000	16-1024
Canada	302	DATAPAC	Passive	1980	75-48000	16-1024
Canada	302	Other public	Passive	1980	75-48000	16-1024
Chile	730	Other public	Active	1980	75-48000	16-1024
China	460	Other public	Active	1980	75-48000	16-1024
Denmark	238	DATAPAC	Active	1980	75-48000	16-1024
Denmark	238	Other public	Passive	1984	75-48000	64-512
Egypt	602	Other public	Active	1984	75-48000	16-4096
Finland	244	DATAPAC	Active	1984	75-48000	16-4096
Finland	244	Other public	Active	1980	75-48000	16-1024
France	208	Other public	Active	1984	75-48000	16-4096
Greece	202	Other public	Active	1984	75-48000	16-4096
Hong Kong	454	DATAPAC	Passive	1980	75-48000	16-1024
Iceland	274	Other public	Active	1980	75-48000	16-1024
Indonesia	510	Other public	Active	1984	75-48000	16-4096
Ireland	272	Other public	Passive	1984	75-48000	64-512
Israel	425	Other public	Passive	1980	75-48000	16-1024
Italy	222	Other public	Passive	1980	75-48000	16-1024
Japan	440	Other public	Active	1980	75-48000	16-1024
Korea	450	Other public	Active	1980	75-48000	16-1024
Luxembourg	270	Other public	Active	1980	75-48000	16-1024
Malasia	502	Other public	Passive	1980	75-48000	16-1024
Mexico	334	Other public	Passive	1980	75-48000	16-1024
Netherlands	204	Other public	Active	1980	75-48000	16-1024
New Zealand	530	Other public	Active	1980	75-48000	16-1024
Norway	242	DATAPAC	Passive	1980	75-48000	16-1024
Portugal	268	Other public	Passive	1980	75-48000	16-1024
Singapore	525	Other public	Passive	1980	75-48000	16-1024
South Africa	655	Other public	Active	1980	75-48000	16-1024
Spain	214	Other public	Passive	1980	75-48000	16-1024
Sweden	240	DATAPAC	Passive	1980	75-48000	16-1024
Switzerland	228	Other public	Passive	1980	75-48000	16-1024
Turkey	286	Other public	Passive	1980	75-48000	16-1024
UK*	234-5	Other public	Passive	1980	75-48000	16-1024
USA	310-6	DDN	Active	1980	75-48000	16-1024
USA	310-6	Other public	Active	1980	75-48000	16-1024
USA	310-6	TELENET	Active	1980	75-48000	16-1024
Germany	218	DATEX	Passive	1980	75-48000	16-1024
Germany	218	Other public	Active	1984	75-48000	16-4096
Yugoslavia	220	Other public	Passive	1980	75-48000	16-1024
Others		DATAPAC	Passive	1980	75-48000	16-1024
		DATEX	Passive	1980	75-48000	16-1024
		DDN	Active	1980	75-48000	16-1024
		TELENET	Active	1980	75-48000	16-1024
		Other public	Active	1980	75-48000	16-1024

*: Also first incoming SVC set to 512, first 2-way SVC set to 1024, first outgoing SVC set to 1536.

Appendix I. Special Notices

This publication is intended to help customers and systems engineers install, customize and use X.25 support of the RISC System/6000. The information in this publication is not intended as the specification of any programming interfaces that are provided by AIX/6000, AIXLink/X.25, AIX X.25 or AIX SNA Services. See the PUBLICATIONS section of the IBM Programming Announcement for AIX/6000, AIXLink/X.25, AIX X.25 or SNA Services for the RISC System/6000 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX	AIX/6000
AIXwindows	AnyNet
APPN	BookManager
BookMaster	C Set ++
IBM	Micro Channel
NetView	Portmaster
PS/2	RISC System/6000
RS/6000	S/370
S/390	SAA
System/390	Systems Application Architecture
VTAM	XWindow System

The following terms are trademarks of other companies:

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

C-bus is a trademark of Corollary, Inc.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Java and HotJava are trademarks of Sun Microsystems, Inc.

BSC	BusiSoft Corporation
CA	Computer Associates
DCE	The Open Software Foundation
Digital, VT	Digital Equipment Corporation
SCSI	Security Control Systems, Incorporated
ADD	Applications Design and Development, Incorporated
SMS	Standard Microsystems Corporation
AFS	Transarc Corporation
ODM	Optical Disk Mastering, Incorporated
TI	Texas Instruments, Incorporated
DCA	Digital Communications Associates, Incorporated
MS	Microsoft Corporation
DynaText	Electronic Book Technologies, Incorporated
Geneva	Apple Computer, Incorporated
Network File System, NFS	Sun Microsystems, Inc.
X-Windows	Massachusetts Institute of Technology
Intel,80186	Intel

Other trademarks are trademarks of their respective companies.

Appendix J. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

J.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How To Get ITSO Redbooks" on page 303.

- *International Technical Support Organization Bibliography of Redbooks*, GG24-3070.

To get listings of ITSO technical publications (known as "redbooks") online, VNET users may type:

TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG

How to Order ITSO Technical Publications

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-284-4721. Visa and Master Cards are accepted. Outside the USA, customers should contact their local IBM office.

Customers may order hardcopy ITSO books individually or in customized sets, called GBOFs, which relate to specific functions of interest. IBM employees and customers may also order ITSO books in online format on CD-ROM collections, which contain books on a variety of products.

J.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RISC System/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RISC System/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

J.3 Other Publications

These publications are also relevant as further information sources:

Manuals written for the X.25 LPP:

- *AIXLink/X.25 1.1 for AIX: Guide and Reference*, SC23-2520-02.

Other manuals:

- *X.25 Interface Co-Processor/2 Technical Reference*, S16F-1879-00.
- *Realtime Interface Co-Processor Portmaster Adapter/A*, S33F-5337-01 and supplements SC28-9598-00 and SC28-9599-00.
- *AIX Communication Concepts and Procedures for RISC*, GC23-2203-01.
- *AIX Version 3.2 Technical Reference Volume 3: Communications*, SC23-2384-02.
- *AIX Version 3.2 System Management Guide: Communications & Networks*, GC23-2487-00.
- *AIX Version 4 System Management Guide: Communications & Networks*, SC23-2526-03.
- *The X.25 DTE/DCE and DTE/DTE Interface*, SC30-3409-02.
- *AIX SNA Server/6000: Transaction Program Reference*, SC31-7003-01.
- *AIX SNA Server/6000: Configuration Reference*, SC31-7014-01.
- *AIX SNA Server/6000: Command Reference*, SC31-7100-00.
- *AIX V4 Software Problem Debugging and Reporting*, GG24-2513-00.

J.3.1 Articles

- Special Thanks to /AIXTRA: IBM's Magazine for AIX Professionals for permission to reprint the article entitled "Networking with X.25", by Cindy Kueck Young and Alan E. Hodel, "Networking with X.25", January 1992 issue, G362-1002-00.

J.4 Non-IBM References

- ISO 8208 - the International Standard on Information Processing Systems - *Data Communications - X.25 Packet Level Protocol for Data Terminal Equipment* (1987).
- ISO 7776 - the International Standard on Information Processing Systems - *Data Communications - High-Level Data Link Control Procedures - Description of the X.25 LAPB-compatible DTE Data Link Procedures*.
- *Yellow Book, Volume VIII - Fascicle VIII.2, Data Communication Networks Services and Facilities, Terminal Equipment and Interfaces, Recommendations X.1-X.29* (7th Plenary Assembly, Geneva 10-21 November 1980).
CCITT Recommendation X.25, *Interface Between Data Terminal Equipment (DTE) and Data Circuit Terminating Equipment (DCE) for Terminals Operating in the Packet Mode and Connected to Public Data Networks by Dedicated Circuit*.
- *Red Book, Volume VIII - Fascicle VIII.3, Data Communication Networks Interfaces, Recommendations X.20-X.32* (8th Plenary Assembly, Malaga-Torremolinos 8-19 October 1984).
CCITT Recommendation X.25, *Interface Between Data Terminal Equipment (DTE) Data and Circuit Terminating Equipment (DCE) for Terminals Operating in the Packet Mode and Connected to Public Data Networks by Dedicated Circuit*.

- *Blue Book, Volume VIII - Fascicle VIII.2, Data Communication Networks: Services and facilities, Interfaces, Recommendations X.1-X.32* (9th Plenary Assembly, Melbourne 14-25 November 1988, ISBN 92-61-03671-6).

CCITT Recommendation X.25, *Interface Between Data Terminal Equipment (DTE) and Data Circuit Terminating Equipment (DCE) for Terminals Operating in the Packet Mode and Connected to Public Data Networks by Dedicated Circuit.*

- Deasington, Richard *X.25 Explained: Protocols for Packet Switching Networks*. 2nd edition. Chichester: Ellis Horwood, 1988.
- *Inside X.25: A Manager's Guide*, by Sherman K. Schlar, McGraw Hill, ISBN-0-07-607007-7, 300 pages, 1990.

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL <http://www.redbooks.ibm.com>.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

To get LIST3820s of redbooks, type one of the following commands:

```
TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
```

To get lists of redbooks:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
```

To register for information on workshops, residencies, and redbooks:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
```

For a list of product area specialists in the ITSO:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
```

- **Redbooks Home Page on the World Wide Web**

<http://w3.itso.ibm.com/redbooks>

- **IBM Direct Publications Catalog on the World Wide Web**

<http://www.elink.ibm.link.ibm.com/pb1/pb1>

IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibm.link.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	Internet
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark
--	--	--

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States) or (+1) 415 855 43 29 (Outside USA)** — ask for:

Index # 4421 Abstracts of new redbooks
Index # 4422 IBM redbooks
Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com

- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibm.com/pbl/pbl

- **Internet Listserver**

With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibm.com with the keyword `subscribe` in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity
-------	--------------	----------

- Please put me on the mailing list for updated versions of the IBM Redbook Catalog.
-

First name	Last name
------------	-----------

Company

Address

City	Postal code	Country
------	-------------	---------

Telephone number	Telefax number	VAT number
------------------	----------------	------------

- Invoice to customer number _____

- Credit card number _____
-

Credit card expiration date	Card issued to	Signature
-----------------------------	----------------	-----------

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

Abbreviations

ABBREVIATION	MEANING		
AIX	advanced interactive executive (IBM's flavor of UNIX)	EMEA	Europe, Middle East and Africa
AIX BOS	AIX base operating system	ESC	the escape character
APAR	authorized program analysis report	FTP	file transfer protocol
API	application program interface	GDLC	Generic data link control (SNA)
ARTIC	a real time interface coprocessor	GOSIP	government OSI profile (USA)
ASCII	American National Standard Code for Information Interchange	HCON	HCON - IBM AIX 3270 host connection program/6000 (licensed program)
BCD	binary coded decimal	HDLC	high-level data link control (X.25 protocol)
BSC	binary synchronous communication	IP	internet protocol (ISO)
BSD	Berkeley software distribution (UC at Berkeley, UNIX)	IPC	inter-programm communication
CCITT	Comite Consultatif International Telegraphique et Telephonique	IPL	initial program load
CD	carrier detect (RS-232)	ISO	International Organization for Standardization
COMIO	Common Input/Output	ITSC	International Technical Support Center
CPU	central processing unit	Kbps	kilobits per second
CTS	clear to send (RS-232)	LAN	local area network
CU	call up (UNIX / AIX utility for using modems and asynchronous lines)	LAPB	link access protocol - balanced
CUD	call/clear user data (XI)	LAPX	link access procedure balanced (XI)
CUG	closed user group (OSI layer 3)	LLC	logical link control (LAN, top sublayer of layer 2, IEEE 802.2)
Ctrl	control	LPP	licensed program product
DCD	data carrier detect	LU	logical unit
DCE	data circuit-terminating equipment (ISO, CCITT, EIA)	MPQP	IBM 4-Port Multiprotocol Communications Controller
DDN	defense data network	MTU	maximum transmission unit (Internet protocols)
DLPI	Data Link Provider Interface	Mbps	megabits per second
DSE	data switching equipment	NIA	network interface adapter
DSR	data set ready	NIS	network information system (Sun Microsystems Inc.)
DTE	data terminal equipment	NPI	network provider interface
DTR	data terminal ready	NTN	national trends network (EPA, USA)
EBCDIC	extended binary coded decimal interchange code	NTU	network terminating/terminator unit
		NUA	network user address
		NUI	network user identifier (PPSN)

ODM	object data manager (AIX)	TELNET	U.S. Dept. of Defense's virtual terminal protocol, based on TCP/IP
OSI	open systems interconnection		
PAD	packet assembler / disassembler (X.25)	TTY	teletypewriter
PING	packet internet groper	TXCLK	transmit clock
PMR	problem management report	TXD	transmit data
PN	part number	UA	unnumbered acknowledgement
PS/2	IBM Personal System 2	UUCP	UNIX-to-UNIX communication protocol
PSDN	packet switching data network	V.24	list of definitions for interchange circuits between terminal equipment and circuit-terminating equipment (CCITT)
PTF	program temporary fix		
PTT	Post Telegraph Telephone (National Post and Telecommunication Authority)	V.35	standards for data transmission at 48 kilobits per second (CCITT)
PU	physical unit (SNA)	VC	virtual circuit (XI)
PVC	permanent virtual circuit	VTAM	virtual telecommunications access method (IBM) (runs under MVS, VM, DOS/VSE)
QLLC	qualified logical link control		
RCM	realtime control microcode	WAN	wide area network
RISC	reduced instruction set computer/cycles	X.21	ISO standard for interface to digital transmission services
RJE	remote job entry	X.25	ISO standard for interface to packet switched communications services
RPOA	recognized private operating agency (CCITT)	X.28	a CCITT recommendation for the DTE / DCE interface for a start-stop DTE accessing the packet assembly/disassembly facility in a public network situated in the same country
RTS	request to send (RS-232)	X.29	a CCITT recommendation for procedures for the exchange of control information between a packet assembly/disassembly facility (PAD), a packet mode DTE or another PAD
RXCLK	receive clock		
RXD	receive data	X.3	a CCITT recommendation for the packet assembly/disassembly facility in a public data network
SABM	set asynchronous balance mode	XID	exchange station identification (SDLC)
SDLC	synchronous data link control (teleprocessing)	bps	bits per second
SMIT	system management interface tool (AIX)	ms	millisecond
SNA	systems network architecture (IBM)		
SNMP	simple network management protocol		
SRC	system resource controller		
SSCPID	system services control point identification		
SVC	switched virtual circuit		
TCP/IP	transmission control protocol internet protocol		

Index

Special Characters

/etc/hosts 139

Numerics

6-Port V.35 Interface Board/A 26, 29
6-Port X.21 Interface Board/A 26, 29
8-Port RS-232-D/CCITT V.24 Interface Board/A 26, 29

A

abbreviations 307
acronyms 307
AIX SNA Server/6000 149
ARGENTINA 295
arpt_killc 143
ARTIC960 Adapter 30
AUSTRALIA 295
AUSTRIA 295

B

backupx25 89
BELGIUM 295
bibliography 299
Bilateral Closed User Group Selection 279
BRAZIL 295

C

Call redirection notification 19, 283
Call User Data 84
Called Address Extension 285
Called Line Address Modified Notification 282
Calling Address Extension 284
CANADA 295
CCITT 1
CCITT Clear and Reset Causes 269
CCITT Diagnostic Codes 269
CCITT Recommendation X.25 5
CCITT Specified Facilities to OSI 284
Changing TCP/IP CUD Values 146
Charging Information Request 280
Charging requesting service 19
chdev command 70, 71
CHILE 295
CHINA 295
Clearing an X.25/IP session 143
Clearing codes 100
closed user group (CUG) 18
Closed User Group Selection 278
Closed User Group with Outgoing Access 279
Co-Processor/2 25

COMIO Emulation 35
Configuring SNMP 147
Consultative Committee on International Telegraph
and Telephone (CCITT) 1
Control Point Profile, changing 163
CUD 84, 146

D

Data Circuit-terminating Equipment (DCE) 5
Data Link Control (DLC) 159
Data Link Control (DLC) profile, defining 164
Data-Switching Equipment (DSE) 5
Data-Terminal Equipment (DTE) 5
DCE 5
DDN 142
Debugging Information 206
DENMARK 295
Differences between X.25 LPP and AIX V3 bos X.25
support 239
DLC 159
DLPI 35
DSE 5
DTE 5

E

EGYPT 295
End-to-End Transmit Delay Facility 286
ESCDELAY variable 194
Expedited Data Negotiation 286

F

Facilities, list of 276
fast select mode 69
fast select option 19
FC 1170 29
FC 1210 29
FC 1211 29
FC 2028 29
FC 5306 29
FC 6590 29
FC 7006 26
FC 7008 26
FC 7042 26
FC 7046 26
FC 7048 26
FINLAND 295
flow-control parameters negotiation 18
Fragmentation 222
FRANCE 295
Function Keys not Working Properly 194

G

GERMANY 295
GREECE 295

H

Hardware Requirements 39
high-level data-link control (HDLC) 8
HONG KONG 295
hooks 203

I

I (information) frames 8
IBM ARTIC Multiport Adapter Model II 29
IBM ARTIC Portmaster Adapter/A 26
IBM ARTIC960 Adapter 30
IBM X.25 Interface Co-Processor 25
ICELAND 295
implied route 191
INDONESIA 295
Initial Node Setup 161
International Organization for Standardization (ISO) 3
Interrupt Level 26
IP sessions on the same virtual channel 143
iptrace 223
IRELAND 295
ISO 3
ISO 8208 Diagnostic Codes 271
ISRAEL 295
ITALY 295

J

JAPAN 295

K

KOREA 295

L

link level 8
Link Station profile, defining 165
logical channel 14
Logical Channel States 274
lsdev command 45
lslpp command 52
LU 0 151
LU 1 151
LU 2 151
LU 2 Session profile, defining 169
LU 3 151
LU 3 Session profile, defining 171
LU 6.2 151
LUXEMBOURG 295

M

MALAYSIA 295
manage unexpected results by ping 192
Maximum facility field length 71
Maximum Transmission Unit 222
mbuf 224
Memory buffers 224
MEXICO 295
Minimum Throughput Class 285
modem setting verification 52
MTU 222

N

NETHERLANDS 295
network user address (NUA) 12
Network user identification 19, 280
NEW ZEALAND 295
NORWAY 295
NPI 35

O

Open Systems Interconnection (OSI) 4
OSI 4

P

Packet Assembler Disassembler (PAD) 91
packet layer 9
packet level protocol 9
packet-switched data network (PSDN) 1
PAD commands 97
PAD parameters 95, 287
PAD Parameters and Commands 287
PAD parameters, detailed description 289
PAD Profiles, parameter settings for predefined profiles 101, 102
PAD start 113
PAD stop 113
PAD Support 35
PAD, basic functions of 92
PAD, commands entered from the prompt 290
PAD, messages from PAD to the terminal 292
Performances 219
permanent virtual circuit (PVC) 15
physical connections to the network 47
ping 192
Portmaster Adapter 26, 29
PORTUGAL 295
Problem Determination 181
Product-Specific Diagnostic Codes 271
Profiles for LU 2, defining 163
PSDN 1
PVC 15

Q

QLLC 150
Qualified Logical Link Control (QLLC) 150

R

RCM 26
Realtime Control Microcode (RCM) 26
reject packet 184
removex25 89
restart packet 184
restorex25 89
reverse charging 70
Reverse Charging and Fast Select 279
reverse charging option 19
RPOA selection 19, 282

S

S (supervisory) frames 8
SINGAPORE 295
SMIT screens 227
SNA APIs 152
SNA Diagnostic Codes 271
SNA profiles 153
SNA Server Problem Determination 194
SNA Server/6000 setup for X.25 158
SNMP 35
SOUTH AFRICA 295
SPAIN 295
start IP/X.25 interface 138
static route 191
SVC 15
SWEDEN 295
switched virtual circuit (SVC) 15
SWITZERLAND 295

T

Throughput Class 278
throughput class negotiation 17
TURKEY 295

U

U (unnumbered) frames 8
UNITED KINGDOM 295
USA 295

V

V.24/X.21bis Pin Assignment 251
V.35/X.21bis Pin Assignment 252
V.36 Pin Assignment 253
virtual circuit 13, 14
VTAM definitions 160

W

Window Size Selection 277
Wrap plug 25
Wrap plug Portmaster and Multiport 262
Wrap plug Co-Processor 255

X

X.21 Connection 47
X.21 Pin Assignment 249
X.21bis Connections: V.24 and V.35 47
X.25 Commands 34
X.25 Device Driver 33
X.25 Optional facilities profile, defining 168
X.25 ports 33
X.28 93
X.28 Facility codes 99
X.29 93
X.3 93
X.32 36
x29d 113
xroute 83
xspad command 114
xtalk 74
XTALK Diagnostic Codes 274

Y

YUGOSLAVIA 295



Printed in U.S.A.

SG24-4475-01

